Relevance feedback and a fuzzy set of search terms in an information retrieval system

A system that uses relevance feedback to modify the user's query is described by **A F Smeaton**

This paper describes a retrieval strategy for a document retrieval system which incorporates relevance feedback to perform a modification of the user's initial query. At any given point in a user's search, the relevant documents found by the user up to that point are used to assign weights to each term in a set of search terms. These search term weights are known as relevance weights, and are in turn, used to assign a 'score' to each document in the collection, for ranking and subsequent presentation to the user. The relevant documents found in the user's search at any point are also used to determine the actual terms which are included in the set of search terms for relevance weighting. Each term in this set of search terms also has associated with it, a degree of membership of the search term set, so the set of search terms is in fact a fuzzy set, whose members are only partial members of the set. The relevant documents found in a user's search at any point are used to compute a relevance weight for each search term, to determine the terms to be included in the set of search terms, and also to compute the 'degree of membership' of each search term, to the set of search terms. This retrieval strategy has been tried out on a test collection of documents and queries, and the results of these experiments, and an analysis, are presented.

Keywords: information retrieval, relevance feedback, fuzzy sets, document retrieval

A document retrieval system (DRS) is an information system in which a collection of documents (e.g., papers, newspaper articles, bibliographic data) is stored for subsequent retrieval in response to a user's query. In some DRS the documents are stored as full text, in which case the retrieval can be a string matching operation between the user's query and the full text documents.

In other document retrieval systems, the documents may be represented as a set of index terms whose combined semantic meaning roughly reflects the information content of each document. When a user wishes to retrieve information from such a system, his information need is formulated as a query, which is usually an incomplete specification of his information need. This query may be a natural language statement and may be represented within the DRS as a set of index terms called the query terms. In order for the document retrieval system to be able to retrieve documents for the user in response to his query, a set of search terms is matched with the sets of index terms representing the documents in the collection, and a set or list of best matched documents is retrieved for the user to examine. At the start of the user's search, the set of search terms will usually be the set of query terms.

The set or list of documents retrieved for the user in response to his query could contain document(s) whose information content satisfies the information need of the user. Such documents are the documents *relevant* to the user's query, and the user may continue to examine the set or list of documents presented by the system, until he finds all the documents relevant to his query.

The operation where the set of search terms are matched with the sets of index terms representing the documents is done using one of several possible *matching functions,* and the overall organization of the user's entire search is called a *retrieval strategy.*

Much research has been done in recent years in designing and testing the effectiveness of retrieval strategies on test collections of documents. Some of these retrieval

Department of Computer Science, University College Dublin, Belfield, Dublin, 4, Ireland
Received 13 May 1983, revised 22 July 1983

strategies have been derived from complex mathematical models of the retrieval processes of a document retrieval system. In particular, a lot of research has been done into deriving some mathematical models of document retrieval, using probability theory[1,2,3].

Stochastic models of document retrieval can differ from each other in many ways. In particular, these models differ with respect to the assumptions made about the dependency relationship between the distributions of index terms of a document collection throughout the documents of that collection. From probability theory, a fundamental identity equation states that each index term is dependent on the full set of *all* other index terms in the collection. For a given term, however, some of these dependencies may not exist, or may be far too complex to estimate. For this reason, approximating expansions have been used to model the probability distributions of index terms throughout the document collection.

One approximating expansion that has been shown to be useful in deriving probability based mathematical models of retrieval, assumes that index terms are independently distributed over the documents. This is a strong assumption to make, but as we shall see later has led to mathematical models and retrieval strategies which have been quite useful.

Retrieval strategies derived from probability based models of retrieval have met with varying degrees of success in their effectiveness. In general, the retrieval strategies have not yielded significant improvements, as the theory would have suggested. It is generally accepted now that the major cause of this has been the inability to accurately estimate probabilities, given very small amounts of sample data. This problem has been described in more detail by van Rijsbergen *et al.*[4] and the situation has been summed up by Smeaton and van Rijsbergen[5]:

> ... unless some better estimation rules are found which can accurately estimate probabilities from small samples of relevant documents, then the theoretical advantages of using some retrieval strategies with sound mathematical backgrounds, may never materialise into significant improvements in retrieval effectiveness.

RELEVANCE FEEDBACK

One of the basic preconditions made when deriving mathematical models of document retrieval using probability theory is that there is a sample of relevant documents available *a priori,* from which estimates of the occurrences of search terms among *all* the relevant documents can be made. This sample of relevant documents may be provided by the user, or the retrieval strategy may incorporate *relevance feedback.*

In relevance feedback, a sample set of relevant documents is obtained by presenting a set of documents found by a simple and unsophisticated matching function to the user. This set of documents is called the *initial set.* The user can then judge these documents for relevance, and feed back to the document retrieval system which of the documents in the initial set (if any) are relevant to his query. The system can then use this relevance information to modify the remainder of the search, and present a second set of documents to the user. After the user has examined the second set of documents, the system can then use *all* the relevance information it has available (i.e., relevant documents from the initial set, and relevant documents from the second set presented to the user), to modify the remainder of his search. The whole process is iterative so that when relevant documents are found by the user in the search, they are added to a continuously increasing set of relevant documents, which is used by the system to refine the remainder of the user's search, until *all* the documents relevant to the user's query have been found. This whole iterative approach to document retrieval is called relevance feedback.

