

A Belief Network Model for IR

Berthier A. Ribeiro-Neto*
berthier@dcc.ufmg.br

Computer Science Department, Federal University of Minas Gerais
Brazil

Richard Muntz
muntz@cs.ucla.edu
Computer Science Department
University of California, Los Angeles

Abstract

We introduce a belief network model for IR which is derived from probabilistic considerations over a clearly defined sample space. This model subsumes the classical models in IR and generalizes the inference network model of Turtle and Croft. Further, we show how to extend the model with information from other queries (which we call contexts) to yield improved retrieval performance.

1 Introduction

The two most traditional schools of thought in probability are based on the *frequentist* view and the *epistemological* view [2, 5]. The frequentist view takes probability as a statistical notion related to the laws of chance. The epistemological view interprets probability as a degree of belief whose specification might be devoid of statistical experimentation. This second viewpoint is important because we frequently refer to probabilities in our daily lives (e.g., probability of raining, probability of a team winning a championship game, etc) without a clear definition of the statistical experiment which yielded those probabilities. Often, even the sample space is undefined or is difficult to specify.

The belief network model we introduce here uses degrees of belief devoid of statistical experimentation (as also done in [11, 13]). However, as in [12], we insist in explicitly defining a sample space which works as the basic foundation of our model. By doing so, we establish a firm ground on which we can build our degrees of belief more intuitively. Further, our model is simple to understand because it is derived from probabilistic considerations over a well defined sample space (as done in [13]).

Our belief model can be used as a general framework for classical IR (as also can the inference network model introduced in [11]). This is important because it allows combining features of distinct models into the same representational scheme. For instance, in [11] it is demonstrated that extending the inference network model with Boolean representations of the user queries might yield improved retrieval performance. Here we show that extending our belief network model with information from other queries might also yield improved retrieval performance.

The set-theoretic argumentation used to develop our model is quite distinct from the epistemological argumentation used in the development of the inference network model. As a consequence, the topology of the two networks differ and so does their ranking strategies. Our belief model is more general than the inference network model in the sense that it can represent any ranking probability function represented by that model (while the converse is not true).

Section 2 presents a very brief introduction to Bayesian belief networks. Section 3 defines the sample space used as the foundation of our network model. Section 4 discusses our belief network model for IR, its application to the classic Vector model, and its relationship to the inference network model. Experimental results are presented in section 5 followed by our conclusions.

*Supported in part by the Brazilian agency CNPq under grant number 300188/95-1.

2 Bayesian Belief Networks

Bayesian belief networks are DAGs in which the nodes represent random variables, the arcs portray relationships between the linked variables, and the strengths of these influences are expressed by conditional probabilities. The *parents* of a node (which is then considered as a *child* node) are those judged to be direct *causes* for it. This causal relationship is represented in the DAG by a link directed from each parent node to the child node. The *roots* of the network are the nodes without parents.

Let x_i be a node in a Bayesian network G and Γ_{x_i} be the set of parent nodes of x_i . The influence of Γ_{x_i} on x_i can be specified by any set of functions $F_i(x_i, \Gamma_{x_i})$ that satisfy

$$\sum_{\forall x_i} F_i(x_i, \Gamma_{x_i}) = 1 \quad (1)$$

$$0 \leq F_i(x_i, \Gamma_{x_i}) \leq 1 \quad (2)$$

This specification is complete and consistent because the product $\prod_{\forall i} F_i(x_i, \Gamma_{x_i})$ constitutes a joint probability distribution for the nodes in G . The reader is referred to [5] for further details.

3 Probability Space

We assume that all documents in the collection are indexed by *index terms* and that the universe of discourse U is the set of all index terms.

Definition 1 *Let t be the number of index terms in the system and k_i be an index term. $U = \{k_1, \dots, k_t\}$ is the set of all index terms and defines our sample space. Let $u \subset U$ be a subset of U .*

We view each index term as an *elementary concept* and U as a concept space (which we adopt as our sample space as also done in [12]). A concept u is a subset of U and might represent a document in the collection or a user query.

In a belief network, set relationships are specified using random variables as follows.

Definition 2 *To each index term k_i is associated a binary random variable which is also referred to as k_i . The random variable¹ k_i is 1 to indicate that the index k_i is a member of a concept/set u . Let $g_i(u)$ be the value of the variable k_i according to the concept/set u .*

The random variables (i.e., k_i) associated to the index terms are *binary* because this is the simplest possible representation for set membership and is a representation which suffices for our purposes (as in [11]). The set u defines a concept in U as the subset formed by the indexes k_i for which $g_i(u) = 1$. Thus, there are 2^t possible concepts definable in U . This association of concepts to subsets is useful because it allows expressing the logical notions of conjunction, disjunction, negation, and implication as the more familiar set-theoretic notions of intersection, union, complementation, and inclusion [13].

Documents and user queries can be defined as concepts in the sample space U as follows.

