An Evaluation of Term Dependence Models in Information Retrieval

G. Salton[*], C. Buckley[*]
[*]Department of Computer Science
Cornell University
Ithaca, NY 14853/USA

C. T. Yu[†]
[†]Department of Information Engineering
University of Illinois/Chicago Circle
Chicago, IL 60680/USA

Abstract

In practical retrieval environments the assumption is normally made that the terms assigned to the documents of a collection occur independently of each other. The term independence assumption is unrealistic in many cases, but its use leads to a simple retrieval algorithm. More realistic retrieval systems take into account dependencies between certain term pairs and possibly between term triples. In this study, methods are outlined for generating dependency factors for term pairs and term triples and for using them in retrieval. Evaluation output is included to demonstrate the effectiveness of the suggested methodologies.

1. Term Dependency Models

From a decision-theoretic viewpoint, the information retrieval task is controlled by two probabilistic parameters which specify for each document of a collection the probability of relevance, and the probability of nonrelevance, with respect to a particular query. The larger the probability of relevance, and the smaller the probability of nonrelevance, the greater is the retrieval probability for the given item.

Consider in particular an item $\underline{x}$ in the data base represented by binary attributes $(x_1,x_2,\ldots,x_n)$, where x_i takes on the values 1 or 0 depending on whether the ith attribute is or is not assigned to item $\underline{x}$. For each item $\underline{x}$ and each query Q, it is in principle possible to generate the two parameters $P(\underline{x}|\text{rel})$ and $P(\underline{x}|\text{nonrel})$, representing the probabilities that a relevant and a nonrelevant item, respectively, has vector representation $\underline{x}$. Using decision theoretic considerations, it is easy to show that an optimal retrieval rule will rank the documents in decreasing order according to the expression

$$\frac{P(\underline{x}|\text{rel})}{P(\underline{x}|\text{nonrel})} \qquad (1)$$

That is, given two items $\underline{x}$ and $\underline{y}$, $\underline{x}$ should be retrieved ahead of $\underline{y}$ whenever the value of expression (1) for $\underline{x}$ exceeds the corresponding value for $\underline{y}$. [1-5]

This study was supported in part by the National Science Foundation under grant IST 81-08696.

The probablistic approach is of course useless in retrieval, unless methods can be found for estimating the probabilities $P(\underline{x}|s)$ for each item in the classes s of relevant and nonrelevant items, respectively. These probabilities will necessarily depend on the occurrence characteristics of the individual attributes x_i in the relevant and nonrelevant items of the collection. The class variable s will be dropped in the remainder of this study whenever possible because the development is identical for the relevant and nonrelevant classes of documents. The computation must be carried out separately for the two classes to obtain the retrieval function of expression (1).

An exact formulation for $P(\underline{x})$ which takes into account term dependencies of any order (that is, between term pairs, triples, quadruples, etc.) is given by the Bahadur-Lazarsfeld expansion (BLE) as follows: [5,6]

$$P(\underline{x}) = \prod_{t=1}^{n} p_t^{x_t} (1-p_t)^{1-x_t} \left[1 + \sum_{i<j} \rho_{ij} \frac{(x_i-p_i)(x_j-p_j)}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} \right.$$

$$+ \sum_{i<j<k} \rho_{ijk} \frac{(x_i-p_i)(x_j-p_j)(x_k-p_k)}{\sqrt{p_i p_j p_k (1-p_i)(1-p_j)(1-p_k)}} + \cdots$$

$$\left. + \rho_{12\ldots n} \frac{(x_1-p_1)(x_2-p_2)\ldots(x_n-p_n)}{\sqrt{p_1 p_2 \ldots p_n (1-p_1)(1-p_2)\ldots(1-p_n)}} \right] \qquad (2)$$

where p_k is the probability of occurrence of attribute k in the class under consideration, that is, $\mathrm{Prob}(x_k=1)$, and ρ_{ij}, ρ_{ijk}, etc., represent the second, third, and higher order correlations between term pairs $x_i x_j$, triples $x_i x_j x_k$, and higher order subsets of terms. Specifically,

$$\rho_{ij} = \frac{E[(x_i-p_i)(x_j-p_j)]}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} = \frac{E(x_i x_j) - p_i p_j}{\sqrt{p_i p_j (1-p_i)(1-p_j)}}$$

$$= \frac{p_{ij} - p_i p_j}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} \qquad (3)$$

$$\text{and} \quad \rho_{ijk} = \frac{E[(x_i-p_i)(x_j-p_j)(x_k-p_k)]}{\sqrt{p_i p_j p_k (1-p_i)(1-p_j)(1-p_n)}}$$

$$= \frac{E(x_i x_j x_k) - E(x_i x_j)p_k - E(x_i x_k)p_j - E(x_j x_k)p_i + 2p_i p_j p_k}{\sqrt{p_i p_j p_k (1-p_i)(1-p_j)(1-p_k)}}$$

$$= \frac{p_{ijk} - p_{ij}p_k - p_{ik}p_j - p_{jk}p_i + 2p_i p_j p_k}{\sqrt{p_i p_j p_k (1-p_i)(1-p_j)(1-p_k)}} \qquad (4)$$

where p_{ij} denotes the occurrence probability of the pair $x_i x_j$, p_{ijk} denotes the probability of the triple $x_i x_j x_k$, and so on. Corresponding expressions apply to the higher-order correlations.

The BLE expansion (2) cannot be used in practice unless the term occurrence probabilities p_i, p_{ij}, p_{ijk}, etc., can be obtained for all term combinations in both the relevant and nonrelevant document sets. In practice it is clearly impossible to compute the correlation coefficients for an exponential number of term combinations. Hence it becomes necessary to use some approximation to the exact formulation of expression (2). A great variety of simplified formulations may serve for this purpose among which the following may be of greatest interest.

a) The document terms could be assumed to occur independently of each other. This reduces expression (2) to the first term only. In particular, for the independent case

$$P(\underline{x}) = P(x_1) \cdot P(x_2) \dots . P(x_n) = \prod_{t=1}^{n} p_t^{x_t}(1-p_t)^{1-x_t} \qquad (5)$$

where x_i is again assumed equal to 1 or 0 depending on whether term x_i occurs in item $\underline{x}$ or not, and $p_i = P(x_i=1)$ and hence $(1-p_i) = P(x_i=0)$.

b) Additional terms beyond the first might be incorporated into the BLE expression. However instead of computing and storing an exponential number of higher order correlations, only the most important second and third order correlations could be used.

