Information Technology: Research and Development (1983), *2* (129–154)

A GENERALIZED TERM DEPENDENCE MODEL IN INFORMATION RETRIEVAL*

C. T. YU

Department of Information Engineering, University of Illinois-Chicago Circle, Chicago, Illinois 60680, USA

C. BUCKLEY

Department of Computer Science, Cornell University, Ithaca, New York 14853, USA

K. LAM

Department of Statistics, Hong Kong University, Hong Kong

AND

G. SALTON

Department of Computer Science, Cornell University, Ithaca, New York 14853, USA

(*Received 4 March 1983, revised 3 June 1983*)

ABSTRACT

The tree dependence model has been used successfully to incorporate dependencies between certain term pairs in the information retrieval process, while the Bahadur Lazarsfeld Expansion (BLE) which specifies dependencies between all subsets of terms has been used to identify productive clusters of items in a clustered database environment. The successes of these models are unlikely to be accidental; it is of interest therefore to examine the similarities between the two models.

The disadvantage of the BLE model is the exponential number of terms appearing in the full expression, while a truncated BLE system may produce negative probability values. The disadvantage of the tree dependence model is the restriction to dependencies between certain term pairs only and the exclusion of higher-order dependencies. A generalized term dependence model is introduced in this study which does not carry the disadvantages of either the tree dependence or the BLE models. Sample evaluation results are included to illustrate the operations of the generalized system.

1. DECISION-THEORETIC RETRIEVAL

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 non-relevance, with respect to a

* This study was supported in part by the National Science Foundation under grant IT-8108696.

0144-817X/83/03 0129-26 $03.00

particular query. For obvious reasons, the larger the probability of relevance of a particular item, and the smaller the probability of non-relevance, the greater will be the retrieval probability for the item.

In particular, consider an item x in the database 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 x. For each item x and each query Q, it is in principle possible to generate the two parameters $P(x \mid \text{rel})$ and $P(x \mid \text{non-rel})$, representing the probabilities that a relevant and a non-relevant item, respectively, has vector representation 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

$$\log \frac{P(x \mid \text{rel})}{P(x \mid \text{non-rel})} \tag{1}$$

That is, given two items x and y, x should be retrieved ahead of y whenever the value of expression (1) for x exceeds the corresponding value for y (Maron and Kuhns, 1960; Robertson, 1977; Kraft and Bookstein, 1978; Salton, 1979; Yu *et al.*, 1979; Chow and Yu, 1982).

The probabilistic approach is of course useless in retrieval unless methods can be found for estimating the probabilities $P(x \mid s)$ for each item in the classes s of relevant and non-relevant items, respectively. These probabilities will necessarily depend on the occurrence characteristics of the individual vector elements x_i in the relevant and non-relevant items of the collection. The class variable s will be dropped in the remainder of this paper because the development that follows is identical for the two classes of documents.

An exact formulation for $P(x)$ is given by the Bahadur Lazarsfeld expansion (BLE) as follows (Duda and Hart, 1973; Yu *et al.*, 1979; Lam and Yu, 1982):

$$P(x) = \prod_{t=1}^{n} p_t^{x_t} (1-p_t)^{1-x_t} \left[1 + \sum_{i<j} \varrho_{ij} \frac{(x_i-p_i)(x_j-p_j)}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} + \sum_{i<j<k} \varrho_{ijk} \frac{(x_i-p_i)(x_j-p_j)(x_k-p_k)}{\sqrt{p_i p_j p_k (1-p_i)(1-p_j)(1-p_k)}} + \ldots . + \varrho_{12\ldots n} \frac{(x_1-p_1)(x_2-p_2)\ldots(x_n-p_n)}{\sqrt{p_1 p_2 \ldots p_n (1-p_1)(1-p_2)\ldots(1-p_n)}} \right] \tag{2}$$

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

$$\varrho_{ij} = \frac{E[(x_i-p_i)(x_j-p_j)]}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} = \frac{E(x_i x_j) - p_i p_j}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} \tag{3}$$

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

$$= \frac{E(x_i x_j x_k) - E(x_i x_j)p_k - E(x_i x_k)p_j - E(x_j x_k)p_i + 2p_i p_j p_k}{\sqrt{p_i p_j p_k (1-p_i)(1-p_j)(1-p_k)}} \quad (4)$$

Corresponding expressions apply to the higher order correlations.

The BLE expansion (2) is of no help unless the term occurrence probabilities p_k can be obtained for all terms k in both the relevant and non-relevant document sets. Furthermore the correlation coefficients ϱ_{ij}, ϱ_{ijk}, etc. must also be available for all term combinations in the two document classes. This last requirement is unfortunately difficult to satisfy in practice for two main reasons:

1. It is in practice impossible to compute the correlation coefficients for an exponential number of term combinations.
2. An injudicious truncation of the BLE series may produce unreliable results; for example, the second order correlations ϱ_{ij} become negative when the joint occurrence probabilities $E(x_i\, x_j)$ for pairs of terms are close to zero, but the individual probabilities p_i and p_j are positive; this may lead to the computation of negative (false) probability values from the BLE formula when third and higher order dependencies are neglected.*

To render the computational task more manageable, one often assumes that the term occurrences are independent of each other in each of the relevant and non-relevant documents of a collection. In that case

$$P(x) = P(x_1)\, P(x_2) \ldots . P(x_n). \quad (5)$$

For the independence case, the BLE expansion reduces to

$$P(x) = \prod_{t=1}^{n} p_t^{x_t} (1-p_t)^{(1-x_t)} \quad (6)$$

since all ϱ values will be equal to 0 (Robertson and Sparck Jones, 1976; Yu and Salton, 1976).

In actual document collections, the assigned keywords and attributes do not of course occur independently of each other. The elimination of term dependencies may then lead to substantial losses of information and to a reduced retrieval effectiveness. This suggests that an approach be used in which certain selected term dependencies are included while the others are disregarded. The *tree dependence* model represents such a compromise solution.

In describing the tree dependence model, the following notation is used:

* A referee has pointed out that a Ph.D. dissertation by D. J. Harper (1980) considers the use of the Bahadur Lazarsfeld expansion and provides a detailed analysis of the tree dependence model introduced later in this section.

1. $P(\underline{x})$ or $P(x_1, x_2, \ldots, x_n)$ represents the actual probability distribution for a vector of n terms. When no ambiguity arises, the vector $(x_1, x_2, \ldots, x_n)$ is replaced by $(1, 2, \ldots, n)$. Thus any distribution $h(x_1, x_2, \ldots, x_n)$ is written as $h(1, 2, \ldots, n)$.
2. The notation $h(j_1, j_2, \ldots, j_t)$ for specific terms $j_1, j_2, \ldots, j_t$ stands for $\sum_{N-J} h(1, 2, \ldots, n)$ where $N = \{1, 2, \ldots, n\}$ and $J = \{j_1, j_2, \ldots, j_t\}$. That is, h represents the probability distribution for the set of terms $J = \{j_1, j_2, \ldots, j_t\}$ and the summation extends over all possible combinations of 0 and 1 for all variables other than those in J. For example when $n = 4$,

$$h(1, 3) = h(x_1, x_2 = 0, x_3, x_4 = 0) + h(x_1, x_2 = 0, x_3, x_4 = 1) + h(x_1, x_2 = 1, x_3, x_4 = 0) + h(x_1, x_2 = 1, x_3, x_4 = 1).$$

In particular,

$$P(i) = \sum_{\substack{k \neq i \\ k \in N}} P(1, 2, \ldots, n); p_i = P(x_i = 1)$$

represents the probability of occurrence of the ith term. An underlined variable denotes a vector of variables; a variable that is not underlined stands for a single variable.

2. PROPERTIES OF THE TREE DEPENDENCE MODEL

The tree dependence model is characterized by the fact that the dependence structure between terms constitutes a tree in which the vertices represent the terms and the edges represent the dependencies between pairs of terms. More specifically, let T be a tree with root v. The tree can be represented by a directed graph $G = (V,E)$, where V is the set of vertices and E is the set of directed edges (away from the root v). Then the probability distribution of the terms on the items is given by the tree dependence model as follows:

$$f(\underline{x};G) = P(v) \left[\prod_{E} P(a \mid b) \right] \tag{7}$$

where b is the parent of a and the product is taken over all edges of E (van Rijsbergen, 1977; Harper and van Rijsbergen, 1978; Robertson *et al.*, 1981). When E is null, i.e., the graph has exactly one vertex, then the product over E is assumed to be 1.