Definition 3 *A document d in the collection is represented as the concept $d = \{k_1, k_2, \dots, k_t\}$ where k_1 to k_t are binary random variables which define the index terms related to d . Analogously, a user query q is represented as the concept $q = \{k'_1, k'_2, \dots, k'_t\}$ where k'_1 to k'_t are binary random variables describing the index terms related to q .*

If an index term k_j is used to describe the document d then it must be $g_j(d) = 1$. Further, if the same index term k_j also describes a user query q then it must be $g_j(q) = 1$.

The task of the IR system is to determine which documents are most relevant to a given user query. Given that documents and queries are represented as concepts in the space U , we can view the IR system fundamentally as a concept matching system (as done in [4]).

A probability distribution P is defined over U as follows. Let c be a generic concept in the space U representing a document or user query. Then,

$$P(c) = \sum_u P(c|u) \times P(u) \quad (3)$$

$$P(u) = \left(\frac{1}{2}\right)^t \quad (4)$$

¹It should always be clear whether k_i refers to the index term or to the random variable.

Equation 3 defines $P(c)$ as the *degree of coverage* of the space U by c . Such coverage is computed by contrasting each of the concepts in U with c (through $P(c|u)$) and by summing up the individual contributions. This sum is weighted by the probability $P(u)$ with which u occurs in U . Since at the beginning the system has no knowledge of the probability with which a concept u occurs in the space U , we can assume that each u is equally likely (as in [3]) which results in equation 4.

It is important to emphasize at this point that each particular specification of the probability function $P(c|u)$ leads to a distinct strategy for computing $P(c)$ (i.e., to a distinct *ranking* strategy). For instance, one might define that $P(c|u)$ is 1 whenever $c \subset u$ and that it is 0 otherwise. Such specification of $P(c|u)$ leads to the classical Boolean method. Another possibility is to allow partial matches (i.e., partial coverages) between c and u and to weigh such matches by *tf-idf* factors² associated to the indexes common to c and u . Such specification of $P(c|u)$ leads to the classical vector method.

Given a user query q and a document d in the collection, the complete scheme for generating a ranking involves computing $P(q)$, $P(d)$, and then determining the degree of coverage of q by d . In the immediate following we introduce a Bayesian belief network as a basic tool for computing such ranking.

4 A Belief Network Model for IR

In this section, we propose a Bayesian belief network model for IR which adopts the concept space U as its sample space. We also show how to tune this belief network to subsume the vector model. Further, we demonstrate that any ranking generated by the inference network model can be reproduced by our belief network while the converse is not true.

4.1 Basic Network Model

A common assumption to all three classical IR models is as follows.

Assumption 1 *The index terms are considered to be independent among themselves.*

Such assumption is also adopted in our belief network because it simplifies the model. In fact, it implies that the set of index terms can be modelled by t independent network nodes where each node is associated to a *binary* random variable.

We model the user query q as a network node to which is associated a binary random variable (as in [5]) which is also referred to as q^3 . This variable is 1 to indicate that q completely covers the concept space U . Thus, when we assess $P(q)$ we compute the degree of coverage of the space U by q (as in [13]). This is equivalent to assessing the degree of belief associated to the following proposition: Is it true that q completely covers U ?

A document d in the collection is also specified as a concept in the space U . This concept contains all the index terms associated to the document d and no more (i.e., index terms not present in d are absent in this concept). The document is modelled as a network node to which is associated a binary random variable which is also referred to as d . This variable is 1 to indicate that d completely covers the concept space U . When we assess $P(d)$, we compute the degree of coverage of the space U by d . This is equivalent to assessing the degree of belief associated to the following proposition: Is it true that d completely covers U ?

According to the above formalism, the user query and the documents in the collection are modelled as subsets of index terms. Each of these subsets is interpreted as a *concept* embedded in the concept space U which works as a common *sample space*. Furthermore, user queries and documents are modelled identically. This is an important observation because it defines the topology of our belief network as discussed in the immediately following.

Figure 1 illustrates our belief network model. The index terms are modelled as independent binary random variables — the k_i variables. A query q is modelled as a network node associated to a binary random variable (also referred to as q). The node q is pointed to by the index term nodes which compose the query concept (as in [5]). The probability $P(q)$ is computed through equation 3 as $P(q) = \sum_u P(q|u) \times P(u)$. In principle, the computation of $P(q)$ must consider the influence of all the 2^t concepts u in U . Fortunately, practical IR systems require considering the influence of a few concepts u only. Documents are treated analogously to user queries

²Term_frequency \times inverse_document_frequency (see [9], for instance).

³It should always be clear whether q refers to the query or the random variable.

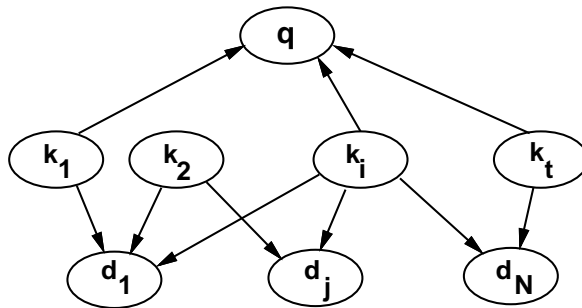


Figure 1: Basic Bayesian belief network model.