This implies that viable methods must be available for choosing the more important dependent term pairs and triples. Furthermore, special provisions must be involved to insure that only positive probability values are generated. Indeed, when the BLE expression (2) is used in a truncated, incomplete form, negative sums can be generated when the value of the negated terms exceeds that of the nonnegated terms. For example, when the joint occurrence probabilities p_{ij} for pairs of terms are close to zero, but the individual probabilities p_i and p_j are positive, expression (3) shows that ρ_{ij} becomes negative. Similarly, ρ_{ijk} and the other higher-order correlations can become negative under corresponding assumptions.

c) A different, simplified term dependence model might be used that is not based on the use of the BLE expression. The well-known <u>tree dependence</u> system which has been used experimentally in previous studies represents such a model. [7-9] In the tree dependence model each term is assumed to be dependent on at most one other term. Such a situation may be represented by a <u>precedence tree</u> where a given node in the tree is dependent only on the immediately preceding parent node on the next higher tree level. In the illustration of Fig. 1, nodes e,b,c,d, and h are dependent only on a; in addition, f and g are dependent on e, and i and j are dependent on h. The root node a is assumed to be independent of the other nodes.

By analogy with the independent case of expression (5), the occurrence probability $P(\underline{x})$ of each item $\underline{x}$ in the relevant and nonrelevant document classes may be expressed as a product of conditional, as opposed to simple probabilities in the tree dependence model. Thus

$$P(\underline{x}) = P(x_a)\left[\prod_E P(x_u|x_v)\right] \tag{6}$$

where term a is the root of the tree and v is the parent of u in the tree, the product being taken over all the edges E of the precedence tree. For the precedence tree of Fig. 1, expression (6) reduces to

$$P(\underline{x}) = P(x_a)P(x_e|x_a)P(x_b|x_a)P(x_c|x_a)P(x_d|x_a)P(x_h|x_a)$$

$$P(x_f|x_e)P(x_g|x_e)P(x_i|x_h)P(x_j|x_h)$$

The tree dependence model exhibits a number of advantages over the exact model provided by the BLE expression. Most obviously, the tree dependence expression (6) is more easily computed than the BLE expansion (2). Furthermore, since expression (6) represents a simple product of probabilities, the result remains positive when the original conditional probabilities are positive. The basic tree model does, however, suffer from the fundamental restriction that at most (n-1) pairwise dependencies are included in he dependence equation (6) for n originally available terms. The BLE system may then be expected to outperform the basic tree model in most retrieval environments.

A number of studies of the tree dependence system indicate, however, that the differences between the basic tree dependence and the BLE models may not be very substantial [10]:

a) In particular, it is known that for any term pair (u,v) which is not directly represented by an edge in the precedence tree, the corresponding correlation coefficient ρ_{uv} (see expression (3)) can be computed in the tree dependence model as the product of the pairwise correlations for all term pairs on the unique path from u to v in the tree. Considering, for example, the pair (f,b) in Fig. 1, the unique path from f to b in Fig. 1 covers edges (f,e), (e,a), and (a,b). Hence one has

$$\rho_{fb} = \rho_{fe} \cdot \rho_{ea} \cdot \rho_{ab} \quad . \tag{7}$$

This formula makes possible the effective inclusion of any term pair into the tree dependence model, whether the pair is represented by an edge in the tree, or not.

b) In addition, the tree dependence model may be extended by including certain dependencies between term triples. In particular, each time an edge is implicitly added to the precedence tree using the extension of equation (7), a triangle is formed which translates into a dependence between those terms. For example, by adding the edge (e,b) in the example of Fig. 1 the triangle (e,a,b) is formed. It is known that whenever a triangle is used in the basic tree dependence system, improved results are obtained provided that the added edge does not produce a cycle of length greater than 3 in the graph (that is, a dependency between more than 3 terms). [10]

These extensions to the tree dependence system should render the tree model equivalent to the BLE system when the higher order dependencies beyond size 3 which normally appear in the BLE expression are all negligible. The experiments described in the remainder of this study are designed to investigate this question.

2. Identification of Term Dependencies

In the tree dependence model provision is made for the use of (n-1) pairwise term dependencies for any given set of n originally available terms. A direct comparison between the BLE process and the tree dependence system thus becomes possible by including in the BLE expansion precisely (n-1) of the more important term pair dependencies. An obvious method for identifying the most important pairwise dependencies consists in constructing a <u>maximum spanning tree</u> (MST) in which the nodes are used to represent the individual terms, and the branches between pairs of nodes designate the pairwise similarities, or dependencies. [11] An MST includes precisely (n-1) branches, chosen so as to cover the whole tree while maximizing the sum of the pairwise similarities. A typical spanning tree for the term nodes used earlier in Fig. 1 is shown in Fig. 2. It may be seen that the same pairwise dependencies are included in the trees of Figs. 1 and 2. However no precedence order is defined for the nodes in the spanning tree.

To construct a maximum spanning tree for a set of entities, it is necessary to identify the most important similarities between pairs of entities. A criterion for measuring the similarity, or the amount of dependence, between pairs is the <u>expected mutual information measure</u> (EMIM), defined as

$$I(x_i,x_j) = \sum_{\substack{x_i=0,1 \\ x_j=0,1}} P(x_i,x_j) \log \frac{P(x_i,x_j)}{P(x_i)P(x_j)} \tag{8}$$

where the sum is taken over all four combinations of values of x_i and x_j (either 0 or 1), and $P(x_i,x_j)$, $P(x_i)$, and $P(x_j)$ are computed as the proportion of documents in the collection containing respectively both terms x_i and x_j, or x_i or x_j alone. [7] A result of 0 log 0 in expression (7) is defined as 0. The EMIM value for term triples may be similarly defined as

$$I(x_i,x_j,x_k) = \sum_{\substack{x_i=0,1 \\ x_j=0,1 \\ x_k=0,1}} P(x_i,x_j,x_k) \log \frac{P(x_i,x_j,x_k)}{P(x_i)P(x_j)P(x_k)} \tag{9}$$

where the sum is now taken over all eight combinations of values for which x_i,x_j and x_k are either 0 or 1, and $P(x_i,x_j,x_k)$ is the proportion of documents in the collection containing all three terms.

