A Layered Bayesian Network Model for Document Retrieval

Luis M. de Campos[1], Juan M. Fernández-Luna[2], and Juan F. Huete[1]

[1] Dpto. de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática, Universidad de Granada, 18071 – Granada, Spain
{lci,jhg}@decsai.ugr.es
[2] Departamento de Informática, Escuela Politécnica Superior Universidad de Jaén, 23071 – Jaén, Spain
jmfluna@ujaen.es

Abstract. We propose a probabilistic document retrieval model based on Bayesian networks. The network is used to compute the posterior probabilities of relevance of the documents in the collection given a query. These computations can be carried out efficiently, because of the specific network topology and conditional probability tables being considered, which allow the use of a fast and exact probabilities propagation algorithm. In the initial model, only direct relationships between the terms in the glossary and the documents that contain them are considered, giving rise to a Bayesian network with two layers. Next, we consider an extended model that also includes direct relationships between documents, using a network topology with three layers. We also report the results of a set of experiments with the two models, using several standard document collections.

1 Introduction

Information Retrieval (IR) is the field that deals with the automated storage and retrieval of information. In our case, the pieces of information considered will always be texts (the textual representations of any objects), referred to as *documents*. An *IR model* is a specification about how to represent documents and queries (formal statements of user's information needs), and how to compare them, whereas an *IR system* is the computer software that implements a model. Probabilistic IR models [4,9,12] use probability theory to deal with the intrinsic uncertainty with which IR is pervaded [3]. Also founded on probabilistic methods, *Bayesian networks* [5] have been proven to be a good model to manage uncertainty, even in the IR environment, where they have already been successfully applied as an extension of probabilistic IR models [13,14,7].

In this paper we introduce new IR models based on Bayesian networks. The retrieval engine of our first model is composed of a Bayesian network with two layers of nodes, representing the documents and the terms in the document collection and the relationships among each other. The second model extends

F. Crestani, M. Girolami, and C.J. van Rijsbergen (Eds.): ECIR 2002, LNCS 2291, pp. 169–182, 2002.

the first one by including a third layer, also composed by documents, with the aim of capturing some relationships between documents.

The rest of the paper is organized as follows: we begin in Section 2 with the preliminaries. In Section 3 we introduce the basic model, the assumptions that determine the network topology being considered, the details about probability distributions stored in the network, and the way in which we can efficiently use the network model for retrieval, by performing probabilistic inference. In Section 4 we study the extended model. In Section 5 we discuss the similarities and differences between our models and other retrieval models also based on Bayesian networks. Section 6 shows the experimental results obtained with the two models, using several standard document collections. Finally, Section 7 contains the concluding remarks and some proposals for future research.

2 Preliminaries

Many IR models usually represent documents and queries by means of vectors of *terms* or *keywords*, which try to characterize their information content. Because these terms are not equally important, they are usually weighted to highlight their importance in the documents they belong to, as well as in the whole collection. The most common weighting schemes are the *term frequency*, tf_{ij}, i.e, the number of times that the i^{th} term appears in the j^{th} document, and the *inverse document frequency*, idf_i, of the i^{th} term in the collection, $idf_i = \lg(N/n_i) + 1$, where N is the number of documents in the collection, and n_i is the number of documents that contain the i^{th} term. The combination of both weights, $\mathrm{tf}_{ij} \cdot \mathrm{idf}_i$, is also a common weighting scheme.

The evaluation of the retrieval performance of an IR system is usually carried out by means of two complementary measures: *recall* and *precision* [10]. The first one measures the ability of the IR system to present all the relevant documents (number of relevant documents retrieved / number of relevant documents). The second one, precision, measures its ability to present only the relevant documents (number of relevant documents retrieved / number of documents retrieved). By computing the precision for a number of fixed points of recall (the average precision values for all the queries being processed), the recall-precision curves are obtained. If a single value of performance is desired, the average precision, for all the points of recall considered, may be used.

