Improving Retrieval Performance by Relevance Feedback*

Gerard Salton
Chris Buckley

88-898
February 1988

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This study was supported in part by the National Science Foundation under grant IST 83-16166 and IRI 87-02

Improving Retrieval Performance by Relevance Feedback

Gerard Salton and Chris Buckley*

Abstract

Relevance feedback is an automatic process, introduced over 20 years ago, designed to produce improved query formulations following an initial retrieval operation. The principal relevance feedback methods described over the years are examined briefly, and evaluation data are included to demonstrate the effectiveness of the various methods. Prescriptions are given for conducting text retrieval operations iteratively using relevance feedback.

*Department of Computer Science, Cornell University, Ithaca, NY 14853.
This study was supported in part by the National Science Foundation under grant IST 83-16166 and IRI 87-02735.

1. Introduction to Relevance Feedback

It is well known that the original query formulation process is not transparent to most information system users. In particular, without detailed knowledge of the collection make-up, and of the retrieval environment, most users find it difficult to formulate information queries that are well designed for retrieval purposes. This suggests that the first retrieval operation should be conducted with a tentative, initial query formulation, and should be treated as a trial run only, designed to retrieve a few useful items from a given collection. These initially retrieved items could then be examined for relevance, and new improved query formulations could be constructed in the hope of retrieving additional useful items during subsequent search operations.

Conventionally, the query formulation, or reformulation process is a manual, or rather an intellectual task. The *relevance feedback* process, introduced in the mid-1960s is a controlled, automatic process for query reformulation, that is easy to use and can prove unusually effective. The main idea consists in choosing important terms, or expressions, attached to certain previously retrieved documents that have been identified as relevant by the users, and of enhancing the importance of these terms in a new query formulation. Analogously, terms included in previously retrieved nonrelevant documents could be deemphasized in any future query formulation. The effect of such a query alteration process is to "move" the query in the direction of the relevant items and away from the nonrelevant ones, in the expectation of retrieving more wanted and fewer nonwanted items in a later search.

The relevance feedback procedure exhibits the following main advantages:

- It shields the user from the details of the query formulation process, and permits the construction of useful search statements without intimate knowledge of collection make-up and search environment.
- It breaks down the search operation into a sequence of small search steps, designed to approach the wanted subject area gradually.

- It provides a controlled query alteration process designed to emphasize some terms and to deemphasize others, as required in particular search environments.

The original relevance feedback process was designed to be used with *vector queries,* that is, query statements consisting of sets of possibly weighted search terms used without Boolean operators. [1-5] A particular search expression might then be written as

$$Q_o = (q_1, q_2, ..., q_t) \tag{1}$$

where q_i represents the weight of term i in query Q_o. The term weights are often restricted to the range from 0 to 1, where 0 represents a term that is absent from the vector, and 1 represents a fully weighted term. A term might be a concept chosen from a controlled vocabulary, or a word or phrase included in a natural language statement of user needs, or a thesaurus entry representing a set of synonymous terms.

Given a query vector of the type shown in (1), the relevance feedback process generates a new vector

$$Q' = (q'_1, q'_2, ..., q'_t) \tag{2}$$

where the q'_i represent altered term weight assignments for the t index terms. New terms are introduced by assigning a positive weight to terms with an initial weight of 0, and old terms are deleted by reducing to 0 the weights of terms that were initially positive. The feedback process can be represented graphically as a migration of the query vector from one area to another in the t-dimensional space defined by the t terms that are assignable to the information items.

Initially, the relevance feedback implementations were designed for queries and documents in vector form. More recently, relevance feedback methods have been applied also to Boolean query formulations. In that case, the process generates term conjuncts such as (Term i *and* Term j) or (Term i *and* Term j *and* Term k) that are derived from previously retrieved relevant documents. These conjuncts are then incorporated in the revised query formulations. [6-9] The application of relevance feedback methods in Boolean query environments is not further discussed in this note.

Many descriptions of the relevance feedback process are found in the literature. With the exception of some special-purpose applications [10], the method has, however, never been applied on a large scale in actual operational retrieval environments. Some recent proposals, originating in the computer science community, do suggest that a relevance feedback system should form the basis for the implementation of modern text retrieval operations in parallel processing environments. [11-12] It is possible that the time for a practical utilization of relevance feedback operations is now finally at hand.

A study of the previously mentioned parallel processing application reveals that the relevance feedback process is easily implemented by using windowing and information display techniques to establish communications between system and users. In particular, ranked lists of retrieved documents can be graphically displayed for the user, and screen pointers can be used to designate certain listed items as relevant to the user's needs. These relevance indications are then further used by the system to construct modified feedback queries. Previous evaluations of feedback procedures have made it clear that some feedback methods are much more effective than others. Indeed, a poorly conceived arbitrary query reformulation can easily produce a deterioration in retrieval effectiveness rather than an improvement. [13]

The current note is thus designed to specify useful relevance feedback procedures, and to determine the amount of improvement obtainable with one feedback iteration in particular cases.