The construction of the MST, or of the dependency tree, for a given set of n nodes requires the generation of $n^2/2$ EMIM values for distinct pairs of nodes. However, these values can be generated consecutively, by first identifying the maximum similarity between a given node a and any of the other nodes; this is followed by the determination of the maximum similarity between the next node, say b, and all nodes other than a, next between node c and all other nodes except a and b, and so on, until all nodes are covered. Thus at most (n-1) pairwise similarities need be processed at any one time during the construction of the spanning tree. To build a precedence, as opposed to a spanning tree, it is necessary in addition arbitrarily to choose some node as the root of the tree; this node is then considered to be independent of any other (father) node.

3. The BLE Term Dependence Process

The MST construction process identifies the term pairs, and by extensions the term triples, which must be taken into account in the computation of the BLE expansion (expression (2)). Since $P(\underline{x})$ is computed separately for the relevant and nonrelevant documents of a collection, relevance information must be available of documents with respect to queries. For experimental purposes one may assume that the relevance of each document with respect to each query is determined a priori outside the retrieval system (the retrospective case). In practice, a relevance feedback process may be used in which the system users stipulate the relevance of certain items retrieved in earlier searches of the collection; alternatively, an estimation process may serve to obtain approximations to the occurrence probabilities of terms in the _relevant_ items, given the corresponding term occurrence statistics in the _whole_ collection. [12,13]

The availability of relevance assessments of documents with respect to queries makes it possible to compute the term occurrence characteristics for single terms (or term pairs and triples) in the relevant and nonrelevant documents of the collection. A typical term occurrence table characterizing the occurrence properties of two terms x_i and x_j in the relevant documents of a collection is shown in Table 1. A corresponding table can be constructed for term occurrences in the nonrelevant documents of the collection. The entries a,b,c, and d in Table 1 represent the number of relevant documents which include both terms, only one of the terms, or neither term, respectively.

The occurrence characteristics of the terms in the relevant and nonrelevant documents of a collection may now be used to generate the various coefficients required for the computation of $P(\underline{x}|\text{rel})$ and $P(\underline{x}|\text{nonrel})$. In principle, the computation of $P(\underline{x})$ for a given document $\underline{x}$ should involve all terms x_i in the document collection, since each x_i is equal to either 0 or 1 in each document. In practice, the computation of $P(\underline{x})$ with respect to a given query can be simplified by using for the evaluation of expression (2) only those document terms which also occur in each particular query under consideration. In these circumstances, it is sufficient to use term occurrence frequencies of the type shown in Table 1 for only the _query_ terms, or query term pairs and triples.

The query term occurrence probabilities for the relevant documents may be obtained in the following way: [7-9]

$$p_i = P(x_i|\mathrm{rel}) = \frac{a+b + 0.5}{a+b+c+d + 1.0}$$

$$p_j = P(x_j|\mathrm{rel}) = \frac{a+c + 0.5}{a+b+c+d + 1.0}$$

and

$$p_{ij} = P(x_i \text{ and } x_j|\mathrm{rel}) = \frac{a + 0.5}{a+b+c+d + 1.0}$$

Corresponding formulas apply to the occurrences of query terms in the nonrelevant documents of a collection, and to the occurrences of term triples in both the relevant and the nonrelevant document terms. The use of the constants (0.5 and 1.0) in the numerators and denominators in (9) is designed to prevent the generation of inaccurate probability values when either the numerators or the denominators in (9) are very small.

The occurrence probabilities of the query terms in the relevant and nonrelevant documents with respect to each query are now used to generate the factors ρ_{ij} of expression (3), and also the term importance values $P(\underline{x}|\mathrm{rel})$ and $P(\underline{x}|\mathrm{nonrel})$ of expression (2). This is done by summing for each document $\underline{x}$ the factors of expression (2) corresponding to the query term appearing in document $\underline{x}$.

In a probabilistic retrieval system, the documents must be ranked at the output in decreasing order according to the function

$$g(\underline{x}) = \frac{P(\underline{x}|\mathrm{rel})}{P(\underline{x}|\mathrm{nonrel})} = \frac{u}{v}. \tag{10}$$

In principle, $g(\underline{x})$ may turn out to be negative when the BLE expansion is injudiciously truncated. The generation of negative values can, however, be avoided by taking special precautions when the values of either u or v in expression (10) are negative or very small. In such a case the quotient $g(\underline{x})$ may not furnish a reliable criterion for the document importance with respect to the query. In the experiments described in this study, special methods are invoked when the values of u or v are smaller than a given threshold T, taken as $1 \cdot 10^{-9}$; $g(\underline{x})$ is then defined in accordance with the rules of Table 2. In particular, for the case represented in the upper right-hand corner of Table 2, where the probability of relevance (u) exceeds the threshold, but the probability of nonrelevance (v) is negative, $g(\underline{x})$ is defined as u'/v' computed in accordance with the formula of expression (5); that is, only the independent part of the BLE equation is taken into account. For the other cases specified on lines 2 and 3 of Table 2, u/v is defined as the expected similarity value obtained by using an a priori probability of relevance of 0.02. That is $g(\underline{x})$

is then computed as (0.02/(1-0.02)) = 0.0204. A ranking procedure also illustrated in Table 2 is used to distinguish the several items which may exhibit identical query-document similarities of 0.0204. In particular documents with a high probability of occurrence in the relevant items of a collection will be retrieved ahead of others whose probability of occurrence is low in the relevant items.