Consider as an example the dependence tree of Figure 1. The root is 1; the immediate descendants are 2, 3 and 4, whose descendants are respectively {5, 6}, {7}, {8}. Then

$$f(\underline{x};G) = P(1)\,P(2 \mid 1)\,P(3 \mid 1)\,P(4 \mid 1)\,P(5 \mid 2)\,P(6 \mid 2)\,P(7 \mid 3)\,P(8 \mid 4).$$

Expression (7) can now be rewritten as follows. Suppose an edge (v,u) incident on

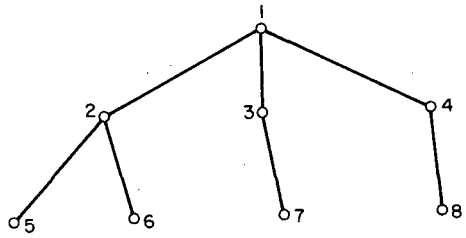

FIG. 1. Typical dependence tree

root v is deleted. Then the tree T is decomposed into two subtrees $G_u=(V_u, E_u)$ and $G_v=(V_v, E_v)$ having roots u and v respectively. It is clear that $E=E_u \cup E_v \cup\{(v, u)\}$. Hence

$$f(\underline{x};G) = P(v)\,P(u\,|\,v)\,[\prod_{E-(v,u)} P(a\,|\,b)]$$

$$= P(v)\,P(u\,|\,v) \left[\frac{P(v)\prod_{E_v} P(a\mid b)}{P(v)}\right]\left[\frac{P(u)\prod_{E_u} P(a\,|\,b)}{P(u)}\right]$$

$$= \frac{P(u, v)}{P(u)\,P(v)}\, f(x_v;G_v)\,f(x_u;G_u) \tag{8}$$

where x_v and x_u are the variables restricted to vertices of V_v and V_u respectively.

Thus, (8) is an inductive definition, equivalent to (7). When the original tree G has 1 vertex only, say v (and no edge),

$$f(x;G) = P(v) \tag{9}$$

When the original tree G contains more than one vertex, expression (8) applies.

The next lemma shows that the tree expansion formulas (7), (8), (9) are well defined in the sense that the same result is obtained if a different root is chosen for expansion or a different edge (v, u) is deleted. In fact, a simple formula is given in terms of the edges and the vertices of the tree.

Proposition 1: For a tree $G=(V, E)$, the tree dependence $f(x; G)$ is given by

$$f(\underline{x};G) = \frac{\prod_{(i,j)\in E} P(i,j)}{\prod_{i\in V} P(i)^{d_i-1}} \tag{10}$$

where d_i is the degree of (the number of edges incident on) vertex i. If E is null, the numerator of (10) gives 1. Expression (10) shows that the tree dependence $f(x;G)$ is independent of the chosen root and of the direction of the edges.

Proof: Since (7) is equivalent to the inductive definition given by (8) and (9), it is sufficient to show that (10) is equivalent to the inductive definition.

The proof is by induction. If G has one vertex only, say vertex v, then both (9) and (10) give $P(v)$.

Consider a connected tree G having more than one vertex. The deletion of an edge (v, u) causes the tree $G=(V, E)$ to be decomposed into two subtrees $G_u=(V_u, E_u)$ and $G_u=(V_v, E_v)$ such that the degree of each of the vertices u and v in the subtrees is one less than the degree of the same vertices in G (*see* Fig. 2). By the inductive hypothesis, (8) gives

$$f(x;G) = \frac{P(u, v)}{P(u)\,P(v)} \left[\frac{\prod_{(i,j)\in E_v} P(i,j)}{P(v)^{d_v-2} \prod_{\substack{i\in V_v \\ i\neq v}} P(i)^{d_i-1}} \right] \left[\frac{\prod_{(i,j)\in E_u} P(i,j)}{P(u)^{d_u-2} \prod_{\substack{i\in V_u \\ i\neq u}} P(i)^{d_i-1}} \right]$$

$$= \frac{\prod_{(i,j)\in E} P(i,j)}{\prod_{i\in V} P(i)^{d_i-1}}$$

which is identical with (10), since $E=E_u \cup E_v \cup \{(v, u)\}$ and $V=V_u \cup V_v$.

It is clear that (8) and hence (10) are identical with (7) for the tree decomposition into subtrees G_v, G_u and edge (u, v). Furthermore, vertex v and edge (u, v) do not appear explicitly in (10). Hence any other decomposition will also produce the formula of expression (10).□

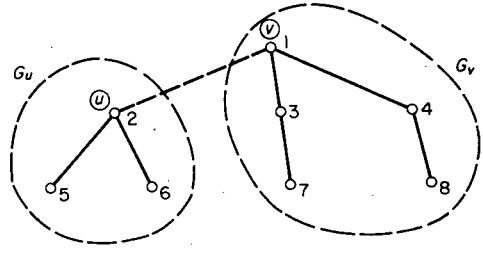

FIG. 2. Tree decomposition using edge (u, v)

For the decomposition of Figure 2, expression (10) can be written as

$$f(x;G) = P(1,2) \cdot \frac{P(1,3)\,P(1,4)\,P(3,7)\,P(4,8)}{P(1)^2\,P(3)^1\,P(4)^1\,P(7)^0\,P(8)^0} \cdot \frac{P(2,5)\,P(2,6)}{P(2)^2\,P(5)^0\,P(6)^0}$$

$$= P(1,2) \cdot P(3\,|\,1)\,P(4\,|\,1)\,P(7\,|\,3)\,P(8\,|\,4) \cdot (P(5\,|\,2)\,P(6\,|\,2))$$
$$= P(1)\,P(2\,|\,1)\,P(3\,|\,1)\,P(4\,|\,1)\,P(7\,|\,3)\,P(8\,|\,4)\,P(5\,|\,2)\,P(6\,|\,2)$$

This is of course identical with the formula derived from the tree of Figure 1.

It may be noted that the factors $P(i)$ and $P(i, j)$ used in (10) represent probabilities. Hence every term in expression (10) is non-negative. The tree dependence model cannot therefore lead to the computation of negative probability factors, no matter how many, or how few, dependent term pairs are used in the computations.

The similarity between the BLE model and the tree dependence model will now be examined. It will be shown that the tree dependence model places a constraint on the second order correlations, ϱ_{ij}, between term pairs. If these correlation parameters (ϱ_{ij}) are set in the BLE model so as to satisfy this constraint, and if the third and higher order dependencies are negligible (that is, ϱ_{ijk}, ϱ_{ijkh}, etc. are set to 0), then the BLE model is for practical purposes equivalent to the tree dependence model. If the third and higher order term dependencies are significant, then the generalized model introduced in Section 3 should be applied.

The formulation of expression (10) leads to the following proposition:

Proposition 2: In the tree dependence model, if i, j and k are vertices of a tree $G=(V, E)$ such that a path exists between i and j passing through k, then i and j are *independent conditional* on k, that is,

$$f(i, j|k) = f(i|k) \cdot f(j|k) \tag{11a}$$

or equivalently

$$f(i, j, k) f(k) = f(i, k) \cdot f(j, k) \tag{11b}$$

Proof: Consider the tree G following the deletion of vertex k. The resulting graph now consists of two or more components, including $G_i=(V_i, E_i)$ containing vertex i, $G_j=(V_j, E_j)$ containing vertex j, and possibly additional components which may collectively be labelled $\bar{G}$. Assume that edge (k, i_1) is the edge connecting vertex k to G_i along the path from k to i, and similarly that (k, j_1) connects vertex k to G_j along the path from k to j.

Restoring vertex k and its incident edges, the decomposition of G leads to the identification of the following subsets of vertices and edges.

for G_i: $(V_i \cup \{k\}, E_i \cup \{k, i_1\})$
for G_j: $(V_j \cup \{k\}, E_j \cup \{k, j_1\})$
for $\bar{G}$: $(\bar{V} = V - (V_i \cup V_j), E - (E_i \cup E_j \cup (k, i_1) \cup (k, j_1)))$.

For the tree previously used as an illustration, the decomposition into three subtrees is shown in Figure 3.