A Bayesian network $G = (V, E)$ is a *Directed Acyclic Graph* (DAG), where the nodes in V represent the variables from the problem we want to solve, and the arcs in E represent the dependence relationships among the variables. In that kind of graph, the knowledge is represented in two ways [5]: *(a)* Qualitatively, showing the (in)dependencies between the variables, and *(b)* Quantitatively, by means of conditional probability distributions which shape the relationships. For each variable $X_i \in V$, we have a family of conditional probability distributions $P(X_i|Pa(x_i))$, where $Pa(X_i)$ represents the parent set of the variable X_i in G. From these conditional distributions we can recover the joint distribution over V:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \quad (1)$$

This expression represents a decomposition of the joint distribution (which gives rise to important savings in storage requirements). The dependence/independence relationships which make possible this decomposition are graphically encoded (through the d-separation criterion [5]) by means of the presence or absence of direct connections between pairs of variables. Bayesian networks can perform efficiently reasoning tasks: the independencies represented in the graph reduce changes in the state of knowledge to local computations. There are several algorithms [5] that exploit this property to perform probabilistic inference (propagation), i.e., to compute the posterior probability for any variable given some evidence about the values of other variables in the graph.

3 A Bayesian Network Model with Two Layers

In IR problems we can distinguish between two different sets of variables (nodes in the graph): The set $\mathcal{T}$ of the M terms, T_i, in the glossary from a given collection, and the set $\mathcal{D}$ of the N documents, D_j, that compose the collection. Each term, T_i, consists on a binary random variable taking values in the set $\{\bar{t}_i, t_i\}$, where $\bar{t}_i$ stands for 'the term T_i is not relevant', and t_i represents 'the term T_i is relevant'[1]. Similarly, a variable referring to a document D_j has its domain in the set $\{\bar{d}_j, d_j\}$, where in this case, $\bar{d}_j$ and d_j respectively mean 'the document D_j is not relevant for a given query', and 'the document D_j is relevant for a given query'[2].

Focusing on the structure of the network, the following guidelines have been considered to determine the topology of the graph [2]:

- For each term that has been used to index a document, there is a link between the node representing that keyword and each node associated with a document it belongs to.
- The relationships between documents only occur through the terms included in these documents.
- Documents are conditionally independent given the terms that they contain. Thus, if we know the relevance (or irrelevance) values for all the terms indexing document D_i then our belief about the relevance of D_i is not affected by knowing that another document D_j is relevant or irrelevant.

These assumptions partially determine the structure of the network: On one hand, links joining terms and documents must be directed from term nodes to

[1] We speak about the relevance of a term in the sense that the user explicitly employs this term when formulating a query. Similarly, a term is not relevant when the user also explicitly employs it, but in this case in a negative sense: he/she is not interested in documents containing this term.

[2] In this case a document is relevant if it satisfies the user's information need.

document nodes and, on the other hand, there are not links between document nodes. The parent set of any document node D_j is then the set of term nodes that belong to D_j, i.e., $Pa(D_j) = \{T_i \in \mathcal{T} \,|\, T_i \in D_j\}$. To completely determine the network topology, we include an additional assumption: the terms are marginally independent among each other, which implies that there are not links between terms nodes (all of them are root nodes). In this way, we get a network composed of two simple layers, the term and document subnetworks, with arcs only going from nodes in the first subnetwork to nodes in the second one (see Figure 1).

