USING PROBABILISTIC MODELS OF DOCUMENT RETRIEVAL WITHOUT RELEVANCE INFORMATION

W. B. CROFT
Department of Computer and Information Science, University of Massachusetts, Amherst
and
D. J. HARPER
Computer Laboratory, University of Cambridge

Most probabilistic retrieval models incorporate information about the occurrence of index terms in relevant and non-relevant documents. In this paper we consider the situation where no relevance information is available, that is, at the start of the search. Based on a probabilistic model, strategies are proposed for the initial search and an intermediate search. Retrieval experiments with the Cranfield collection of 1,400 documents show that this initial search strategy is better than conventional search strategies both in terms of retrieval effectiveness and in terms of the number of queries that retrieve relevant documents. The intermediate search is shown to be a useful substitute for a relevance feedback search. Experiments with queries that do not retrieve relevant documents at high rank positions indicate that a cluster search would be an effective alternative strategy.

INTRODUCTION

THE PROBABILISTIC models of document retrieval which have recently been proposed in the literature[1–5] have been successful both in improving the retrieval performance of experimental systems and in providing a theoretical basis for methods which have previously relied on heuristics. A major assumption made in these models is that *relevance information* is available. That is, some or all of the relevant and non-relevant documents have been identified. Partial relevance information can be obtained by retrieving documents on the basis of the query and presenting them to the user for judgement as relevant or non-relevant. This process of obtaining relevance information and using it in a further search is called *relevance feedback*. In general it is the information about the relevant documents that is the most important since the characteristics of the non-relevant documents can be approximated by those of the entire collection.[4] For relevance feedback to be effective, the initial search using the query should present relevant documents to the user in as many cases as possible. This paper investigates the application of probabilistic models to the initial search with the aim of improving the retrieval effectiveness of this search.

The design of methods for improving the initial search has been one of the major research topics in information retrieval. The simplest approach to this search is to rank the documents according to the number of index terms in common with the query (sometimes called a co-ordination level search). Some of the modifications which have been proposed are

Journal of Documentation, Vol. 35, No. 4, December 1979, pp. 285–295.

(*a*) Using a normalized query–document similarity measure, such as the Cosine Correlation.[6]
(*b*) Weighting the terms using inverse document frequencies.[7]
(*c*) Searching clusters of documents rather than the documents themselves.[8]

These methods differ from the approach taken in this paper in that we use the same probabilistic model for the initial search as is used for the relevance feedback search. The retrieval process can be viewed as a two-stage application of a probabilistic model where the main difference between the stages is the increase in the amount of relevance information available.

The second stage of the retrieval process depends heavily on the relevance information obtained in the first stage by presenting documents to the user. Since the user will only want to judge a relatively small number of documents, the evaluation of the retrieval experiments reported here will emphasize the ability of the initial search to retrieve relevant documents at the top end of the ranked list of documents. This leads to the associated problem of methods for dealing with queries which do not retrieve relevant documents at the top of the ranked list.

THE PROBABILISTIC MODEL

Each document is assumed to be described by a binary vector $\mathbf{x} = (x_1, x_2, \ldots x_v)$ where $x_i = 0$ or 1 indicates the absence or presence of the ith index term. A decision rule can be formulated by which any document can be assigned to either the relevant or the non-relevant set of documents for a particular query. The obvious rule is to assign a document to the relevant set if the probability of the document being relevant given the document description is greater than the probability of the document being non-relevant, that is if

$$P(\text{Relevant}|\mathbf{x}) > P(\text{Non-Relevant}|\mathbf{x})$$

A more convenient form of the decision rule can be found by using Bayes' theorem. This new rule, when expressed as a weighting function is,

$$g(\mathbf{x}) = \log P(\mathbf{x}|\text{Relevant}) - \log P(\mathbf{x}|\text{Non-Relevant})$$

This means that instead of making a strict decision on the relevance of a document, the documents are ranked by their $g(.)$ value such that the more highly ranked a document is, the more likely it is to be relevant.

