IX. Negative Response Relevance Feedback

John Kelly

1. Introduction

Relevance feedback is a method for improving the performance of an automatic information retrieval system so that the system better satisfies the needs of its users. [1,2] Such a method is required for at least two reasons. First, because of unfamiliarity with the system on the part of many users, the initial queries are often misstated and hence do not reflect the user's true needs. Secondly, because of the particular indexing scheme employed by the system, an initial query may not properly distinguish concepts which the user is capable of distinguishing. In particular, relevance feedback attempts to increase the precision and recall of a user's query by using feedback information furnished in response to the documents originally retrieved in order to modify the query so that it better distinguishes the relevant and non-relevant documents of the collection. Of course, the process may be applied to the new query so that, by a sequence of iterations, an optimal query is reached which best separates the relevant and non-relevant document sets.

Previous research [1,2] has concentrated on the case where at least one relevant document is retrieved and the goal is to obtain the others. The results obtained show that a few feedback iterations significantly increase the precision and recall of most queries. However, there are enough initial queries which fail to retrieve any relevant documents to make it useful to design a query modification algorithm capable of effectively handling these cases. Hence, the purpose of this project is to determine

and test a query modification procedure which can cause a relevant document to be retrieved in as few iterations as possible whenever the initial query retrieves only non-relevant documents.

2. Principal Algorithm

A document or query is represented by a t-component vector in which the value of the $i^{\underline{th}}$ component gives the importance or weight for the document or query of the $i^{\underline{th}}$ concept. The weights are non-negative and a weight of zero represents absence of the corresponding concept. Let $\underline{d}^j$ be the $j^{\underline{th}}$ document vector, $\underline{q}^i$ the query vector of the $i^{\underline{th}}$ iteration ($i = 1$ for the initial query), and $\underline{v}_k$ the $k^{\underline{th}}$ component of vector $\underline{v}$. Throughout this paper all vectors are assumed to be normalized to unit length unless stated otherwise, i.e.

$$\sum_{k=1}^{t} (\underline{v}_k)^2 = 1.$$

Let D be the set of integers from 1 to the number of documents d such that each document has a unique integer assigned to it. It is assumed that the user designates all retrieved documents as either relevant or non-relevant. Hence, let D^i be the set of documents retrieved by $\underline{q}^i$, R^i those designated relevant, and N^i those not relevant, where $D^i \leqq D$, $R^i \leqq D^i$, and $N^i \leqq D^i$. Finally, each concept k is assigned a weight reflected by the magnitude of the corresponding components $\underline{v}_k$ in the various document vectors. For example, it may occur in ten of the d documents. These frequencies may be ranked in descending order, and $f(k)$ may

represent the concept number, $1 \leq f(k) \leq t$, with the $k^{\underline{th}}$ highest frequency.

The set D^i is determined by the matching function $p(\underline{d}^j, \underline{q}^i)$ which correlates $\underline{d}^j$ and $\underline{q}^i$, and by the number s of new documents sent to the user. A useful matching function is the cosine correlation

$$p(\underline{d},\underline{q}) = \underline{d} \cdot \underline{q} = \sum_{k=1}^{t} \underline{d}_k \underline{q}_k = \cos(\underline{d},\underline{q}).$$

As usual, $\underline{q}^i$ may be correlated with each $\underline{d}^j$, $j=1,2,\ldots,d$, and the correlations may be ranked in decreasing order. In addition, a list may be maintained which contains the document numbers and the user's responses for those documents sent to him on previous iterations. On the second iteration, D^i consists of the r most highly correlated documents (with $\underline{q}^i$) where s documents are not on the list, $r \geq s$, and the $r^{\underline{th}}$ ranked document is not on the list. The s new documents are sent to the user and added to the list along with his responses.

Assuming that the user wishes to perform another search of the document collection, the modified query $\underline{q}^{i+1}$ is computed from $\underline{q}^i$, N^i, and R^i by the following algorithm. Two sums of document vectors are formed, one over R^i and one over N^i. Should either set be empty, its sum is set to zero. Let $g(j) = r+1-h(j)$ where $h(j)$ is the rank of the $j^{\underline{th}}$ document as determined by the correlations with $\underline{q}^i$. Then

$$\underline{S}_R = \sum_{j \in R^i} g(j)\underline{d}^j \quad \text{and} \quad \underline{S}_N = \sum_{j \in N^i} g(j)\underline{d}^j \quad .$$

Highly ranked documents $(g(j) \approx 1)$. In addition, two sums of document weights may be formed as follows:

$$W_R = \sum_{j \in R^i} g(j) \quad \text{and} \quad W_N = \sum_{j \in N^i} g(j) \quad .$$

Now $\underline{S}_R/W_R$ and $\underline{S}_N/W_N$ form the weighted means of the documents in R^i and N^i.