2. Basic Feedback Procedures

A) Vector Processing Methods

In a vector processing environment both the stored information items D and the requests for information Q can be represented as t-dimensional vectors of the form D = $(d_1, d_2, \ldots, d_t)$ and Q = $(q_1, q_2, \ldots, q_t)$. In each case, d_i and q_i represent the weight of term i in D and Q, respectively. A typical query-document similarity measure can then be computed as the inner product between corresponding vectors, that is

$$Sim(D,Q) = \sum_{k=1}^{t} d_i \cdot q_i \qquad (3)$$

It is known that in a retrieval environment that uses inner product computations to assess the similarity between query and document vectors, the best query leading to the retrieval of n particular items (the relevant ones) from a collection of N documents is of the form [1,2]

$$Q_{opt} = \frac{1}{n} \sum_{\substack{relevant \\ items}} \frac{D_i}{|D_i|} - \frac{1}{N-n} \sum_{\substack{non \\ relevant \\ items}} \frac{D_i}{|D_i|} \tag{4}$$

The D_i used in (4) represent document vectors, and $|D_i|$ is the corresponding Euclidian vector length.

The formula of expression (4) cannot be used in practice as an initial query formulation, because the set of n relevant documents is of course not known in advance of the search operation. Expression (4) can however help in generating a feedback query after relevance assessments are available for certain items previously retrieved in answer to a search request. In that case, the sum of *all* normalized relevant or nonrelevant documents used in (4) is replaced by the sum of the *known* relevant or nonrelevant items. In addition, experience shows that the original query terms should be preserved in a new feedback formulation. An effective feedback query following the retrieval of n_1 relevant and n_2 nonrelevant items can then be formulated as

$$Q_1 = Q_o + \frac{1}{n_i} \sum_{\substack{known \\ relevant}} \frac{D_i}{|D_i|} - \frac{1}{n_2} \sum_{\substack{known \\ non- \\ relevant}} \frac{D_i}{|D_i|} , \tag{5}$$

where Q_o and Q_1 represent the initial and first iteration queries, and the summation is now taken over the known relevant and nonrelevant documents. More generally, the following query formulation can be used for suitable values of the multipliers α, β, and γ.

$$Q_{i+1} = \alpha\, Q_i + \beta \sum_{rel} \frac{D_i}{|D_i|} - \gamma \sum_{nonrel} \frac{D_i}{|D_i|} \tag{6}$$

In expressions (4) to (6), normalized term weights are used whose values are restricted to the range from 0 to 1. When larger weights greater than 1 can be accommodated, unnormalized weights are also usable.

The relevance feedback operation is illustrated in Fig. 1 for the two-dimensional case where vectors carry only two components. The items used in the example are identified by the weighted terms "information" and "retrieval". Assuming that document D_1 is specified as relevant to the initial query Q_o, the feedback formula shown in Fig. 1 produces a new query Q' that lies much closer to document D_1 than the original query. An analogous situation exists when document D_2 is identified as relevant and the new query Q" replaces the original Q_o. Such new queries may be expected to retrieve more useful items similar to the previously identified documents D_1 and D_2, respectively.

The vector modification method outlined earlier is conceptually simple, because the modified term weights are directly obtained from the weights of the corresponding terms in documents known to be relevant or nonrelevant to the respective queries. When the weight assignments available for initial queries and stored documents accurately reflect the values of the terms for content identification, the standard vector modification process provides a powerful query construction method.

B) Probabilistic Feedback Methods

An alternative relevance feedback methodology is based on the probabilistic retrieval model. [14-18] In that case an optimal retrieval rule is used to rank the documents in decreasing order according to expression

$$\log \frac{Pr(x \mid rel)}{Pr(x \mid nonrel)} \tag{7}$$

where $Pr(x \mid rel)$ and $Pr(x \mid nonrel)$ represent the probabilities that a relevant or nonrelevant item, respectively, has vector representation x.

Assuming that the terms are independently assigned to the relevant and to the nonrelevant documents of a collection, and that binary term weights restricted to 0 and 1 are assigned to the documents, a query document similarity value can be derived from (7) between the query and each document D $= (d_1, d_2, \ldots, d_t)$, using two parameters p_i and u_i that represent the probabilities that the ith term has value 1 in a relevant and nonrelevant document, respectively:

$$sim(D,Q) = \sum_{i=1}^{t} d_i \log \frac{p_i(1-u_i)}{u_i(1-p_i)} + constants \tag{8}$$

$$p_i = Pr(x_i = 1 \mid relevant)$$
$$u_i = Pr(x_i = 1 \mid nonrelevant)$$

The similarity formula of expression (8) cannot be used in practice without knowing the values of p_i and u_i for all document terms. A number of different methods have been suggested to estimate these quantities. For the initial search, when document relevance information is not available, the assumption is often made that the values of p_i are constant for all terms (typically 0.5), and that the term distribution in the nonrelevant items is closely approximated by the distribution in the whole collection. [19] Referring to the term occurrence data of Table 1 specifying the occurrences of a typical term i in the relevant and nonrelevant document subsets, u_i can then be set equal to n_i/N, the proportion of documents in the collection that carry term i. For the initial run, expression (8) is then reduced to