The probabilities $P(\mathbf{x}|\text{Relevant})$ and $P(\mathbf{x}|\text{Non-Relevant})$ are difficult to calculate directly. However, they can be approximated in a number of different ways. If the assumption is made that the index terms occur *independently* in the relevant and non-relevant documents then

$$P(\mathbf{x})|\text{Relevant}) = P(x_1|\text{Relevant})\, P(x_2|\text{Relevant}) \ldots P(x_v|\text{Relevant})$$

and similarly for $P(\mathbf{x}|\text{Non-Relevant})$.

Let $\quad p_i = P(x_i = 1|\text{Relevant})$ and $q_i = P(x_i = 1|\text{Non-Relevant})$

where these are the probabilities that an index term occurs in the relevant and non-relevant sets respectively.

Then

$$P(\mathbf{x}|\text{Relevant}) = \prod_{i=1}^{v} p_i^{x_i} (1 - p_i)^{1-x_i}$$

$$P(\mathbf{x}|\text{nNon-Relevant}) = \prod_{i=1}^{v} q_i^{x_i} (1 - q_i)^{1-x_i}$$

and

$$g(\mathbf{x}) = \sum_{i=1}^{v} x_i \log \frac{p_i(1 - q_i)}{(1 - p_i) q_i} + \sum_{i=1}^{v} \log \frac{1 - p_i}{1 - q_i}$$

The second term of this function will be constant for a given query and will not affect the ranking of the documents. The first term involves a summation over all the terms in the document collection but for reasons explained by van Rijsbergen,[3] this summation is usually restricted to just the query terms. This function is then equivalent to a simple matching function between the query and the documents where query term i has the weight $\log p_i(1 - q_i)/(1 - p_i)q_i$. This model was first used by Robertson and Sparck Jones.[1] If the terms are assumed to be *not* independently distributed, then more accurate approximations for $P(\mathbf{x}|\text{Relevant})$ and $P(\mathbf{x}|\text{Non-Relevant})$ are possible.[3] In this paper we shall (mainly) use the simpler model based on the independence assumption.

When the model is applied to a retrieval system, the binomial parameter p_i is estimated from the sample of relevant documents obtained by user judgements and q_i is usually estimated from the total collection of documents. In the initial search, that is, prior to relevance feedback, we have no information about the relevant documents and we could therefore assume that all the query terms had equal probabilities of occurring in the relevant documents. The effect of this assumption can be seen by splitting the first term of $g(\mathbf{x})$ into two parts.

$$\sum_i x_i \log \frac{p_i}{1 - p_i} + \sum_i x_i \log \frac{1 - q_i}{q_i}$$

where the summation is over all query terms. If we assume that all p_i are the same, then the first part of the above expression is simply a constant ($\log p_i/1 - p_i$) times the number of common terms between the query and the document. If we estimate q_i by n_i/N where n_i is the number of documents in which the term i occurs and N is the size of the collection, then the second part of the expression is

$$\sum x_i \log \frac{N - n_i}{n_i}$$

For large N this query term weight is very similar to the inverse document frequency weight used by Sparck Jones and Bates,[7] $\log N/n_i$.* Therefore this probabilistic model indicates that in the case where no relevance information is available, the best function for ranking the documents is a combination of a simple match and a match using inverse document frequency weights. Previous

* They implement this weight as $\log (\max n_i)/n_i$, where $(\max n_i)$ is the maximum value of n_i observed for the particular document collection. The same implementation is adopted in the experiment reported in this paper.

experiments have of course used either one or the other of these methods but not both together. This function shall be referred to as the *combination match*.

As an example of the difference between the initial searches mentioned, consider the binary query description $\mathbf{r} = (r_1, r_2, \ldots r_v)$ and a document description $\mathbf{x}$. The simple match for this pair is

$$\sum x_i r_i$$

The summation is now over all terms because the query appears explicitly. The match with inverse document frequency weights is

$$\sum x_i r_i \log N/n_i \text{ implemented as } \sum x_i r_i \log (\max n_i)/n_i$$

The match with the Cosine Correlation is

$$\sum x_i r_i / (\sum x_i \sum r_i)^{\frac{1}{2}}$$

and the combination match is

$$C \sum x_i r_i + \sum x_i r_i \log (N\text{-}n_i)/n_i$$

where C is a constant. An interesting special case of the combination match is obtained by allowing p_i to approach 1·0, in which case the constant C becomes very large. The combination match then specializes to ranking the documents by inverse document frequency weighting (approximately) within co-ordination level, which will be referred to as the co-ordination/IDF match.