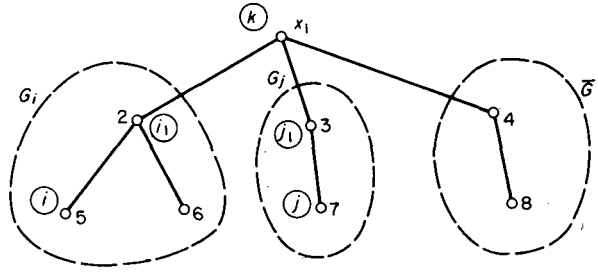

FIG. 3. Decomposition into three subtrees following removal of vertex k

The result of Lemma 1 shows that the tree expansion $f(x;G)$ is independent of any particular node v used for expansion. Furthermore, the numerator of expression (10) can be divided into three parts involving the edge sets associated with G_i, G_j and $\bar{G}$ (that is, $E_i \cup (k, i_1)$, $E_j \cup (k, j_1)$, and $E-(E_i \cup E_j \cup (k, i_1) \cup (k, j_1))$); similarly the denominator of (10) can be divided into three parts, consisting of the vertex sets associated with G_i, G_j, and $\bar{G}$, with vertex k appearing in all three sets. Expression (10) can then be rewritten as

$$f(\underline{x};G) = f(\underline{x}) = h_1(\underline{x}_1) \cdot h_2(\underline{x}_2) \cdot h_3(\underline{x}_3)$$

where $\underline{x}_1$, $\underline{x}_2$, $\underline{x}_3$ involve variables in the three subsets of nodes and edges, and h_1, h_2, h_3 are suitable functions representing the products included in (10).

Using the notation introduced earlier, one obtains

$$f(i, k) = \sum_{\underline{x}-\{i, k\}} f(\underline{x})$$

$$= \sum_{\underline{x}-\{i, k\}} h_1(\underline{x}_1) \cdot h_2(\underline{x}_2) \cdot h_3(\underline{x}_3) \tag{12}$$

With the formulation of expression (12), the four terms of expression (11b) can now be rewritten as:

$$f(i, k) = \left[\sum_{V_i-\{i\}} h_1(\underline{x}_1)\right]\left[\sum_{V_j} h_2(\underline{x}_2)\right]\left[\sum_{\bar{V}-\{k\}} h_3(\underline{x}_3)\right] \tag{12a}$$

$$f(j, k) = \left[\sum_{V_i} h_1(\underline{x}_1)\right]\left[\sum_{V_j-\{j\}} h_2(\underline{x}_2)\right]\left[\sum_{\bar{V}-\{k\}} h_3(\underline{x}_3)\right] \tag{12b}$$

$$f(i, j, k) = \left[\sum_{V_i-\{i\}} h_1(\underline{x}_1)\right]\left[\sum_{V_j-\{j\}} h_2(\underline{x}_2)\right]\left[\sum_{\bar{V}-\{k\}} h_3(\underline{x}_3)\right] \tag{12c}$$

$$\text{and} \quad f\{k\} = \left[\sum_{V_i} h_1(\underline{x}_1)\right]\left[\sum_{V_j} h_2(\underline{x}_2)\right]\left[\sum_{\bar{V}-\{k\}} h_3(\underline{x}_3)\right] \tag{12d}$$

It is clear that the product of (12a) and (12b) is identical with the product of (12c) and (12d). This proves the proposition of expression (11). □

Consider as an example the tree of Figure 3. In that case, $f(\underline{x}) = h_1(\underline{x}_1) \cdot h_2(\underline{x}_2) \cdot h_3(\underline{x}_3)$

$$= \left[\frac{P(1, 2)\,P(2, 5)\,P(2, 6)}{P(2)^2\,P(1)^2}\right]\left[\frac{P(1, 3)\,P(3, 7)}{P(3)}\right]\left[\frac{P(1, 4)\,P(4, 8)}{P(4)}\right]$$

where $P(1)^2$ is arbitrarily included in $h_1(x_1)$. From 12(a) to 12(d) it follows that

$$f(1,5) = \left[\sum_{(2,6)} \frac{P(1,2)\,P(2,5)\,P(2,6)}{P(2)^2\,P(1)^2}\right]\left[\sum_{(3,7)} \frac{P(1,3)\,P(3,7)}{P(3)}\right]\left[\sum_{(4,8)} \frac{P(1,4)\,P(4,8)}{P(4)}\right]$$

$$f(1,7) = \left[\sum_{(2,5,6)} \frac{P(1,2)\,P(2,5)\,P(2,6)}{P(2)^2\,P(1)^2}\right]\left[\sum_{(3)} \frac{P(1,3)\,P(3,7)}{P(3)}\right]\left[\sum_{(4,8)} \frac{P(1,4)\,P(4,8)}{P(4)}\right]$$

$$f(1,5,7) = \left[\sum_{(2,6)} \frac{P(1,2)\,P(2,5)\,P(2,6)}{P(2)^2\,P(1)^2}\right]\left[\sum_{(3)} \frac{P(1,3)\,P(3,7)}{P(3)}\right]\left[\sum_{(4,8)} \frac{P(1,4)\,P(4,8)}{P(4)}\right]$$

$$f(1) = \left[\sum_{(2,5,6)} \frac{P(1,2)\,P(2,5)\,P(2,6)}{P(2)^2\,P(1)^2}\right]\left[\sum_{(3,7)} \frac{P(1,3)\,P(3,7)}{P(3)}\right]\left[\sum_{(4,8)} \frac{P(1,4)\,P(4,8)}{P(4)}\right]$$

Thus, $f(1,5)f(1,7) = f(1,5,7)f(1)$. □

Using these results, it is now easy to show that a relationship exists in the tree dependence model between the correlation coefficients which measure the dependencies between term pairs. In particular for any term triplet, the correlation coefficient of a given term pair included in the triplet is automatically derivable from the coefficients of the other two term pairs in the triplet. The following proposition states the result more formally:

Proposition 3: If the joint distribution of terms follows a tree dependence structure and i, j and k are vertices of the tree such that there is path from i to j passing through k, then

$$\varrho_{ij} = \varrho_{ik} \cdot \varrho_{kj} \tag{13}$$

Remark: Van Rijsbergen (1979: 137–138) has pointed out that a formula by Kendall and Stuart (1967: 318) could be used to prove the result of Proposition 3. However the formula in Kendall is defined only for multivariate random variables, and not for the discrete random variables used here.

The result of Proposition 3 could be proved using the log-linear model and techniques similar to those given by Bishop *et al.* (1974). A direct proof is given in this study.

Proof:

$$\varrho_{ik} = \frac{E[(x_i - p_i)(x_k - p_k)]}{\sqrt{p_i p_k (1-p_i)(1-p_k)}} = \frac{f(i=1, k=1) - p_i p_k}{\sqrt{p_i p_k (1-p_i)(1-p_k)}} \tag{14}$$

Similarly

$$\varrho_{jk} = \frac{f(j=1, k=1) - p_j p_k}{\sqrt{p_j p_k (1-p_j)(1-p_k)}} \tag{15}$$

From (14) and (15) one obtains that $\varrho_{ik} \cdot \varrho_{jk}$ equals

$$\frac{[f(i=1,k=1)(f(j=1,k=1)-p_i p_k f(j=1,k=1)-p_j p_k f(i=1,k=1)+p_i p_j p_k^2]}{\sqrt{p_i p_j (1-p_i)(1-p_j)} \cdot p_k (1-p_k)} \tag{16}$$

Since i, j are independent conditional on k, by Proposition 2, the left-hand side of (11b) can be substituted in (16) for $f(i, k) f(j, k)$. Following cancellation of p_k from both numerator and denominator of (16), one obtains

$$\varrho_{ik} \cdot \varrho_{jk} = \frac{f(i=1,j=1,k=1)-p_i f(j=1,k=1)-p_j f(i=1,k=1)+p_i p_j p_k}{(1-p_k)\sqrt{p_i p_j (1-p_i)(1-p_j)}} \tag{17}$$

The numerator N of (17) may now be transformed in the following way:

$$N = f(i=1,j=1,k=1)-p_i f(j=1,k=1)-p_j f(i=1,k=1)+p_i p_j - p_i p_j (1-p_k) \tag{18}$$

Since $p_i = P(i=1) = f(i=1,k=0)+f(i=1,k=1)$, (18) is further transformed into

$$\begin{aligned} & f(i=1,j=1,k=1)-p_i f(j=1,k=1)+p_j f(i=1,k=0)-p_i p_j (1-p_k) \\ & = f(i=1,j=1,k=1)-p_i f(j=1,k=1)-p_i p_j (1-p_k) \\ & \qquad +[f(j=1,k=0)+f(j=1,k=1)] f(i=1,k=0) \\ & = f(i=1,j=1,k=1)-p_i p_j (1-p_k)+f(j=1,k=0) f(i=1,k=0) \\ & \qquad -f(j=1,k=1)[p_i - f(i=1,k=0)] \\ & = f(i=1,j=1,k=1)-p_i p_j (1-p_k)+f(k=0) f(i=1,j=1,k=0) \\ & \qquad -f(j=1,k=1) f(i=1,k=1) \end{aligned}$$

using the independent conditional property of expression (11b) with $i=1$, $j=1$, and $k=0$.