$$initial\ sim(D,Q) = \sum_{i=1}^{t} d_i \log \frac{N - n_i}{n_i} . \qquad (9)$$

For the feedback searches, the accumulated statistics relating to the relevance or nonrelevance of previously retrieved items are used to evaluate expression (8). This is done by assuming that the term distribution in the relevant items previously retrieved is the same as the distribution for the complete set of relevant items, and that all nonretrieved items can be treated as nonrelevant. Applying the statistics of Table 1 to the retrieval portion of the collection, one finds that

$$p_i = \frac{r_i}{R} \quad \text{and} \quad u_i = \frac{n_i - r_i}{N - R} .$$

When these expressions are substituted in (8), one obtains the new form

$$feedback\ sim(D,Q) = \sum_{i=1}^{t} d_i \log \left[\frac{r_i}{R - r_i} \div \frac{n_i - r_i}{N - R - n_i + r_i} \right] \qquad (10)$$

where R now represents the total number of relevant retrieved items, r_i is the total number of relevant retrieved that include term i, and n_i is the total number of retrieved items with term i.

Formula (10) poses problems for certain small values of R and r_i that frequently arise in practice--for example, R = 1, r_i = 0--because the logarithmic expression is then reduced to 0. For this reason, a 0.5 adjustment factor is often added in defining p_i and u_i, and the following formulas are used in the conventional probabilistic system to obtain p_i and u_i:

$$p_i = \frac{r_i + 0.5}{R+1} \quad \text{and} \quad u_i = \frac{n_i - r_i + 0.5}{N - R + 1} \quad . \tag{11}$$

The conventional probabilistic system has been criticized for a variety of reasons. For example, the 0.5 adjustment factor may provide unsatisfactory estimates in some cases, and alternative adjustments have been proposed to compute p_i and u_i, such as n_i/N or $(n_i - r_i)/(N - R)$. [18, 20-21] When no relevant items are initially retrieved (that is, R = 0), the best estimate for p_i, the probability that a term occurs in a relevant document, is simply its probability of occurrence in the complete collection. In that case, $p_i = n_i/N$. The test results for the adjusted probabilistic derivation presented later in this study correspond to the following estimates for p_i and u_i:

$$\begin{aligned} p'_i &= Pr(x_i = 1 \mid rel) = \frac{r_i + n_i/N}{R+1} \\ u'_i &= Pr(x_i = 1 \mid nonrel) = \frac{n_i - r_i + n_i/N}{N - R + 1} \end{aligned} \tag{12}$$

In expression (12), the adjustment factor is n_i/N instead of 0.5 as before. An alternative adjustment factor of $(n_i - r_i)/(N - R)$ is valid when the number of relevant documents not yet retrieved is assumed to be small. In practice, the output obtained with that factor differs only marginally from that obtainable with expressions (12).

An additional *ad-hoc* adjustment, similar to the one previously used to transform expression (4) into (5), may be made in (12) by enhancing the importance of document terms that also occur in the queries. This is achieved by assuming that a term occurrence in a query is equivalent to a term occurrence in 3 relevant documents (that is, for query terms $r'_i = r_i + 3$, $R' = R + 3$).

The advantage of all probabilistic feedback models compared with the conventional vector modification methods, is that the feedback process is directly related to the derivation of a weight for query terms. Indeed, the document similarity function of expression (8) increases by a weighting factor of log $[p_i(1-u_i)/u_i(1-p_i)]$ for each query term i that matches a document, and this term weight is optimal under the assumed conditions of term independence and binary document indexing.

On the other hand, a good deal of apparently useful information is disregarded in the probabilistic environment in determining the form of the feedback query, including, for example, the weight of the terms assigned to the documents, and the weight of terms in the original query formulation. Furthermore, the set of relevant retrieved items is not used directly for query adjustment in the probabilistic environment, as it is in the vector model. Instead the term distribution in the relevant retrieved items is used indirectly to determine a probabilistic term weight. These indirections may account for the fact that the probabilistic relevance feedback methods do not in general operate as effectively as the conventional vector modification methods.

3. Relevance Feedback Evaluation

The relevance feedback methods are evaluated by using six document collections in various subject areas for experimental purposes. The collections ranging from a small biomedical collection (MED) consisting of 1033 documents and 30 queries to a large computer engineering collection (INSPEC) of 12684 documents and 84 queries are characterized in Table 2. In all cases the query vectors carry fewer terms than the corresponding document vectors.

A high-quality *initial* search was used for experimental purposes in all cases, consisting of the vector match of query and document vectors (expression (3)) using weighted query and document vectors. The term weights used for both documents and queries in the initial search were computed as the product of the term frequency multiplied by an inverse collection frequency factor, defined as

$$w_i = \frac{(0.5+0.5\ \dfrac{tf_i}{\max\ tf}) \cdot \log \dfrac{N}{n_i}}{\sqrt{}(0.5+0.5\dfrac{tf_i}{\max\ tf})^2\ (log\dfrac{N}{n_i})^2} \tag{14}$$

where tf_i is the occurrence frequency of term i in the document (or in the query), and N and n_i are defined in Table 1. It is known that the term weight assignment of expression (14) produces a high order of performance. [22]

For the experiments, the assumption is made that the top 15 items retrieved in the initial search are judged for relevance, and the information contained in these relevant and nonrelevant retrieved items is then used to construct the feedback query.