One of the first uses of relevance feedback as part of a retrieval strategy for a document retrieval system was by Rocchio[6], as part of the SMART project. Rocchio's work was done using a vector space model of document retrieval. In his work. Rocchio started a search with an initial user query, and used each iteration of the feedback loop to modify the query by adding the normalized vectors of the relevant documents found by the user in all iterations, to the original query. He also subtracted the normalized vectors of the documents judged by the user to be nonrelevant, over all feedback loop iterations, from the query. Thus Rocchio hoped to eventually 'home in' on the optimal query for a given user's search.

When using probability theory to derive mathematical models of document retrieval, a fundamental task is to estimate the probability of a search term occurring among the relevant documents, and to estimate the probability of the search term occurring among nonrelevant documents for each given search term. The relevant documents found at any point of a user's search, may be used as a sample of *all* the documents relevant to a given query.

In estimating the probability of a search term occurring in the nonrelevant documents, the documents judged by the user to be nonrelevant could be used. It is more useful, however, for the system to use the whole document collection as the sample of nonrelevant documents[1]. This is because the whole collection (although it will contain relevant documents) is a larger and more unbiased sample.

SEARCH STRATEGIES

One of the most commonly used retrieval mechanisms for obtaining an initial set of documents for presentation to the user is to weight the search terms by their inverse document frequency (IDF) weights. Each document is then given a score, computed as the sum of the IDF weights of the search terms which index that document. The collection is then ranked by the document scores, and the top ranked documents are presented to the user as a list for him to examine until he has found some documents relevant to his query.

The IDF weight of an individual search term is given by:

$$\text{IDF} = -\log \frac{n^*}{N} \qquad (1)$$

where N is the number of documents in the entire collection, and n is the number of documents indexed by the search term being weighted.

IDF weighting has been used as the retrieval mechanism for obtaining an initial set as part of an overall ret-

*n is assumed greater than zero

rieval strategy which incorporates relevance feedback, or it can form a retrieval strategy of its own, with no relevance feedback. IDF weighting uses no relevance information to modify the search, so once the initial ranking of the documents has been done all the computation work to implement this strategy is finished.

One retrieval strategy which does use relevance feedback to modify the remainder of the search is that proposed by Robertson and Sparck Jones[1]. In this strategy, the relevant documents identified by the user, possibly from an IDF ranking, are used to assign term weights called *relevance weights* to each of the search terms in the user's query. When some relevant documents have been found by the user, the search term relevance weights are computed. All those documents in the collection not yet seen by the user (i.e., the whole collection, excluding those in the initial set — called the residual collection) are then given new document scores, computed as the sum of the relevance weights of the search terms which index those documents. These new scores are used to rerank the residual collection, and the user examines this new ranking until he finds some more relevant document(s). These, in turn, are then fed back into the system, and new relevance weights can be computed, and the cycle can be repeated until all documents relevant to the user's query have been found.

The relevance weights for a search term, as proposed by Robertson and Sparck-Jones[6], are computed using a relevance weighting formula, which will be known as the binary independence weight (BIW). The BIW for a search term is implemented as:

$$\text{BIW} = \log \frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - R + r + 0.5)} \quad (2)$$

where N is the total number of documents in the collection, n is the total number of documents indexed by the search term, R is the number of relevant documents found so far, and r is the number of the relevant documents found so far, which are indexed by the search term. The 0.5s are included in the formula as a necessary heuristic probability estimation technique and the formula has got some drawbacks in that it tends to overestimate some of the probabilities[4]. Thus this formula can lead to inaccurate probability estimates when the amount of sampling data (i.e., the small and incomplete set of relevant documents available at any time during the user's search) is small. Nevertheless, the implementation of the BIW as given above will be used in this paper.

In all of the previous retrieval experiments in the IR literature using relevance feedback and relevance weighting of search terms, one of the most crucial factors has been neglected, and that is the point (or points) at which the relevance weights are recomputed and the residual collection is reranked. For example, in the paper by Robertson and Sparck-Jones[1], the user examines for relevance the documents of a fixed size initial set, and feeds back his judgements to the system. The retrieval strategy then uses this information to compute relevance weights and rerank the residual collection, but this is the only time in the overall retrieval strategy that this is done. The sum total of the relevance information used by the retrieval strategy to modify the user's search is the user's relevance judgements of the initial set.

It is quite possible than an improvement in retrieval effectiveness could be obtained if the relevance feedback loop was iterated more often. An obvious candiate retrieval strategy would be for the user to examine the initial set until he found the first relevant document. The system could then compute search term relevance weights and rerank the residual collection *every time* the user finds another relevant document. This increased amount of relevance feedback looping could help to home in on a set of search term relevance weights which would improve overall retrieval effectiveness.