The new query $\underline{q}^{i+1}$ is computed in three steps. First, if N^i is non-empty, one forms $\underline{q}' = \underline{q}^i - a_N \underline{S}_N/W_N$, where a_N is an arbitrary weighting factor determined by experience with the system. In forming $\underline{q}'$, any element $\underline{q}'_k$ less than zero is set to zero. The second step, taken when R^i is non-empty, is to form $\underline{q}^{i+1} = \underline{q}' + a_R \underline{S}_R/W_R$, where a_R is another arbitrary weighting factor. The third step, taken when R^i is empty, is to add an arbitrary weight W to the $i^{\underline{th}}$ most frequent concept $f(i)$, i.e. $\underline{q}_k^{i+1} = \underline{q}'_k$ for $k \neq f(i)$ and $\underline{q}_{f(i)}^{i+1} = \underline{q}'_{f(i)} + W$. Finally, $\underline{q}^{i+1}$ is normalized to unit length.

The two factors a_N and a_R control the amount of feedback used in forming $\underline{q}^{i+1}$. Since $\underline{S}_N/W_N$ and $\underline{S}_R/W_R$ are mean vectors of N^i and R^i, their component values are of about the same order of magnitude as those of $\underline{q}^i$. Hence, a reasonable range for a_N and a_R is 0.5 to 2.0. The added weight W is designed to cause the successive queries $\underline{q}^i$, i=1,2,..., to sweep through the document space until a relevant document is retrieved. Thus, W should be large enough to cause a significant change in the query. On the other hand, $\underline{q}^i$ and in particular $\underline{q}^1$ may be fairly close to a relevant document, so that W should not be too large. A reasonable choice for

W is thus $W \approx \frac{1}{2}\left[\max_{k=1,\ldots,t}(q_k')\right]$

3. Experimental Method

The algorithm outlined above has been tested on a simulation of the ADI collection (82 documents, 35 queries, 601 concepts, produced by a thesaurus run on abstracts and titles), i.e., on a document collection with approximately the same characteristics as the ADI collection already used with SMART but with a larger number of documents and a smaller number of concepts. There are two important characteristics which must be simulated. One is the concept frequency as discussed above and used in the algorithm. The other is the concept weight frequency, i.e. the probability that a non-zero concept weight has a particular value. For example, the non-zero weight 12 occurs with a frequency of 0.656 in the ADI collection. Because concept numbers are arbitrarily assigned in the experimental collection, the ranking function $f(i)$ becomes unnecessary if the concept numbers in the new collection are assigned in decreasing frequency order, so that concept one occurs most often. Both frequencies are independent of d while the weight frequency is also independent of t. The concept frequency becomes approximately independent of t by appropriate scaling. For example, if one-sixth as many concepts are desired, then every sixth concept of the original collection is used (after the concepts are ranked by frequency).

A new random document collection with arbitrary d and t but with the same frequency characteristics may be generated by means of a uniform random number generator which generates numbers with equal probability in

the range 0 to 1. The frequencies are now simply probabilities, and thus need only be normalized to the interval (0,1). Hence, the non-zero concept weights are assigned to distinct successive subintervals of (0,1) so that the length of a subinterval equals the probability of that weight being selected. For example, weights 12, 24, and 36 with frequencies 1/2, 1/3 and 1/6 respectively would be assigned intervals (0,1/2), (1/2, 5/6), and (5/6,1). The concept frequencies are already normalized. The value of each concept of each document generated is then determined by two random numbers. If the first number is larger than the concept frequency, the value is set to zero. Otherwise, the second number selects that weight within whose subinterval it lies. For instance, 2/3 selects 24 in the above example. Finally, the vectors are normalized to unit length. Note that typical initial queries may be generated in the same manner, the characteristic frequencies now being those of a collection of queries for the document collection.

The use of a random document collection rather than an actual collection brings with it one main advantage: flexibility. Thus d and t may be adjusted so that the collection of vectors fills exactly the main memory of a computer. By storing vectors, one concept per word of memory, the program for implementing and testing an algorithm becomes easily modifiable. The described query modification algorithm is the result of trial and error experimentation in which the ease of alteration has proved worthwhile. Finally, note that actual document collections vary widely in their characteristics, so that the simulation may not need to be very accurate.