To evaluate the effectiveness of the relevance feedback process, it is necessary to compare the performance of the first iteration feedback search with the results of the initial search performed with the original query statements. Normally, recall (R) and precision (P) measures are used to reflect retrieval effectiveness, where recall is defined as the proportion of relevant items that are retrieved from the collection, and precision is the proportion of retrieved items that are relevant. In evaluating a relevance feedback process, the evaluation is complicated by the fact that an artificial ranking effect must be distinguished from the true feedback effect. Indeed, any originally retrieved relevant item that is used for feedback purposes will necessarily be retrieved again in the first iteration feedback search, normally with a much improved retrieval rank. This occurs because the feedback query has been constructed so as to resemble the previously obtained relevant items. When an item originally retrieved with a retrieval rank of 7 or 8 is again obtained with a rank of 1 or 2 in the feedback search, the resulting improvement in recall and precision is not a true reflection of user satisfaction, because a relevant item brought to the user's attention for a second time is of no interest to the user. Instead, the relevance feedback operation must be judged by the ability to retrieve *new* relevant items, not originally seen by the user.

Various solutions offer themselves for measuring the true advantage provided by the relevance feedback process. [23] One possibility is the so-called *residual collection* system where all items previously seen by the user (whether relevant or not) are simply removed from the collection, and both the initial and any subsequent searches are evaluated using the reduced collection only. This depresses the absolute performance level in terms of recall and precision, but maintains a correct relative difference between initial and feedback runs. The residual collection evaluation is used to evaluate the relevance feedback searches examined in this study.

Twelve different relevance feedback methods are used for evaluation purposes with the 6 sample collections, including 6 typical vector modification methods, and 6 probabilistic feedback runs. Six of these feedback methods are characterized in Table 3. In the first two vector modification methods, termed "Ide dec-hi" and "Ide regular" in Table 3, the terms found in previously retrieved relevant (or nonrelevant) documents are added to (or subtracted from) the original query vectors without normalization to obtain the new query statements. [3-4] In the "dec-hi" system, all identified relevant items but only one retrieved nonrelevant item (the one retrieved earliest in a search) are used for query modification. The single nonrelevant item provides a definite point in the vector space from which the new feedback query is removed. The "ide regular" method is identical except that additional previously retrieved nonrelevant documents are also used in the feedback process.

The vector adjustment methods termed "Rocchio" in Table 3 uses reduced document weights to modify the queries as shown earlier in expression (6). [1,2] Several different values are used experimentally for the β and γ parameters of equation (6) to assign greater or lesser values to the relevant items compared with the nonrelevant, including $\beta=1$, $\gamma=0$; $\beta=0.75$, $\gamma=0.25$; and $\beta=\gamma=0.5$.

Three probabilistic feedback systems are also included in Table 3, including the conventional probabilistic approach with the 0.5 adjustment factor, the adjusted probabilistic derivation with adjustments of n_i/N, and finally the adjusted derivation with enhanced query term weights.

A total of 72 different relevance feedback runs were made using the 12 chosen feedback methods. All the feedback methods produce weighted query terms. However, the weights of the terms attached to the documents are not specified by the feedback process. The document vectors may thus be weighted, using a weighting system such as that of expression (14); alternatively, the document vectors used in the feedback searches may carry binary weights, where terms that are present receive a weight of 1 and terms that are absent from a vector are assigned a weight of 0.

In addition to using weighted as well as binary document terms in the experiments, a number of *query expansion* methods can be applied in the feedback operations. The first possibility consists in not using any query expansion at all, and preserving only the original query terms appropriately reweighted for feedback purposes. Alternatively, a full query expansion can be used where all terms contained in the relevant previously retrieved items are added to formulate the new feedback query, as suggested in the previously given feedback equations. Finally, several partial query expansion methods can be used, where only some of the terms present in the previously identified relevant items are incorporated into the query. The partial query expansions are designed to produce superior query formulations at much reduced storage cost by restricting the query length to the average length of the relevant retrieved documents. In the expansion system presented in this study, the *most common* terms chosen for addition to the original query are those with the highest occurrence frequencies in the previously retrieved relevant items. Alternatively, the *highest weighted* terms--those with the highest feedback weight--could have been used for query expansion.

Three measures are computed for evaluation purposes, including the *rank order* of each particular feedback method out of the 72 different feedback procedures tried experimentally. A rank order of 1 designates the best method exhibiting the highest recall-precision value, and a rank of 72 designates the worst process. In addition, a search precision figure is computed representing the average precision at three particular recall points of 0.75, 0.50, and 0.25 (representing high recall, medium recall, and low recall points, respectively). Finally the percentage improvement in the three-point precision between feedback and original searches is also shown. Typical evaluation output for 5 of the 6 collections is shown in Table 4 for the runs using weighted document and query terms, and in Table 5 for runs with binary document terms.

