A GENERALIZED TERM DEPENDENCE MODEL IN INFORMATION RETRIEVAL

C.T. Yu*
C. Buckley**
K. Lam***
G. Salton**

February 1983
TR 83-543

*Department of Information Engineering,
University of Illinois-Chicago Circle
Chicago, IL 60680

**Department of Computer Science
Cornell University
Ithaca, NY 14853

***Department of Statistics
Hong Kong University
Hong Kong

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

A Generalized Term Dependence Model in Information Retrieval

C.T. Yu,* C. Buckley**, K. Lam,*** and G. Salton**

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 data base 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 appearings 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 nigher-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 demonstrate the usefulness of the generalized system.

---

*Department of Information Engineering, University of Illinois-Chicago Circle, Chicago, Illinois 60680.
**Department of Computer Science, Cornell University, Ithaca, New York 14853.
***Department of Statistics, Hong Kong University, Hong Kong.

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

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

In particular, consider 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

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

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]

The probabilistic 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 vector elements x_i in the relevant and nonrelevant 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 $P(\underline{x})$ is given by the Bahadur Lazarsfeld expansion (BLE) as follows: [5-7]

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

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

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

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,

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

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

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

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 nonrelevant 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:

a) it is in practice impossible to compute the correlation coefficients for an exponential number of term combinations;

b) 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 nonrelevant documents of a collection. In that case

$$P(\underline{x}) = P(x_1)\ P(x_2)....P(x_n). \tag{5}$$

For the independence case, the BLE expansion reduces to

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

since all ρ values will be equal to 0. [8-9]

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

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:

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

where b is the parent of a and the product is taken over all edges of E. [10-12] 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

$$f(x;G) = P(1)\ P(2|1)\ P(3|1)\ P(4|1)\ P(5|2)\ P(6|2)\ P(7|3)\ P(8|4).$$

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

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

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

$$= \frac{P(u,v)}{P(u)P(v)}\ f(\underline{x}_v;G_v)\ f(\underline{x}_u;G_u) \tag{8}$$

where $\underline{x}_v$ and $\underline{x}_u$ 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),

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

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.

Lemma 1: For a tree G = (V,E), the tree dependence $t(\underline{x};G)$ given by (7) is independent of the chosen root and of the direction of the edges. In particular $f(\underline{x};G)$ 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.

Proof: Since (7) in 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 Figure 2). By the inductive hypothesis, (8) gives

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

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

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). □

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

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

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

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

This is of course identical with the formula derived from the tree of Fig. 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 nonnegative. 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,

$$f(i,j|k) = f(i|k) \cdot f(j|k) \qquad (11a)$$

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 Fig. 3.

The result of Lemma 1 shows that the tree expansion $f(\underline{x};G)$ 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

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

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

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

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

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

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

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

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 Fig. 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

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

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

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

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

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

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

Remark: It has been pointed out ([13], p. 137-138) that a formula by Kendall ([14], p. 318) could be used to prove the result of proposition 3. However the formula in Kendall is defined only for multi-variate 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 [15]. A direct proof is given in this study.

Proof:

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

Similarly

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

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

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

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

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

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

$$f(i=1,j=1,k=1) - p_i f(j=1,k=1) + p_j f(i=1,k=0) - p_i p_j (1-p_k)$$

$$= f(i=1,j=1,k=1) - p_i f(j=1,k=1) - p_i p_j (1-p_k) + [f(j=1,k=0) + f(j=1,k=1)] f(i=1,k=0)$$

$$= f(i=1,j=1,k=1) - p_i p_j (1-p_k) + f(j=1,k=0) f(i=1,k=0) - f(j=1,k=1) [p_i - f(i=1,k=0)]$$

$$= f(i=1,j=1,k=1) - p_i p_j (1-p_k) + f(k=0) f(i=1,j=1,k=0) - f(j=1,k=1) f(i=1,k=1)$$

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

$$= f(i=1,j=1,k=1) + f(k=0) f(i=1,j=1,k=0) - p_i p_j (1-p_k) - p_k f(i=1,j=1,k=1)$$

$$= f(i=1,j=1,k=1)(1-p_k) + (1-p_k) f(i=1,j=1,k=0) - (1-p_k) p_i p_j \qquad (19)$$

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

produce

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

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

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

$$\rho_{38} = \rho_{34} \cdot \rho_{48}$$

but $\rho_{34} = \rho_{13} \cdot \rho_{14}$.

