SOME CONSIDERATIONS FOR APPROXIMATE OPTIMAL QUERIES

K. L. Kwok

Computer Science Department, Queens College,
City University of New York, Flushing, New York 11367, USA

Abstract

An optimal query has been defined as one which will recover all the known relevant documents of a query in their best probability of relevance ranking. We have slightly modified the definition so that it also allows one to trace its evolution from the original to the optimal via the various feedback stages. Such a query can be constructed by modifying the original query with terms from the known relevant documents. It is pointed out that such a term addition strategy differs materially from other approaches that add terms based on term association with all query terms, and calculated from the whole document collection. The effect of viewing a document as constituted of components, and hence affecting the weighting and retrieval results of of the optimal query, is also discussed.

1. Introduction

Given a query statement representing the needs of a user, an Information Retrieval (IR) system attempts to arrange the items in a document collection in descending order of probability of relevance to the user. In this paper, we will be interested in queries that are represented simply as a vector of weighted index terms [1]. It is generally true that the query first presented by the user may not be the best formulation of his or her needs, and that some modifications (such as term addition, deletion, or simply term weight changes) is necessary to obtain better retrieval results. There are many ways by which an original query may be modified such as using a predefined thesaurus, which may be manually generated based on term meaning [2], or automatically generated based on term co-occurrence statistics [1,3,4]. Recent investigations have relied more heavily on user feedback information, rendering the modifications more adaptive. For example, in [5,6], query terms were re-weighted (based on a term independence model) after some relevance information was obtained from the user, and this could iteratively approach an optimal upper-bound retrieval result. In [7,8,9,10], the above technique was extended to include query term additions as well. First a structure of term association, which may be a maximum spanning tree or a graph, was determined from the whole document collection, and this was used to identify candidate terms for addition to the query. Feedback relevance information was then employed to provide term dependency weighting for the expanded query term set. Significant improvements to retrieval results have been observed in all these methods compared to those obtained without query re-weighting or modification.

In these approaches, the structure of term association that has to be constructed initially for each collection can be quite expensive. In a real-life environment, the collection can grow substantially in a period of time, and one has to provide for a strategy of whether and how to modify this structure dynamically, or to re-structure from scratch. In addition, the mathematical formulation of term dependency leads to a non-linear similarity function between the query and a document, which is not easy to deal with. It is also not clear how much improvements to retrieval results is due to the incorporation of term dependency, and how much is due to term addition only. The latter is indirectly dependent on term dependency information. For example, although in one experiment in [8] it was shown that the explicit term dependency formulation was found to be more important than term addition, in another [8] it has been found that a linear similarity function with the appropriate term

weighting, together with query term expansion, could be as effective as the non-linear similarity function with the same term expansion.

Since the explicit term dependency formulation requires much more parameter estimation and hence difficult to use, it is generally preferable to assume term expansion can take care of a large part of the dependency, and to use term independent weighitng for the expanded query [9,11]. In this paper we would like to explore less expensive methods of identifying terms to add to a query. In [12], the concept of an optimal query has been introduced. It was shown that, if one knows in advance the answer documents to a query, then the best query that can retrieve these answers can be defined. In [13], Yu pursued the same reasoning, and in [14] Chow and Yu defined an optimal query that lead to retrieval results ordered on the probability of relevance. These approaches can be viewed from the idea of document indexing based on a Principle of Document Self-Recovery that we recently introduced [15], and in this paper we would like to re-examine these issues.

2. Indexing based on a Principle of Document Self-Recovery

In [15,16] we have introduced a method for the indexing and weighting of a document. The idea is that to index a document, we need to consider its relationship with the other documents of the collection. Given a document called the source, if a few associated ones relevant to it are also known, then its topics and contents could be represented (indexed and weighted) in such a way that its representation would recover this relevant set of documents optimally amongst the collection. Optimal here means that if this representation is regarded as a 'query', then the relevant set of documents would be retrieved in their best order based on the odds function of the probability of relevance, namely, $P(+R|doc)/[1-P(+R|doc)]$, where $P(+R|doc)$ is the probability that given the representative features of 'doc', it will be found relevant to the topics of the source. Thus, it is found that the source document should be indexed by all terms present in the relevant set including the source itself, and that each term, say term k, should be weighted by the familiar function:

$$w_k = \log \left[\frac{p_k}{1-p_k} \Big/ \frac{q_k}{1-q_k}\right] = wp_k - wq_k \qquad (1a)$$