Using expression (11b) again with $i=1$, $j=1$, $k=1$, this is further transformed into

$$\begin{aligned} & = f(i=1,j=1,k=1)+f(k=0) f(i=1,j=1,k=0)-p_i p_j (1-p_k) \\ & -p_k f(i=1,j=1,k=1) = f(i=1,j=1,k=1)(1-p_k)+(1-p_k) \\ & \qquad f(i=1,j=1,k=0)-(1-p_k) p_i p_j \end{aligned} \tag{19}$$

The last expression can now be substituted for the numerator of (17) to produce

$$\varrho_{ik} \cdot \varrho_{jk} = \frac{f(i=1,j=1,k=1)+f(i=1,j=1,k=0)-p_i p_j}{\sqrt{p_i p_j (1-p_i)(1-p_j)}}$$

$$= \frac{f(i=1,j=1)-p_i p_j}{\sqrt{p_i p_j (1-p_i)(1-p_j)}} = \varrho_{ij} \quad \square$$

Consider as an example ϱ_{38} in the tree used as an example in Figures 1 to 3. Using the result derived in Proposition 3 one has from Figure 4:

$$\varrho_{38} = \varrho_{34} \cdot \varrho_{48}$$
$$\text{but} \quad \varrho_{34} = \varrho_{13} \cdot \varrho_{14}$$
$$\text{Thus} \quad \varrho_{38} = \varrho_{13} \cdot \varrho_{14} \cdot \varrho_{48}.$$

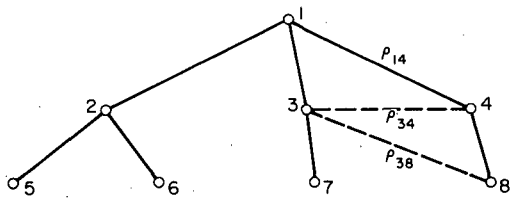

FIG. 4. Composition of correlation coefficients $\varrho_{38} = \varrho_{13} \cdot \varrho_{14} \cdot \varrho_{48}$

When Eq. (13) is valid, as it is in a pure tree dependence model, third and higher order correlations ϱ_{ijk}, ϱ_{ijh} are equal to zero. Hence when the higher order correlations are negligibly small in a practical case, the probabilities computed with the tree dependence model are about the same as those obtained with the BLE model where the actual ϱ_{ij} values are used for term pairs (i, j) that are explicitly included in the dependence tree and ϱ values for term pairs (k, h) not represented by an edge in the tree are then computed as the product of the ϱ values for the unique path leading from k to h in the tree.

Unfortunately, dependencies between term triplets and higher order term sets may not always be small. In that case the tree dependence model may still be usable in an extended form as explained in the next section.

3. A GENERALIZED DEPENDENCE MODEL

In the last section, a probabilistic expression was constructed for a given set of the tree dependencies by decomposing the tree into two subtrees connected by edge (u, v). This resulted in expressions (8) and (9). It is useful to extend the inductive construction to render it applicable to connected graphs containing triangles (that is, dependencies between term triplets). The development which follows is applicable in suitably altered form to higher order dependencies; however as a practical matter it may suffice to extend the tree dependence model by inclusion of certain third order dependencies only.

Let G be a graph consisting of three or more vertices and containing the triangle (u, v, w), but not a cycle of length four or more. A cycle of length i contains exactly i vertices and i edges. Expressions analogous to (8) and (9) may then be constructed using the triangle (u, v, w). Specifically, the following definition applies using the expansion about triangle (u, v, w):

$$f(x;G) = \frac{P(u, v, w)}{P(u)\,P(v)\,P(w)}\, f(x_u;G_u)\, f(x_v;G_u)\, f(x_w;G_w) \tag{20}$$

where G_u, G_v, and G_w are the connected subgraphs containing vertices u, v, and w, respectively, after the three edges (u, v), (u, w) and (v, w) have been removed.

When an expansion is performed about a triangle, then expression (20) can be used to represent the probability distribution of the terms. Otherwise, expression (8) which is based on the expansion about an edge not included in a triangle can be used.

It is necessary to show that the inductive definition of expression (20) is well defined and compatible with that given earlier in (8) and (9). This can be done in the following four steps:

1. An expression identical with (8) must be obtained no matter what edge incident on vertex v, other than (u, v) is chosen for expansion.
2. An expression identical with (20) must be obtained no matter what triangle incident on vertex v other than (u, v, w) is chosen for expansion.
3. The two expansions of expressions (8) and (20) about vertex v, one using an edge (u, v) and the other a triangle (u, x, y), where u and v are different from x and y, must be identical.
4. The expansions must be independent of the chosen vertex v.

Proposition 4: The inductive definitions of expressions (8) and (20) are well defined if the connected graph G has no cycle of length 4 or more.

Proof:

1. It has already been shown in Proposition 1 that the tree dependence approximation of expression (8) is independent of any particular edge chosen for expansion in the absence of triangles. Consider a graph G with two edges (u, v) and (v, w) incident on vertex v, such that neither edge is part of a triangle. Expression (8) applies in this case. After removal of edge (u, v) from the graph, two connected components remain, consisting of G_u and $G_v \cup G_w \cup (v, w)$, where G_u, G_v and G_w are edge-disjoint subgraphs which together with the edges (u, v) and (v, w) form the original graph G. (If the graph G were still connected following removal of the edge (u, v), (u, v) would be part of a cycle of length three or more in the original graph contrary to assumption.) The situation is represented schematically in Figure 5. Removal of edge (v, w) from the graph of Figure 5 will similarly produce two subgraphs consisting of G_w and $G_v \cup G_u \cup (u, v)$.

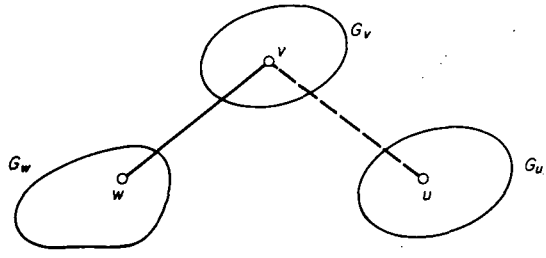

FIG. 5. Decomposition following removal of edge (u, v)

Consider now the expansion about vertex v using edge (u, v). Applying (8) one has

$$f(\underline{x};G) = \frac{P(u, v)}{P(u)\,P(v)}\, f(\underline{x}_u;G_u)\, f(\underline{x}_v \cup \underline{x}_w;\, G_v \cup G_w \cup (v, w))$$

where $\underline{x}_v \cup \underline{x}_w$ represents the union of the variables in $\underline{x}_v$ and $\underline{x}_w$.
When the last factor is itself expanded using edge (v, w) the above expression produces

$$f(\underline{x}:G) = \frac{P(u, v)}{P(u)\,P(v)}\, f(\underline{x}_u;G_u)\, \frac{P(v, w)}{P(v)\,P(w)}\, f(\underline{x}_v;G_v) f(\underline{x}_w;G_w)$$

$$= \frac{P(v, w)}{P(v)\,P(w)}\, f(\underline{x}_w;G_w) \left[\frac{P(u, v)}{P(u)\,P(v)} f(\underline{x}_u;G_u) f(\underline{x}_v;G_v) \right]$$

But the above expression is the expansion using edge (v, w). Obviously the expression about (u, v) is identical with the one about (v, w).

2. Consider now the application of expressions (20) to a situation involving triangles. Let the two triangles be (u, v, w) and (v, x, y) with common vertex v, and consider the decomposition obtained by deletion of triangle (u, v, w). The illustration of Figure 6 shows that three connected subgraphs are produced consisting of G_u, G_w, and $G_v \cup G_x \cup G_y \cup (v, x, y)$, respectively. On the other hand, deletion of triangle (v, x, y) produces the three subgraphs G_x, G_y, and $G_v \cup G_u \cup G_w \cup (u, v, w)$.

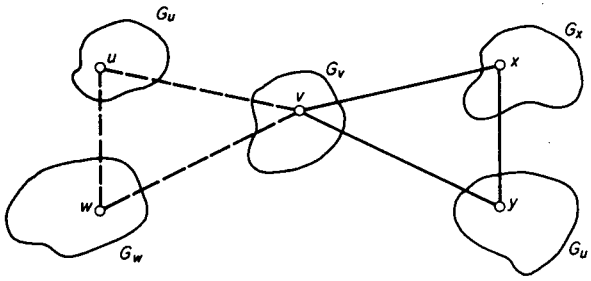

FIG. 6. Decomposition following removal of triangle (u, v, w)

A transformation similar to that carried out earlier for the edges makes clear that the expansions for the two triangles are identical.