Thus $\rho_{38} = \rho_{13} \cdot \rho_{14} \cdot \rho_{48}$.

When equation (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):

$$f(\underline{x};G) = \frac{P(u,v,w)}{P(u)P(v)P(w)} f(\underline{x}_u;G_u)\, f(\underline{x}_v;G_u)\, f(\underline{x}_w;G_w) \qquad (20)$$

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:

a) an expression identical with (8) must be obtained no matter what edge incident on vertex v, other than (u,v) is chosen for expansion;

b) 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;

c) 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;

and d) 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: a) 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 Fig. 5. Removal of edge (v,w) from the graph of Fig. 5 will similarly produce two subgraphs consisting

of G_w and $G_v \cup G_u \cup (u,v)$.

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).

b) 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 Fig. 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 $G_v \cup G_u \cup G_w \cup (u,v,w)$.

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

c) 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 Fig. 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).

d) 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 [10]), 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 depen-

dence 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 $h(\underline{x})$ and $g(\underline{x})$ in n variables be measured by the information theoretical measure as

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

$\underline{x}$ is a vector in n variables and $h(\underline{x})$ and $g(\underline{x})$ are the distributions whose difference must be measured. [10]. 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

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

Consider the situation in Fig. 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 4 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

$$f(G^i) = \frac{P(u,w)}{P(u)P(w)} \; f(G_u) \; f(G_w \cup G_v \cup (v,w))$$

$$= \frac{P(u,w)}{P(u)P(w)} \; \frac{P(v,w)}{P(v)P(w)} \; f(G_u) \; f(G_w) \; f(G_v) \tag{24}$$

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:

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

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

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

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

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

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

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

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 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

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

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 [16].

4. Experimental Results

The generalized term dependence model is evaluated by using a small sample collection of 1033 documents and 30 queries in biomedicine. Specifically for each query the various probabilistic models (term independence, standard tree dependence, and generalized term dependence) are used to obtain a ranking of the documents $\underline{x}$ in decreasing order of the expression $P(\underline{x}|rel)/P(\underline{x}|nonrel)$. For each document, expression (5) or (6) is 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.

To insure that a sufficient number of dependent term pairs are available for use in the tree dependence systems, the original user queries are expanded before the probabilistic calculations are actually made by adding new related terms to the ones originally present. The following sequence of steps is

used: [17]

a) 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.

b) 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.

c) The pairwise occurrence probabilities P(i,j|rel) and P(i,j|nonrel) 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.

d) 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|rel) and P(i,j,k|nonrel) are computed as well as the corresponding W value of expression (27).

e) For each document $\underline{x}$, the factors $P(\underline{x}|rel)$ and $P(\underline{x}|nonrel)$ 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 $\underline{x}$. 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:

a) the tree specifies the subset of term pairs, and by extension of term triples, which can be taken into account in the tree dependence system;

b) the tree is used to supply an adequate number of dependent query term pairs using the previously mentioned query expansion process.

The query expansion process has nothing as such to do with the operations of a probabilistic retrieval system. In fact, the query expansion will be injurious if the added query terms are not reflective of the user's original information needs. Unfortunately, the term pairs defined by an unexpanded query are not likely to be explicitly present in the MST. In that case, the tree dependence model reduces by default to the term independence model since no pairs are then available for use. For this reason some procedure must be used in any probabilistic term dependence model to insure that term dependence information is in fact available for an adequate number of subsets of terms.

An example of the query expansion process is shown in simplified form in Fig. 9. Given an initial query $Q = (x_2, x_3)$ and the maximum spanning tree of

Fig. 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 Fig. 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).

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

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

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

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

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 nonrelevant 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 <u>retrospective</u> experiment can be performed in which the (unrealistic) assumption is made that all relevant and

nonrelevant 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 nonrelevant to the query. Instead of using the full set of relevant and nonrelevant 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.

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 nonrelevant 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 nonrelevant 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 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: [8,12]

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

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. 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 $p_i' \ll 0.25$ 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 nonrelevant items.