The following sequence of processing steps represents a summary of the methods used in generating a ranked retrieval output using the truncated BLE expansion system:

a) a maximum spanning tree representing the more important pairwise term dependencies is generated for the terms included in a given document collection;

b) relevance assessments are obtained for all documents with respect to a given set of user queries;

c) expanded queries are generated by taking the original query terms and adding all terms that are immediately adjacent in the MST; for example, given query (b,g,i), the MST of Fig. 2 produces an expanded query consisting of terms (a,b,e,g,h,i) as in Fig. 3(b).

d) the pairwise term dependencies ρ_{ij} are obtained for all term pairs i and j included in the expanded query such that each pair (i,j) is represented by an edge in the spanning tree; for the previously used example, this includes the pairs (a,b), (a,e), (a,h), (e,g) and (h,i) shown in Fig. 3(c);

e) term triples can be identified by selectively adding edges to the reduced spanning tree identified in part (d); in particular a triple may be defined whenever all three terms occur in the expanded query, and two of the three possible edges appear adjacently in the MST; for the case illustrated earlier the added edges (a,g), (b,e), (e,h), (b,h), and (a,i) create the triples (a,e,g), (a,b,e), (a,e,h), (a,b,h) and (a,h,i); the correlation factor ρ_{ijk} of expression (8) may be computed for each such triple;

f) for each document $\underline{x}$, the factors $P(\underline{x}|rel)$ and $P(\underline{x}|nonrel)$ are computed by summing the values of expression (2) for all query terms included in document $\underline{x}$; all documents are then ranked in decreasing order according to expression (10).

In the experiments covered in this study, the BLE expansion was computed for each expanded query using first only the single terms, then the single terms with dependent term pairs, and finally the single terms with added pairs and triples.

4. The Tree Dependence Process

The process used to obtain the document ranking for the tree dependence model is substantially similar to that described earlier for the BLE model, except that the probabilities P($\underline{x}$|rel) and P($\underline{x}$|nonrel) for each item $\underline{x}$ are computed using the tree dependence formula (6) instead of the BLE formula (2). In particular a maximum spanning tree is constructed as before for the terms included in a given document collection, based on the computation of the EMIM values (expression (7)) for pairs of terms. A particular term x_i is then randomly chosen as the root of the dependence tree. [8] The dependence tree is then used to expand the query in a manner identical with that previously explained for the BLE case. For example, starting with query terms (c,j), the tree expansion process illustrated in Fig. 4 yields an expanded query consisting of (a,c,f,g,h,j). This query expansion leads to the identification of term pairs (f,j), (a,c), (c,g), and (c,h) as shown in Fig. 4(c). Finally, three added edges yield the triples (a,c,h), (a,c,g) and (c,g,h) represented in Fig. 4(d).

The document importance factors P($\underline{x}$|rel) and P($\underline{x}$|nonrel) may be generated for the tree dependence in a manner analogous to that described earlier for the BLE method. For each term pair in the expanded query included in the precedence tree, a term occurrence table similar to that shown for the BLE case in Table 1 can be generated. Such a table specifies the occurrence characteristics for each pair (x_i, x_{j_i}), where x_{j_i} represents the immediate predecessor (the father node) of x_i in the dependence tree. The statistics in Table 1 apply to query term occurrences in the relevant documents; similar tables can be constructed for term occurrences in the nonrelevant items of a collection.

The occurrence statistics of the query terms in the relevant and nonrelevant documents may now be used as before to generate the coefficients needed to evaluate the document importance factors P($\underline{x}$) of equation (6). Specifically, the parameters p_i and r_i are defined as follows:

$$p_i = P(x_i=1|x_{ji}=1) = \frac{a + 0.5}{a+c + 1.0}$$

$$1-p_i = P(x_i=0|x_{ji}=1) = 1 - \frac{a + 0.5}{a+c + 1.0}$$

$$r_i = P(x_i=1|x_{ji}=0) = \frac{b + 0.5}{b+d + 1.0} \qquad (11)$$

$$1-r_i = P(x_i=0|x_{ji}=0) = 1 - \frac{b + 0.5}{b+d + 1.0}$$

The added factors of 0.5 and 1.0 in expression (11) are used again to prevent the generation of near-zero probability values.

To compute the probability $P(\underline{x}|\text{rel})$ for a given document $\underline{x}$ using the definition of expression (6), it is then simply necessary to choose one of the four formulas in (11) for each edge (x_i,x_{ji}) in the expanded query depending on the particular values of x_i and x_{ji} in document $\underline{x}$. When the parent x_{ji} of a given query term x_i does not appear in the expanded query, then $P(x_i|x_{ji})$ is defined as $P(x_i)$; that is, such a term is considered to be independent of the other query terms. In that case, the data of Table 1 are used to define $P(x_i=1)$ as $(a+b)/(a+b+c+d)$, and $P(x_i=0)$ as $(c+d)/(a+b+c+d)$. For the sample query used as an illustration, terms a and f do not have a predecessor in the tree. Hence the expression for P(x) is obtained as

$$P(\underline{x}) = P(x_a)\ P(x_c|x_a)\ P(x_g|x_c)\ P(x_h|x_c)\ P(x_f)\ P(x_j|x_f)$$

Equivalently, log $P(\underline{x})$ can be computed as the sum of the logarithms of the corresponding probabilities.

Following the computation of $P(\underline{x}|\text{rel})$ and $P(\underline{x}|\text{nonrel})$ for each doument $\underline{x}$ with respect to a given query, the documents are once again ranked in decreasing order according to the value of $g(\underline{x})$ in expression (10). A comparison of the query expansion methods illustrated in Figs. 3 and 4 for the BLE and the tree dependence methods shows that precisely the same terms are added to the queries in both cases. This implies that the dependent term pairs used by both methods are also identical. The formulas used to compute $P(\underline{x})$ are, however, different for the BLE and tree dependence cases since a distinction is made in the tree dependence case between terms that exhibit a predecessor in the tree and those that do not.