3. Consider now a comparison of the expansion using a particular edge (x, v) with the expansion using a triangle (u, v, w) both incident on vertex v as shown in the sample graph of Figure 7. Using edge (x, v) one obtains from (8)

$$f(\underline{x};G) = \frac{P(x, v)}{P(x)\,P(v)} f(\underline{x}_x;G_x) f(\underline{x}_u \cup \underline{x}_w \cup \underline{x}_v;G_u \cup G_v \cup G_w \cup (u, v, w))$$

Using (20) this becomes

$$\frac{P(x, v)}{P(x)\,P(v)} f(\underline{x}_x;G_x)\, \frac{P(u, v, w)}{P(u)\,P(v)\,P(w)}\, f(\underline{x}_u;G_u) f(\underline{x}_v;G_v) f(\underline{x}_w;G_w)$$

$$= \frac{P(u, v, w)}{P(u)\,P(v)\,P(w)} f(\underline{x}_u;G_u) f(\underline{x}_w;G_w) f(\underline{x}_v \cup \underline{x}_x;G_v \cup G_x \cup (v, x))$$

The last expression is precisely the expansion using the triangle (u, v, w) whose removal decomposes the graph into components G_u, G_w and the connected part consisting of G_v and G_x and the edge (v, x).

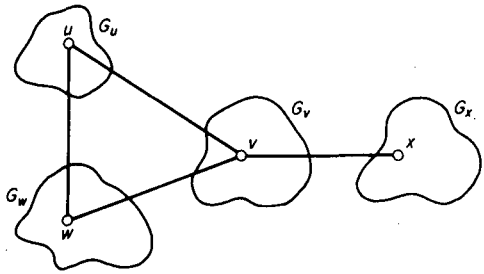

FIG. 7. Comparison of triangle and edge decomposition

4. It remains to show that in a connected graph G the expansion about any vertex v is the same as that about some adjacent vertex u. If the edge (u, v) is not part of a triangle, expression (8) produces identical expansions about either vertex u or vertex v. Similarly, expression (20) produces identical expansions for any triangle u, v, w regardless of how the vertices u, v, and w are chosen. □

Using the inductive definition for the approximating distribution of a graph dependence structure that does not include any cycles of length four or more, it is now possible to show that for any tree, say G^0, (and in particular also for the maximum spanning tree that includes the most important dependencies for pairs of terms: van Rijsbergen, 1977), the tree dependence approximation can be improved by the addition to the original graph of t edges, $t \geq 1$. Each edge added to the original tree will produce a triangle, representing the dependence between a group of three terms (a triplet). In the present development the added edges are chosen in such a way that no higher order cycles are formed in the graph, that is no cycles of length four or more.

Let the difference between two distributions $h(x)$ and $g(x)$ in n variables be measured by the information theoretical measure as

$$I(h(\underline{x}), g(\underline{x})) = \sum_{\underline{x}} h(\underline{x}) \log \frac{h(\underline{x})}{g(\underline{x})} . \tag{21}$$

x is a vector in n variables and $h(\underline{x})$ and $g(\underline{x})$ are the distributions whose difference must be measured (van Rijsbergen, 1977). It is known that $I(h(\underline{x}), g(\underline{x})) \geq 0$, the equality holding when $h(\underline{x}) = g(\underline{x})$ for all $\underline{x}$. The smaller the value of $I(h(\underline{x}), g(G^t))$ the closer the two distributions are to each other.

Consider, in particular, the original tree G^0 and the graph G^t formed by adding t edges (producing t triangles) to G^0. If $P(\underline{x})$ represents the true probability distribution which presumably includes information about the occurrence characteristics of all subsets of terms, and $f(G^0)$ and $f(G^t)$ are the dependence approximations using the tree G^0 and the graph G^t, respectively, it is possible to show that

$$I(P(\underline{x}), f(G^0)) \geq I(P(\underline{x}), f(G^t)). \tag{22}$$

The next proposition shows that each additional triangle gives a better approximation.

Lemma 5: Consider two graphs G^i and G^{i+1} such that G^{i+1} differs from G^i by addition of edge (u, v) which forms the triangle (u, v, w). Then

$$f(G^i)/f(G^{i+1}) = [P(w)\,P(u \mid w)\,P(v \mid w)/P(u, v, w). \tag{23}$$

Consider the situation in Figure 8 showing the two graphs G^i and G^{i+1}. The original edge (u, w) cannot be part of a triangle (u, w, x) in G^i, because otherwise the addition of edge (u, v) would create a cycle (x, u, v, w) of length four in G^{i+1}, contrary to assumption. Similarly, the original edge (v, w) cannot be part of a triangle in G^i. Thus by (8) the expansion in G^i about vertex w using edge (u, w) is

$$\begin{aligned} f(G^i) &= \frac{P(u, w)}{P(u)\,P(w)}\, f(G_u) f(G_w \cup G_v \cup (v, w)) \\ &= \frac{P(u, w)}{P(u)\,P(w)}\ \frac{P(v, w)}{P(v)\,P(w)}\, f(G_u) f(G_w) f(G_v) \end{aligned} \tag{24}$$

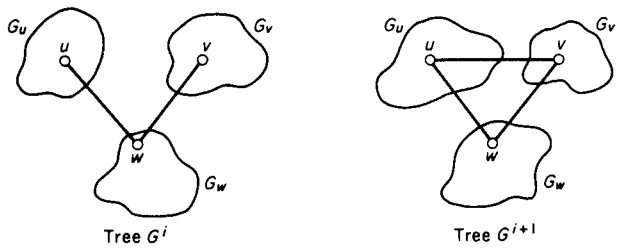

FIG. 8. Addition of one edge (u, v) forming triangle (u, v, w)

An expansion in G^{i+1} using triangle (u, v, w) can be written by (20) as

$$f(G^{i+1}) = \frac{P(u, v, w)}{P(u)\,P(v)\,P(w)}\ f(G_u) f(G_w) f(G_v) \tag{25}$$

The lemma follows immediately by division of (24) by (25).

Using (23) it is now easy to establish (22).

Proposition 6:

$$I(P(\underline{x}), f(G^i)) \geq I(P(\underline{x}), f(G^{i+1})).$$

Proof:

$$\begin{aligned} &I(P(\underline{x}), f(G^i)) - I(P(\underline{x}), f(G^{i+1})) \\ &= \sum_{\underline{x}} P(\underline{x}) \log \frac{P(\underline{x})}{f(G^i)} - \sum_{\underline{x}} P(\underline{x}) \log \frac{P(\underline{x})}{f(G^{i+1})} \end{aligned}$$

$$= \sum_{\underline{x}} P(\underline{x}) \log \frac{f(G^{i+1})}{f(G^i)}$$

$$= \sum_{\underline{x}} P(\underline{x}) \log \frac{P(u, v, w)}{P(w)\, P(u \mid w)\, P(v \mid w)} \quad \text{from (23)}$$

$$= \sum_{(u, v, w)} P(u, v, w) \log \frac{P(u, v, w)}{P(w)\, P(u \mid w)\, P(v \mid w)}$$

$$= I(P(u, v, w), P(w)\, P(u \mid w)\, P(v \mid w)) \tag{26}$$

The last expansion is necessarily greater or equal to zero because the information theoretic measure is always non-negative. □

The foregoing development shows that the information theoretic measure for the two distributions using G^i and G^{i+1} differs precisely by the difference due to the use of triangle (u, v, w) on the one hand, and the edges (u, w) and (v, w) on the other. An improved approximation to the distribution can be obtained by selectively adding edges to the dependence tree in such a way that at each point the value of

$$W = \sum_{u, v, w} P(u, v, w) \log \frac{P(u, v, w)}{P(w)\, P(u \mid w)\, P(v \mid w)} \tag{27}$$

is maximized. The first triangle to be formed could be the one for which W is maximum; the next triangle could produce the next highest value of W, and so on, until no further triangles can be generated without adding cycles of length four or more.

In summary, the tree dependence model is a computationally attractive method for including dependencies between certain pairs of terms in a probabilistic retrieval system. The computed probabilities are guaranteed to produce positive values, and the differences between the tree dependence model and the optimum probabilistic model will be small when the higher order term dependencies are small.

When dependencies between term triplets, quadruplets and higher order term subsets become substantial, it is possible to improve the tree dependence model by selective consideration of term triplets in addition to term pairs. The triplets to be added could be chosen in decreasing order of the values of W in expression (27). When triplets that do not form cycles of length four are exhausted, further improvements may be obtainable by adding dependencies between term quadruplets that do not produce cycles of length five, and so on for the higher order dependencies. Eventually the extended tree dependence distribution converges with the true distribution given by the Bahadur Lazarsfeld expression. However, in practice, it is unlikely that fourth or higher order dependencies can be easily determined. The extended tree dependence model described here is a product approximation of the kind introduced in Lewis (1959).