(i.e., both are concepts in the space U). Thus, a document node is pointed to by the index term nodes which compose the document. The probability $P(d)$ is computed through equation 3 as $P(d) = \sum_u P(d|u) \times P(u)$.

Given our belief network, we still need to specify how to rank the documents in the collection relative to a given query q . As suggested in [13], two possibilities arise immediately: (a) adopt $P(q|d)$ as the rank of the document d and (b) adopt $P(d|q)$ as the rank of the document d . In [13], $P(q|d)$ is interpreted as a measure of the *precision* of d with respect to q while $P(d|q)$ is interpreted as a measure of the *recall* index of d with respect to q . We find both interpretations hard to follow. Recall and precision are measures related to the set of relevant documents (provided by experts) and the set of retrieved documents respectively and it is unclear how these two measures relate to the probabilities $P(d|q)$ and $P(q|d)$ (whose computations consider neither the expert-provided relevant documents nor the retrieved ones).

We simply interpret $P(q|d)$ and $P(d|q)$ as concept matching relationships. As demonstrated in [13], $P(q|d)$ and $P(d|q)$ can be made equivalent by proper normalization (at least for the vector model). Thus, it is not critical whether we take one or the other as the document ranking. However, from the point of view of the application, we find it more natural to consider that the query is provided as evidence of the user’s information needs (as in [5]). Therefore,

Assumption 2 We adopt $P(d|q)$ as the ranking of the document d with respect to the query q .

The above decision is not restrictive. Using it we are able to show that our belief network model subsumes all three classical IR models — Boolean, vector, and probabilistic (see [7]). In the following section we show how to specify the parameters in our network to subsume the vector model (because our experiments adopt it).

There is a close resemblance between our network of figure 1 and the inference network proposed in [6, 11]. However, this resemblance hides important differences between the two models. First, our model is based on a set-theoretic view of the IR ranking problem and adopts a clearly defined sample space. The inference network model takes a purely epistemological view of the IR problem which is more difficult to grasp (because, for instance, the sample space is not clearly defined). Second, our network is more general than the inference network in [6, 11] because it is able to reproduce any ranking strategy generated by this last one while the converse is not true (see section 4.3).

To complete our belief network we need to specify the conditional probabilities $P(q|u)$ and $P(d|u)$. As shown in [11], distinct specifications of these probabilities allow modelling different ranking strategies (corresponding to different IR models). In [7] we discuss how to specify these probabilities to subsume the classical Boolean and probabilistic models. In the immediate following we discuss how to specify these probabilities to subsume the vector model.

4.2 Belief Network for the Vector Model

In the vector model, documents and user queries are represented as index term vectors. Let $\vec{d} = (k_1, \dots, k_t)$ be the vector representing the document d and $\vec{q} = (k'_1, \dots, k'_t)$ be the vector representing the user query q . To improve precision and recall levels, the vector model associates weights to document and query indexes. Let $w_{i,d}$ be the weight associated to the index k_i of document d and let $w_{j,q}$ be the weight associated to the index k'_j of query q . These weights are usually specified as a variation of *tf-idf* factors [9]. The similarity of a document d with respect to a user query q is measured as the cosine of the angle between the weighted query

and document vectors. This *cosine similarity ranking formula* can be written as

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} \bullet \vec{q}}{|\vec{d}| \times |\vec{q}|} \quad (5)$$

$$= \frac{\sum_{i=1}^t w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,d}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (6)$$

In our network of figure 1, the similarity (i.e., rank) of a document d with respect to a user query q is computed through the coverage relationship $P(d|q)$. Applying Bayes theorem, we can write $P(d|q) = \frac{P(d \wedge q)}{P(q)}$. Since $P(q)$ is a constant for all documents in the collection, we can write $P(d|q) \propto P(d \wedge q)$ i.e., the rank assigned to a document d is directly proportional to $P(d \wedge q)$. This last probability is computed through application of equation 3 which yields $P(d|q) \propto \sum_{\forall u} P(d \wedge q|u) \times P(u)$. In the belief network of figure 1, instantiation of the index term variables logically separates the nodes q and d making them mutually independent. Therefore,

$$P(d|q) \propto \sum_{\forall u} P(d|u) \times P(q|u) \times P(u) \quad (7)$$

$P(d|q)$ can be made equivalent to $\text{sim}(\vec{d}, \vec{q})$ through proper specification of the probabilities $P(d|u)$ and $P(q|u)$ as follows.

Lemma 1 *Let*

$$P(q|u) = \begin{cases} 1 & \text{if } \forall k_i, g_i(q) = g_i(u) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$P(\vec{q}|u) = 1 - P(q|u) \quad (9)$$

Further, define

$$P(d|u) = \frac{\sum_{i=1}^t w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,d}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (10)$$

$$P(\vec{d}|u) = 1 - P(d|u) \quad (11)$$

where $w_{i,d}$ and $w_{i,q}$ are the weights used in the vector model. This specification is valid and consistent because $P(d|u)$ measures the cosine of the angle between two vectors which is a number between 0 and 1. Then, the ordering of relevant documents (i.e., ranking) defined by $P(d|q)$ coincides with the ordering defined by $\text{sim}(\vec{d}, \vec{q})$.