With some minor exceptions, the results of Tables 4 and 5 are homogeneous for the 5 collections, in the sense that the best results are produced by the same relevance feedback systems for all collections, and the same holds also for the poorest results. These results differ, however, from those described later for the sixth collection (the NPL collection). The following main performance results are evident:

- A comparison of the results of Tables 4 and 5 for weighted and unweighted document vectors, respectively, shows that the weighted terms produce much better results in a feedback environment. This confirms results produced by earlier term weighting studies. [22]

- The paired comparisons included in Table 4 between full query expansion (where all terms from the previously retrieved relevant documents are incorporated in the feedback query) and restricted expansion by the most common terms from the relevant items, shows that full expansion is often preferable. However, the performance difference is modest, so that the expansion by most common terms should be used when storage requirements and processing times needed for fully expanded queries appear excessive. Other possible expansion systems (no expansion, or expansion by highest weighted terms) are inferior.

- The best overall relevance feedback method is the "Ide dec hi" method, where terms are directly added to the queries and only one nonrelevant item is used in the process. This method is also computationally least demanding.

- Other effective vector modification methods are the regular Ide process where additional nonrelevant items are used, and the Rocchio modification using normalized term weight adjustments. Relatively more weight should be given to terms obtained from the relevant items than to those extracted from the nonrelevant ($\beta=0.75$, $\gamma=0.25$). Other choices of the parameters (e.g. $\beta=\gamma=0.5$, and $\beta=1$, $\gamma=0$) produce less desirable results.

- The probabilistic feedback system is in general not as effective as the vector modification method. Of the probabilistic methods implemented here, the adjusted derivation with extra weight assignments for query terms was only somewhat less effective than the better vector processing methods. However, the probabilistic methods are, in any case, computationally more demanding than the vector processing methods.

The results of Tables 4 and 5 demonstrate that relevance feedback represents a powerful process for improving the output of retrieval system operations. The average improvement in the 3-point precision obtained for a single search iteration is nearly 90 percent for the five test collections. Furthermore additional improvements of up to 100 percent may be obtainable when additional feedback searches are carried out. [9]

The actual amount of improvement produced by one iteration of the feedback process varies widely, ranging from 47 percent for the CISI collection to 160 percent for the CRAN collection for the "Ide dec hi" system. The following factors may be of principal importance in determining the improvement obtained from the feedback process in particular collection environments:

- The average length of the original queries is of main interest. Because the feedback process involves the addition to the queries of new terms extracted from previously retrieved relevant documents, collections with short (often incomplete) queries tend to gain more from the feedback procedure than collections using longer, more varied initial query statements. The statistics of Table 2 show that query length is directly correlated with relevance feedback performance (for the CRAN collection with an average query length of 9.2 an improvement of over 150 percent is obtainable, but the gain is limited to about 50 percent for CISI with an average query length of 28.3 terms).
- Collections that perform relatively poorly in an initial retrieval operation can be improved more significantly in a feedback search than collections that produce satisfactory output in the initial search. For example, the MED collection with an initial average precision performance of 0.3346 has less potential for improvement than collections with an initial performance of 0.15 or less.
- Technical collections used with precisely formulated queries may be better adapted to the feedback process, than more general collections used with more discursive queries. In the former case, the set of relevant documents for any query may be concentrated in a small area of the document space, making it easier to construct high-performance queries in a feedback operation.

The relevance feedback results for the NPL collection shown in Table 6 do not follow the complete pattern established for the other collections. While the relative performance order for the feedback methodologies and the query expansion systems remains generally unchanged, the NPL results obtained for binary document vectors are in each case superior to those for the corresponding weighted term assignments. It was noted in earlier studies that the characteristics of the NPL collection differ substantially from those of the other collections. [22] The data of Table 2 show that both the document and the query vectors are much shorter for NPL than for the other collections, and the variation in query length (standard deviation of 2.36 for a mean number of 7.16 query terms) is very small. Furthermore, the term frequencies are especially low for the NPL collection: each query term appears precisely once in a query, and the average frequency of the terms in the documents is only 1.21. In these circumstances, the term frequency weighting and length normalization operations cannot perform their intended function and binary term assignments are preferred. One may conjecture that the NPL index terms are carefully chosen, and may in fact represent specially controlled terms rather than freely chosen natural language entries. Because of the short queries, a substantial performance improvement of over 100 percent is, however, noted for the NPL collection, confirming the earlier results obtained for other collections with short query formulations.

In conclusion, the relevance feedback process provides an inexpensive method for reformulating queries based on previously retrieved relevant and nonrelevant documents. A simple vector modification process that adds new query terms and modifies the weight of existing terms appears most useful in this connection. Weighted document vectors should be used except when the occurrence characteristics of all terms are uniform as in NPL. The probabilistic feedback methods that disregard the original query term weights as well as the document term weights are not generally competitive with the simpler vector modification methods. Improvements from 50 percent to 150 percent in the three-point precision are attainable in the first feedback iteration. In view of the simplicity of the query modification operation, the relevance feedback process should be incorporated into operational text retrieval environments.

References

[1] J.J. Rocchio, Jr., Document Retrieval Systems--Optimization and Evaluation, Doctoral Dissertation, Harvard University, in Report ISR-10, to the National Science Foundation, Harvard Computation Laboratory, Cambridge, MA, March 1966.

[2] J.J. Rocchio, Jr., Relevance Feedback in Information Retrieval, in The Smart System-- Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, NJ, 1971, 313-323.

[3] E. Ide, New Experiments in Relevance Feedback, in The Smart System--Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, NJ, 1971, 337-354.