4. EXPERIMENTAL WORK

4.1 Parameter estimation process

To carry out experiments using the various probabilistic models (term independence, standard tree dependence, and generalized term dependence), it is necessary to obtain for each query a ranking of the documents x in decreasing order of the expression $P(\underline{x} \mid \text{rel})/P(\underline{x} \mid \text{non-rel})$. For each document, expressions (5) or (6) may be used for the calculations in the term independence model. Expressions (7) and (8) serve similarly in the tree dependence system, and expression (20) is used in the generalized system for the term triplets. In each case, only those document terms which are also included in the corresponding query are used in the calculations since these may be of greatest importance in retrieval.

To illustrate the operations of the tree dependence systems it is necessary to include in the calculations a certain number of dependent term pairs in addition to the individual, single query terms. A sufficient number of usable term pairs can be generated by modifying the original user queries before carrying out the probabilistic calculations through addition of new query terms related to the ones originally present. The following sequence of steps may be used for this purpose (van Rijsbergen, 1977; Harper and van Rijsbergen, 1978; Salton *et al.*, 1983):

1. A maximum spanning tree (MST) is constructed for the terms included in a given document collection in such a way that each vertex represents a term, each edge represents a dependent term pair, and the sum of the edge weights identifying the amount of useful dependency information between pairs of terms is maximized.
2. The original available queries are expanded by using the MST to add to each query all terms that are immediately adjacent to the vertices representing the original query terms.
3. The pairwise occurrence probabilities $P(i, j \mid \text{rel})$ and $P(i, j \mid \text{non-rel})$ are obtained for all pairs (i, j) included in the expanded query (that is, for each query term pair represented by an edge in the spanning tree). The co-occurrence and dependency information allow these values to be calculated for pairs included in the MST.
4. Term triples are identified for all sets of three terms for which the individual terms occur in the expanded query, and two of the three possible edges appear adjacently in the MST (that is, they share a common vertex). For example, the triple (x_i, x_j, x_k) is identified if the three terms are included in the expanded query and vertices (x_i, x_j) and (x_j, x_k) (or alternatively, pairs (x_i, x_k) and (x_k, x_j), or pairs (x_j, x_i) and (x_i, x_k)) appear in the MST. For each identified triple, the probability factors $P(i, j, k \mid \text{rel})$ and $P(i, j, k \mid \text{non-rel})$ are computed as well as the corresponding W value of expression (27).
5. For each document $\underline{x}$, the factors $P(\underline{x} \mid \text{rel})$ and $P(\underline{x} \mid \text{non-rel})$ are computed, assuming either the term independence model, the tree dependence model, or the generalized term dependence model, by summing the values of the corresponding probability expression for all query terms included in document x. The documents are then ranked in decreasing order according to expression (1), and the corresponding recall and precision values are computed.

In the experimental process, the maximum spanning tree is used for two distinct purposes:

1. The tree specifies $(m-1)$ term pairs out of the $m(m-1)/2$ possible pairs, and by extension a subset of term triples, which can be taken into account in the tree dependence system.
2. The tree is used to supply an adequate number of dependent query term pairs using the previously mentioned query expansion process.

The spanning tree structure thus supplies a manageably small number of dependent term pairs to be used in the tree dependency model, and the query expansion system ensures that some of these same term pairs included in the spanning tree are also present in an expanded query. On the other hand, certain term pairs derivable from the original query terms may well not be included in the spanning tree even though these pairs might constitute important indicators of query content. Obviously such pairs cannot be included in the tree dependency calculations. Furthermore, some term pairs derivable from an expanded query which are usable in the tree dependence calculations might well not be helpful in retrieving relevant documents. Whether the terms added by the query expansion process will actually help in the retrieval activity depends on whether the added query terms are reflective of the user's information needs, or more precisely, on the occurrence characteristics of the added terms in the relevant and non-relevant documents of the collection.

An example of the query expansion process is shown in simplified form in Figure 9. Given an initial query $Q=(x_2, x_3)$ and the maximum spanning tree of Figure 9(b), only the term independence model is directly applicable, since pair (x_2, x_3) is not available in the spanning tree. The expanded query $Q=(x_1, x_2, x_3, x_4, x_5)$ leads to the use of the four pairs specified in the tree $((x_1, x_2), (x_1, x_3), (x_3, x_4)$ and $(x_3, x_5))$. In Figure 9(c) a single dependent term triple (x_1, x_2, x_3) is used instead of the two pairs (x_1, x_2) and (x_1, x_3).

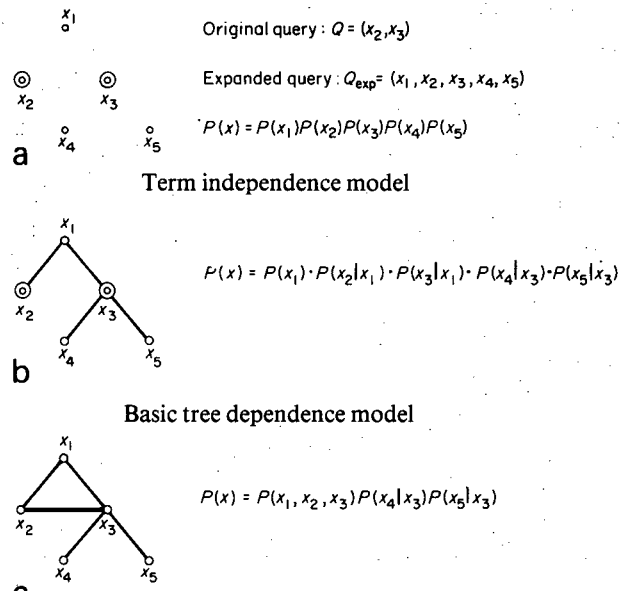

FIG. 9. Operations of extended tree dependence system

In computing the formula of expression (1), it is necessary to estimate values of

$$\begin{aligned} p_i &= P(x_i = 1 \mid \text{rel}) \\ 1-p_i &= P(x_i = 0 \mid \text{rel}) \\ p_i' &= P(x_i = 1 \mid \text{non-rel}) \\ \text{and} \quad 1-p_i' &= P(x_i = 0 \mid \text{non-rel}). \end{aligned} \tag{28}$$

Normally, the occurrence probabilities p_i and p_i' of terms x_i in the relevant and non-relevant documents of a collection are obtained by using actual occurrence frequencies of the terms in the respective document subsets. In particular

$$p_i \approx r_i/R \text{ and } p_i' \approx \frac{n_i - r_i}{N-R} \tag{29}$$

where r_i and n_i represent the occurrence frequencies of term x_i in the relevant document set and in the whole collection, respectively, and R and N represent the size of the relevant document set and the total collection size.

It is clear that unless the relevant and non-relevant document subsets with respect to each query are properly identified, problems will arise in the evaluation of expression (1). Two possibilities offer themselves for obtaining the values of p_i and p_i' in (29). A *retrospective* experiment can be performed in which the (unrealistic) assumption is made that all relevant and non-relevant documents with respect to each query are known in advance of each search. In that case, the values of p_i and p_i' are readily computable for all terms x_i. Alternatively, in a more realistic *predictive* experiment the initial queries are first used to retrieve a subset $R' \subseteq R$ of documents identified as relevant to the query, and a subset $N' \subseteq N-R'$ of documents identified as non-relevant to the query. Instead of using the full set of relevant and non-relevant documents R and $N-R$ for the parameter estimation process, the partial subsets of initially retrieved items R' and N' are used for the predictive calculations (Robertson and Sparck Jones, 1976; van Rijsbergen, 1977; Harper and van Rijsbergen, 1978).

Two problems arise in performing the predictive experiments: on the one hand, not enough information may be available to permit an accurate estimation of the parameters p_i and p_i' for the terms x_i; in particular the subset of relevant or non-relevant items actually available may be very small, leading to inaccurate occurrence probability estimates. The evaluation process is also complicated by the fact that the relevant and non-relevant items initially retrieved and used to derive the p_i and p_i' values should not be used again in evaluating the results of the subsequent probabilistic searches.