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 nave not yet been seen, one obtains the following probability estimates: [18]

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

and

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

In the previously cited limiting cases, the new estimates of expression (31) provide the following more sensible values: when no relevant items are intially retrieved and R = r = 0, one finds $P(x_i|\text{rel}) = P(x_i|\text{nonrel})$. When the only relevant item does not contain a given term x_i, one has $p_i = 1/2\, p_i'$. The formulas of expression (31) were used to estimate the probability values in the experiments described in this section.

The experimental output for the generalized term dependence model is included in Table 1 for the Medlars collection of 1033 documents and 30 queries. Instead of showing full recall-precision output, Table 1 contains the average precision values for three recall points, corresponding to a low recall of 0.25, a medium recall of 0.50, and a high recall of 0.75, averaged over the 30 Medlars queries. Both the retrospective and the predictive output are shown in Table 1. In the latter case, a vector provessing run was initially performed using a cosine similarity measure to compare documents and queries expanded by use of the maximum spanning tree. The top 20 documents retrieved by the cosine run were judged for relevance in each case. The term occurrence information obtained from these 20 documents was then used to compute the values of p_i and p_i' for all terms x_i, and a new ranking was obtained for all the documents using the probabilistic term independence, tree dependence, and generalized term dependence models.

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 nonrelevant items retrieved in the top 20 ranks, since these are utilized to estimate the parameters needed for the probabilistic formulas. This is done by using a rank freezing process which fixes the relevant items originally retrieved at their initial ranks, while discarding the nonrelevant items initially seen and replacing them by new items retrieved at lower ranks. [19]

It may be seen that for both the retrospective and the predictive experiments the probabilistic retrieval system performs better than the initial cosine run used in the rank freezing mode. Moreover, for the retrospective case where full relevance information is available and exact values can be

computed for the probabilistic parameters, the generalized term dependence theory is confirmed. That is, the tree dependence model provides a 13 percent improvement over the term independence model; additional small improvements are obtained when one, two, and finally all term triples defined by the spanning tree are taken into account. While the percentage improvement is small because the high level of performance of all the runs leaves little room for amelioration, a consistent improvement is nevertheless in evidence for the generalized model including term triples. When one triple is added to the term pairs, 18 out of 30 queries show improvement, 6 show deterioration, and the performance of 6 is not changed. When all triples are used, the performance improves for 24 out of 30 queries, with 3 additional queries remaining unchanged, and only 3 showing deterioration.

In the predictive case where little relevance information is available, the computed probability information for term triples obtained from the retrieved documents did not help in the subsequent retrieval of additional relevant items. The average precision values of 0.69 to 0.70 are very close to each other for the predictive case, but the best output was obtained for the standard tree dependence model using all term pairs without added triples.

The use of the maximum spanning tree to define all usable term pairs and triples and to expand the query terms may not be felicitous in practice. When the original query contains high-frequency, common terms, a great many related terms are added to the queries that might better be left out. Additional experiments are under way using structures other than the maximum spanning tree to define the term dependencies, and more discrimination query expansion methods.

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, to appear.

[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] K. Lam and C.T. Yu, A Clustered Search Algorithm Interpreting Arbitrary Term Dependencies, ACM Transactions on Data Base Systems, Vol. 7, No. 3, September 1982, p. 500-508.

[8] S.E. Robertson and K. Sparck Jones, Relevance Weighting of Search Terms, Journal of the Am. Soc. for Information Science, Vol. 27, No. 3, 1976, p. 129-146.

[9] C.T. Yu and G. Salton, Precision Weighting--An Effective Automatic Indexing Method, Journal of the ACM, Vol. 23, No. 1, 1976, p. 76-88.

[10] 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.

[11] 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.

[12] 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.

[13] C.J. van Rijsbergen, Information Retrieval, Butterworths, London, Second Edition, 1979.

[14] M.G. Kendall and A. Stuart, Advanced Theory of Statistics, Vol. 2, C. Griffin, London, Second Edition, 1967.

