2 QUERY MODIFICATION THROUGH RELEVANCE FEEDBACK

2.1 Retrieval techniques

This chapter outlines some of the research and practice which underlies the system development and experiments described in later chapters. For good and largely non-overlapping reviews of research in the area three useful references are Bookstein on probability and fuzzy set applications to information retrieval [BOOK85], Belkin and Croft on retrieval techniques [BELK87a] and Efthimiadis and Robertson on feedback and interaction [EFTH89].

The central problem in information retrieval is matching, or helping the user to match, the user's terminology to the terminology used in describing (indexing) the records which are relevant to the query. There is an ideal, and unattainable, search on a perfect system which finds all the items which are about the user's topic, and no other items. In practice things are very different. Relevant documents may be described in many different ways. Different searchers express the same search differently. Documents may be inadequately, inconsistently or incorrectly indexed. It is clearly impossible in general to achieve an ideal search. In any case many users do not want an exhaustive search.

For our present purposes there are just two categories of document retrieval techniques. The first is typified by conventional reference retrieval systems, DIALOG for example, which usually require boolean search statements in the form of words separated by AND, OR, NOT, and other, stronger, AND-like connectives like WITH and ADJ. These will be referred to as *boolean* systems and they use what Belkin and Croft call *exact match* retrieval techniques. Most of the more recent online catalogues fall into this category, although users are not in most cases expected to be able to construct boolean searches. Some online catalogues only allow searching on one "feature" at a time, where a feature might be a subject heading or a single title or subject word. Others assume an AND between the words of the user's search. The distinguishing characteristic of boolean systems is that a search either "succeeds" or fails. Every search results in a set of records (an empty set in the case of failure) which exactly match the search. The records in the retrieved set are unordered in the sense that they all match the request to the same degree, although they may be ordered in some other way - alphabetically, or by date or accession number. (Note that this category includes systems where the search key is chosen by the user from a displayed list. It includes systems with touch screen interaction as well as ones where an initial search results in the display of an index or a list of headings from which the user selects one for display of the bibliographic records.)

The other type of retrieval system will be referred to as *ranked output* systems. Most of these use what Belkin and Croft call *partial match* retrieval techniques [BELK87a]. The distinguishing feature of ranked output systems is that records are output in a sequence which is supposed to reflect either their probability of relevance, in the case of *probabilistic* systems, or, for *vector space models*, the

degree to which they match the query.

While almost all of the operational retrieval systems are of the exact match type they are of little theoretical interest. It can be said that they give acceptable, although very far from ideal, results so long as searching is done by professionals. But to be operable by a wide range of end users their functionality has to be reduced, as is illustrated by the existence of online catalogue systems which retrieve only records indexed by all the user's query terms or which accept only one term at a time. Apart from the well-documented inability of many people to learn to compose boolean queries which reflect what they mean (see, for example, [BORG86]), many writers have pointed out the irrationality of a system which treats all the terms in a query as of equal importance or usefulness.

Ranked output techniques have been the subject of a large amount of research and development, but for reasons which it is not appropriate to discuss here there are relatively very few live implementations. Some of them are mentioned in later sections. There are many varieties of partial match system. What most of them have in common is that records are not required to match the query exactly, and output is produced in order of decreasing goodness of fit to the query. There are many different theories and techniques of matching records to queries.

The vector space models, associated with the SMART projects described by Salton and others in many publications ([SALT68] and [SALT71] for example), treat both queries and documents as points in a many-dimensional space in which each dimension corresponds to a query or index term. Both query and index terms are given numerical weights which are intended to be a measure of their degree of importance or the extent to which they represent a topic. Matching is performed by computing some measure of the similarity between the vector representing the query and those representing the documents in the collection. In principle vector models have some clear advantages over exact match techniques. In practice they have been little used, partly because they are not easy to implement on a large scale, but also because probabilistic models are thought to have a sounder theoretical basis.

Probabilistic models aim to attach a numerical measure, or weight, to each document in the collection which reflects the probability that the document will be relevant to the query. The truth of the "probability ranking principle" is usually assumed. This is given by Robertson [ROBE77], quoting W S Cooper, as

> If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best which is obtainable on the basis of that data.