[4] E. Ide and G. Salton, Interactive Search Strategies and Dynamic File Organization in Information Retrieval, in The Smart System-- Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, NJ, 1971, 373-393.

[5] G. Salton, Relevance Feedback and the Optimization of Retrieval Effectiveness, in The Smart System-- Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, NJ, 1971, 324-336.

[6] M. Dillon and J. Desper, Automatic Relevance Feedback in Boolean Retrieval Systems, Journal of Documentation, 36:3, September 1980, 197-208.

[7] G. Salton, E. Voorhees and E.A. Fox, A Comparison of Two Methods for Boolean Query Relevance Feedback, Information Processing and Management, 20:5-6, 1984, 637-651.

[8] E.A. Fox, Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types, Doctoral Dissertation, Cornell University, Department of Computer Science, August 1983.

[9] G. Salton, E.A. Fox and E. Voorhees, Advanced Feedback Methods in Information Retrieval, Journal of the American Society for Information Science, 36:3, May-June 1985, 200-210.

[10] C. Vernimb, Automatic Query Adjustment in Document Retrieval, Information Processing and Management, 13:6, 1977, 339-353.

[11] C. Stanfill and B. Kahle, Parallel Free-Text Search on the Connection Machine, Communications of the ACM, 29:12, December 1986, 1229-1239.

[12] D.L. Waltz, Applications of the Connection Machine, Computer, 20:1, January 1987, 85-97.

[13] G. Salton and C. Buckley, Parallel Text Search Methods, Communications of the ACM, 31:2, February 1988, also Technical Report 87-828, Department of Computer Science, Cornell University, Ithaca, NY, April 1987.

[14] C.J. van Rijsbergen, Information Retrieval, Butterworths, London, Second Edition, 1979.

[15] D.J. Harper, Relevance Feedback in Document Retrieval Systems, Doctoral Dissertation, University of Cambridge, England, 1980.

[16] S.E. Robertson and K. Sparck Jones, Relevance Weighting of Search Terms, Journal of the American Society for Information Science, 27:3, May-June 1976, 129-146.

[17] S.E. Robertson, C.J. van Rijsbergen and M.F. Porter, Probabilistic Models of Indexing and Searching, in Information Retrieval Research, R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen, and P.W. Williams, editors, Butterworths, London, 1981, 35-56.

[18] C.T. Yu, C. Buckley, K. Lam and G. Salton, A Generalized Term Dependence Model in Information Retrieval, Information Technology: Research and Development, 2:4, October 1983, 129-154.

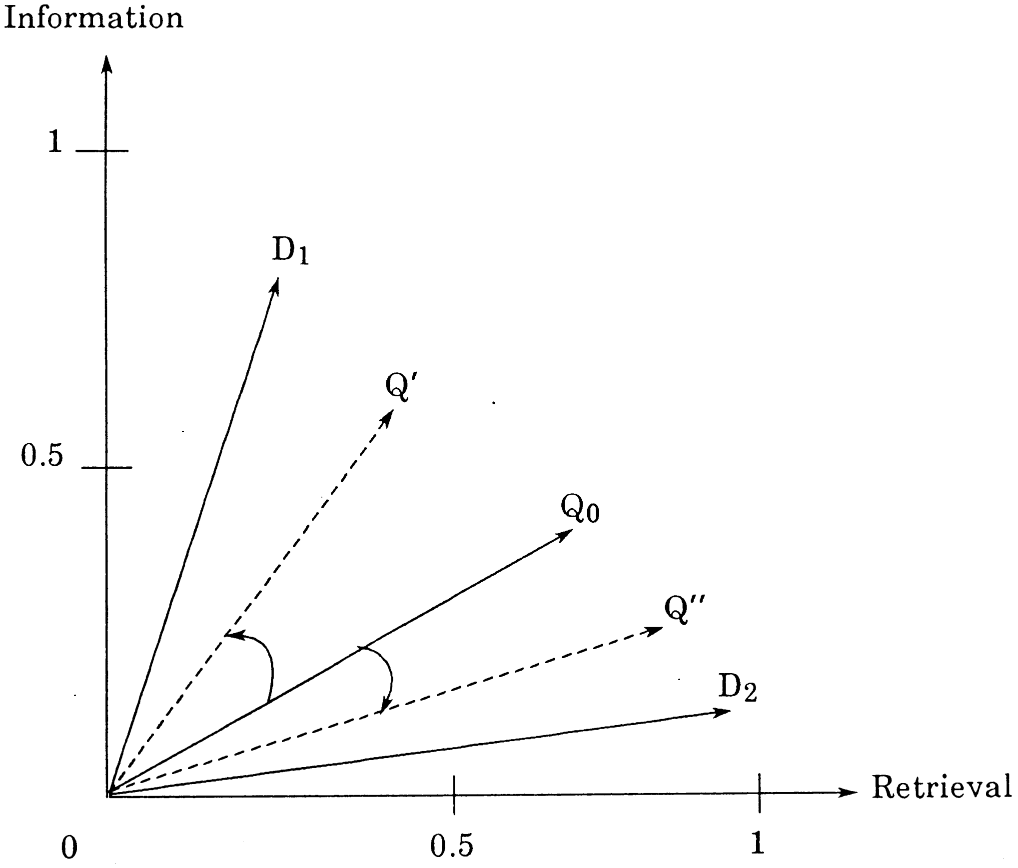
[19] W.B. Croft and D.J. Harper, Using Probabilistic Models of Document Retrieval Without Relevance, Journal of Documentation, 35:4, December 1979, 285-295.