Another possible application of the model is to introduce an intermediate search between the initial search and the relevance feedback search. The documents at the top of the ranking produced by the initial search have a high probability of being relevant. The assumption underlying the intermediate search is that these documents *are* relevant, whether in actual fact they are or not. Therefore before asking the user to give relevance judgements, a search would be performed in which the top few documents are used to provide estimates for p_i. The effect of this search will vary widely from query to query. For a query which retrieved a high proportion of relevant documents at the top of the ranking, this search would probably retrieve additional relevant documents. For a query which retrieved a low proportion of relevant documents, this search may actually downgrade the retrieval effectiveness. The intermediate search would therefore be most useful if the process of interacting with the user to obtain relevance judgements is considered very expensive or even impractical by either the user or the system designer.

A different application arises when the query does not retrieve any relevant documents at a particular cutoff. In this case the user has judged all the retrieved documents as being non-relevant, whereas the intermediate search is performed before the user looks at any documents. The technique of using the characteristics of the judged non-relevant documents to perform another search for relevant documents (negative feedback) has been shown to have some merit by Ide.[9] The application of the probabilistic model to negative feedback could be done by performing an initial search with either of the following changes,

(*a*) The terms in the judged non-relevant set of documents could provide estimates for q_i, in which case all p_i values remain equal.
(*b*) The terms in the non-relevant documents could be used to vary the estimates for p_i. If a query term occurs frequently in the judged non-relevant set, the p_i values for the term would be reduced. Experimental evidence suggests that the latter method is to be preferred.

In summary then, the combination match, the intermediate search and the negative feedback search are all based on the same probabilistic model and in each case no relevant documents are known. The difference in the searches is in the method of estimating the p_i values. The three searches are tested with retrieval experiments in the following section.

THE EXPERIMENTS

The experiments reported here were done with the Cranfield C1400I collection of 1,400 documents and 225 queries with binary indexing. A full description of this collection can be found in Sparck Jones and Bates.[7] Three evaluation methods are used. The first is a precision-recall table which gives average precision values at standard recall levels. The exact method of calculation of these values is described by Harper and van Rijsbergen.[4] To emphasize the performance of the searches at the top end of the rankings, the E measure[8] is used as the second evaluation method. The E measure is a weighted combination of precision and recall such that *the lower the* E *value, the better the effectiveness*. The parameter β is used to reflect the emphasis on precision or recall. $\beta = 1$ corresponds to attaching equal importance to precision and recall. $\beta = 0{\cdot}5$ and 2 corresponds to attaching half and twice as much importance to recall as to precision respectively. The E measure evaluates a set of retrieved documents so a ranked document list must be evaluated at cutoff points. However this has the advantage that the ranks within the retrieved set do not affect the evaluation whereas precision-recall figures can be very sensitive to the exact rankings. For example, at a cutoff of say ten documents the E measure considers simply the number of relevant documents in this set rather than their positions. Since the user must examine the ten documents anyway for the relevance feedback process, the E measure does seem appropriate.

Another advantage of the E measure is that significance tests are easy to apply because there is a single E value (for a given value of β) for each query rather than a set of recall-precision values. The significance test used here is the Wilcoxon signed-ranks test with a significance level of 5%. The E evaluation in the experiments is presented as average E values (over 225 queries) for $\beta = 0{\cdot}5,1,2$ at two arbitrary cutoffs of ten and twenty documents from the top of the document ranking.

The third evaluation method is simply to list the number of queries that do not retrieve relevant documents and the total number of relevant documents retrieved over all queries at a particular cutoff in the ranking. This gives a direct indication of the amount of relevance information that can be obtained from the user after the initial or intermediate search. The first figure is the most important because relevance feedback can be effective even with a single relevant document.[5]

The first experiment was to compare the usual methods for the initial search—the simple match (COORD), the match with inverse document frequency

weights (INVWT) and the match using the Cosine Correlation (COS). Table 1 gives the results for these searches on the C1400I collection. The cutoff values used in the evaluation were ten and twenty documents. Values any higher than this would probably require too much effort for the user to judge documents for relevance.