The same term triples are also identified by both methods. However the order in which the triples are incorporated into the term dependence expressions may differ. For the BLE system, the triples are added in decreasing order according to their EMIM values (expression (8)). In the tree dependence case, it is known that an optimal ordering of the term triples is determined by the values of W_{ijk}, where

$$W_{ijk} = \sum_{i,j,k=0,1} P(i,j,k) \log \frac{P(i,j,k)}{P(i)\ P(j|i)\ P(k|i)} \tag{12}$$

and (j,k) is the edge added to the two originally existing edges (i,j) and (i,k) to form the triple (i,j,k). [10] The ordering of the term triples according to the EMIM and W values (expressions (8) and (12)) should be similar in most cases. The BLE and tree dependence methodologies are therefore directly comparable.

5. Experimental Results

The term dependence methodologies using either the BLE or the tree dependence methods are compared in a number of experiments performed with a document collection in biomedicine (Medlars 1033). A summary of the collection and query statistics appears in Table 3. It may be seen that the average number of terms describing the 1033 documents and the 30 expanded queries is approximately the same (55.8 and 48.6, respectively). The average number of term pairs obtained from the expanded queries is 42.6 and the average number of term triples is 177.1. The previously mentioned results for the tree dependence model are valid only for term triples that do not produce dependencies among four or more terms. [10] This implies that the added term triples should not exhibit common term pairs. For the experiments described in this study no triples were added that included common term pairs. With that restriction, the average number of triples obtained from the expanded queries decreases from 177.1 to 18.4 as shown in Table 3.

The experimental results are presented in terms of recall-precision tables giving average precision values at various levels of the recall for the 30 sample queries. [14] At the bottom of each table the percentage of improvement is given in each case over a base run using the truncated BLE expression (5) with the term independence assumption (no added pairs or triples). Table 4 contains a comparison of the base run with a standard cosine vector matching process using objectively determined frequency-based term weights. The loss of thirty percent in average precision between the BLE run based on term independence and the cosine matching system is due to the retrospective nature of the probabilistic retrieval runs. Specifically, the computation of the probability values p_i, p_{ij}, p_{ijk}, etc. used in the experiment involves full knowledge of the relevance of all documents with respect to all queries. In these circumstances exact probability values can be obtained, leading to an optimal term weighting system. On the other hand for the vector matching system, no assumption is made about the relevance of documents with respect to queries. The retrospective case thus represents an upper-bound of the performance obtainable under ideal operating conditions.

The retrospective search results using a truncated BLE formula (expression (2)) are covered in Table 5, and the corresponding tree dependence experiments (expression (6)) appear in Table 6. The BLE run with term independence is used as a standard (case 1) in each Table. It may be seen that for the BLE experiments the addition of term pairs and term triples provides improvements over the base case ranging from about ten percent for term pair addition to 17 percent for the addition of pairs and triples. Case 3 of Table 5 covers the addition of the four best term triples (added in decreasing order of the EMIM value of expression (9)) in addition to the term pairs.

Comparable tree dependence results are included in Table 6. The improvement over the base case involving term independence exceeds 38 percent when all dependent term pairs are included in the computation. However, the further addition of term triples does not provide additional advantages except at the very high recall end of the performance range when the recall exceeds 90 percent.

It appears from these results that the tree dependence method must be preferred over the BLE implementation used in ths study, because the tree dependence system is easier to implement and produces better results. It must be remembered, however, that the restrictions on the available term pairs and triples in the BLE system, imposed by the maximum spanning tree process is not inherent in the use of the standard BLE formula (2). It may well be that additional improvements can be obtained for the BLE method when pairs and triples are considered that are not specified within the MST. No such improvements are, of course, available for the tree dependence system since the term dependencies are then necessarily tied to the precedence tree. Further experiments using different implementations of the BLE system remain to be carried out.

6. Summary

An attempt is made in this study to compare two methodologies for the use of term dependence information in a retrieval environment. The methods examined in this study represent one possible approach among many that could have been used. The tree dependence system described in section 4 could, for example, be replaced by a simple query term weighting process which does not explicitly consider any dependent term pairs and triples. [8] Such a term weighting process may be based on the construction of global term occurrence tables giving the number of relevant and nonrelevant documents in a collection in which each query term occurs. Typical statistics of this kind valid for query term x_i are shown in Table 7. The number of relevant and nonrelevant documents with respect to the given query identified in Table 7 is R and N-R, respectively. Correspondingly, the number of relevant and nonrelevant items containing query term q_i is r (out of R) and f-r (out of N-R).

The data of Table 7 may now be used to define a term weight w_{q_i} for each query term q_i which increases with the number of term occurrences in the relevant documents, and decreases with the number of term occurrences in the nonrelevant items. In particular

$$w_{q_i} = \frac{r}{N-R} \log \frac{r/I}{\frac{f}{N}\cdot\frac{R}{N-R}} + \frac{(N-R-f+r)}{N} \log \frac{(N-R-f+r)/N}{\frac{N-f}{N}\cdot\frac{N-R}{N}}$$

$$- \frac{f-r}{N} \log \frac{(f-r)/N}{\frac{f}{N}\cdot\frac{N-R}{N}} - \frac{R-r}{N-R} \log \frac{(R-r)/(N-R)}{\frac{N-f}{N}\cdot\frac{R}{N-R}} \qquad (13)$$

Following the computation of the term weights for all query terms in accordance with the formula of equation (13), the documents ($\underline{x} = x_1, x_2, \ldots, x_n$) can be ranked in decreasing order according to the value of the inner product between document vectors and weighted query vectors, that is, $\sum_{i=1}^{n} x_i \cdot w_{q_i}$. It is claimed that this process produces retrieval results that are equivalent to what is obtainable by using the tree dependence process described earlier. [8]

Many additional modifications in the experimental design may be made in an attempt to evaluate the use of term dependencies in information retrieval. The following possibilities may be of greatest interest:

a) the relevance assessments of documents with respect to queries may be obtainable directly from the user population by implementing an interactive search system based on relevance feedback;

b) the expanded queries may be generated in various ways, for example, by adding to the queries all terms located within a distance of 2 (instead of a distance of 1) from the original query terms in the spanning tree;

c) a different maximum spanning tree program may be used; for example, a new MST could be built for each query using only the terms included in that query; alternatively, two maximum spanning trees might be used, one obtained from terms occurring in the relevant items and the other from terms in the nonrelevant documents;

d) the term pairs used in the BLE or the tree dependence expressions may be chosen in various ways, for example by defining a threshold for the EMIM values and adding only those pairs whose EMIM value exceeds the given threshold;

e) the term triples to be added to the process can also be obtained in a variety of ways, for example by adding only triples that do not exhibit a commond edge (a common term pair) in the graph structure, or by using triples whose similarity value exceeds a given threshold;

f) the adjustments made in the BLE computation to account for small or negative probability values may be different from those used in the present experiments; in particular, different values may be chosen for the entries of Table 2;

g) different document collections may be used as a test bed for the experiments.