[20] S.E. Robertson, On Relevance Weight Estimation and Query Expansion, Journal of Documentation, 42:3, September 1986, 182-188.

[21] H. Wu and G. Salton, The Estimation of Term Relevance Weights Using Relevance Feedback, Journal of Documentation, 37:4, December 1981, 194-214.

[22] G. Salton and C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, Technical Report 87-881, Department of Computer Science, Cornell University, November 1987.

[23] Y.K. Chang, C. Cirillo and J. Razon, Evaluation of Feedback Retrieval Using Modified Freezing, Residual Collection, and Test and Control Groups, in The Smart Retrieval System--Experiments in Automatic Document Processing, G. Salton, editor, Prentice Hall Inc., Englewood Cliffs, NJ, 1971, Chapter 17, 355-370.



Q_o	=	retrieval of information	(0.7, 0.3)
D_1	=	information science	(0.2, 0.8)
D_2	=	retrieval systems	(0.9, 0.1)

$$Q' = \frac{1}{2} Q_o + \frac{1}{2} D_1 = (0.45, 0.55)$$

$$Q'' = \frac{1}{2} Q_o + \frac{1}{2} D_2 = (0.80, 0.20)$$

Relevance Feedback Illustration

Fig. 1

	relevant items	nonrelvant items	all items
$d_i = 1$	r_i	$n_i - r_i$	n_i
$d_i = 0$	$R - r_i$	$N - R - n_i + r_i$	$N - n_i$
all items	R	$N - R$	N

Occurrences of Term i in a Collection of N Documents

Table 1

Collection	Number of Vectors (Documents or Queries)	Average Vector Length (Number of Terms)	Standard Deviation of Vector Length	Average Frequency of Terms in Vectors	Percentage of Terms in Vectors with frequency 1
CACM					
documents	3204	24.52	21.21	1.35	80.93
queries	64	10.80	6.43	1.15	88.68
CISI					
documents	1460	46.55	19.38	1.37	80.27
queries	112	28.29	9.49	1.38	78.36
CRAN					
documents	1398	53.13	22.53	1.58	69.50
queries	225	9.17	3.19	1.04	95.69
INSPEC					
documents	12684	32.50	14.27	1.78	61.06
queries	84	15.63	8.66	1.24	83.78
MED					
documents	1033	51.60	22.78	1.54	72.70
queries	30	10.10	6.03	1.12	90.76
NPL					
documents	11429	19.96	10.84	1.21	84.03
queries	100	7.16	2.36	1.00	100.00

Collection Statistics (including average vector length and standard derivation of vector lengths)

Table 2

Feedback Method	Description
Vector Adjustment (Ide dec-hi)	Add document term weights directly to query terms; use all relevant retrieved for feedback purposes, but only the top-most nonrelevant items $Q_{new} = Q_{old} + \sum_{\substack{all\\relevant}} D_i - \sum_{\substack{one\\nonrelevant}} D_i$
Vector Adjustment (Ide regular)	Add actual document term weights to query terms; use all previously retrieved relevant and nonrelevant for feedback: $Q_{new} = Q_{old} + \sum_{\substack{all\\relevant}} D_i - \sum_{\substack{all\\nonrelevant}} D_i$
Vector Adjustment (Standard Rocchio)	Add reduced term weights to query following division of term weights by number of documents used for retrieval; choose values of β, γ in range 0 to 1 so that $\beta + \gamma = 1.0$ $Q_{new} = Q_{old} + \beta \sum_{\substack{n_1 \; rel\\docs}} \frac{D_i}{n_i} - \gamma \sum_{\substack{n_2 \; non\\relevant\\docs}} \frac{D_i}{n_2}$
Probabilistic conventional	$Q_{new} = \log\,[p_i(1-q_i)/q_i(1-p_i)]$ $p_i = P(x_i \mid rel) = \frac{r_i + 0.5}{R + 1.0}$ $u_i = P(x_i \mid nonrel) = \frac{n_i - r_i + 0.5}{N - R + 1.0}$
Probabilistic adjusted derivation	$Q_{new} = \log\,[p_i(1-q_i)/q_i(1-p_i)]$ $p'_i = P(x_i \mid rel) = \frac{r_i + n_i/N}{R + 1}$ $u'_i = P(x_i \mid nonrel) = \frac{n_i - r_i + n_i/N}{N - R + 1}$
Probabilistic adjusted derivation revised	Same as adjusted derivation, but for query terms use r'_i and R' instead of r_i and R, where $r'_i = r_i + 3$, $R' = R + 3$