Thus it is necessary to estimate the probability that each document in the collection will be relevant to the query given the description of the document. This estimation is impossible without making simplifying assumptions. Almost all practical implementations have assumed that document terms, or features, occur independently of each other, both among the relevant documents in the collection and among the non-relevant documents. This is a strong assumption because it is

obviously far from true: one would expect a positive association between "yachting" and "boating" and perhaps a negative one between "yachting" and "diseases of the kidney". Once this independence assumption has been made the problem of estimating the probability of relevance from a document description can be reduced to one of estimating the probabilities of each feature occurring in relevant and non-relevant documents. From these term-probabilities one can calculate a weight for each document such that if documents are placed in order of descending weight they are in descending order of probability of relevance. The remaining problem is one of how to get a search off the ground. Initially there is no relevance information, so there is no obvious way of estimating the probability of terms occurring in relevant and non-relevant documents. It is clear that probabilistic systems need to make use of relevance feedback before their potential can be realised.

2.2 Term weighting and relevance feedback

Experienced searchers often try to find one or a few relevant references, look at terms by which these records are described, and choose some of these as new search terms. This is an example of (positive) relevance feedback (1.2). (Negative feedback - the exclusion of terms which have been seen only in non-relevant records - is also possible, but has not been much used.)

Looked at from a common sense point of view, it is clear that relevance feedback ought to be effective. It is giving the system additional information about what constitutes a relevant document. It seems obvious that if a document is relevant to a query (or an information need) then if there are other documents which are similar to it, these are also likely to be relevant. "Similar" documents might be ones by the same author, ones with a high proportion of subject descriptors in common with the relevant document, ones which cite some of the same references, and so on. Clearly there must be more or less reliable ways of measuring the degree of similarity between two documents. This is connected with the number of common features, and is complicated by the fact that different types of feature vary in importance. In computing the degree of similarity between documents some account may also need to be taken of the "length" of the documents.

In the SMART projects (2.1) documents and queries were represented by vectors, the elements of which represented the presence or absence, and the importance, if present, of each term in the document or query. In the system proposed by Rocchio [ROCC71], one of the SMART project workers, an initial user query would retrieve documents in decreasing order of similarity with the query vector. The user would then make relevance assessments, and the query would be modified by adding the normalized vectors of relevant documents to the query vector, and subtracting the vectors of non-relevant documents. A second search would then be made and the retrieved records assessed for relevance. This process may be iterated as many times as required.

As mentioned above, relevance information plays a more central part in probabilistic systems. The most difficult problem in the implementation of probabilistic systems is that of obtaining estimates of the parameters of the probability distributions involved. Assuming document terms are independent of each other, we need to estimate two parameters for each term: p - the probability that it will occur in relevant documents, and q - the probability that it will occur in

non-relevant documents. It can be shown that if each term is assigned a weight

$$w = \log((p / (1 - p)) / (q / (1 - q))) \qquad (1)$$

and each document is given a weight W equal to the sum of the weights w of all the terms which it has in common with the query, then if documents are output in order of decreasing W they will be in decreasing order of probability of relevance. This formula can be arrived at in a number of ways. The most frequently cited paper is probably that of Robertson and Sparck Jones, published in 1976 [ROBE76]. Apart from taking the logarithm, which is really done so that term weights may be added to obtain a document weight, this formula seems sensible. It represents the ratio of the odds on a relevant record containing the term to the odds on a non-relevant record containing the term. The greater this ratio, the better the term ought to be for discriminating between relevant and non-relevant records.

If some relevant documents are known there is an obvious way of estimating the parameter p as the proportion of known relevant documents containing the term. The reliability of this estimate depends on the method used to obtain the relevant documents, and the number of known relevant documents. If they were obtained by searching for occurrences of the term in question, then the probability will often be an overestimate. This leads to consideration of techniques for performing the initial search.