The following procedure was used for testing the feedback algorithm: First, a document collection is generated. Next, a query is generated to be used as the target or goal of the modification process. The relevant

documents are those most highly correlated with the target query. Such a tight clustering of the relevant documents implies that if a given document is retrieved, then the others will, also, usually be retrieved within one or two more iterations; furthermore, the query must lie within one small sector of the document space to retrieve any relevant documents. In practical applications, the relevant documents often occupy larger areas of the document space, thereby increasing the chance of finding at least one relevant document while simultaneously decreasing the likelihood of retrieving all of them. The third step consists in generating an initial query whose correlation with the target (TIQ correlation) lies within a specified range, e.g. 0.500 to 0.800. The final step is to begin the iterations of search, retrieval, response, and query modification.

4. Results

An experiment consists of the performance of the above procedure for one target-initial query pair. The following table shows the parameters used in the experiments, all of which were performed on the same document collection using the cosine correlation function.

150 documents.
100 concepts.
3 documents selected as relevant.
2 new documents sent to user per iteration.

The tables in the Appendix give the results of 89 experiments using the principal algorithm and six modifications of it. Method 1 is the principal algorithm with $a_N = .9$, $a_R = 1$, and $W = \frac{1}{2} \max (\underline{q}_k')$. In all other methods, rank weighting does not occur, i.e., $g(j) = 1$ for all j. In methods 2, 3,

4, and 7, $a_N = 1$. Methods 2, 3, and 7, have W's of 0, (1/10) min $(\underline{q}_k')$, and $\frac{1}{2}$ max $(\underline{q}_k')$ respectively, while Method 4 allows the concept weights to be negative and has $W = \frac{1}{2}$ max $(\underline{q}_k')$. Finally methods 5 and 6 add $W = \frac{1}{2}$ max $(\underline{q}_k')$ into two concepts, namely i and i+5 for Method 5, and i and i+4 for Method 6, where i is the iteration number.

Because the goal is to retrieve a relevant document when the initial query fails, suitable measures of performance are the percentage of successful experiments (relevant document found) and the average number of iterations required to succeed. Precision and recall measures are not appropriate because both are zero until a relevant document is found. The table which follows summarizes these measures for the seven methods.

Method	No. of Successes	No. of Failures	Percent Success	Percent Failure	Average No. Iterations for Success	TIQ Correlation Range
1	30	12	71	29	3.0	.0-.8
2	5	5	50	50	12.4	.0-.7
3	1	2	33	67	5.0	.0-.1
4	1	2	33	67	25.0	.0-.1
5	3	3	50	50	4.0	.2-.7
6	4	8	33	67	2.8	.5-.8
7	4	5	44	56	4.5	.0-.7

Clearly, the principal method (1) gives the best overall results. Examination of the table for Method 1 in the Appendix shows that this method works well over the entire TIQ correlation range .0-.8. When the added concept weight W is zero or small (methods 2 and 3), success usually occurs only because a large portion of the document collection has been retrieved. Allowing query concept weights to become negative (Method 4)

also fails because in very few iterations all of the weights become negative, so that the query is well outside the document space. Double insertion of W (methods 5 and 6) fails because it tends to overpower the query by forcing too many of the concept weights to approximately the same value (the maximum value). However, the low success of Method 7 ($g(j) = 1$) is due to a lack of experiments in the .5-.8 TIQ correlation range. Methods with large W sweep through the document space and retrieve new documents only on 70-90% of the iterations.

5. Conclusion

The use of the non-relevant documents retrieved and the insertion of extra concept weights as described in the principal algorithm appears to be an effective method of query modification for retrieving relevant documents in a few iterations, even if the initial query retrieves nothing useful. It is recommended that the method be tested on several actual document collections, in particular, the ADI collection. In addition, Method 7 should also be further tested since it may give equally good results but with less computation. Another untested possibility consists in using only the one or two highest ranked non-relevant documents rather than all of those retrieved. In any case, the insertion of large concept weights into successively ranked concepts is vital to the success of the algorithm. Without it, a query is restricted to the concepts of the initial query, and these are often eliminated because of their prominence in the retrieved non-relevant documents. Finally, it is suggested that the use of concept weight insertion to locate widely separated relevant documents not retrievable by a single query be investigated.

References

[1] J. J. Rocchio, Document Retrieval Systems -- Optimization and Evaluation, Report ISR-10 to the National Science Foundation, Chapter 3, March 1966.

[2] W. Riddle, T. Horwitz, and R. Dietz, Relevance Feedback in an Information Retrieval System Report ISR-11 to the National Science Foundation, Section 6, June 1966.