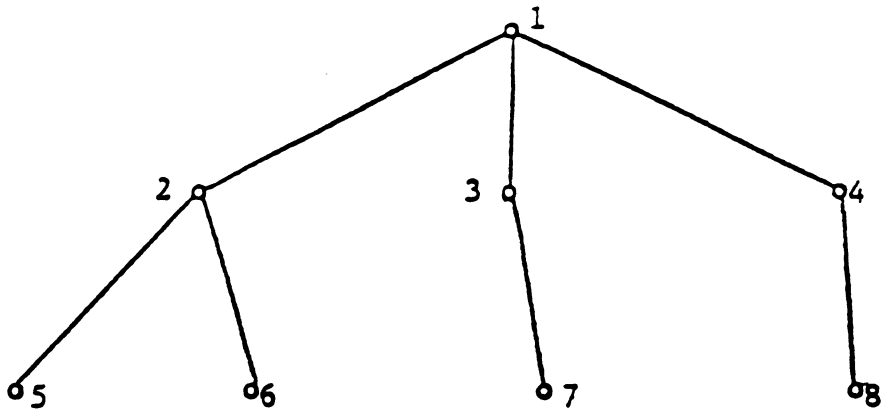
[15] Y.M.M. Bishop, S.E. Fienberg and P.W. Holland, Discrete Multivariate Analysis: Theory and Practice, MIT Press, Cambridge, Massachusetts, 1974.

[16] P.M. Lewis, Approximating Probability Distributions to Reduce Storage Requirements, Information and Control, Vol. 2, No. 3, 1959, p. 214-225.

[17] G. Salton, C. Buckley and C.T. Yu, An Evaluation of Term Dependence Models in Information Retrieval, in Research and Development in Information Retrieval, Lecture Notes in Computer Science, Vol. 146, G. Salton and H.J. Schneider, editors, Springer Verlag, Berlin 1983, p. 151-173.

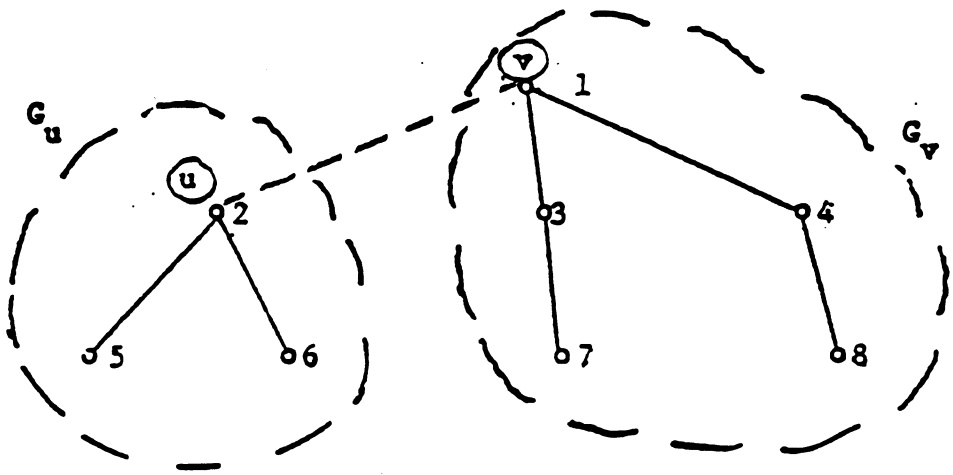
[18] C. Buckley, Probability Estimation, Technical Report, Department of Computer Science, Cornell University, Ithaca, New York, 1983.

[19] G. Salton, E.A. Fox, C. Buckley, and E. Voorhees, Boolean Query Formulation with Relevance Feedback, Technical Report 83-539, Department of Computer Science, Cornell University, Ithaca, New York, January 1983.