Proof $P(q|u)$ is 1 if $u = q$ and is 0 otherwise. Refer to the concept $u = q$ as u_q . Then, equation 7 reduces to $P(d|q) \propto P(d|u_q) \times P(u_q)$. The probability $P(d|u_q)$ is computed using the equation 10 while the probability $P(u_q)$ is a constant. Therefore, $P(d|q) = K \times \text{sim}(\vec{d}, \vec{q})$ where K is a constant. \square .

In section 4.3 we discuss the modeling of the vector model according to the inference network model [11].

4.3 Relationship with the Inference Network Model

The inference network model [11] is the belief network model first proposed for IR. It introduced nice theoretical results and has yielded good retrieval performance with many collections [6, 11]. In this section we discuss its relationship with our network model.

An inference network models index terms, documents, and user queries as nodes associated to binary random variables. According to the authors, *an index term variable corresponds to the event that an index term has been assigned to a document*. Further, *a document variable corresponds to the event that a document has been observed* [11]. Observation of a document is the *reason* (i.e., the cause) for observing any of its assigned index terms. This causal relationship is modelled by directing the edges in the network from document nodes towards index term nodes. The query portion of the inference network is analogous to the query portion of our network model (both follow the query modeling approach suggested in [5]). Figure 2 illustrates the basic inference network model for IR [6]. Notice that the directionality of the dependency links between a document

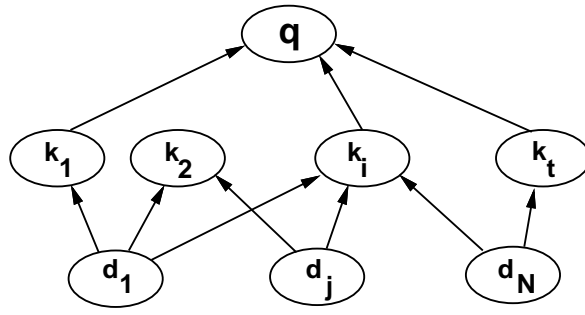


Figure 2: Basic inference network model.

node and its index term nodes is *opposite* to that in our network model. The key implication is that the rank of a document d is computed as $P(q|d)$ (in fact, according to the model's formulation, it is not clear what $P(d|q)$ means). To complete the inference network model one must specify the conditional probabilities $P(k_i|d)$ and $P(q|u)$ where $u = \{k_1, k_2, \dots, k_t\}$. The rank $P(q|d)$ of a document d is obtained by considering that the document d is being observed (i.e., the random variable d is set to 1) while all the other documents in the collection are not being looked at. Thus, given d as the sole piece of evidence, the network in figure 2 allows us to write $P(q|d) = \sum_{\forall u} P(q|d, u)P(u|d)$. Since instantiation of the index term variables isolates q from d , we have

$$P(q|d) = \sum_{\forall u} P(q|u)P(u|d) \quad (12)$$

where $P(q|u)$ and $P(u|d)$ must be true probability functions for which $P(q|u) + P(\bar{q}|u) = 1$ and $P(u|d) + P(\bar{u}|d) = 1$.

Lemma 2 *Any ordering of documents specified by the inference network model through equation 12 can also be reproduced in the context of our network model by properly specifying the conditional probabilities in equation 7.*

Proof To distinguish the probabilities in the two models, we briefly adopt the notation P_f to refer to the probabilities in the inference network model (i.e., $P_f(q|d)$, $P_f(q|u)$, $P_f(u|d)$, and $P_f(k_i|d)$). From figure 2 we observe that instantiation of a document node d in the inference network model isolates the index term nodes making them mutually independent. Thus, $P_f(u|d)$ can be always computed as $P_f(u|d) = \prod_{\forall k_i|g_i(d)=1} P_f(k_i|d)$. Consider now our network ranking as specified by equation 7. Define $P(q|u) = P_f(q|u)$. Further, restrict the form of $P(d|u)$ to $P(d|u) = \prod_{\forall k_i|g_i(d)=1} P(d|k_i)$. Let $P(d|k_i) = P_f(k_i|d)$ i.e., the degree to which the concept d is covered by the index k_i is evaluated as the probability of observing the index k_i given that the document d has been observed. Then, $P(d|u) = P_f(u|d)$ and the rankings $P_f(q|d)$ and $P(d|q)$ differ only by the constant factor $P(u)$. \square

The above lemma is a consequence of the particular topology of the inference network model in which instantiation of a document node d logically separates its index term nodes making them mutually independent. However, there are document rankings generated by our network model which find no counterpart in the network inference model.