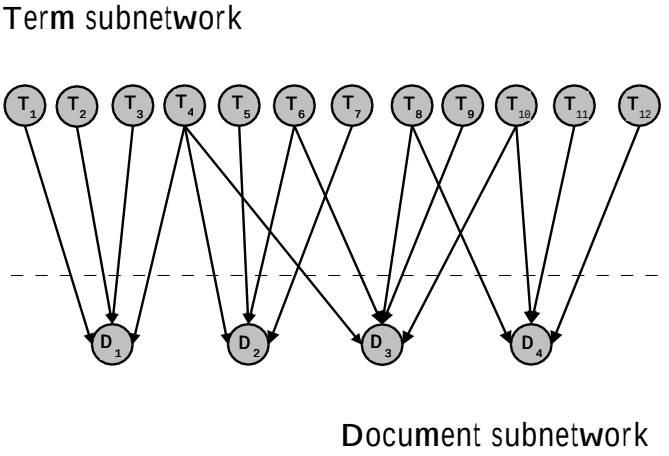

Fig. 1. Two-layered Bayesian network.

The final step to completely specify a Bayesian network is to estimate the probability distributions stored in each node. Two different cases have to be considered:

• Term nodes: In this case we store marginal distributions, estimated as follows:

$$p(t_i) = \frac{1}{M} \quad \text{and} \quad p(\bar{t}_i) = \frac{M-1}{M} \tag{2}$$

M being the number of terms in a given collection.

• Document nodes: In this case, the estimation of the conditional probabilities $p(D_j|Pa(D_j))$ is more problematic because of the huge number of parents that a document node has. For example, if a document has been indexed by 30 terms, we need to estimate and store 2^{30} probabilities. Therefore, instead of explicitly computing these probabilities, we use a *probability function*, also called a canonical model of multicausal interaction [5], which returns a conditional probability value when it is called during the inference stage, each time that a conditional probability is required. We have developed a new general canonical model: for

any configuration $pa(D_j)$ of $Pa(D_j)$ (i.e., any assignment of values to all the term variables in D_j), we define the conditional probability of relevance of D_j as follows:

$$p(d_j|pa(D_j)) = \sum_{\substack{T_i \in D_j \\ t_i \in pa(D_j)}} w_{ij} \tag{3}$$

where the weights w_{ij} have to verify $0 \leq w_{ij}$ and $\sum_{T_i \in D_j} w_{ij} \leq 1$. So, the more terms are relevant in $pa(D_j)$ the greater is the probability of relevance of D_j.

Given a query Q submitted to our system, the retrieval process starts placing the evidences, i.e., the terms T_Q belonging to Q, in the term subnetwork by setting their states to t_Q (relevant). The inference process is run, obtaining for each document its probability of relevance given that the terms in the query are also relevant, $p(d_j|Q)$. Then, the documents are sorted by their posterior probability to carry out the evaluation process.

Taking into account the number of nodes in the Bayesian network and the fact that, although the network topology seems relatively simple, it contains cycles and nodes with a great number of parents, general purpose inference algorithms cannot be applied due to efficiency considerations, even for small document collections. To solve this problem, we have designed a specific inference process that takes advantage of both the topology of the network and the kind of probability function used for document nodes, eq. (3): the propagation process is substituted by a single evaluation for each document node, but ensuring that the results are the same that the ones obtained using exact propagation in the entire network [2]:

$$p(d_j|Q) = \sum_{T_i \in D_j} w_{ij}\ p(t_i|Q) \tag{4}$$

Moreover, as terms nodes are marginally independent, we know, using eq. (2), that

$$p(t_i|Q) = \begin{cases} 1 & \text{if } T_i \in Q \\ \frac{1}{M} & \text{if } T_i \notin Q \end{cases} \tag{5}$$

Therefore, the computation of $p(d_j|Q)$ can be carried out as follows:

$$p(d_j|Q) = \sum_{T_i \in D_j \cap Q} w_{ij} + \frac{1}{M} \sum_{T_i \in D_j \setminus Q} w_{ij} \tag{6}$$

A simple modification of this model is to include the information about the frequency of the terms in the query Q, qf_i, with the aim of giving more importance to the terms more frequently used (as is usual in other IR models). This can be done by duplicating qf_i times in the network each term T_i appearing in the query. Then, eq. (6) is transformed in

$$p(d_j|Q) = \sum_{T_i \in D_j \cap Q} w_{ij}\ \mathrm{qf}_i + \frac{1}{M} \sum_{T_i \in D_j \setminus Q} w_{ij} \tag{7}$$