with

$$p_k = P(y_k=1|+R),\ q_k = P(y_k=1|-R) \qquad (1b)$$

Here, p_k (q_k) is the probability that given relevance +R (non-relevance -R) to the source document, that term k will be present (y_k=1). The usual assumption of term independence in the relevant and non-relevant set has been used. Thus, every document can be represented in an optimal fashion that makes use of all known information about it and its relationship with the other items of the collection. When a collection does not contain knowledge of the relevant associates of a source document (which is usually the case), one can still apply the theory to the components of a document as discussed in [15], the set of components playing the part of a relevant set. Components are text units of unambiguous concepts and are to be independent. Depending on what we choose as document components, we would obtain different weighting formulae based on Equation 1. This has been called indexing based on a Principle of Document Self-Recovery [15]. The candidate components may be single stems, phrases, sentences, or the whole document (abstract).

In the next section, we will apply the above reasoning to the relationship of a query (being modified via feedback information) with its answer documents, and to its representation. It turns out that this view has certain advantages compared to some previous studies of the same situation.

3. Query Modification during Relevance Feedback

As discussed earlier, it is useful to consider the retrieval operation as an iterative process. The query is modified as more relevance feedback information is gained, and eventually approaches the optimal one when all relevant documents are known. Since the original query statement first proposed by the user describes the topics wanted (perhaps approximately or incompletely), it should be taken as fully relevant to the user's intended content needs. Thus we would regard the original query itself as a fictitious but relevant (short) document and is now added to the document collection, thus incrementing the size of the collection from N to N+1, and slightly perturbing the other term frequencies as well. This is done so that we could make use of all available information, especially information about relevant item characteristics, and the query is regarded as one of them. In what follows we will consider the query as it goes through the various stages of relevance feedback iteration.

3.1 Initial Stage

At this stage, which we may call the 0-th iteration, the only item known to be relevant is the fictitious document (query) itself. When this is considered in relation to the collection which has been augmented to size N+1, we end up with the following 2x2 table for each query term, say term k:

	+R	-R	
+Idx	1	n_k	n_k+1
-Idx	0	$N-n_k$	$N-n_k$
	1	N	N+1

(Here n_k is the document frequency for term k of the original collection, $\pm$R denote relevance/non-relevance to the query, and $\pm$Idx denote the term being used for indexing or not). Substituting for the probability definitions in Equation 1, we are led to the following weighting formula for each query term:

$$w_k = C + \log (N-n_k)/n_k \qquad (2)$$

where C is (theoretically) a large constant. This is exactly the Combination Match formula introduced by Croft and Harper in [17]. However, the estimation of C is unsound at this stage because the number of relevant item is only one. It is usual to set this constant to 0, leaving only the logarithm term, which is essentially the Inverse Document Frequency (IDF) weighting introduced by Sparck-Jones [18], and which has been experimentally determined to give good retrieval results under many circumstances. Thus, to start the retrieval process, the best one could use is the original query weighted as in Equation 2. The above argument is equivalent to using the whole document as a component; if other linguistic constructs are used as components, other formulae will result, see [15].

3.2 The i-th Iteration Stage

At this stage, we would have accumulated all the known relevant documents up to the (i-1)-th stage, numbering c^1. Our aim is to make use of these c^1 relevant items and the original query (totalling c^1+1) to define an effective query representation for the i-th retrieval. The original query is intrinsically and fully relevant. Each of the c^1 documents however, although identified as relevant in an overall fashion by the user, may contain components that are not useful or necessary for the retrieval. As explained in [15], documents can generally be multi-disciplinary, involving many topics and concepts, and that a user may find only a portion of the item useful and relevant. These documents we would call them external relevant items. It would be extremely useful if relevance feedback can involve the user to provide information on which components of a document (such as terms, phrases, sentences, etc.) are actually describing the topics of retrieval, thus allowing for much sharper and more precise feedback. Unfortunately current systems or evaluated databases generally do not have these finer details available. Under this limited circumstance, methods must be devised to select from the external relevant items only those index terms that are deemed most useful for modifying the original query. Indiscriminate additions of terms may make the modified query migrate to unwanted territories and may lead to worse retrieval than without modification [11]. In addition, although one is supposed to incorporate all index terms from all relevant items to define the query at this stage, this can lead to a representation that may be hundreds of terms long and would be unwieldly large. Some truncation method is therefore a necessity, and hence this also calls for more specific feedback information from the user.