Lemma 3 *There exist orderings of documents generated by our network model through equation 7 which cannot be reproduced in the context of the network inference model. The ordering generated by the vector space model according to equation 6 is one such example.*

Proof Our network subsumes the vector model through equations 8 and 10. According to these two equations, we can write

$$P(d|q) = K \times \frac{\sum_{i=1}^t w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,d}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (13)$$

Consider now that the inference network model also subsumes the vector model. Then, it must be

$$P_f(q|d) = K_1 \times \frac{\sum_{i=1}^t w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,d}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (14)$$

The factor $\sqrt{\sum_{i=1}^t w_{i,d}^2}$ in the denominator depends on all the index terms in the document d . However, the topology of the inference network model asserts that a document node d can only affect the belief in the query node q through the index term nodes common to q and d . In fact, corollary 4 on page 120 of reference [5] clearly states: *Given a DAG D and a probability distribution P , a necessary and sufficient condition for D to be a Bayesian network of P is that each variable X be conditionally independent of all its non-descendants, given its parents Π_X , and that no proper subset of Π_X satisfy this condition.* Thus, strict adherence to the Bayesian formalism requires that $P_f(q|d)$ be independent of the index terms which do not occur in the query q . As a result, the inference network model is unable to strictly reproduce the ranking generated by the vector model. \square

One should not infer that the above restriction prevents the inference network model from yielding good retrieval performance. In fact, as already demonstrated in [1, 6, 11], the inference network model is able to accomplish top level retrieval performance with many different collections. Further, it is possible to specify the conditional probabilities $P_f(u|d)$ and $P_f(q|u)$ using the $w_{i,q}$ and $w_{i,d}$ weights to closely approximate the behavior of the vector model. The point that we make here is that, from a theoretical point of view, our belief model for IR is more general than the inference network model.

5 Empirical Results

In this section we describe some experiments we carried out with our network model. The experiments use a Cystic Fibrosis database [10]. The choice of a particular database is *not* critical for our purposes. The main contribution of this work is *not* an empirical comparative study of different retrieval strategies. Instead, our main intention is to propose a general belief network model for IR. The examples we illustrate here aim at highlighting the flexibility and adequacy of this model and at exposing the possibilities it opens for further exploration.

5.1 The Cystic Fibrosis Database

The Cystic Fibrosis (CF) database [10] is composed of a collection of 1,239 documents describing clinical and research studies associated with CF. The documents were published from 1974 through 1979 and indexed with the term *Cystic Fibrosis* in the National Library of Medicine's *MEDLINE* database. The CF database also includes a collection of 100 *information requests* (i.e., queries) with exhaustive relevance evaluations. A typical example of a query is

What is the association between liver disease (cirrhosis) and vitamin A metabolism in CF?

Queries are stated in natural language and the CF database does *not* include vectors of index terms for them. Thus, we had to generate the query index term vectors ourselves. For instance, the query vector that we generated for the above query (see [7]) is

(liver-cirrhosis, liver-cirrhosis-alcoholic, liver-cirrhosis-biliary, vitamin-a, vitamin-a-deficiency, energy-metabolism)

which can then be used for experimentation.

For each query, the CF database includes an expert-provided set of most relevant documents. Further, a set of four relevance scores for each relevant document is also provided. Three of these scores were issued by subject experts while the fourth one was issued by a medical bibliographer. Relevance scores were based on examination of the full text of the document and range from 0 to 2 with the following meaning: 2 indicates that the document is *highly relevant*, 1 indicates that the document is *marginally relevant*, and 0 indicates a *non-relevant* document. Thus, the total relevance score for each document ranges from 0 to 8. A document is considered relevant (and retrieved) if its total relevance score for a query exceeds a pre-established *relevance threshold* (RT). For the experiments described below, we adopted $RT = 1$.

Relevance assessments were influenced by tables and figures in the document. In several cases, these assessments reflected information present in a table or figure but *not* described in the text of the document. Consequently, for several queries we were unable to generate a query vector that retrieves a minimal portion of the documents considered relevant by the experts⁴. We decided to exclude these queries from our test collection. As a result, we were left with a final set of 64 example queries.

⁴This suggests the need to automate the examination of tables, figures, and images if performance is to be further improved.

5.2 Experiments

Using the 64 example queries, we first ran experiments for three distinct retrieval strategies: (1) the classic vector model (called Vector), (2) the vector model using queries expanded with *related indexes* (this model is called Vector_exp), and (3) our belief network model using queries expanded with related indexes which are also weighted by correlation factors (this model is called Network).

A *related index* here is any index term which has a relevance relationship with the index terms in the query. In our work, this relevance relationship is established automatically as follows. Let q_{other} be any query other than the current query q and let d_{other}^r be a document relevant to q_{other} (according to the experts). Index terms which co-occur in d_{other}^r are said to have a relevance relationship. Further, the number of relevant documents in which two index terms co-occur is used here to quantify the strength of this relevance relationship (which we call *correlation factor*).

The Vector_exp strategy takes the original query, expands it with index terms related to the query indexes, and applies the cosine similarity formula to the expanded query. Query expansion is limited to a maximum of 100 related index terms. The Network strategy expands the query with the same set of related indexes used by the Vector-exp strategy. However, the index terms used in the expansion are weighted with correlation factors. This weighting scheme is based on an adaptation of the Rochio formula for relevance feedback [8] and works as follows.