One possible drawback with having an increased amount of relevance feedback looping is the increased requirement on computing resources to compute relevance weights (mostly CPU time, but also memory space), score documents of the residual collection, and rerank those documents for presentation to the user. A possible solution to this, which still retains the advantage of increased relevance feedback looping, is to reiterate the feedback loop every time the user finds *two,* or even *three,* relevant documents. Thus each residual collection reranking would be examined by the user until, say, two relevant documents have been found, and at that point all the documents relevant to the user's query could be used to compute the relevance weights, and the residual collection could be reranked. This would effectively halve the amount of computation resources needed to implement the overall retrieval strategy. The effects of such variances on the amount of feedback looping will be investigated experimentally later in the paper.

There is one other point concerning the amount of relevance feedback iteration which may affect the effectiveness of the retrieval strategy. Harper (1980) has shown that there is a threshold number of relevant documents (five in fact) which must be found by the user from the initial set before reliable probability estimates can be made for one of his proposed relevance weighting formulae. The possible existence of such a threshold number of relevant documents from the initial sample, which must be found by the user before BIW relevance weighting can be effective, shall also be investigated experimentally later on in the paper.

A FUZZY SET OF SEARCH TERMS

In all of the retrieval strategies using relevance feedback described in the previous section, relevance information (i.e., the relevant documents found at any given point in the user's search) has been used to modify the weights assigned to search terms, by using relevance weighting formula. In these strategies, the search term set, has been the set of original query terms. Smeaton and van Rijsbergen[5] have tried to use the relevance information to *modify* the set of search terms by term addition or term deletion. In that paper, various query modification strategies were tried on a test collection of documents, and the effects of this query modification on the overall retrieval strategy were analysed.

The query modification strategies presented by Smeaton and van Rijsbergen[5] have had a detrimental effect on the overall retrieval effectiveness, and there are various reasons put forward for this surprising result. One reason is that the query modification strategies could have been implemented at too early a point in the user's overall retrieval strategy.

In all of the retrieval strategies mentioned so far, each of the search terms has been treated as equally important

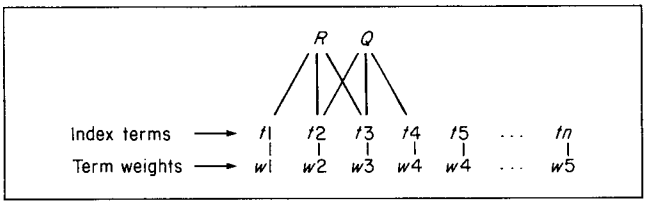

Figure 1. Overlap of terms indexing R and Q

to the user's search. An improvement in retrieval effectiveness copuld be obtained by assigning a secondary weight, besides the relevance weight, to each of the search terms, which would indicate that term's relative degree of importance to the user's search. These 'degree of importance' weights for the search terms, and the relevance weights, could be combined in some way to score and rerank the residual document collection, for presentation to the user.

Having degree of importance weights for the search terms would effectively make the search term set a partial membership, or *fuzzy* set. Here numerical values indicating how much each search term is a member of the search term set, could be obtained directly from the user, or could be computed automatically by the system.

The concept of degree of membership of an item within a set is not new to information retrieval research. Fuzzy sets have been used by Bookstein[7] and others in trying to model retrieval, but to date no great insights have been obtained using this approach.

In order to describe the proposed modification to the retrieval strategy in more detail, let us consider the case where one relevant document has been found in the user's search so far, and there are more relevant documents in the collection still to be found. This represents a midpoint in the user's overall search where some (i.e., one), but not all, of the documents relevant to the user's query have been found. Let us assume that, instead of recomputing BIW relevance weights for each search term every time the strategy reiterates around the feedback loop, we use IDF weights as the search term weights to score the documents of the residual collection. This is done to reduce the computation resources needed to implement the strategy because if IDF weights are used then for each relevance feedback iteration the residual document collection scores need only be incremented by the search term weights of the last relevant document to have been found, whereas if BIW relevance weights are used, the set of residual document scores has to be totally recomputed. Later on, however, I shall present results obtained when BIW relevance weights are used instead of IDF weights.

The collection of documents that the user is searching through has associated with it, a set of terms, say $t_1 \ldots t_n$. Both the user's query (Q) and the single relevant document identified so far (R) are indexed by some of this set of index terms. There will be some overlap between the sets of terms indexing Q and those indexing R (since R was found by IDF weighting the terms of Q). Each of the index terms has an IDF term weight associated with it, say w. This situation is shown in Figure 1.

What is needed from a document retrieval strategy is a complete reranking of the residual document collection by rescoring all these documents, and presenting the top-scored documents to the user. Let us examine the case for the 'scoring' of one document only, say document n (D_n), and let us further assume that D_n is indexed by:

- *a term indexing R* but not Q (t1)
- a term indexing R and Q (t3)
- a term indexing Q but not R (t4)
- a term indexing neither R nor Q (t5)

These four cases of index terms are mutually exclusive and cover all possible combinations with respect to the index term lists of R and Q. Figure 1 can now be updated as in Figure 2.