Some of the variations suggested in the foregoing list will be incorporated in future experiments designed to validate the use of term dependencies in information retrieval.

References

[1] M.E. Maron and J.L. Kuhns, On Relevance, Probabilistic Indexing and Information Retrieval, Journal of the ACM, Vol. 7, No. 3, July 1960, p. 216-244.

[2] D. Kraft and A. Bookstein, Evaluation of Information Retrieval Systems: A Decision Theory Approach, Journal of the ASIS, Vol. 29, No. 1, January 1978, p. 31-40.

[3] D. Chow and C.T. Yu, "On the Construction of Feedback Queries", Journal of the ACM, Vol. 29, No. 1, January 1982, p. 127-151.

[4] G. Salton, Mathematics and Information Retrieval, Journal of Documentation, Vol. 35, No. 1, March 1979, p. 1-29.

[5] C.T. Yu, W.S. Luk and M.K. Siu, On Models of Information Retrieval Processes, Information Systems, Vol. 4, No. 3, p. 205-218, 1979.

[6] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, J. Wiley and Sons, New York, 1973.

[7] C.J. van Rijsbergen, A Theoretical Basis for the Use of Cooccurrence Data in Information Retrieval, Journal of Documentation, Vol. 33, No. 2, June 1977, p. 106-119.

[8] D.J. Harper and C.J. van Rijsbergen, An Evaluation of Feedback in Document Retrieval using Co-occurrence Data, Journal of Documentation, Vol. 34, No. 3, September 1978, p. 189-216.

[9] 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, p. 35-56.

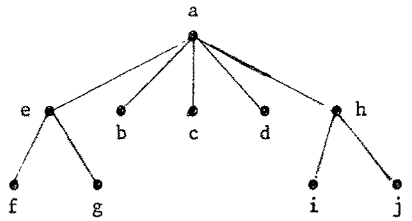
[10] C.T. Yu, K. Lam, and G. Salton, Extensions to the Tree Dependence Model in Information Retrieval, Technical Report, Computer Science Department, Cornell University, Ithaca, New York, 1982.

[11] K.V.M. Whitney, Minimal Spanning Tree, Communications of the ACM, Vol. 15, No. 4, April 1972, p. 273-274.

[12] C.T. Yu, K. Lam, and G. Salton, Term Weighting in Information Retrieval Using the Term Precision Model, Journal of the ACM, Vol. 29, No. 1, January 1982, p. 152-170.

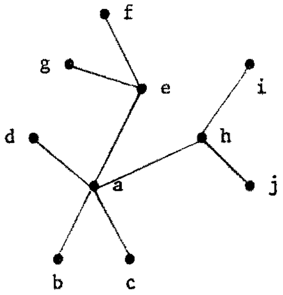
[13] H. Wu and G. Salton, The Estimation of Term Relevance Weights Using Relevance Feedback, Journal of Documentation, Vol. 37, No. 4, December 1981, p. 194-214.

[14] G. Salton, Dynamic Information and Library Processing, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1975, Chapter 6.