Croft and Harper, in another much-cited paper [CROF79], proposed that in the absence of any relevance information one might as well assign identical relevance-probabilities (p) for all the terms in the query. Since most documents in a collection will be non-relevant for most queries, q, the non-relevance probability, can reasonably be estimated as the proportion of documents in the whole collection which contain the term. The p-component in the above formula is then the same for all the terms, and the term weight becomes

$$w = c + \log((N - n)/n) \qquad (2)$$

where c is log(p / (1 - p)), N is the number of documents in the collection and n the number of documents containing the term. If p approaches 1 the first term becomes large compared with the second, and documents will be ordered in accordance with the number of query terms they contain. This is often known as *coordination level* matching. If p is taken to be 1/2, c becomes zero, and

$$w = \log((N - n)/n) \qquad (3)$$

which is almost the same as the well known inverse frequency weighting log(N/n) [SPAR72]. Experiments have been made with different values of p. Croft and Harper [CROF79] suggest values between 0.6 and 0.9, and the latter is the value used in INSTRUCT (2.4.4). It is believed that CITE (2.4.2) also uses a value greater than 0.5. Okapi systems have always used 0.5, so take no direct account of the number of terms common to the query and the retrieved documents.

Once relevance information is available p can be estimated rather than guessed. If there are R relevant records and r of them contain the term, then r/R is an estimate of p. If it is assumed that all the non-assessed documents are non-relevant q can be

estimated as (n - r)/(N - R) (where N and n have the same meaning as above). The log-odds formula (1) above then becomes

$$w = \log((r / R - r)/(n - r / N - n - R + r)) \qquad (F4)$$

This is the widely used F4 formula of Robertson and Sparck Jones [ROBE76]. It reduces to (3), the inverse term-frequency formula, when there is no relevance information. It is usual to add 0.5 to each of the components. This avoids infinite or indeterminate values when some of the components are zero, and may be more accurate when there is little relevance information. This gives

$$w = \log((r + 0.5 / R - r + 0.5)/(n - r + 0.5 / N - n - R + r + 0.5)) \qquad (F4')$$

This F4' formula has been widely used in probabilistic systems with relevance feedback. It has been pointed out that it tends in practice (with collections of realistic size) to give unduly high weight to rare terms (ones with small n) which have occurred in few relevant records. Robertson has proposed a modification which somewhat reduces this effect [ROBE86a]. Van Rijsbergen and others experimented with a version of F4 in which its components are individually weighted [VAN77], [HARP78], [VAN81]. In their experiments this **Eiq** formula gave results which were better than F4, but no theoretical justification has been found. Many researchers have also considered models which incorporate dependences between terms. In practical situations there is rarely enough relevance information to estimate the large numbers of parameters involved, and most experiments have had negative results.

2.3 Query modification

2.3.1 Sources of terms for query modification

There are a number of ways in which queries may be modified in the light of relevance information. Perhaps the most obvious source of additional terms is the user, but this is outside the scope of the present project. It is noticeable that the intermediaries' technique of thinking of other ways in which a query might be expressed does not come naturally to the majority of untrained searchers. Efthimiadis and Robertson [EFTH89] distinguish four other ways in which queries may be modified. The first does not add to or subtract from the query terms, but merely adjusts their weights [ROBE86b]. This method is not likely to be useful if the query contains only a few terms - up to four or five, say. The second method retains the query terms and adds additional terms from some source other than the relevant records, such as terms which are closely associated with the query terms in the collection as a whole [VAN81, SMEA83]. In this case the relevance information is only used in the assignment of weights to the selected terms. The third method is to use terms from relevant documents in addition to the query terms [SALT85]. Finally, query terms may be abandoned and replaced by terms from relevant documents.

End users of interactive systems tend to enter very short queries, so the method of using query terms only was not considered for the present project. The second method has not proved very successful in experiments [SMEA83], nor does it seem to fit well with the assumption in most of the weighting schemes used in probabilistic systems that terms occur independently. The third method, that of adding to the query terms from

relevant records, has the merit of resembling a technique which experienced searchers use. However, this type of query expansion also did not perform well in one set of experiments [SMEA83]. The final method seems to involve discarding information for no obvious reason, but is used in systems such as a few of the online catalogues where the user is given the option of seeing other records with the same subject heading or class mark.