In relevance weighting as used in the BIW retrieval strategy, the 'score' given to D_n *would be* $(w3 + w4)$, i.e., the sum of the weights of the terms assigned to both Q and D_n (index terms 3 and 4). This is a function of both Q and D_n, which will define as $M(Q, D_n)$, and we shall call this the document weighting function WF0.

In order to avail of the added information provided by the known relevant document, R, we can define a new document weighting function called WF1 in which the score of the document D_n would be:

$$\begin{aligned} \mathrm{WF1}(D_n) &= M(D_n, Q) + M(D_n, R) \\ &= w1 + 2.w3 + w4 \end{aligned} \tag{3}$$

This can interpreted as matching D_n with the terms of the original query, and also with the terms of the single relevant document found so far. This is a form of query expansion where the new set of search terms is the set of original query terms plus the terms of the single relevant document found so far. For an index term which indexes both Q and R, e.g., t3, this term would effectively be included twice in the new set of search terms, i.e., its degree of membership of the search term set is twice that of the other terms.

All documents of the residual collection are scored in the same way as D_n and the ranking of the top-scored documents is presented to the user until another relevant document is found. When more than one relevant document has been fed back to the system in total, the document weighting function WF1 can be generalized so that when a document is being rescored, its terms are matched with the terms of the original query and also with the terms of *all* the relevant documents identified in the search so far. Formally, this generalized document weighting function can be defined as WF1'

$$\mathrm{WF1}'(D_n) = M(D_n, Q) + \sum_{i=1}^{x} M(D_n, R_i) \tag{4}$$

where there are x relevant documents found by the user so far, i.e., $R_1 \ldots R_x$. This overall retrieval strategy is similar to that presented by Rocchio[6].

One of the obvious disadvantages associated with the document weighting function WF1' is that although it uses

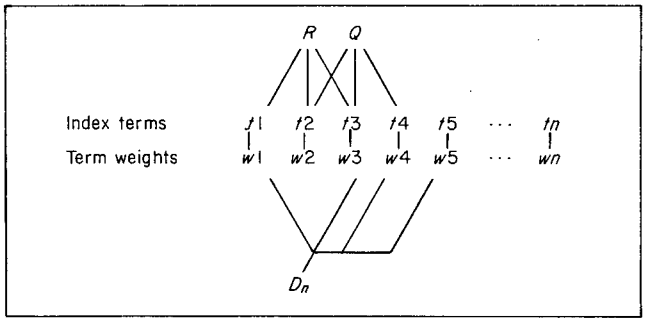

Figure 2. Update of terms indexing R and Q

relevance information provided by known relevant documents to modify the remainder of the user's search, this relevance information may tend to 'swamp' the original query, which, after all, is a direct expression by the user of his information need.

What is needed is a way of measuring how much of a contribution each relevant document should make to the remainder of the search and consequently, how important the terms of each relevant document should be, to the overall retrieval. This can be done by measuring the similarity between the original query, Q, and each relevant document R_i, and grading the contribution of the terms of each relevant document to the remainder of the search accordingly. The similarity measures between Q and all the R_is can be interpreted as a computation of the degrees of membership of the terms of the R_is, to the set of search terms.

The whole idea can be captured formally as a new document weighting formula WF2. Suppose there is only 1 relevant document, R, identified in the search so far. WF2 can be defined as:

$$\begin{aligned} \mathrm{WF2}(D_n) &= M(D_n, Q) + k.\mathrm{M}(D_n\, R) \\ &= k.w1 + (1 + k).w3 + w4 \end{aligned} \tag{5}$$

where k is a measure of the similarity between R and Q, defined as $S(R, Q)$. The 'degree membership' of the set of search terms for term 3 is $(1 + k)$, while for term 4, say, it is 1, and for term 1 it is k. When there are x relevant documents identified, WF2 can be generalized to:

$$\mathrm{WF2}'(D_n) = M(D_n, Q) + \sum_{i=1}^{x} k_i M(D_n, R_i) \tag{6}$$

where $k_i = S(R_i, Q)$

There are several ways of measuring $S(R_i, Q)$, the similarity between a document and a query. Most formulae which measure such a similarity use 3 parameters:

- a — the number of terms in the query
- b — the number of terms in the document
- c — the number of terms in common

Some examples of such formulae are the Cosine measure, the Dice measure and the Ivie measure[8], which are defined as:

$$\text{Cosine} = \frac{c^2}{a.b} \tag{7}$$

$$\text{Dice} = \frac{2.c}{(a + b)} \tag{8}$$

$$\textit{Ivie} = \frac{c}{a.b} \tag{9}$$

Any of these measures could be used to compute the degrees of membership of the terms of the relevant documents found by the user in his search so far. Before I describe the experimental results obtained with the proposed retrieval strategies, i.e., varying the amount of relevance feedback looping *and* including a fuzzy set of search terms, I will first outline the test collection of documents and the evaluation procedure to be used in the experiments.