Methods for helping to select the useful terms for query modification from the set of candidate terms formed from all the external relevant items appear to be very limited, because of the scarce information available at this stage. Incorporation of the original query, which is the only intrinsic relevant item, should have a stabilizing effect to prevent the modified query from deviating from the original content needs. This paper does not deal with this problem, but rather looks at the easier final stage when all relevant items are known, as discussed in the next section.

3.3 The Final (Optimal) Stage

At the final stage, all relevant documents are known and they have been employed to define an optimal query in [14]. This definition specifically excludes the original query. However, as part of the iteration chain presented in the previous sections, it appears to be logical and would be very useful to include the original query (i.e. the fictitious document) among the relevant items, especially in the intermediate stages. We will therefore count this fictitious document in when defining the optimal query. The advantages of doing this for weighting purposes will be apparent in our discussion in Section 4. In [5], Robertson and Sparck-Jones also used an optimal query for their retrospective retrieval, leading to their upper bound results. The difference is that they considered the query terms as fixed and static, only the

weighting was modified.

4. Approximations to the Final Stage Optimal Query

At the final feedback stage, all relevant items are known. Chow and Yu [14], as well as our Principle of Document Self-Recovery (Section 2), have shown how the optimal query Q^* should be constructed. At this point, we would like to make an observation. Let us partition the optimal query into three sets of terms: $Q^* = Q^*_1 \cup Q^*_2 \cup Q^*_3$, where Q^*_1 contains original query terms that also occur in the (external) relevant documents, Q^*_2 contains query terms that do not, and Q^*_3 contains terms from the relevant documents but not in the original query, see Fig. 1. The optimal query therefore implies that any additional terms to be used for expanding the original query should be terms associated only with those of Q^*_1, and occurring in the relevant set only. This differs materially from the approach taken in [7,8,9,10]. There, the method of term addition is based on the assumption that all the original query terms are necessary, and that terms highly associated with them in the whole collection (i.e. relevant and non-relevant) would also be useful. Highly associated terms are usually regarded as semantically associated as well. However, these terms are dependent on the characteristics of the collection as a whole, but may not necessarily reflect the specific content required of a particular query. Thus, if we are to believe in the optimal query, then the latter approach might not be adding the most effective terms. This observation is also suggested in [11].

Considering the three sets of terms in the optimal query again, we see that the first set is the most significant because they are used in the original query and confirmed in some relevant documents. The second set contains terms that do not appear in any relevant documents (except in the fictitious one). They would directly affect the ranking of non-relevant documents only, and some or all of them may perhaps be truncated. The third set are terms associated with the first set and appearing in some relevant documents, but there are usually too many of them and have to be truncated. Hence, as in Section 3.2, it is desirable to have user relevance feedback information at the component level. In [14], a method of approximating the optimal query has been considered. With each of the terms in the three sets, we can estimate a weight given by Equation 1. Let us sort the terms in descending order of their $|w_k|$ values. Chow and Yu have proved that if a single term is to be removed from the optimal query, then the last one with the smallest $|w|$ value should be chosen, because it will lead to the least disturbance from the optimal arrangement of documents. The proof was involved, and it does not seem to have been extended to justify using the same consideration for removing additional terms. We do not have a proof either. A trivial case that will satisfy the above process for all terms is when the sequence of $|w_k|$ values obey the folowing criteria:

$$|w_k| = \Sigma_{j>k} |w_j| \quad \text{for all } k. \qquad (3)$$

For then, when we consider creating the query by adding one term at a time according to the sequence of descending $|w_k|$ values, the pairs of new document index term patterns spawned at each step never cross each other on the w scale, and the optimal arrangement of documents will be retained at all steps of single term removal in ascending order of $|w_k|$ (Fig. 2). However, we cannot expect the condition of Equation 3 to be satisfied in real situations. Intuitively, extending Chow and Yu's method to include the removal of additional terms seems to make sense. For, a term with positive w_k (i.e. $wp_k > wq_k$, and hence $p_k > q_k$) will serve to promote a larger proportion of relevant items than non-relevant ones, and a term with negative w_k will serve to demote also a larger proportion of non-relevant items than relevant ones on the average. The larger a value it is, the more effective it will be. Hence a workable strategy may be to retain candidate terms that have the largest $|w|$ values, and is simple to implement. Either a cut-off value of $|w|$ (obtained from a $|w|$ versus term rank curve), or simply a predetermined query size can be used for truncation. What is a standard size is of course highly subjective, but our experience with profiles of document clusters [19] suggests that a size of an average dcoument (somewhere between 15 to 35 terms) would probably be appropriate.