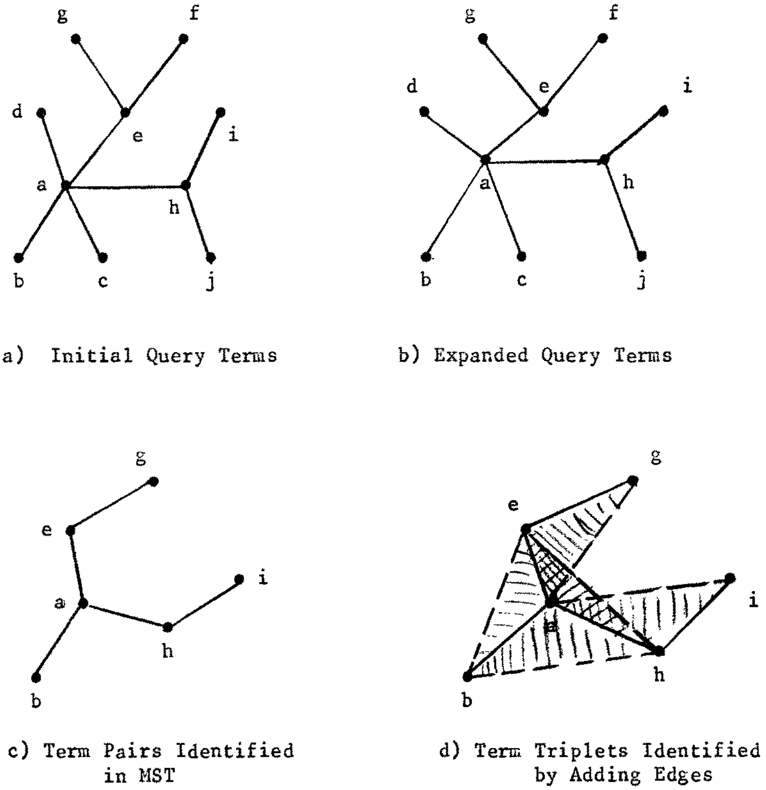
Typical Precedence Tree

Fig. 1



Typical Spanning Tree for Ten Terms

Fig. 2



a) Initial Query Terms

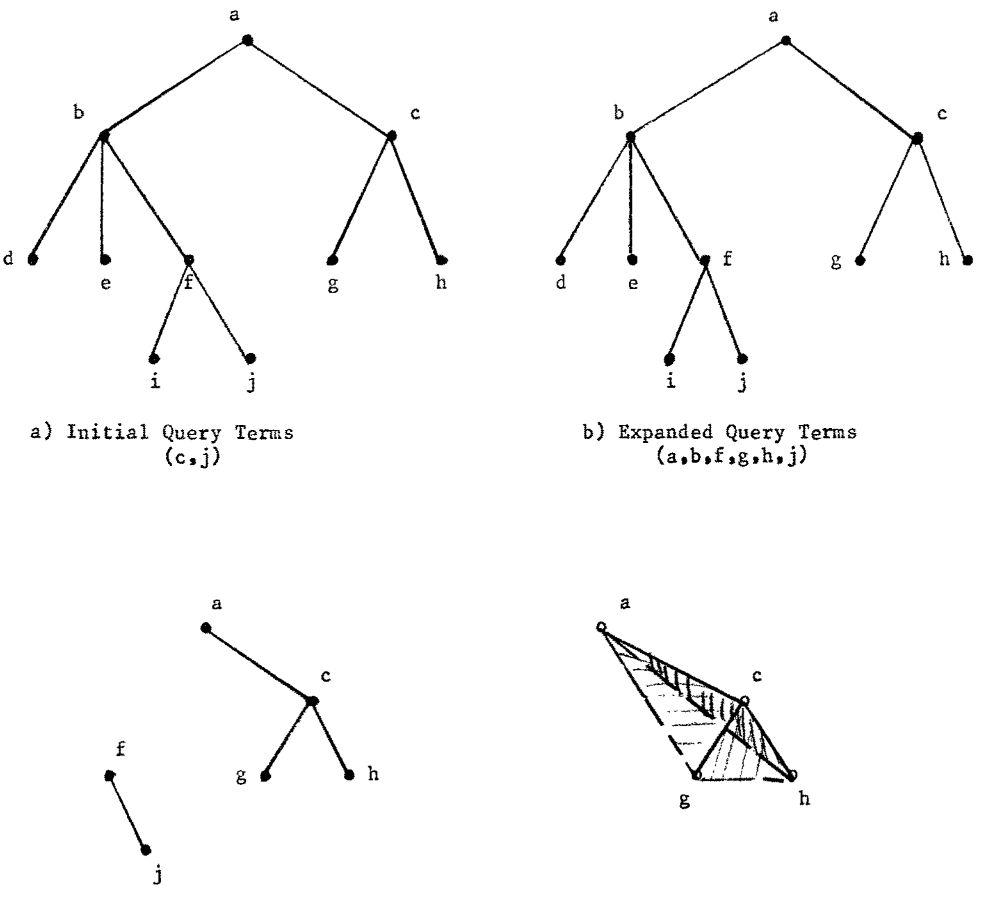
b) Expanded Query Terms

c) Term Pairs Identified in MST

d) Term Triplets Identified by Adding Edges

Determination of Term Pairs and Triples Using MST

Fig. 3



a) Initial Query Terms
(c,j)

b) Expanded Query Terms
(a,b,f,g,h,j)

c) Term Pairs Identified in
Precedence Tree
[(f,j),(a,c),(c,g),(c,h)]

d) Term Triplets Identified
by Adding Edges
[(a,c,h),(a,c,g),(c,g,h)]

Determination of Term Pairs and Trips Using
Precedence Trips

Fig. 4

	$x_i=1$	$x_i=0$
$x_j=1$ (or $x_{ji}=1$)	a	c
$x_j=0$ (or $x_{ji}=0$)	b	d

Term Occurrence Characteristics of Query Terms x_i and x_j

in the Relevant Documents

Table 1

	$T \le v$	$0 \le v < T$	$v < 0$
$T \le u$	u/v	u/v	u'/v' (independent)
$0 \le u < T$	.0204(3)	.0204(2)	.0204(1)
$u < 0$	.0204(6)	.0204(5)	.0204(4)

Ranking Provisions of Documents for Small $g(\underline{x}) = u/v$

(Items (i) are ranked ahead of items (j) for $i < j$)

Table 2

	Relevant Items	Nonrelevant Items	
$q_i = 1$	r	f-r	f
$q_i = 0$	R-r	N-f-R+r	N-f
	R	N-R	N

Occurrences of Query Term x_i in the Relevant
and Nonrelevant Documents
Table 7

Medlars (1033 items in biomedicine)	
Number of documents	1033
Average number of terms per document	55.8
Number of queries	30
Average number of query terms (original)	10.7
Average number of query terms (expanded)	48.6
Average number of relevant documents per query	23.2
Average number of term pairs determined from expanded queries	42.6
Average number of term triples determined from expanded queries	177.1
Average number of term triples without common edges (triples not leading to dependencies for four or more terms)	18.4

Statistics for Experimental Collection

(Medlars 1033 documents, 30 queries)

Table 3

BLE retrospective runs on med1033

1) Independent Case: uses EXP_SIM
2) Cosine sim with term frequency weights

recall	precision for cases:	
level	1	2
0.00	0.9685	0.8958
0.05	0.9619	0.8288
0.10	0.9265	0.7809
0.15	0.8946	0.7359
0.20	0.8874	0.7007
0.25	0.8685	0.6733
0.30	0.8487	0.5886
0.35	0.8276	0.5517
0.40	0.8025	0.5227
0.45	0.7563	0.4721
0.50	0.7206	0.4456
0.55	0.6816	0.4156
0.60	0.6551	0.3864
0.65	0.6056	0.3657
0.70	0.5426	0.3278
0.75	0.5086	0.3018
0.80	0.3984	0.2650
0.85	0.3080	0.2042
0.90	0.2210	0.1542
0.95	0.1647	0.1027
1.00	0.1261	0.0869

Percentage change from base case

Recall	Base	Case
level	case	2
0.00	0.9685	-7.5
0.05	0.9619	-13.8
0.10	0.9265	-15.7
0.15	0.8946	-17.7
0.20	0.8874	-21.0
0.25	0.8685	-22.5
0.30	0.8487	-30.6
0.35	0.8276	-33.3
0.40	0.8025	-34.9
0.45	0.7563	-37.6
0.50	0.7206	-38.2
0.55	0.6816	-39.0
0.60	0.6551	-41.0
0.65	0.6056	-39.6
0.70	0.5426	-39.6
0.75	0.5086	-40.7
0.80	0.3984	-33.5
0.85	0.3080	-33.7
0.90	0.2210	-30.2
0.95	0.1647	-37.6
1.00	0.1261	-31.1
Average Change:		-30.4%