INTRODUCTION TO EXPERIMENTS

The overall retrieval strategies outlined in the previous sections, and the suggested possible modifications which could improve retrieval effectiveness, were implemented on a test collection of documents, queries and relevance assessments. The test collection used was the NPL test collection used previously in experiments reported in Robertson *et al.*[8] and Smeaton and van Rijsbergen[5]. It consists of 11 429 documents composed of 7491 unique index terms, and 93 test queries, each with between 1 and 84 known relevant documents. Table 1 gives a statistical summary of the collection.

The reason that this collection of documents was chosen were twofold. Firstly, work related to that reported in this paper had been done on this particular collection, and, if the same evaluation method was used, direct comparisons could be made with work in Smeaton and van Rijsbergen[5] and others. Secondly, the collection is of medium size and exhibits characteristics fairly representative of a real life document collection. Details of how the collection was formed can be found in Robertson *et al.*[8].

The method I have chosen to evaluate the performances of different retrieval strategies is similar to that used previously by Robertson *et al.*[8], Harper[3], Harper and van Rijsbergen[9], Smeaton and van Rijsbergen[5], etc. For each retrieval strategy, a list of the positions of *all* the relevant documents in all the queries is found. For each query, a set of precision–recall pairs of figures is generated where:

$$\text{Precision} = \frac{\text{No. documents relevant and retrieved}}{\text{No. documents retrieved}} \tag{10}$$

$$\text{Recall} = \frac{\text{No. documents relevant and retrieved}}{\text{No. documents relevant}} \tag{11}$$

The sets of precision–recall pairs for the queries are then pessimistically interpolated and averaged to give a set of precision values at the standard recall values of 10%, 20%, ... 100%. The particular averaging technique used is called recall cutoff evaluation and is described in more detail in Harper and van Rijsbergen[9].

There is one point that should be mentioned concerning recall cutoff evaluation and that is that the high recall figures are somewhat useless and should generally be ignored. This is because the set of precision values is for the rank positions of *all* the relevant documents in all the queries of the test collection. Some of these relevant documents will have very low rank positions. For example, some relevant documents may not be indexed by *any* of the search terms; hence they will have a document score of zero, and will be retrieved at the *last* rank position (although query modification may reduce this problem somewhat). It is most likely that a user will perservere in his search for such documents at the last rank position,

Table 1. Statistical summary of documents

	Documents	Terms	Queries	Relevance assessments
	11429	7491	93	93
Maximum length	105	2511	13	84
Minimum length	1	1	2	1
Average length	19.96	30.45	7.14	22.40

and try to get all the documents relevant to his query, without some form of query modification. Thus, the high recall values are not very indicative of the true performance of the retrieval strategies being evaluated. Because of this, comparisons between sets of precision figures will mostly be made with respect to the high precision areas.

The system of programs to implement the different retrieval strategies was written in the programming language C, under the EUNICE operating system, and the machine used was a Digital Equipment Corp. VAX–11/750.

INITIAL EXPERIMENTAL RESULTS

In this section I shall present the experimental results obtained by varying the amount of iteration of the relevance feedback loop of a document retrieval strategy. One of the candidate strategies outlined was to have the user find one document relevant to his query from an initial set generated by an IDF weighting of the original query terms. The BIW search term relevance weighting formula could then be used to assign new weights to all the search terms (i.e., the query terms) and a residual collection reranking done *every* time the user finds one more relevant document. I shall give this overall retrieval strategy the mnemonic I1B1 (IDF weighting to get the initial set, user finds *one* relevant document from this sample, BIW weighting used to rerank every time *one* relevant document is found.

The experimental results obtained with the I1B1 retrieval strategy will form the yardstick results against which subsequent results to be preseented will be compared. These results are given below:

I1B1:	55.9	47.6	39.7	33.4	27.2	21.2	16.1	11.8	7.9	3.9

In order to test the effects of having a larger set of relevant documents available before relevance weighting, the next set of results will be for a retrieval strategy which requires that there be three (or five) relevant documents from the initial sample, before re-ranking (I3B1, I5B1). If there are less than three (or five) relevant to the user's query, then no relevance weighting of search terms will be done at all, as the user's information need will have been satisfied by the relevant documents from the initial set.

I3B1:	56.0	47.8	39.7	33.3	27.0	20.7	15.9	11.7	7.9	3.8
I5B1:	55.7	47.0	38.9	32.9	26.7	20.4	15.7	11.6	7.5	3.7

These results show that there does seem to be a threshold number of relevant documents from the initial sample which will yield an improvement in overall retrieval effectiveness, since the I3B1 results are better than either the I1B1 or I5B1 results. Nevertheless, the margin of improvement is quite small.