In the experiments of Smeaton and Van Rijsbergen reported in [SMEA83], no method of query expansion was found to be beneficial. No method of modification was noticeably better than the unmodified queries and most were worse. On the whole, the more terms were used the worse the results. Adding random terms was no worse than most of the other techniques tried. The experiments were carried out on a small test collection, with pre-assessed queries and documents. The queries were long compared to those usually collected from interactive system use (2 - 13 terms, mean about 7). Despite these results, on the basis of the feasibility studies described in 1.6 we chose the third method for the present project - that of selecting expansion terms from the combined set of query terms and terms from relevant records.

2.3.2 Weighting and term selection

When query expansion using index terms from relevant records is done manually by searchers, they apply selection criteria. These would not be easy to formalize, but would usually involve linguistic and subject knowledge, and, in the case of professional searchers, some knowledge of the nature of the current database. Selection would also take account of results already obtained in the current session.

In automatic and semi-automatic query expansion in a probabilistic retrieval system, weights are normally assigned by the system. Various weighting schemes have been used. Several of the operational systems are known to use the Robertson/Sparck Jones F4' formula (above).

2.4 Query modification in end user retrieval systems

2.4.1 Term selection by the user

Several of the commercially available retrieval systems offer a type of "related record" facility, where a single index term from a displayed record may be selected by the user and used as a new search key. The BLCMP online catalogue offers shelf-order browsing using the shelfmark of a selected record as the starting point. The Dynix integrated library system (the installation referred to here is the one at the University of Stirling in September 1988) gives a "related works" option at the foot of a full record display. The user can choose (by means of a two stage menu selection process) any heading in any authority controlled field. When the heading has been chosen the system looks it up. If there is only one posting for the heading it displays "This is the only title containing ...", otherwise it shows brief author/title entries for any other records indexed under the chosen heading. This facility may be a little ponderous to use, but it certainly provides a way of doing some structured browsing. A serious disadvantage is that it is not usually possible to return to the previous set of records without rekeying the search. The INNOPAC online catalogue system allows both shelf marks and subject headings to be used as new search keys (Fig 2.1).

Fig 2.1 A screen from the INNOPAC online catalogue system

```
================================================================================
  You searched for the Words: health

TITLE : X-rays, health effects of common exams.


Related items may be found under SUBJECTS

  1  Tumors, Radiation-induced.
  2  Radiography, Medical -- Complications and sequelae.
  3  X-rays -- Toxicology.


Key a number or
S > SHOW items nearby on shelf        R > RETURN to Browsing
D > Re-DISPLAY record                 N > NEW search
                    Choose one (1-3,S,R,D,N)

================================================================================
```

The Silver Platter CD-ROM search software offers the same sort of facility in a rather different way. When a record is displayed the user can select terms from the displayed text for searching, by means of cursor movement and function keys. The chosen terms are looked up, and each one gives rise to a set of postings for later display or combination.

All these systems are completely passive, the initiative being entirely with the user. It is likely that a large proportion of users would not make the effort to learn how to make use of the feedback facilities. No evaluation results are known.

2.4.2 The CITE catalogue at the NLM

CITE is the outstanding example of a reference retrieval system which offers semi-automatic query expansion using terms from relevant records.

CITE is, or was, the public access catalogue for most of the National Library of Medicine's (NLM's) monograph collection (more than 500,000 titles). It allows search by subject, personal author, title, corporate author etc., series or call number. When a subject search has led to the display of records the user is asked to provide relevance information. If the user chooses a query expansion search CITE presents a list of index terms associated with the relevant records. The user is invited to select from and rank these terms, and may also add terms. The query expansion is thus of a user-aided, semi-automatic type (1.2).

The interaction proceeds as follows. CITE invites the user to "type your search question". CITE processes the terms and then displays a list of MeSH headings and textwords on which it proposes to search, giving each term a rank. At this stage the user has to "type the rank numbers of the search terms you want to use, in their order of importance, or type ALL". (If the user types "ALL", the system will continue the search with the terms and weights it has assigned). Weights are re-assigned,

using the rank order which the searcher has indicated, some type of "best match" search is performed, and CITE starts to display records. Each record is headed by a sequence number, which the user may have to remember or make a note of. After each screen the system asks whether the user wants to continue or not. When the user says no, CITE requests the user to "choose the items in which you are most interested" - again there is an "ALL" option. Following this choice, the user can see again the chosen records or "find items similar to the ones you chose". To find similar items, CITE uses the MeSH headings of the selected records, and then re-searches after asking the user to rank the terms and/or add further terms. If there are fewer than ten terms, CITE will also include call numbers from some of the selected records.