These retrieval results indicate that there is little difference between the three initial searches. However, it does seem that the COS search does slightly better in terms of precision whereas the INVWT search does slightly better in terms of recall. In fact the only significant difference in the E values according to the Wilcoxon test is that the INVWT value for $\beta = 2$ (recall-oriented), cutoff = 20 is better than the other searches. In terms of the relevance information provided, the COS search is the best at cutoff = 10 whereas the INVWT search is the best at cutoff = 20.

TABLE 1. *Comparison of the COORD, INVWT and COS initial searches*

Recall	*Precision*		
	COORD	*INVWT*	*COS*
10	47·5	45·8	49·1
20	41·2	39·7	43·2
30	34·4	33·5	34·9
40	30·0	30·3	29·9
50	27·6	27·7	27·5
60	20·5	22·1	21·0
70	16·5	18·2	16·1
80	12·9	15·1	13·7
90	10·1	11·5	10·0
100	9·5	11·0	9·2

Average E *values*

Search	*Cutoff* = 10			*Cutoff* = 20		
	β = 0·5	1·0	2·0	0·5	1·0	2·0
COORD	0·79	0·77	0·73	0·84	0·80	0·73
INVWT	0·80	0·78	0·73	0·84	0·79	0·71
COS	0·79	0·77	0·73	0·84	0·80	0·72

Relevance Information

Search	*Cutoff* = 10		*Cutoff* = 20	
	# queries that fail	*Total rels retrvd*	*# queries that fail*	*Total rels retrvd*
COORD	49	447	32	621
INVWT	50	432	28	648
COS	46	454	32	633

The next experiment tested the combination match. A number of searches were performed with the constant value of p_i set to values between 0·3 and 0·9. As p_i was increased from 0·3 to 0·6, material improvements in the search results were obtained. Increasing p_i above 0·6 improved the searches, but only margin-

ally. As expected the results for $p_i = 0{\cdot}5(C = 0)$ were almost identical to the INVWT results. The best* search results (COMB) for p_i set to 0·9 are given in Table 2. Search results (COOINV) are also given for the co-ordination/IDF match. This table shows that these match functions produce very similar results, and more importantly that they are the best overall initial searches. At a cutoff of ten documents, the COMB search is not significantly different to the other three searches but it provides approximately the same relevance information as the COS search. At a cutoff level of twenty the E values from the COMB search are significantly better (according to the Wilcoxon test) than the values for the other searches at each β value. The relevance information provided at this cutoff is much better than any other search.

TABLE 2. *The COMB and COOINV initial searches*

Recall	*Precision*	
	COMB	*COOINV*
10	48·1	48·2
20	41·6	41·6
30	35·3	35·2
40	31·8	31·7
50	28·5	28·5
60	22·7	22·2
70	18·6	18·0
80	15·5	14·8
90	11·4	11·1
100	10·9	10·6

Average E *values*

Search	*Cutoff* = 10			*Cutoff* = 20		
	$\beta = 0{\cdot}5$	1·0	2·0	0·5	1·0	2·0
COMB	0·79	0·77	0·72	0·83	0·79	0·70
COOINV	0·79	0·77	0·72	0·83	0·79	0·70

Relevance Information

Search	*Cutoff* = 10		*Cutoff* = 20	
	# queries that fail	*Total rels retrvd*	*# queries that fail*	*Total rels retrvd*
COMB	44	449	23	670
COOINV	45	447	24	663

That the best results for the combination match were obtained for p_i almost equal to 1·0, is a consequence of the collection used for the experiment. The queries of the C1400I collection were manually indexed, with the indexers instructed to select 'significant' words from the query texts.[7] It is reasonable to suppose that significant words are those with potentially high probabilities of

* The idea of trying the combination match with various values for p_i, and then regarding the best of the searches as the result of the model is a little suspect. It would be better to optimize p_i on one set of queries, and then test the combination match with the best value on another set. However, given the similar search results observed over a wide range of p_i values (0·6–0·9), it would seem that such sophistication is unnecessary.

occurring in relevant documents. Therefore the combination match with a high constant value for p_i would be expected to produce the best results. It is conjectured that for automatically indexed queries the best value for p_i will be much less than 1·0, and that the resulting combination match will be more like inverse document frequency weighting than co-ordination/IDF weighting. The advantage of the combination match over the other match functions is that it can be tailored to a particular document collection. Note that for some collections it may be that the best constant value for p_i will be less than 0·5.