Let \mathcal{R} be the set of index terms related to a given query q and let \vec{q} be the query vector for q . The expanded query vector \vec{q}_{exp} used by the Network strategy is an adaptation of the Rochio formula for relevance feedback and is given by

$$\vec{q}_{exp} = \alpha \vec{q} + \beta \sum_{k_i, k_j \in \mathcal{R}} l_{k_i, k_j} \times \vec{k}_j \quad (15)$$

where α and β are constants, l_{k_i, k_j} is a correlation factor between the indexes k_i and k_j , and \vec{k}_j is a singleton vector. The expanded query \vec{q}_{exp} incorporates index terms correlated to the query indexes, weighting them with the respective correlation factors. The correlation factor l_{k_i, k_j} quantifies the number of expert-provided relevant documents (relative to queries other than q) which include a query index k_i and the index k_j . Therefore, equation (15) is equivalent to the sum of the expert-provided relevant document vectors (relative to queries other than q) which include at least one of the index terms in q . The main difference to the Rochio formula for relevance feedback is that the set of relevant documents is relative to *other* queries and not to the query being processed. The expanded query \vec{q}_{exp} in equation (15) is then used to retrieve and rank the documents using the cosine similarity formula. Notice that this strategy could be implemented outside our network model. However, as we later demonstrate, the presence of an underlying belief network is important because it allows the inclusion of new pieces of evidence which might lead to improved retrieval performance.

Figure 3 displays recall and precision figures for the three retrieval strategies we are considering. At low recall values, the Vector_exp strategy performs poorer than the Vector model because of the excessive interference of the related index terms. At higher recall values, the Vector_exp strategy is able to take advantage of the expanded query to outperform the Vector model. The Network strategy is able to minimize the influence of related index terms at low recall values by setting α to a high mark (for instance, $\alpha = 200$). Further, the strategy is able to take advantage of the expanded query to improve precision at high recall levels. The final result is improved precision at all recall levels.

We observe, however, that the nice performance of the Network strategy is due exclusively to the utilization of the correlation factors l_{k_i, k_j} as weights for the related indexes. The presence of an underlying belief network did not affect the performance. The immediate question is whether the belief network can be used to further improve the retrieval performance. The answer is affirmative. As demonstrated in [11], the belief network allows including new pieces of evidence into the model which might yield improved retrieval performance.

The CF database provides an important additional piece of evidence which has not been included in the network model yet — the relevance scores provided by the panel of experts. As discussed in section 5.1, relevance scores are provided for each document relevant to a given example query q and vary from 1 to 8. The query q defines the *context* used by the experts for deciding the relevance score for each document. Our approach is to include these expert-provided relevance scores into the network.

Let q be the current query and let $\{c_0, c_1, \dots, c_p\}$ be a set of binary random variables associated to queries (which we call *contexts*) related to q . We say that two queries are related if they have at least one index term in common (which means that the internal product of their query vectors is greater than zero). Figure 4

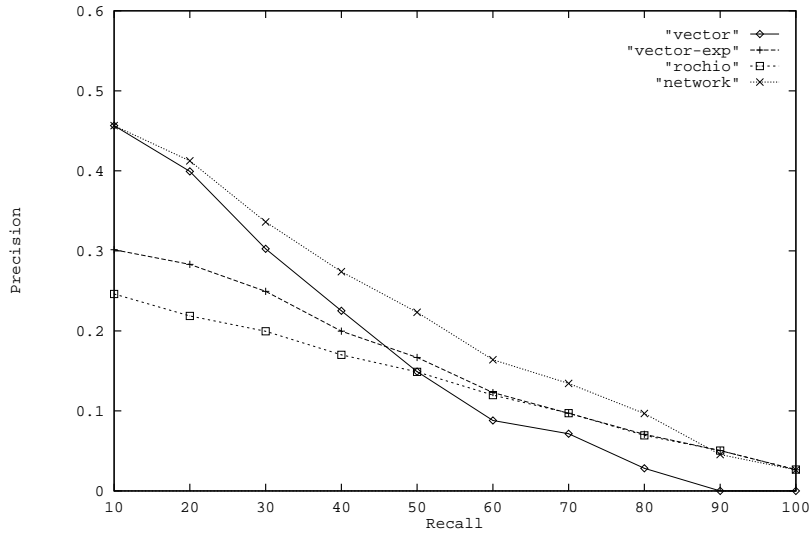


Figure 3: Comparison of 3 strategies: Vector, Vector_exp, and Network.

illustrates a belief network extended with context nodes. The square box delimits the belief network used in our previous experiment (with query and document nodes relabelled). The nodes c_0, c_1, \dots, c_p model the contexts (i.e., queries) related to q . The nodes dc_1, dc_2, \dots, dc_N model the expert-provided relevant documents for the various contexts. As before, all variables are binary. The query node q provides two pieces of evidence: (1) evidence derived from the expert-provided relevance scores through node q_c and (2) evidence derived from related index terms through node q_v . A document node d_j combines the rank generated by the evidence q_c with the rank generated by the evidence q_v . The approach is entirely modular because the new evidence introduced by q_c does not disturb the previous network. The context node c_0 is not linked to any document node because it models the context of the query q itself (and thus, the expert-provided answers for it cannot be used).