There is one distinct advantage in requiring that there be a larger number of relevant documents found by the user from the initial sample, and that is that there will be a saving on the computation resources needed the more relevant documents that must be found from the initial set before relevance weighting is used. This saving is obtained because there will be fewer sets of relevance weights to compute, fewer scorings of the residual document collection to be done, and fewer rerankings to be completed for the user.

The next point to be dealth with concerning the amount of relevance feedback looping is whether or not reiteration of the feedback loop less often, will have a detrimental effect on overall retrieval effectiveness. In the following sets of results, the feedback loop will be reiterated every time two or three, relevant documents have been found by the user.

I2B2:	56.7	47.8	40.4	33.4	27.1	20.8	15.8	11.5	7.7	3.8
I3B3:	55.9	47.3	39.8	33.2	27.1	20.6	16.3	12.1	7.8	3.8

These results, when compared with the figures for I1B1, show that in the important high precision region of the figures, reiterating the feedback loop less often has very little effect on the overall results. In fact the figures for I2B2 are *better* than those for I1B1 in the high precision region. This could be because in I2B2, two relevant documents are found from the initial sample before relevance weighting is used, and, as the earlier results have shown, there does seem to be a threshold number of relevant documents from the initial sample which will yield better overall retrieval performance figures.

Nevertheless, the above results show that, if an implementation of the I1B1 retrieval strategy proves to be too expensive in terms of computer resources needed, then a realistic alternative which will not have a significant detrimental effect on retrieval performance would be to reiterate the feedback loop less often.

The final set of experiments illustrates how much improvement in retrieval effectiveness can be obtained by reiterating the relevance feedback loop. To discover this, the next set of results will be for retrieval strategies which have no relevance feedback to the retrieval strategy once the initial sample has been used to find three or five relevant documents, and BIW relevance weighting is used to rerank the residual collection.

I3B0:	55.6	47.2	39.3	33.0	25.6	19.6	15.3	11.4	7.2	3.5
I5B0:	55.3	45.7	38.7	31.9	25.7	19.9	15.3	11.6	7.4	3.5

The above results can be compared with the I3B1 and I5B1 results. Such a comparison shows that reiterating the feedback loop more often does, as expected, improve overall retrieval effectiveness (I3B1 better than I3B0, and I5B1 better than I5B0). The surprising thing about these results is that the amount of improvement is remarkably small.

The amount of improvement in retrieval effectiveness obtained by reiterating the feedback loop of the user's search more often is so small that investing the necessary extra computer resources needed to perform the reiteration of the feedback loop in an operation retrieval system may not be worthwhile. To illustrate this point, the graph in Figure 3 shows some of the results obtained for some of the retrieval strategies as presented in this section. The point to note about this graph is that all the results are very close together, and no set of figures is significantly better than any other.

One of the reasons the margins of improvement obtained so far have been so small could be that the evaluation figures presented for each retrieval strategy are for the complete document collection ranking. Since the improvements in retrieval effectiveness that are being obtained to date are in the second or subsequent iteration of the relevance feedback loop, these improvements may not show up as prodominently as if the evaluation had been done solely on the residual collection ranking. In other words, the relevant documents found from the initial sample are at the top of the overall collection ranking, and they must tend to 'swamp' the overall rank position of the

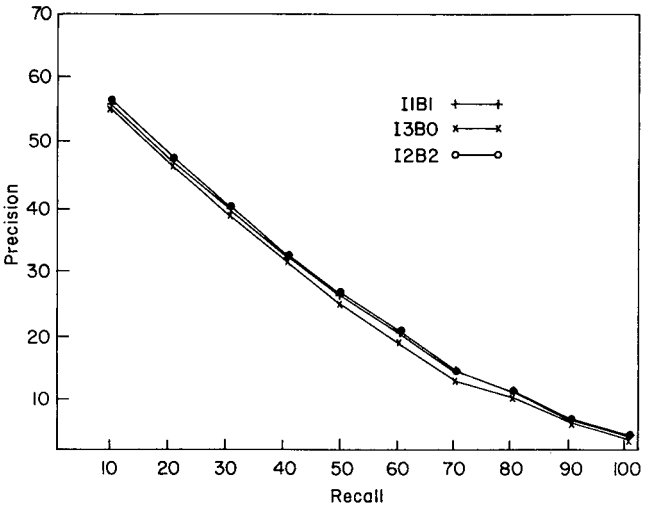

Figure 3.

relevant documents further down the collection ranking, where the improvements in retrieval effectiveness are to be found. This point should be considered when comparing precision–recall figures throughout this paper.

FURTHER EXPERIMENTAL RESULTS

In order to test the retrieval effectiveness of a retrieval strategy which uses relevance information to modify the set of search terms into a fuzzy set, as described earlier, the points at which the relevance feedback loop is to be reiterated must be determined.