The next experiment was to perform an intermediate search by assuming that the top few documents from the ranking of the initial search are relevant. Two main variations of this search were used. The first used just the query terms in the intermediate search but with p_i estimates based on a specified number of the top documents. For this collection, the best performance was obtained by using the top five documents (and the search will be referred to as TOP5). The second variation of the intermediate search was to expand the query with terms from a maximum spanning tree of all terms in the collection. This maximum spanning tree is the tree of maximum edge weight connecting the index terms where the edge weight between two terms is a measure of the dependence of those terms. This form of query expansion, which comes from assuming that terms do not occur independently, is described in Harper and van Rijsbergen.[4] The p_i estimates for the expanded set of terms are found from the top few documents of the initial search, as for TOP5. The effect of this expansion is to provide extra terms which are not in the query but which may be in relevant documents. This search performed best when the top three documents of the initial search were used and it shall be referred to as EXTOP3. The initial search used for this experiment is the simple match (COORD) and the results for the intermediate searches given in Table 3 should be compared to the results for this search.

The significance tests on the E values showed that EXTOP3 is more effective than COORD at both cutoff levels and for all values of β. Even the simple TOP5 strategy was significantly better at a cutoff of ten documents, but not at twenty. However, the relevance information shows that while the total number of relevant documents retrieved has increased compared to the COORD search, the number of queries which do not retrieve any relevant documents at these cutoff levels also increases. Therefore there would not seem to be any advantage in having an intermediate search before a relevance feedback search. Any extra relevant documents retrieved by the intermediate search would almost certainly be retrieved by a relevance feedback search directly after an initial search. The intermediate search could however be used as a substitute for the relevance feedback search in systems where the interaction with the user was considered impractical.

The final experiment investigates methods of dealing with queries that retrieve no relevant documents at high cutoff levels. Even for the best initial search (COMB) there were still twenty-three of these queries at a cutoff of twenty documents. The best strategy used for the probabilistic model was to estimate p_i for the query terms by

$$p_i = 0{\cdot}5 + 0{\cdot}4 \times (1 - f_i)$$

where f_i is the ratio of the number of occurrences of term i in the judged non-relevant set to the size of that set. Therefore the p_i values range from 0·5 for a

TABLE 3. *The TOP5 and EXTOP3 intermediate searches*

Recall	*Precision*	
	TOP5	*EXTOP3*
10	49·4	47·3
20	43·9	41·6
30	36·7	35·1
40	32·1	31·2
50	29·1	29·0
60	23·1	24·0
70	18·6	20·3
80	15·1	17·2
90	11·4	13·1
100	10·7	12·5

Average E *values*

Search	*Cutoff* = 10			*Cutoff* = 20		
	$\beta = 0{\cdot}5$	1·0	2·0	0·5	1·0	2·0
TOP5	0·78	0·76	0·72	0·84	0·80	0·72
EXTOP3	0·78	0·76	0·71	0·83	0·79	0·70

Relevance Information

Search	*Cutoff* = 10		*Cutoff* = 20	
	# queries that fail	*Total rels retrvd*	*# queries that fail*	*Total rels retrvd*
TOP5	53	470	36	644
EXTOP3	54	469	39	673

query term which occurred in all the judged non-relevant documents to 0·9 for a term which occurred in none. This strategy is evaluated by giving the number of queries (out of twenty-three) that retrieved relevant documents in the top ten and twenty positions of the new ranking, excluding documents already seen by the user. The results for this search (NEGFD) and two other searches appear in Table 4. The search called EXTRA is simply looking at the next ten and twenty documents (after the first twenty) of the original COMB search. The other search, CLUSTER, uses the twenty-three queries for a bottom-up search of a cluster hierarchy.[10] It has been shown that this type of cluster search which retrieves one cluster of documents can achieve very high precision results but in many cases may not retrieve any relevant documents at all. Therefore the cluster search is not suitable for the initial search prior to relevance feedback in general but because it uses a different approach it may be a useful alternative strategy.