As discussed in Section 3.3, our definition of the optimal query includes the original query, so that we can trace the query evolution from the initial to the final stage. We believe it has value in that it plays a part in stabilising the retrieval to the content of the user's needs (except in the case when the original query is completely misleading -- probably very rare). It also has the advantage that some of the probability estimates for Equation 1 may be made definite under some circumstances. For example, all candidate terms would have occurrence frequency of at least one in the relevant set and the problems of estimation mentioned in [9] could be avoided somewhat. When the total number of known relevant documents (c^*) is small, say two or three, many candidate terms will appear in all of them, leading to a pathological estimate of $p_k = 1$ if only relevant documents are used. Inclusion of

the original query in these circumstances however give an estimate of $p_k = c^*/(c^*+1)$, if these terms do not appear in the original query. When c^* is large, the presence of this query will have immaterial effect. In the case of a term present in both the original query and all the relevant documents, we have a degenerate case of $p_k = 1$; in this situtation however, we have reason to assign a value approaching 1 for p_k. If the terms in the Q^*_2 set are ignored, our definition of an optimal query would be similar to that of [14], except for the weighting.

We are currently designing programs to perform experiments on this process of optimal query construction and using various approximation methods. We hope that results and observations with this final stage may help us in the construction of effective queries for the intermediate stages as well. The above discussion has been considered with the whole document regarded as a monolithic unit for weighting purposes. As discussed in [15], it is also reasonable to view each document as constituted of components, such as single stems or sentences. These various modes can lead to quite different weighting schemes and hence to different optimal queries. The mode that gives the best retrieval result at this final stage may also provide judgment as to which component type best describes the document collection.

Acknowlegment

This work was partially supported by a grant from the Professional Staff Congress of the City University of New York (PSCUNY 6-65238).

Reference

1. Salton,G.;McGill,M.J. Introduction to modern information retrieval. New York: McGraw Hill; 1983.
2. Salton, G. Automatic information organization and retrieval. New York: McGraw Hill; 1968.
3. Salton, G.; Yang, C. S.; Yu, C.T. "A theory of term importance in automatic text analysis." J. of ASIS. 26:33-44; 1975.
4. Sparck Jones, K. Automatic keyword classification for information retrieval. Connecticut: Archon Books; 1971.
5. Robertson, S. E.; Sparck Jones, K. "Relevance weighting of search terms." J. of ASIS. 27:129-146; 1976.
6. Sparck Jones, K. "Experiments in relevance weighting of search terms." Info. Proc. Mgmnt. 15:133-144; 1979.
7. van Rijsbergen, C.J. "A theoretical basis for the use of co-occurrence data in information retrieval." J. of Doc. 33:106-119; 1977.
8. Harper, D.J.; van Rijsbergen, C.J. "An evaluation of feedback in document retrieval using co-occurrence data." Journal of Documentation. 34:189-216; 1978.
9. van Rijsbergen, C. J.; Harper, D.J.; Porter, M. F. "The selection of good search terms." Info. Proc. Mgmnt. 17:77-91; 1981.
10. Yu, C. T.; Buckley, C.; Lam, K.; Salton, G. "A generalized term dependence model in information retrieval." Info. Tech.: R&D. 2:129-154; 1983.
11. Smeaton A.F.;van Rijsbergen, C.J. "The retrieval effects of query expansion on a feedback document retrieval system." Computer J. 26:239-246; 1983.
12. Rocchio, J.J."Relevance feedback in information retrieval." in The SMART Retrieval Systems, ed. Salton, G. Englewood Cliffs: Prentice-Hall; 1971.
13. Yu, C. T.; Luk, W. S.; Cheung, T.Y. "A statistical model for relevance feedback in information retrieval." J. ACM. 23:273-286; 1976.
14. Chow, D.; Yu, C. T. "On the construction of feedback queries." Journal of the ACM. 29:127-151; 1982.
15. Kwok, K. L. "An interpretation of index term weighting schemes based on document components." 1986 ACM Conference on R&D in Information Retrieval, ed. Rabitti, F. pp.275-283. Pisa, Italy, September 8-10, 1986.
16. Kwok, K.L."A probabilistic theory of indexing and similarity meausre based on cited and citing documents." J. of ASIS. 36:342-351; 1985.
17. Croft, W. B.; Harper, D. J. "Using probabilistic models of document retrieval without relevance information." J. of Doc. 35:285-295; 1979.
18. Sparck Jones, K. "A statistical interprestation of term specificity and its application in retrieval." J. of Doc. 8:11-21; 1972.
19. Feinman,R.;Kwok, K.L. "Classification of scientific documents by means of self-generated groups employing free language." J. of ASIS. 24:382-396; 1973.