Consider first the problem of deriving the values for p_i and p_i' in the predictive case. When a term does not occur in the relevant or non-relevant documents with respect to some query, then p_i or p_i' are equal to 0. In that case $P(x)$ will be 0 and the value of expression (1) may not be computable. Furthermore, when by mischance no relevant items at all are initially retrieved in response to a given query, both r_i and R are equal to 0, and the first expression in (29) is computed as 0/0. To avoid such an undesirable result, it is customary to adjust expressions (29) by addition of constants as follows (Robertson and Sparck Jones, 1976; van Rijsbergen, 1977; Harper and van Rijsbergen, 1978; Robertson *et al.*, 1981):

$$p_i \approx \frac{r_i + 0.5}{R + 1} \text{ and } p_i' \approx \frac{n_i - r_i + 0.5}{N - R + 1} \tag{30}$$

The adjusted parameter estimation process of expression (30) has been widely used in practice, but when r_i and R are small, unsatisfactory estimates are often produced (Sparck Jones, 1979). Consider, for example, the common situation where $R = 1$ and $r_i = 0$ (that is, one relevant document has been retrieved which does not contain term x_i). In that case, one finds that $p_i = 0.25$ and $p_i' << 0.25$ since N, the total number of retrieved documents, is necessarily larger than n_i, the number of retrieved documents with term x_i. So from the information that a term x_i does not occur in a relevant document, one reaches the unusual conclusion that term x_i is more likely to occur in the relevant than in the non-relevant items.

Instead of using the conventional adjustments of expression (30) it is desirable to introduce modified expressions that are compatible with the probabilistic model and produce more reasonable values for p_i and p_i' than the conventional estimation process. If one assumes that the number of relevant documents not yet retrieved is not much larger than the number of relevant items retrieved in the initial search, and that each term x_i is randomly distributed in the relevant items that have not yet been seen, one obtains the following probability estimates (Buckley, 1983):

$$p_i = \frac{r_i + \dfrac{n_i - r_i}{N - R}}{R + 1} \tag{31a}$$

and

$$p_i' = \frac{n_i - r_i - \dfrac{n_i - r_i}{N - R}}{N - R - 1}. \tag{31b}$$

The formulae of expression (31) add or subtract a very small amount to the values of expression (29), the error correction being centred around the expected value where $r_i/R \approx n_i/N$. When $r_i = 0$ and $R \neq 0$, $p_i \approx p_i' \cdot 1/(R+1)$; that is $p_i < p_i'$ as desired. Furthermore when $r_i = R = 0$, $p_i = p_i' = n_i/N$. The formulae of expression (31) were used to estimate the probability values in the experiments described in the remainder of this section.

4.2 Retrospective experiments

To obtain a general idea of the operations of the generalized term dependence model, a number of experiments were carried out using two sample document collections:

1. The Medlars collection consisting of 1033 documents in biomedicine used with 30 queries.
2. The ISI collection consisting of 1460 highly cited documents in library science and documentation extracted from the Social Science Citation Index and used with 76 user queries.

Both retrospective and predictive experiments were carried out, the output being presented as recall–precision tables giving the precision values at certain fixed values of the recall averaged over the respective query sets. For each experimental run a composite precision value is also shown, representing the average precision at three different recall points including recall values of 0.25, 0.50, and 0.75, respectively. For the retrospective experiments, advance knowledge is assumed of the relevance, or non-relevance, of all documents in the collection. In that case it is unnecessary to perform any retrieval runs at all because the original query (or alternatively the expanded query obtained by adding adjacent terms from the maximum spanning tree) may be used directly to compute the necessary probabilities.

More specifically, for each query term i, the values for p_i and p_i' are calculated using the occurrence properties of the terms in the full set of relevant and non-relevant documents. Similarly, the values of the pairwise and tripletwise probabilities are obtained for the term pairs and triples included in the spanning tree. Finally, the documents may be ranked in decreasing order of the optimal decision rule (1), $P(\underline{x} \mid \text{rel})$ and $P(\underline{x} \mid \text{non-rel})$ being computed by using formulae (5), (7), and (20) for the term independence, tree dependence, and generalized term dependence cases, respectively. Since the document ranking for the retrospective experiments is obtained by using the full relevance information for all documents, the corresponding evaluation results represent an upper bound of what can be achieved with the given query and document collections for the corresponding methods.

The evaluation output for the retrospective case is presented in Tables 1 and 2. The output of Table 1 shows that for both the Medlars and ISI collections, the query expansion process using the maximum spanning tree helps in retrieving additional relevant documents. Table 1 covers original as well as expanded queries under the term independence model of equations (5) and (6). It may be noted that the ISI collection performs relatively poorly even under the optimality assumptions inherent in the retrospective case, the average precision for the expanded queries reaching only 0.5797.

Table 1. Retrospective experiments comparing expanded and unexpanded queries

	Medlars 1033, 30 queries term independence, single terms (no pairs, no triples)		*ISI 1460, 76 queries term independence, single terms (no pairs, no triples)*	
Recall	*Expanded queries*	*Original unexpanded*	*Expanded queries*	*Original queries*
0.1	0.9603	0.9263	0.8911	0.7190
0.2	0.9375	0.8832	0.8023	0.5853
0.3	0.8913	0.8381	0.7244	0.4722
0.4	0.8703	0.8080	0.6489	0.4145
0.5	0.8415	0.7455	0.5949	0.3749
0.6	0.8084	0.6517	0.5175	0.3302
0.7	0.7499	0.5946	0.4260	0.2601
0.8	0.6661	0.4915	0.3499	0.2150
0.9	0.4738	0.3113	0.2627	0.1534
1.0	0.2708	0.1479	0.1937	0.1157
Average at $R = 0.25$, 0.50, 0.75	0.8239	0.7205 (−12.6%)	0.5797	0.3797 (−34.5%)

The retrospective output of Table 2 shows that when full relevance information is available and the probabilistic parameters can therefore be computed exactly, the theoretical results outlined in this study are precisely confirmed. For both the Medlars and the ISI collections, the basic tree dependence system is substantially better than the term independence model. As additional term triples are taken into account small improvements are obtained, and this advantage grows with the number of added triples. It is clear that in the retrospective case, where no probability estimation problems arise, the added term pairs and term triples make it increasingly easier to distinguish the relevant from the non-relevant items. For the Medlars collection, the average precision value rises from an already nearly perfect value of 0.9314 for the tree dependence to 0.9538 for the extended dependence

Table 2. Retrospective experiments with expanded queries comparing various term dependence cases

Medlars 1033, 30 queries with tree expansion					
Recall	*Term independence (single terms)*	*Term dependence all pairs*	*Dependence all pairs one triple*	*Dependence all pairs two triples*	*Dependence all pairs all triples*
0.1	0.9603	0.9917	0.9917	1.0000	1.0000
0.2	0.9375	0.9846	0.9849	0.9921	0.9898
0.3	0.8913	0.9685	0.9710	0.9823	0.9843
0.4	0.8703	0.9590	0.9594	0.9719	0.9746
0.5	0.8415	0.9445	0.9488	0.9560	0.9717
0.6	0.8085	0.9222	0.9342	0.9410	0.9536
0.7	0.7499	0.8896	0.8976	0.9046	0.9257
0.8	0.6661	0.8410	0.8446	0.8491	0.8605
0.9	0.4738	0.6379	0.6471	0.6495	0.6876
1.0	0.2708	0.3706	0.3769	0.3797	0.4530
Average at $R=0.25$, 0.50, 0.75	0.8239	0.9314 (+13.1%)	0.9336 (+13.3%)	0.9405 (+14.2%)	0.9538 (+15.8%)
ISI 1460, 76 queries with tree expansion					
Recall	*Term independence (single terms)*	*Term dependence all pairs*	*Dependence all pairs one triple*	*Dependence all pairs two triples*	*Dependence all pairs all triples*
0.1	0.8911	0.9436	0.9446	0.9520	0.9711
0.2	0.8023	0.8994	0.9012	0.9013	0.9332
0.3	0.7244	0.8479	0.8448	0.8555	0.8940
0.4	0.6489	0.7750	0.7757	0.7803	0.8310
0.5	0.5949	0.7253	0.7271	0.7333	0.7835
0.6	0.5175	0.6752	0.6807	0.6852	0.7291
0.7	0.4260	0.6103	0.6145	0.6184	0.6657
0.8	0.3499	0.5335	0.5355	0.5413	0.5921
0.9	0.2627	0.4089	0.4154	0.4233	0.4909
1.0	0.1937	0.2731	0.2838	0.2927	0.3275
Average at $R=0.25$, 0.50, 0.75	0.5797	0.7229 (+24.7%)	0.7247 (+25.0%)	0.7297 (+25.9%)	0.7774 (+34.1%)

system which includes all the triples derivable from the spanning tree. The corresponding figures for ISI are 0.7229 and 0.7774, respectively. One may expect that under the ideal retrospective conditions similar results are obtainable for other document collections.

4.3 Predictive experiments

The predictive experiments differ from the retrospective ones in that they are designed to simulate an actual retrieval operation. In particular, a number of documents are retrieved in some initial search operation. These initially retrieved items are examined for relevance or non-relevance with respect to each query, and the resulting information is used to compute the probability values p_i and p_i' for each term, as well as the pairwise and tripletwise probabilities for pairs and triples included in the maximum spanning tree. The probabilistic parameters are then used to rank the documents in decreasing order of the values of expression (1), as previously explained.