As the basis of the retrieval strategies to be dealth with in this section, I shall use a retrieval strategy which includes in the fuzzy set of search terms all terms from all relevant documents found by the user. Some of these search terms will have a degree of membership of the search terms set, which is twice, three times, or even more than other terms. Such terms will have occurred in more than one relevant document or in the original query. The individual search term weights will be the IDF term weights and the relevance feedback loop will be reiterated *every* time a relevant document is found by the user. This corresponds to scoring the documents of the residual document collection by the WF1′ document weighting function, defined earlier as (4). I will call this overall retrieval strategy I1I1F (F indicating a fuzzy set of search terms) and present precision–recall figures for this strategy first.

The second sets of results to be presented will be for a retrieval strategy in which the documents of the residual collection are scored by the WF2′ document weighting function (6), i.e., where the search terms' degrees of membership are normalized by the similarity between the query (Q) and the relevant documents (R_is) as computed using the Cosine, Ivie or Dice similarity measure. The individual search term weights are, once again, the IDF term weights. These retrieval strategies will be called I1I1F(C), I1I1F(I) and I1I1F(D), respectively (see Table 2).

There are several points to notice about these sets of results. The first point I would like to draw attention to is the fact that all the strategies where the search terms' degrees of membership are normalized by the Q,R_i similarity measures, e.g., I1I1F(C), yield much better results than when the search terms' degrees of membership are not normalized, i.e., I1I1F. This shows that normalizing the contribution of the terms of relevant documents to the remainder of the search by using a similarity measure between query and relevant document works very well in practice.

Among the three I1I1F strategies, I1I1F(C) (using the Cosine similarity measure) yields the best set of figures for retrieval effectiveness. Because of this, and also because of the fact that the Cosine similarity measure has become the 'standard' similarity measure in experimental information retrieval[10], the Cosine measure will be used to compute the similarity between Q and the R_is in the next set of experiments.

The most surprising result from the above figures, however, is that the I1I1F(C) figures are noticably *better* than those from the I1B1 retrieval strategy. This means that incorporating normalized (by $S(Q,R_i)$) terms from all relevant documents in the user's search so far, into a fuzzy set of search terms each weighted by an IDF term weight will yield better retrieval effectiveness than the relevance weighting of the original query terms, by BIW weighting. This fact alone suggests that if BIW relevance weights were used instead of IDF weights in the I1I1F(C) retrieval strategy once some relevant documents had been found (call such a strategy I1B1F(C)), even further improvements in retrieval effectiveness could be obtained. Such a retrieval strategy would work as follows:.

When one relevant document from an initial sample has been identified by the user, the Cosine similarity between this relevant document (R) and the query (Q), is computed giving the value k. The set of search terms now becomes a fuzzy set consisting of the original query terms (degrees of membership 1) plus the terms indexing R (degrees of membership k). Some index terms will have degrees of membership k (terms in R, not in Q), others will have degrees of membership 1 (terms in Q, not in R), and others will have degrees of membership $(1 + k)$ (terms in R and in Q). The search terms are each given a BIW relevance weight, and the documents in the residual document collection are rescored, ranked, and presented to the user in that order. An individual document score is computed as the sum of the product weights of the search terms indexing that document, each product weight being the product of the BIW relevance weight and that term's degree of membership of the search term set.

Such a retrieval strategy as outlined above would demand a large amount of computing resources since the residual document collection scores would have to be

Table 2. Results with different retrieval strategies

I1I1F	I1I1F(C)	I1I1F(I)	I1I1F(D)
53.0	57.1	55.1	57.0
40.5	48.6	47.8	46.7
31.6	41.0	41.0	39.8
26.2	34.6	34.1	34.0
18.9	27.5	27.2	26.6
14.2	21.5	20.8	19.8
11.4	16.7	16.3	15.1
8.5	12.7	12.7	11.4
5.7	8.7	8.7	7.9
3.2	4.5	4.4	4.3

recomputed every time the feedback loop was to be reiterated. Such a demand on resources is not so severe in the I1I1F(C) retrieval strategy since the residual document collection scores do not have to be entirely recomputed (the new document scores are the scores from the previous iteration, incremented to include the scores of the extra search terms from the last relevant document to have been found).

To ease this demand on computer resources, I will also include the results of the retrieval strategy incorporating a fuzzy set of search terms and relevance weighting, but the feedback loop to be reiterated every time two relevant documents are found from each residual collection reranking (I2B2F(C)) (see Table 3).

These results do, as expected, yield a noticable improvement over the results for the I1I1F(C) strategy. A direct comparison between the results of the I1B1 and I1B1F(C) retrieval strategies show that the I1B1F(C) strategy yields a significant improvement (about 6%) over I1B1. This improvement can be seen from the graph in Figure 4. Thus relevance weighting each search term of a fuzzy set of search terms yields the best retrieval effectiveness obtained of all the results presented in this paper. Reiterating the feedback loop of such a retrieval strategy every time two relevant documents are found degrades retrieval effectiveness only marginally but saves a considerable amount of computer resources needed.