Since it uses automatic stemming, automatically maps users' query terms to MeSH (Medical Subject Headings), performs best match, ranked output searches and allows relevance-based query expansion, it can be seen that CITE is functionally a very rich system. Although it is specifically designed for databases of medical references (the stemming procedure is geared to medical terminology), there is no doubt that it could be adapted for use on other types of database. CITE has been one of the strongest influences on the Okapi systems. But from the above brief account of the interaction it seems likely that many catalogue users would not be able to make good use of a CITE-like system. The systems designed for the present project, which will be described in Chapter 3, were largely an attempt to produce something functionally like CITE but requiring a smaller amount of user involvement.

Although there have been many published articles and papers about CITE some of the material is rather repetitive [DOSZ79a], [DOSZ79b], [DOSZ83a], [DOSZ83b], [DOSZ86]. We do not know of any evaluation results, nor do we have details of the algorithms. It seems likely that it uses a Croft-Harper formula (2.2) for weighting the terms used in the initial search. We do not know how ranking by the user affects the weighting of query expansion terms.

2.4.3 The online catalogue at the Scott Polar Research Institute

This system accesses the machine readable portion of the Scott Polar Research Institute's catalogue. It is based on the Muscat system of Martin Porter and is very clearly described by Porter and Galpin in [PORT88]. In its more advanced form (there is also a kind of novice mode) it incorporated a semi-automatic query expansion function.

The catalogue records have no subject headings, and are indexed by title words and UDC numbers (and names). Records are presented singly to the user for assessment by means of the question "Relevant?". After some records have been chosen as relevant the user may request query expansion. The system displays a ranked list of title words and UDC numbers from relevant records for selection or rejection. A "match" command then results in a new search using the selected terms. Records previously seen are removed from those retrieved by the expanded search. The process may be iterated, contributing to a single list of chosen records until the user decides to quit or do a different search.

Porter and Galpin report that "it takes a good deal of learning in order that it may be used to its fullest extent", but that a number of researchers have done so. Results in use are said to be encouraging.

Functionally, this system is not unlike CITE. The Robertson/Sparck Jones

F4 formula (2.2) is used for the weighting of query and expansion terms. However, an *ad hoc* formula, apparently due to Porter, is used for ranking potential expansion terms for display to the user. This is

term weight = r/R - n/N

where the symbols have the same meanings as in 2.2.

2.4.4 *The INSTRUCT system at the University of Sheffield*

INSTRUCT (INteractive System for Teaching Retrieval Using Computational Techniques) is a text retrieval system developed as a teaching package for demonstrating a number of information retrieval techniques [HEND86a], [HEND86b], [WADE88]. It accesses a database of 26,000 title and abstract records from LISA (Library and Information Science Abstracts). It allows several varieties of query expansion using both user-supplied relevance judgments and system-derived term co-occurrence information. It performs best match searches on natural language queries, using automatic stemming and a large stoplist of some 300 words. As well as the query expansion options various other devices are incorporated in INSTRUCT, mainly intended to demonstrate different methods of improving recall. INSTRUCT is intended for the demonstration of information retrieval techniques, and perhaps also for testing algorithms for implementing these techniques. It is not an operational system for general users, and the user interaction side of the system is undeveloped. There are menu and command language versions.

One variety of query expansion in INSTRUCT is automatic, but it is based on a single retrieved document. For this "seed search" INSTRUCT extracts the 25 least frequent stems from the seed or pivot record and uses them in a new search. Following a seed search it is always possible to return to the original list. There is also semi-automatic or user assisted query expansion. This can display up to 20 terms extracted from relevant records for user selection. The procedure involves the user slightly less than CITE's similar option: INSTRUCT does not allow the user to influence the rank ordering, and thus the weight, of query expansion terms. F4' term weights are used (2.2). There has been some evaluation of INSTRUCT in use, described in [ELLI86], but this does not appear to have included the use of query expansion.