Let C be the set of all context variables. As done for U , we interpret C as a concept space. Each c_i is interpreted as a *basic concept* and $c = \{c_0, c_1, \dots, c_p\}$ is viewed as a composed concept in the space C . To simplify the computation, only basic concepts are considered in our experiments (i.e., we do not look at the joint impact of two or more contexts in the evaluation of query q). Let c_{+i} refer to the basic concept in C for which $c_i = 1$ and $c_j = 0$ for all $j \neq i$. Given these considerations, the ranking of a document d_j in the network of figure 4 is computed as follows.

$$\begin{aligned}
P(d_j|q) &= \frac{1}{P(q)} \times \sum_{u, c_{+i}} P(d_j|u, c_{+i}) \times P(q|u, c_{+i}) \times \\
&\quad P(u) \times P(c_{+i}) \\
&= \frac{1}{P(q)} \times \sum_{u, c_{+i}} [1 - (1 - P(dk_j|u)) \times \\
&\quad (1 - P(dc_j|c_{+i}))] \times P(q|u, c_{+i}) \times \\
&\quad P(u) \times P(c_{+i}) \\
&= \frac{1}{P(q)} \times \sum_{u, c_{+i}} [1 - (1 - P(dk_j|u)) \times \\
&\quad (1 - P(dc_j|c_{+i}))] \times P(q_v|u) \times P(q_c|c_{+i}) \times \\
&\quad P(u) \times P(c_{+i})
\end{aligned} \tag{16}$$

where $P(dk_j|u)$ is given by equation (10) and $P(q_v|u)$ is given by equation (8). The probabilities $P(dc_j|c_{+i})$ and $P(q_c|c_{+i})$ are computed as follows. Let $rs(c_i, dc_j)$ be the expert-provided relevance score for the document

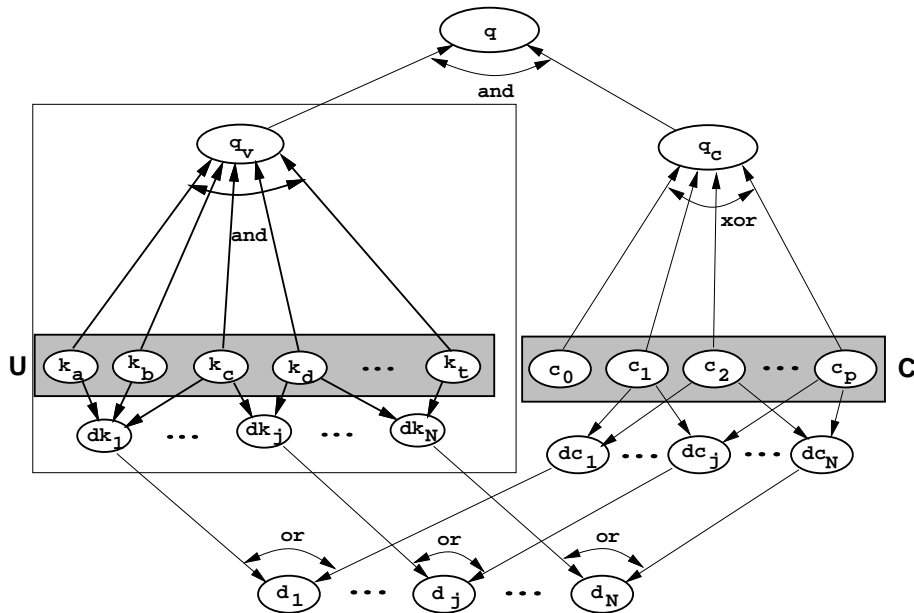


Figure 4: Belief network model expanded to include context nodes.

d_j in the context c_i . Then,

$$P(dc_j|c_{+i}) = \frac{rs(c_i, dc_j)}{8} \quad (17)$$

because the maximum relevance score for any document is 8. Further, let \vec{q}_c be the query vector corresponding to q_c (recall the vector model formulation) and let \vec{q}_i be the query vector for the context c_i . Due to our above definition of relationship between two queries, we write

$$P(q_c|c_{+i}) = \frac{\vec{q}_c \cdot \vec{q}_i}{|\vec{q}_c| \times |\vec{q}_i|} \quad (18)$$

Figure 5 compares the retrieval performance of three ranking strategies: (1) the Vector strategy, (2) the Network strategy in figure 3, and (3) the Network-contexts strategy of figure 4. The results show that our network model expanded with context nodes outperforms the two other strategies at all recall levels. The table in figure 6 compares the numeric values (extracted from figure 5) of precision for the Network and Network-contexts strategies. The improvements observed are due to the inclusion of the expert-provided relevance scores (and their respective contexts) as new evidence in the network.

The expert-provided relevance scores quantify a semantic relationship between a context and its set of relevant documents. This relationship is quite different from the relationships among correlated index terms and cannot be easily incorporated outside the environment of a belief network. One may think of adaptations to equation 15 to include relevance scores but it is not clear how this could be accomplished and whether such adaptations would yield positive results. For instance, we considered changing the calculation of the correlation factors to include relevance scores. Instead of increasing the correlation factor by 1 for each correlation between two index terms, we increased it by the expert-provided relevance score associated to the document in which the correlation was observed. However, no improvements in retrieval performance were observed. In fact, the performance was slightly worst.