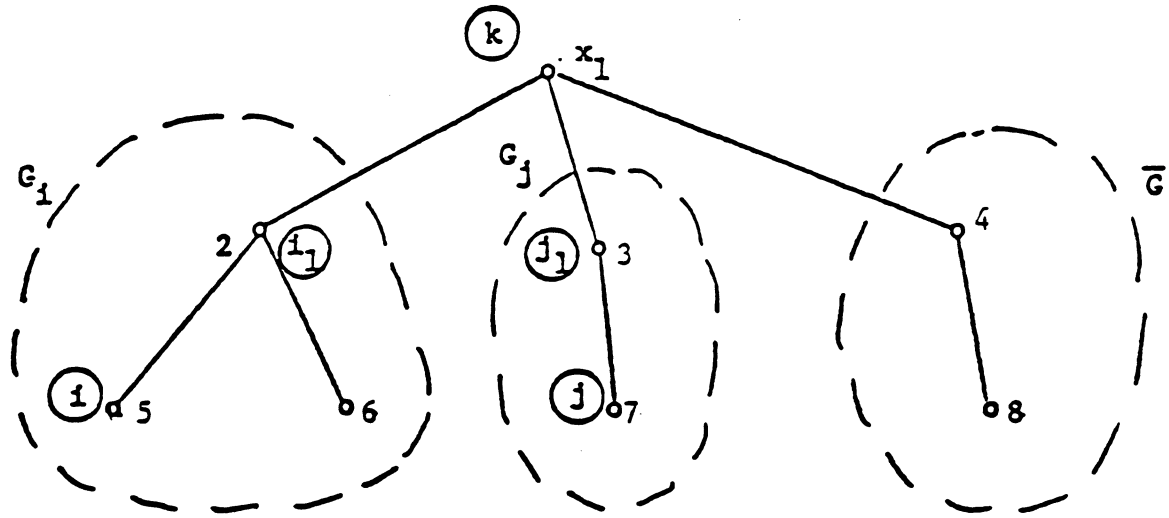
Typical Dependence Tree

Fig. 1



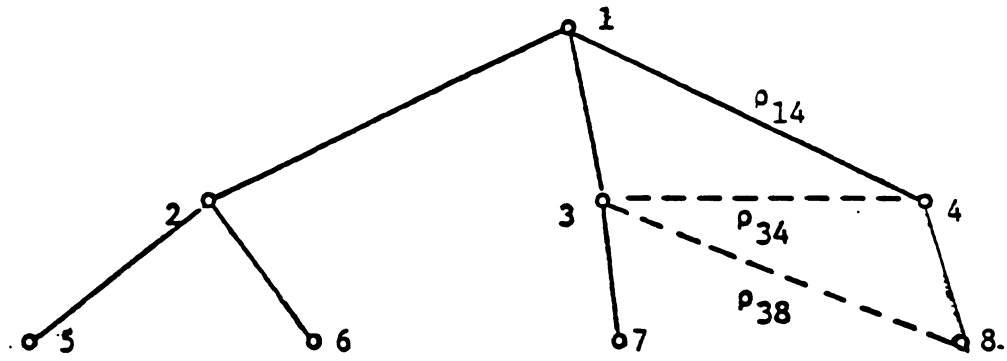
Tree Decomposition Using Edge (u,v)

Fig. 2



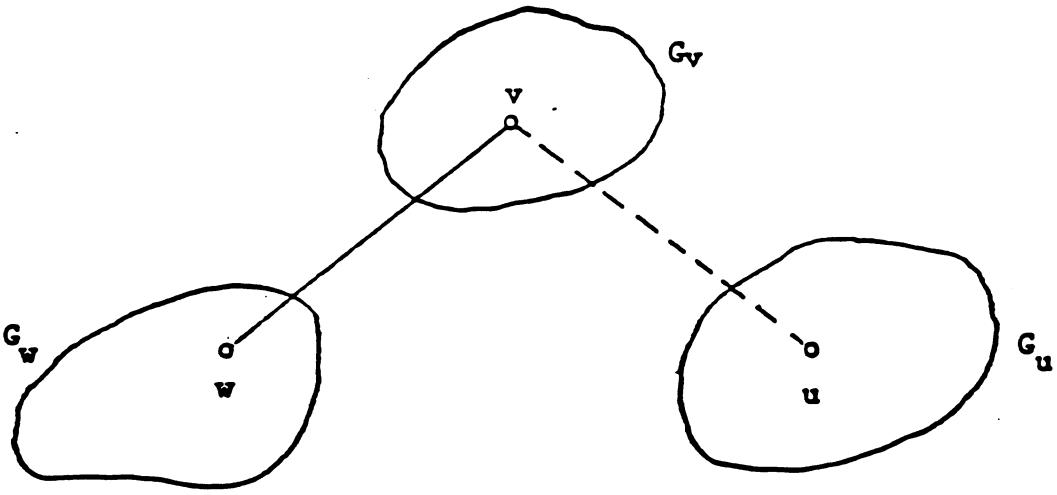
Decomposition into Three Subtrees Following Removal of Vertex k