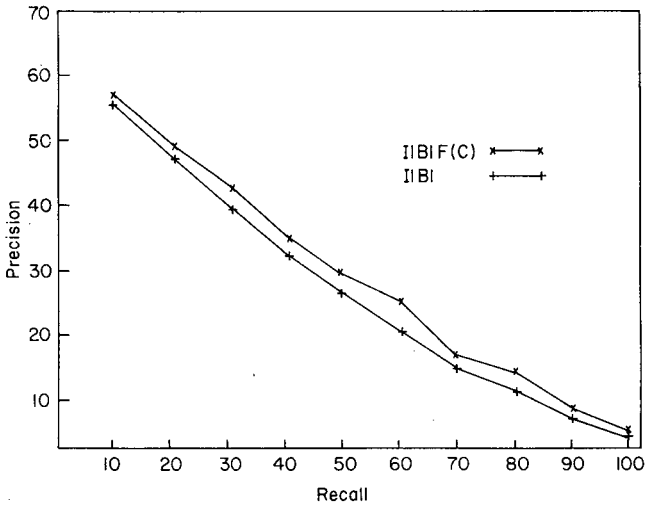

Figure 4.

CONCLUSIONS

In this paper I have presented a series of experiments in which the amount of relevance feedback looping has been modified to test the effectiveness of such modifications on overall retrieval. I have also described a retrieval strategy which uses relevance feedback from the user to compute a fuzzy set of search terms composed of the query terms plus all the terms from all known relevant documents. Known relevant documents are also used to estimate a relevance weight for each of the search terms in the proposed retrieval strategy, and the residual document collection is scored, reranked and presented to the user. This retrieval strategy has yielded improvements in retrieval effectiveness over the more conventional relevance weighting of the original query terms.

One of the primary considerations in the derivation of the new retrieval strategy, as presented in this paper, has been the cost effectiveness of implementing the retrieval strategy. Such a retrieval strategy requires a considerable amount of computer power to compute new document scores when these are many search terms and also requires computer resources to estimate relevance weights and rerank the residual collection.

These considerations have motivated some variations of the new retrieval strategy in which more than one relevant document has to be found from each collection reranking, before reiterating the relevance feedback loop. The results of such experiments have shown that such a modification of the retrieval strategy does not cause noticable detrimental effects in the overall retrieval effectiveness, but results in a saving on the computer resources needed to implement the retrieval strategy.

The most disappointing aspect of all the results presented in this paper has been the fact that a retrieval strategy which uses a fuzzy set of search terms has not yielded any great improvements in retrieval effectiveness over a retrieval strategy which uses relevance weighting of the original query terms. Nevertheless, using a fuzzy set of search terms yields better retrieval effectiveness than using the original query terms as search terms, even if it is considerably more expensive.

In conclusion I would like to say that the experiments reported in this paper will be carried out on the other test collections of documents to see if the overall results can be generalized. Such results were not included in this paper for reasons of space; furthermore the inclusion of many experimental results would possibly move the emphasis of the paper onto a comparison between two test collections.

Table 3. Further retrieval strategies

I1B1F(C)	I2B2F(C)
57.7	56.9
49.6	49.3
43.4	43.2
35.8	36.0
30.0	30.5
24.9	24.0
18.4	17.9
14.5	14.1
9.4	9.2
4.8	4.7

ACKNOWLEDGEMENTS

I would like to thank Keith van Rijsbergen for his helpful comments on earlier drafts of this paper.

REFERENCES

1 **Robertson, S E and Sparck Jones, K** 'Relevance weighting of search terms' *J ASIS* Vol 27 (1976) pp 129–146

2 **van Rijsbergen, C J** 'A theoretical basis for the use of co-occurrence data in information retrieval' *J Doc.* Vol 33 (1977) pp 106–119
3 **Harper, D J** *Relevance feedback in document retrieval systems: An evaluation of probabilistic strategies* PhD thesis, Cambridge University (1980)
4 **van Rijsbergen, C J, Harper, D J and Porter, M F** 'The selection of good search terms' *Inf. Proc. Management* Vol 17 (1981) pp 77–91
5 **Smeaton, A F and van Rijsbergen, C J** 'The retrieval effects of query expansion on a feedback document retrieval system' *The Computer J.* Vol 26 (1983) pp 239–246
6 **Rocchio, J J** 'Relevance feedback in information retrieval' *The SMART retrieval system* (G. Salton ed.) Prentice hall, Englewood Cliffs, New Jersey (1971) pp 313–323
7 **Bookstein, A** 'Fuzzy requests, an approach to weighted boolean searches' *J ASIS* Vol 31 (1980) pp 240–247
8 **Robertson, S E, van Rijsbergen, C J and Porter, M F** 'Probabilistic models of indexing and searching' *Information retrieval research* (R N Oddy, S E Robertson, C J van Rijsbergen and P W Williams eds.) pp 35–56, Butterworths, London (1981)
9 **Harper, D J and van Rijsbergen, C J** 'An evaluation of feedback in document retrieval using co-occurrence data' *J. Doc.* Vol 34 (1978) pp 189–216
10 **Salton, G** (Ed.) *The SMART retrieval system* Prentice Hall, Englewood Cliffs, New Jersey (1971)