6 Conclusions

We have proposed a Bayesian belief network model for IR whose main advantages are as follows. First, our model is founded on a clearly defined sample space which should make it intuitive. Second, our model is derived from probabilistic considerations over this sample space which should simplify its understanding. Third, our model can be viewed as an alternative to the inference network model proposed in [11]. It is not clear whether

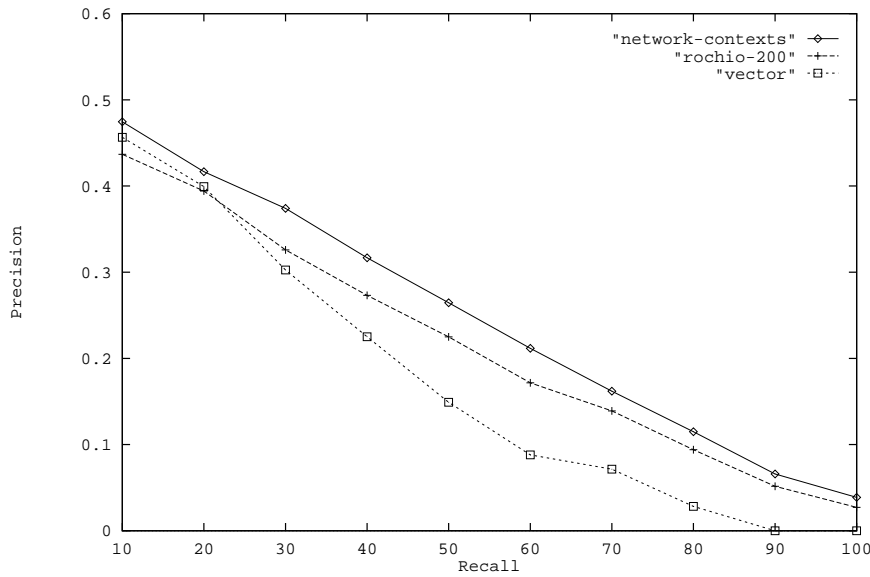


Figure 5: Comparison of 3 strategies: Network-contexts, Network, and Vector.

P R E C I S I O N

	Network	Network-c	Improvement	
R	10	0.4368	0.4745	+08.6%
E	20	0.3942	0.4166	+05.7%
C	30	0.3259	0.3740	+14.8%
A	40	0.2732	0.3167	+15.9%
L	50	0.2250	0.2646	+17.6%
L	60	0.1716	0.2117	+23.4%
	70	0.1390	0.1620	+16.5%
	80	0.0941	0.1149	+09.4%
	90	0.0516	0.0659	+27.7%
	100	0.0272	0.0387	+42.2%

Figure 6: Numeric figures for 2 strategies: Network and Network-contexts.

the two models differ for practical purposes. However, from a theoretical point of view, our model is more general because it is able to reproduce any ranking strategy generated by the inference network model while the converse is not true.

We also made some experimentation with our network model using a Cystic Fibrosis (CF) database. The results confirmed that extending the network to include new pieces of evidence might yield improved retrieval performance (a fact first determined in [11]). In the case of the CF database, the improvements were obtained by extending the network with information from other query contexts.

Our plans for the near future include experimentation with the TREC data and an empirically based comparison with the inference network model.

References

- [1] J. Broglio, J.P. Callan, W.B. Croft, and D.W. Nachbar. Document retrieval and routing using the inquiry system. In D.K. Harman, editor, *Overview of the Third Retrieval Conference (TREC-3)*, pages 29–38. NIST Special Publication 500-225, 1995.
- [2] T.L. Fine. *Theories of Probability: An Examination of Foundations*. Academic Press, 1973. New York.

- [3] E.T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4:227–241, 1968.
- [4] K.L. Kwok. Experiments with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Transactions on Information Systems*, 8(4):363–386, October 1990.
- [5] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.
- [6] T.B. Rajashekar and W.B. Croft. Combining automatic and manual index representations. *JASIS*, 46(4):272–283, May 1995.
- [7] Berthier A.N. Ribeiro. *Approximate Answers in Intelligent Systems*. PhD thesis, University of California, Los Angeles, 1995.
- [8] J.J. Rochio. *Relevance Feedback in Information Retrieval*. Prentice Hall Inc., 1971. In: The SMART Retrieval System: Experiments in Automatic Document Processing, chapter 14.
- [9] G. Salton and C. Buckley. Term-weighting approaches in automatic retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
- [10] W.M. Shaw, J.B. Wood, R.E. Wood, and H.R. Tibbo. The cystic fibrosis database: Content and research opportunities. *LISR*, 13:347–366, 1991.
- [11] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, July 1991.
- [12] S.K.M. Wong and Y.Y. Yao. A probabilistic inference model for information retrieval. *Information Systems*, 16:301–321, 1991.
- [13] S.K.M. Wong and Y.Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):39–68, 1995.