[3] D. J. Wilde, Optimum Seeking Methods, Prentice-Hall, 1964.

Appendix

The subsequent tables list the experiments performed for each of the seven methods tested. The column headings have the following meanings:

Expt. No.	=	Experiment Number.
TIQ Corr.	=	Target-Initial Query Correlation.
Target Terms	=	the generated non-zero target query concepts. (underlined concept numbers indicate larger weights)
IQ Terms	=	the Initial Query concepts.
SF	=	Succeeded in finding a relevant document or Failed. (IS means the initial query succeeded)
No. Iter.	=	the Number of Iterations to find a relevant document if successful, otherwise the number of iterations performed before arbitrarily stopping.

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
44	.707	1	1,3	F	20
45	.500	1,3	1,2	S	4
46	.500	1,5	1,13	F	20
47	.500	1,13	1,11	S	2
48	.500	1,8	1,4	S	2
49	.500	1,2	1,8	S	3
50	.707	1,10	1	S	2
51	.707	2	1,2	S	3
52	.707	1	1,3	F	20
53	.500	2,3	1,2	IS	1
54	.707	1,2	2	S	2
55	.500	1,7	1,3	F	20
56	.500	9,10	1,10	S	2
57	.500	1,3	1,11	S	5
58	.500	1,5	1,3	F	20
59	.500	1,13	1,4	S	2
60	.500	1,8	1,2	S	12
61	.500	1,2	1,13	S	3
62	.707	1,10	1	S	2
63	.577	2	1,2,13	S	2
64	.707	6	1,6	IS	1
65	.707	2,3	2	S	2
66	.707	1,2	1	S	3
67	.500	1,7	1,2	S	2
68	.707	1,10	1	S	2
69	.707	1,6	1	F	10
70	.577	1,4	2,3,4	IS	1
71	.707	5	1,5	S	3
72	.707	1,11	1	F	10
73	.500	2,3	1,3	S	2
74	.707	2	1,2	S	3
75	.500	1,2,7,12	1	F	10
76	.700	1,4	1,3,4	S	2
77	.707	1	1,7	S	2
78	.500	1,2	1,10	S	3
79	.671	1,5	1,11	S	2
80	.577	1,2,4	1	S	3
81	.707	1,3	3	IS	1
82	.500	1,11	1,6	F	15
83	.577	1	1,3,10	F	15
84	.408	8	1,4,8	S	2
85	.577	1	1,3,7	F	15
86	.000	1	2	F	15
87	.000	3	1,2,6,7	S	5
88	.000	2	1,11	S	4
89	.000	1	9	S	3

Method 1

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
1	.000	1	3,13	F	25
2	.000	1	2	F	25
3	.000	1,3	5,10	S	22
4	.000	1,5	2	S	15
20	.577	1	1,2,11	F	25
21	.500	1,3	1,11	F	2
22	.408	1,5	1,3,12	S	8
23	.500	1,13	1,8	S	2
24	.500	1,8	1,2	F	25
25	.500	1,5	1,2	S	15

Method 2

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
5	.000	1	2	F	25
6	.000	1,3	5,10	S	5
7	.000	1,5	2	F	25

Method 3

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
11	.000	1	2	S	25
12	.000	1,3	5,10	F	25
13	.000	1,5	11	F	25

Method 4

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
26	.577	1	1,2,11	F	25
27	.500	1,3	1,11	F	25
28	.408	1,5	1,3,12	F	25
29	.500	1,13	1,8	S	2
30	.500	1,8	1,2	S	6
31	.500	1,2	1,3	S	4

Method 5

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
32	.707	1	1,3	F	20
33	.500	1,3	1,2	F	20
34	.500	1,5	1,13	F	20
35	.500	1,13	1,11	S	2
36	.500	1,8	1,4	S	2
37	.500	1,2	1,8	F	20
38	.707	1,10	1	F	20
39	.707	2	1,2	S	4
40	.671	1,11	1,3	F	20
41	.500	1	1,3,5,12	F	20
42	.707	1,2	2	S	3
43	.500	1,7	1,3	F	20

Method 6

Expt. No.	TIQ Corr.	Target Terms	IQ Terms	SF	No. Iter.
8	.000	1	2	S	3
9	.000	1,3	5,10	S	5
10	.000	1,5	11	F	25
14	.577	1	1,2,11	F	25
15	.408	1,3	1,2,11	S	5
16	.408	1,5	1,2,11	F	25
17	.333	1,3,12	1,2,11	S	5
18	.577	1	1,2,11	F	25
19	.577	1	1,2,11	F	21

Method 7