Comparison of Probabilistic Retrieval Using BLE with Vector Processing
(retrospective case, term independence assumption)

Table 4

BLE retrospective runs

1) Independent case:
2) Pairs no triples
3) All pairs, best 4 (emim) triples
4) All pairs all triples

recall level	precision for cases: 1	 2	 3	 4
0.00	0.9685	0.9833	1.0000	1.0000
0.05	0.9619	0.9606	0.9944	0.9944
0.10	0.9265	0.9302	0.9578	0.9700
0.15	0.8946	0.9293	0.9337	0.9460
0.20	0.8874	0.9100	0.9268	0.9396
0.25	0.8685	0.8970	0.9113	0.9214
0.30	0.8487	0.8705	0.8818	0.8957
0.35	0.8276	0.8473	0.8641	0.8834
0.40	0.8025	0.8270	0.8519	0.8728
0.45	0.7563	0.7815	0.8121	0.8417
0.50	0.7206	0.7425	0.7716	0.7938
0.55	0.6816	0.7230	0.7441	0.7618
0.60	0.6551	0.6918	0.7221	0.7377
0.65	0.6056	0.6620	0.6902	0.7064
0.70	0.5426	0.6288	0.6466	0.6661
0.75	0.5086	0.5684	0.5913	0.6371
0.80	0.3984	0.5347	0.5573	0.5747
0.85	0.3080	0.4323	0.4599	0.4742
0.90	0.2210	0.3157	0.3025	0.3225
0.95	0.1647	0.1756	0.1786	0.2128
1.00	0.1261	0.1349	0.1371	0.1484

Percentage change from base case

Recall level	Base case	Cases: 2	 3	 4
0.00	0.9685	1.5	3.3	3.3
0.05	0.9619	-0.1	3.4	3.4
0.10	0.9265	0.4	3.4	4.7
0.15	0.8946	3.9	4.4	5.7
0.20	0.8874	2.5	4.4	5.9
0.25	0.8685	3.3	4.9	6.1
0.30	0.8487	2.6	3.9	5.5
0.35	0.8276	2.4	4.4	6.7
0.40	0.8025	3.1	6.2	8.8
0.45	0.7563	3.3	7.4	11.3
0.50	0.7206	3.0	7.1	10.2
0.55	0.6816	6.1	9.2	11.8
0.60	0.6551	5.6	10.2	12.6
0.65	0.6056	9.3	14.0	16.6
0.70	0.5426	15.9	19.2	22.8
0.75	0.5086	11.8	16.3	25.3
0.80	0.3984	34.2	39.9	44.3
0.85	0.3080	40.4	49.3	54.0
0.90	0.2210	42.9	36.9	45.9
0.95	0.1647	6.6	8.4	29.2
1.00	0.1261	7.0	8.7	17.7
Average Change:		9.8%	12.6%	16.7%

Probabilistic Retrieval using BLE (retrospective case)

Table 5

Tree retrospective runs

1) Independent case
2) Tree retro all pairs no triples
3) Tree retro all pairs 4 triples (no cycles)
4) Tree retro all pairs all triples (no cycles)

recall	precision for cases:			
level	1	2	3	4
0.00	0.9685	0.9958	1.0000	0.9939
0.05	0.9619	0.9958	1.0000	0.9939
0.10	0.9265	0.9875	0.9886	0.9902
0.15	0.8946	0.9832	0.9796	0.9810
0.20	0.8874	0.9773	0.9755	0.9810
0.25	0.8685	0.9747	0.9755	0.9687
0.30	0.8487	0.9630	0.9588	0.9551
0.35	0.8276	0.9492	0.9521	0.9476
0.40	0.8025	0.9378	0.9447	0.9254
0.45	0.7563	0.9192	0.9316	0.9117
0.50	0.7206	0.9084	0.9146	0.8943
0.55	0.6816	0.8807	0.8873	0.8575
0.60	0.6551	0.8742	0.8773	0.8348
0.65	0.6056	0.8472	0.8380	0.7857
0.70	0.5426	0.8221	0.7921	0.7376
0.75	0.5086	0.7806	0.7341	0.6999
0.80	0.3984	0.7247	0.6598	0.6333
0.85	0.3080	0.6421	0.5798	0.5650
0.90	0.2210	0.5010	0.4716	0.4553
0.95	0.1647	0.3078	0.3380	0.3643
1.00	0.1261	0.1992	0.2042	0.2135

Percentage change from base case

Recall	Base	Cases:		
level	case	2	3	4
0.00	0.9685	2.8	3.3	2.6
0.05	0.9619	3.5	4.0	3.3
0.10	0.9265	6.6	6.7	6.9
0.15	0.8946	9.9	9.5	9.7
0.20	0.8874	10.1	9.9	10.5
0.25	0.8685	12.2	12.3	11.5
0.30	0.8487	13.5	13.0	12.5
0.35	0.8276	14.7	15.0	14.5
0.40	0.8025	16.9	17.7	15.3
0.45	0.7563	21.5	23.2	20.5
0.50	0.7206	26.1	26.9	24.1
0.55	0.6816	29.2	30.2	25.8
0.60	0.6551	33.4	33.9	27.4
0.65	0.6056	39.9	38.4	29.7
0.70	0.5426	51.5	46.0	35.9
0.75	0.5086	53.5	44.3	37.6
0.80	0.3984	81.9	65.6	59.0
0.85	0.3080	108.5	88.2	83.4
0.90	0.2210	126.7	113.4	106.0
0.95	0.1647	86.9	105.2	121.2
1.00	0.1261	58.0	61.9	69.3
Average Change:		38.4%	36.6%	34.6%

Probabilistic Retrieval Using Tree Dependence (retrospective case)

Table 6