Fig. 3



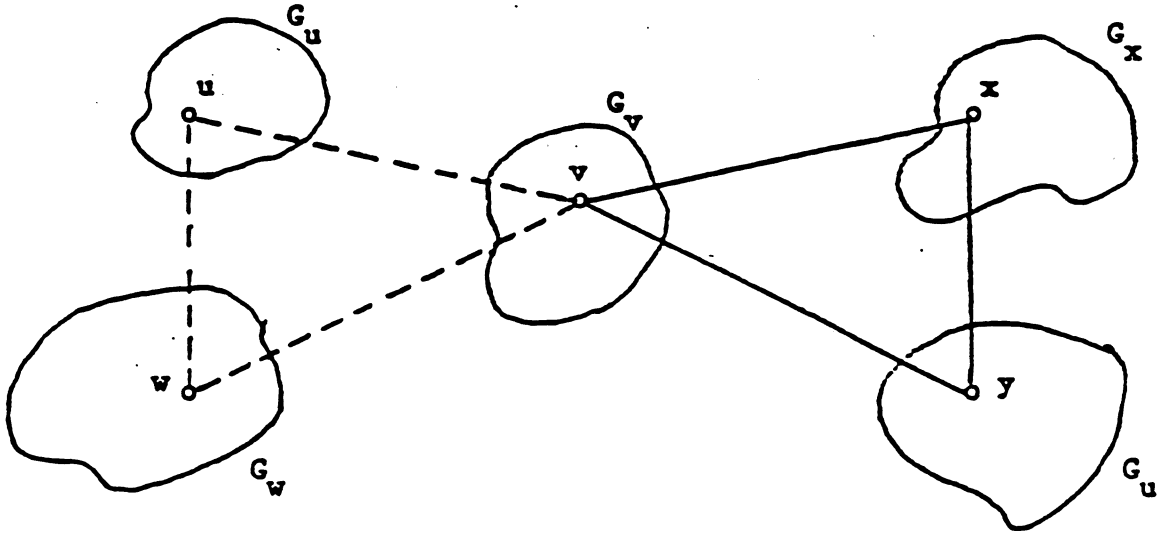
Composition of Correlation Coefficients
$\rho_{38} = \rho_{13} \cdot \rho_{14} \cdot \rho_{48}$

Fig. 4



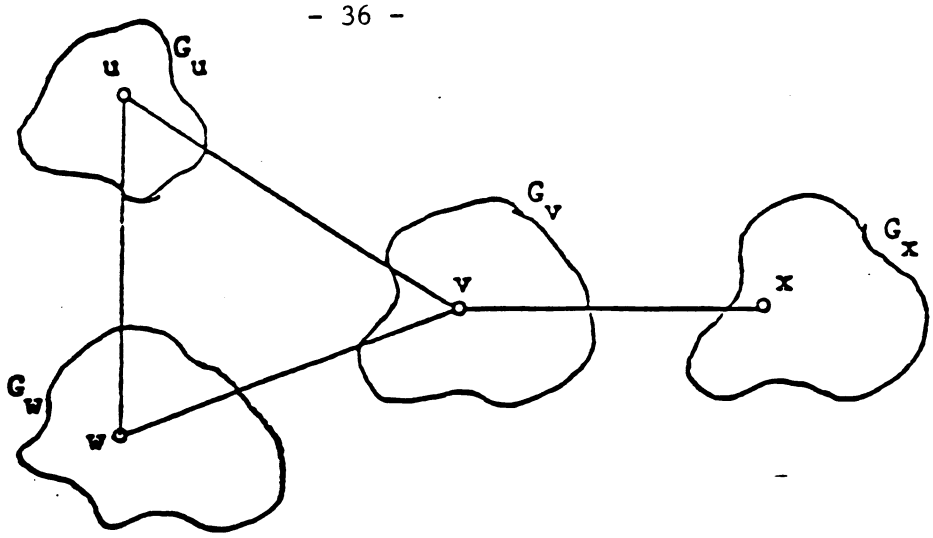
Decomposition Following Removal of Edge (u,v)