Description of Some Relevance Feedback Methods

Table 3

Relevance Feedback Method	Rank of Method and Avg Precision	CACM 3204 docs 64 queries	CISI 1460 docs 112 queries	CRAN 1397 docs 225 queries	INSPEC 12684 docs 84 queries	MED 1033 docs 30 queries	Average
Initial Run (reduced collection)		.1459	.1184	.1156	.1368	.3346	
Ide (dec hi)							
	Rank	1	2	6	1	1	2.2
- expand by	Precision	.2704	.1742	.3011	.2140	.6305	
all terms	Improvement	+86%	+47%	160%	+56%	+88%	+87%
- expand by most	Rank	4	1	13	2	3	4.6
common terms	Precision	.2479	.1924	.2498	.1976	.6218	
	Improvement	+70%	+63%	+116%	+44%	+86%	+76%
Ide (regular)							
	Rank	7	18	15	4	2	9.2
- expand by	Precision	.22417	.1550	.2508	.1936	.6228	
all terms	Improvement	+66%	+31%	+117%	+42%	+86%	+68%
- expand by	Rank	17	5	17	17	4	12
most common	Precision	.2179	.1704	.2217	.1808	.5980	
terms	Improvement	+49%	+44%	+92%	+32%	+79%	+59%
Rocchio (standard $\beta = .75$, $\alpha .25$)							
	Rank	2	39	8	14	17	16
- expand by	Precision	.2552	.1404	.2955	.1821	.5630	
all terms	Improvement	+75%	+19%	+156%	+33%	+68%	+70%
- expand by	Rank	3	12	12	10	24	12.2
most common	Precision	.2491	.1623	.2534	.1861	.5297	
terms	Improvement	+71%	+37%	+119%	+36%	+55%	+64%
Probabilistic (adjusted revised derivation)							
	Rank	11	36	3	32	5	17.4
- expand by	Precision	.2289	.1436	.3108	.1621	.5972	
all terms	Improvement	+57%	+21%	+169%	+19%	+78%	+69%
- expand by	Rank	14	10	18	9	14	13
most common	Precision	.2224	.1634	.2120	.1876	.5643	
terms	Improvement	+52%	+38%	+83%	+37%	+69%	+56%
Conventional Probabilistic							
	Rank	18	56	1	55	13	28.6
- expand by	Precision	.2165	.1272	.3117	.1343	.5681	
common terms	Improvement	+48%	+ 7%	+170%	- 2%	+70%	+59%
- expand by	Rank	12	4	11	19	8	10.8
most common	Precision	.2232	.1715	.2538	.1782	.5863	
terms	Improvement	+53%	+45%	+120%	+30%	+75%	+65%

Evaluation of Typical Relevance Feedback Methods for 5 Collections
(weighted documents, weighted queries)

Table 4

Relevance Feedback Method	Rank of Method and Avg Precision	CACM 3204 docs 64 queries	CISI 1460 docs 112 queries	CRAN 1397 docs 225 queries	INSPEC 12684 docs 84 queries	MED 1033 docs 30 queries	Average 5 Collections
Initial Run (reduced collection)		.1459	.1184	.1156	.1368	.3346	
Ide (dec hi) -expand by most common terms	Rank Precision Improvement	24 .1901 +30%	8 .1653 +40%	28 .1878 +62%	5 .1905 +39%	23 .5317 +59%	17.6 +46%
Ide (regular) -expand by most common terms	Rank Precision Improvement	32 .1812 +25%	30 .1445 +22%	36 .1751 +51%	22 .1734 +27%	28 .5061 +51%	29.6 +35%
Rocchio (standard $\beta=.75$, $\gamma=.25$) -expand by most common terms	Rank Precision Improvement	31 .1843 +26%	49 .1311 +11%	35 .1752 +52%	45 .1526 +12%	56 .4033 +21%	43.2 +24%
Probabilistic (adjusted revised derivation) -expand by most common terms	Rank Precision Improvement	43 .1669 +14%	52 .1305 +10%	33 .1777 +54%	35 .1616 +18%	38 .4766 +42%	40.2 +28%
Conventional Probabilistic -expand by most common terms	Rank Precision Improvement	33 .1800 +23%	28 .1484 +25%	24 .2042 +77%	23 .1732 +27%	29 .4962 +48%	27.4 +40%

Relevance Feedback Evaluation for 5 Collections (binary documents)

Table 5

Processing Method	Binary Documents Weighted Queries			Weighted Documents Weighted Queries		
	Rank	Precision	Improvement	Rank	Precision	Improvement
Initial run (reduced collection)					.1056	
Ide (dec hi)						
-expanded by all terms	1	.2193	+108%	35	.1540	+46%
-expanded by most common terms	3	.2126	+101%	40	.1334	+26%
Rocchio method						
$\beta=0.75$, $\gamma=0.25$ expansion by all terms	8	.1985	+88%	30	.1618	+53%
$\beta=0.75$, $\gamma=0.25$ expansion by most common terms	6	.2037	+93%	42	.1287	+22%
Probabilistic adjusted revised derivation						
- expanded by all terms	21	.1879	+78%	33	.1547	+46%
- expanded by most common terms	10	.1984	+84%	53	.1174	+11%
Probabilistic conventional derivation						
- expanded by all terms	37	.1417	+34%	49	.1259	+19%
- expanded by most common terms	18	.1916	+81%	43	.1285	+22%

Relevance Feedback Evaluation for NPL Collection
(11429 documents, 100 queries)

Table 6