Table 4 shows that for this collection there is no advantage in using a negative feedback strategy because similar results can be obtained by simply looking at more documents from the initial search. The CLUSTER search seems to be particularly useful both in terms of the number of queries that retrieved relevant documents and the small number of documents that would have to be examined. The disadvantage of the cluster search is the overhead involved in setting up and maintaining the cluster hierarchy. This is discussed in detail by Croft.[10] The CLUSTER search and the EXTRA search together retrieved relevant documents

for fourteen out of the twenty-three queries. Therefore it would seem advantageous to both retrieve a cluster and examine a few more documents from the initial search.

TABLE 4. *The NEGFD, EXTRA and CLUSTER searches with twenty-three queries that retrieved no relevant documents at cutoff twenty in the initial search*

Search	*#queries that retrieved relevant docs in*	
	next 10 docs	*next 20 docs*
NEGFD	5	4
EXTRA	5	4
CLUSTER	11*	–

* The average size of the retrieved clusters was three documents.

CONCLUSION

A probabilistic model of document retrieval was applied to two searches which can occur before relevance feedback—the initial search and the intermediate search. For the initial search there is no relevance information available whereas for the intermediate search the relevance information is derived from the top-ranking documents of the initial search. The experiments using the C1400I collection showed that the initial search based on this model and called the combination match performed better than the simple match, the match using inverse document frequency weights and the match using the Cosine Correlation. The improvements in retrieval effectiveness obtained were small but significant and this search also provided the most relevance information. However, this result should be taken as just an indication that the combination match is the most effective initial search because the experiments have only been done with one medium-size collection. The intermediate search, because it increases the number of queries that do not retrieve relevant documents at high ranks, would only be useful when a relevance feedback search was not possible or was too expensive.

A strategy based on the probabilistic model was also devised for queries which did not retrieve any relevant documents at high cutoff levels. This negative feedback strategy was shown to be ineffective for this collection since the same results could be achieved by examining more documents from the initial search. Finally, the results indicated that a cluster search would be a good alternative initial search strategy.

ACKNOWLEDGEMENTS

The work was carried out while D. J. Harper was in receipt of a CSIRO Postgraduate Studentship. Facilities were kindly provided by the Cambridge University Computer Laboratory. The authors thank Keith van Rijsbergen and Karen Sparck Jones for many helpful discussions. The document collection was generously made available by Karen Sparck Jones. The authors would like to thank the referees for their helpful suggestions.

REFERENCES

1. ROBERTSON, S. E. *and* SPARCK JONES, K. Relevance weighting of search terms. *Journal of the ASIS, 27*, 1976, 129–46.
2. YU, C. T. *and* SALTON, G. Precision weighting—an effective automatic indexing method. *Journal of the ACM, 23*, 1975, 76–88.
3. VAN RIJSBERGEN, C. J. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation, 33*, 1977, 106–19.
4. HARPER, D. J. *and* VAN RIJSBERGEN, C. J. An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation, 34*, 1978, 189–216.
5. SPARCK JONES, K. Search term relevance weighting given little relevance information. *Journal of Documentation, 35*, 1979, 30–48.
6. SALTON, G. *Automatic information organization and retrieval.* New York: McGraw-Hill, 1968.
7. SPARCK JONES, K. *and* BATES, R. G. Research on automatic indexing 1974–6. 2 vols. Computer Laboratory, University of Cambridge, 1977.
8. VAN RIJSBERGEN, C. J. *Information retrieval,* 2nd edition, London: Butterworths, 1979.
9. IDE, E. Relevance feedback in an automatic document retrieval system. M.Sc. thesis, Report ISR-15 to the National Science Foundation, Department of Computer Science, Cornell University, Ithaca, N.Y., 1969.
10. CROFT, W. B. Organizing and searching large files of document descriptions. Ph.D. thesis, Computer Laboratory, University of Cambridge, 1979.

(*Revised version received 19 October 1979*)