To carry out the predictive experiments a standard vector processing run can be performed based on a simple automatic indexing process in which word stems extracted from document abstracts, or from natural language query formulations, are used to represent document or query content. A term weight can be automatically assigned to each term, consisting of the product of the frequency of the term in each document multiplied by the inverse document frequency of the term in the collection under discussion. Finally the similarity of each document and each query can be determined as the cosine of the corresponding term vectors (Salton and McGill, 1983). In the predictive experiments the documents retrieved by the vector processing run in the top 20 ranks can be used for the computation of the probabilistic parameters. To obtain a fair comparison between the probabilistic retrieval runs and the initial cosine run, it is necessary to discount the performance of the relevant and non-relevant items retrieved in the top 20 ranks, since these are utilized to estimate the parameters needed for the probabilistic formulae. This is done by using a rank freezing process which fixes the relevant items originally retrieved at their initial ranks, while discarding the non-relevant items initially seen and replacing them by new items retrieved at lower ranks (Salton *et al.*, 1983).

The predictive experiments carried out for purposes of this study produced essentially negative results (*see* Table 3) in the sense that the best results were obtained with the term independence method where only single terms are used. The tree dependence method based on singles as well as pairs was less effective, and additional small losses were produced when term triples were taken into account. Three main explanations may be offered for the failure of the output in the predictive situation:

1. The probability estimation problems necessarily grow worse when higher order dependencies must be included than for the term independence case where only the single term probabilities are needed.
2. The relevant documents are characterized more precisely when the higher order term dependencies are taken into account than when they are not; hence any new relevant documents not yet retrieved are constrained to look increasingly similar to the originally retrieved relevant items when the higher order dependencies are included. This is all to the good when the relevant items not yet seen are indeed similar to the relevant originally retrieved in the top 20 ranks; but

Table 3. Predictive experiments with expanded queries comparing various term dependence cases

Medlars 1033, 30 queries with tree expansion						
Recall	*Term independence single terms*	*Term dependence all pairs*	*Dependence all pairs one triple*	*Dependence all pairs two triples*	*Dependence all pairs all triples*	*Vector processing cosine match (continued)*
0.1	0.9539	0.9499	0.9455	0.9461	0.9359	0.8908
0.2	0.9305	0.9104	0.9070	0.9049	0.9028	0.8627
0.3	0.9086	0.8848	0.8794	0.8758	0.8687	0.8211
0.4	0.8858	0.8374	0.8414	0.8342	0.8289	0.7729
0.5	0.8601	0.7546	0.7508	0.7374	0.7471	0.7016
0.6	0.7955	0.6453	0.6431	0.6173	0.7313	0.6199
0.7	0.7195	0.5040	0.4782	0.4714	0.4941	0.5342
0.8	0.6273	0.3805	0.3808	0.3364	0.3473	0.4344
0.9	0.3988	0.1680	0.1540	0.1491	0.1512	0.2366
1.0	0.1942	0.0981	0.0859	0.0900	0.0878	0.1198
Average at $R=0.25$, 0.50, 0.75	0.8242	0.7066 (−14.3%)	0.6978 (−15.3%)	0.6938 (−15.8%)	0.6961 (−15.5%)	0.6739 (−18.2%)
ISI 1460, 76 queries with tree expansion						
Recall	*Term independence single terms*	*Term dependence all pairs*	*Dependence all pairs one triple*	*Dependence all pairs two triples*	*Dependence all pairs all triples*	*Vector processing cosine match (continued)*
0.1	0.3501	0.3181	0.3132	0.3092	0.3276	0.3952
0.2	0.2513	0.2149	0.2196	0.2212	0.2285	0.2962
0.3	0.1983	0.1186	0.1186	0.1158	0.1411	0.2238
0.4	0.1539	0.0727	0.0713	0.0697	0.0883	0.1657
0.5	0.1241	0.0484	0.0488	0.0491	0.0571	0.1347
0.6	0.1007	0.0408	0.0414	0.0407	0.0417	0.1102
0.7	0.0777	0.0346	0.0340	0.0339	0.0345	0.0841
0.8	0.0626	0.0317	0.0314	0.0312	0.0316	0.0681
0.9	0.0495	0.0303	0.0301	0.0300	0.0296	0.0514
1.0	0.0393	0.0290	0.0290	0.0250	0.0291	0.0374
Average at $R=0.25$, 0.50, 0.75	0.1415	0.0822 (−41.9%)	0.0829 (−41.4%)	0.0813 (−42.5%)	0.0923 (−34.8%)	0.1560 (+10.2%)

when that condition is not met, as it may not be met for the experimental collections, the term independence system may be expected to outperform the dependency models.

3. The query expansion process using the maximum spanning tree may not supply all the term pairs and/or triples that are required accurately to represent the relevant items in the collection.

Experiments are currently under way designed to improve the probability estimation process and to take into account term pairs and higher order term

dependencies other than those specified by a maximum spanning tree. The corresponding results may be presented in a subsequent note.

REFERENCES

Bishop, Y. M. M., Fienberg, S. E. and Holland, P. W. (1974) *Discrete Multivariate Analysis: Theory and Practice.* Cambridge, Massachusetts: MIT Press.

Buckley, C. (1983) *Probability estimation.* Department of Computer Science, Cornell University, Ithaca, New York (Technical Report).

Chow, C. K. and Liu, C. N. (1968) Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory IT-14*, 462–467.

Chow, D. and Yu, C. T. (1982) On the construction of feedback queries. *Journal of the ACM 29*, 127–151.

Duda, R. O. and Hart, P. E. (1973) *Pattern Classification and Scene Analysis.* New York: J. Wiley and Sons.

Harper, D. J. (1980) *Relevance feedback in document retrieval systems.* Doctoral dissertation, University of Cambridge, England.

Harper, D. J. and van Rijsbergen, C. J. (1978) An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation 34*, 189–216.

Kendall, M. G. and Stuart, A. (1967) *Advanced Theory of Statistics*, Vol. 2 (2nd edn). London: C. Griffin.

Kraft, D. and Bookstein, A. (1978) Evaluation of information retrieval systems: A decision theory approach. *Journal of the American Society for Information Science 29*, 31–40.

Lam, K. and Yu, C. T. (1982) A clustered search algorithm interpreting arbitrary term dependencies. *ACM Transactions on Data Base Systems 7*, 500–508.

Lewis, P. M. (1959) Approximating probability distributions to reduce storage requirements. *Information and Control 2*, 214–225.

Maron, M. E. and Kuhns, J. L. (1960) On relevance, probabilistic indexing and information retrieval. *Journal of the ACM 7*, 216–244.

Robertson, S. E. (1977) The probability ranking principle in information retrieval. *Journal of Documentation 33*, 294–304.

Robertson, S. E. and Sparck Jones, K. (1976) Relevance weighting of search terms. *Journal of the American Society for Information Science 27*, 129–146.

Robertson, S. E., van Rijsbergen, C. J. and Porter, M. F. (1981) Probabilistic models of indexing and searching. *Information Retrieval Research.* (R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen and P. W. Williams, eds.) pp. 35–56. London: Butterworths.

Salton, G. (1979) Mathematics and information retrieval. *Journal of Documentation 35*, 1–29.

Salton, G., Buckley, C. and Yu, C. T. (1983) An evaluation of term dependence models in information retrieval. *Research and Development in Information Retrieval. Lecture Notes in Computer Science*, Vol. 146. (G. Salton and H. J. Schneider, eds.) pp. 151–173. Berlin: Springer-Verlag.

Salton, G., Fox, E. A., Buckley, C. and Voorhees, E. (1983) *Boolean query formulation with relevance feedback.* Department of Computer Science, Cornell University, Ithaca, New York (Technical Report 83-539).

Salton, G. and McGill, M. J. (1983) *Introduction to Modern Information*, Chapter 3. New York: McGraw-Hill Book Company.

Sparck Jones, K. (1979) Search term relevance weighting given little relevance information. *Journal of Documentation 35*, 30–48.

Van Rijsbergen, C. J. (1977) A theoretical basis for the use of cooccurrence data in information retrieval. *Journal of Documentation 33*, 106–119.

Van Rijsbergen, C. J. (1979) *Information Retrieval* (2nd edn). London: Butterworths.

Yu, C. T., Luk, W. S. and Siu, M. K. (1979) On models of information retrieval processes. *Information Systems 4*, 205–218.

Yu, C. T. and Salton, G. (1976) Precision weighting—An effective automatic indexing method. *Journal of the ACM 23*, 76–88.