Fig. 5



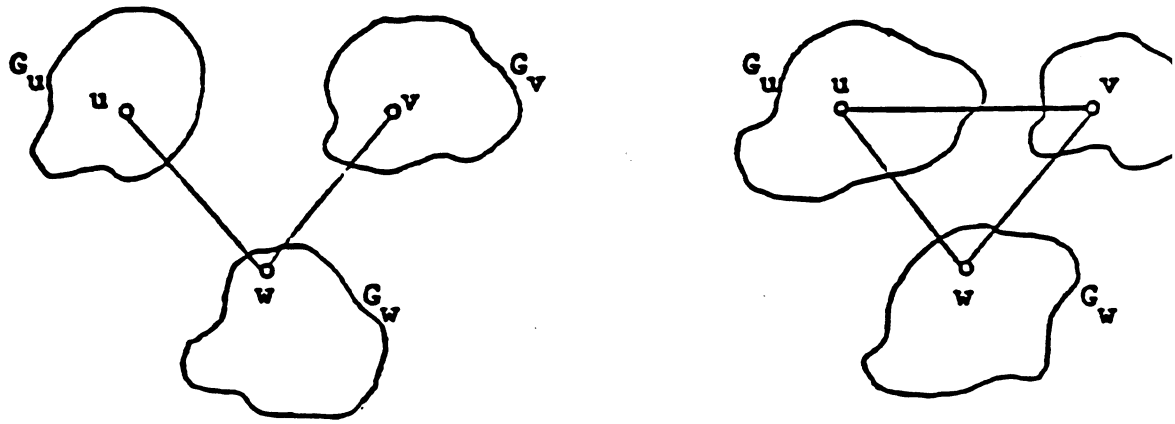
Decomposition Following Removal of Triangle (u,v,w)

Fig. 6



Comparison of Triangle and Edge Decomposition

Fig. 7

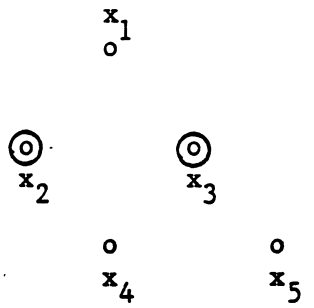


a) Tree G^i

b) Tree G^{i+1}

Addition of One Edge (u,v) Forming Triangle (u,v,w)

Fig. 8

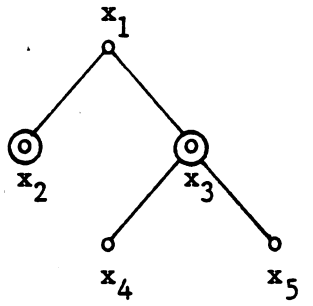


Original Query : $Q = (x_2, x_3)$

Expanded Query : $Q_{exp} = (x_1, x_2, x_3, x_4, x_5)$

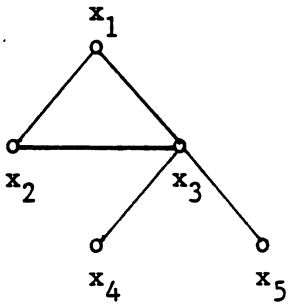
$P(\underline{x}) = P(x_1)P(x_2)P(x_3)P(x_4)P(x_5)$

a) Term Independence Model



$P(\underline{x}) = P(x_1) \cdot P(x_2|x_1) \cdot P(x_3|x_1) \cdot P(x_4|x_3) \cdot P(x_5|x_3)$

b) Basic Tree Dependence Model



$P(\underline{x}) = P(x_1, x_2, x_3)P(x_4|x_3)P(x_5|x_3)$

c) Generalized Tree Dependence With One Added Triple

Operations of Extended Tree Dependence System

Fig. 9

Medlars 1033 Documents, 30 Queries	Retrospective Experiment	Predictive Experiment
Initial Vector Processing Run, cosine similarity ranking, weighted terms; retrieved additional items after freezing relevant retrieved in top 20 ranks	.6739	.6739
Probabilistic Retrieval, natural language terms, query expansion through spanning tree, term independence	.8241	.8242
Probabilistic Retrieval, query expansion, all dependent pairs from spanning tree	.9314	.9066
Probabilistic Retrieval, query expansion, all pairs plus one triple	.9336 (0%)	.6979(-1%)
Probabilistic Retrieval, query expansion, all pairs plus two best triples	.9405(+1%)	.6938(-2%)
Probabilistic Retrieval, query expansion, all pairs plus all triples	.9538(+2%)	.6961(-1.5%)

Evaluation of Generalized Term Dependence Model
(average precision values at recall of 0.25, 0.50 and 0.75)

Table 1