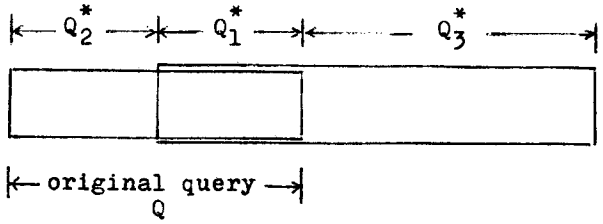

Fig. 1: Optimal Query $Q^* = Q_1^* \cup Q_2^* \cup Q_3^*$

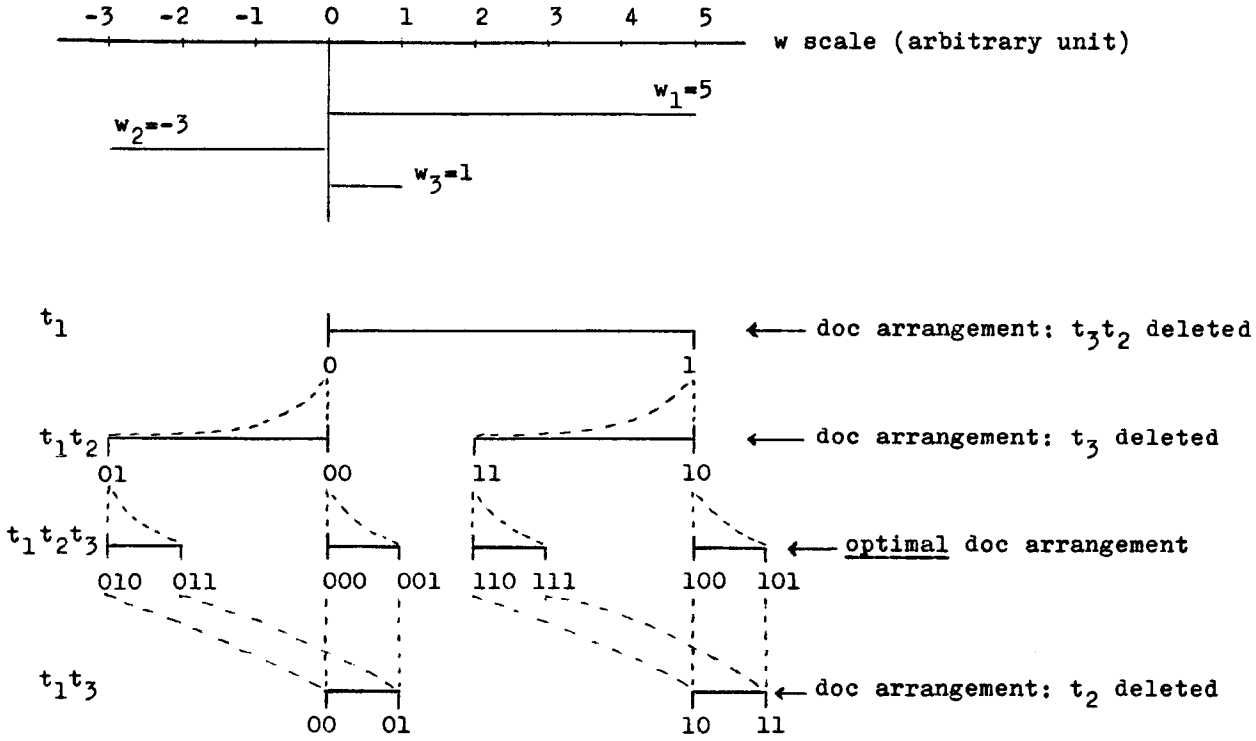

Fig. 2: Document Arrangement and Term Deletion for a Three Term Set: t_1w_1, t_2w_2, t_3w_3 Satisfying Equation 3