4 A Bayesian Network Model with Three Layers

In the previous model, document nodes are related only through terms in common. This fact makes almost impossible to retrieve a document that does not contain any of the terms used to formulate the query, even in the case that these terms are related (in some way) to the ones indexing the document. One approach to deal with this situation could be to include arcs in the term subnetwork modeling direct relationships between terms [1]. Using these relationships, the instantiation of the query terms would increase the probability of relevance of other terms, which in turn would increase the probability of relevance of some documents containing them. A different approach, which is the one considered in this paper, is to directly include in the model relationships between documents. These relationships will play in our model a role similar to the clustering techniques used in other IR models [8,10].

In the absence of information about direct and obvious relationships between documents in the form of, for example, citations or common references, these relationships in our model will be based on measuring (asymmetric) similarities between documents, by means of the estimation of the conditional probabilities of relevance of every document given that another document is relevant. These probabilities will be computed using the Bayesian network with two layers described previously.

So, given any document D_j, if we compute the probabilities $p(d_j|d_i)\ \forall D_i \in \mathcal{D}$, then the documents giving rise to the greatest values of $p(d_j|d_i)$ are the ones which are more closely related with D_j (in the sense that D_j has a high probability of being relevant when we know that D_i is relevant for a given query). Let $R_c(D_j)$ be the set of the c documents more related with D_j[3]. These relationships would be represented in the document subnetwork as arcs going from the documents $D_i \in R_c(D_j)$ to document D_j.

However, instead of using a document subnetwork with one layer, we will use two layers: we duplicate each document node D_k in the original layer to obtain another document node D'_k, thus forming a new document layer, and the arcs connecting the two layers go from $D_i \in R_c(D_j)$ to D'_j (i.e., $Pa(D'_j) = R_c(D_j)$). In this way we obtain a new Bayesian network with three layers (see Figure 2). We use this topology for two reasons: (1) the network with two layers used so far is maintained without changes as a subnetwork of the extended network, and therefore we do not have to redefine the conditional probabilities associated to the document nodes (eq. 3); (2) the new topology contains three simple layers, without connections between the nodes in the same layer, and this fact will redound to the efficiency of the inference process.

Now, we have to define the conditional probabilities $p(D'_j|pa(D'_j))$ for the documents in the second document layer. We use a probability function of the type defined in eq. (3), more precisely:

[3] Note that D_j will always belong to $R_c(D_j)$.

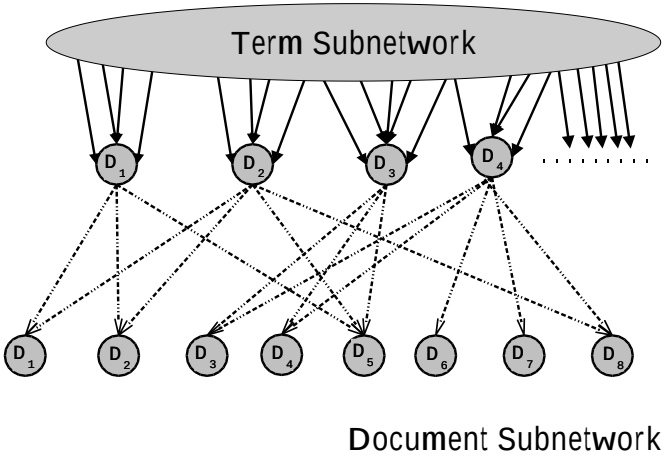

Fig. 2. Three-layered Bayesian network.

$$p(d'_j|pa(D'_j)) = \frac{1}{S_j} \sum_{\substack{D_i \in Pa(D'_j) \\ d_i \in pa(D'_j)}} p(d_j|d_i) \tag{8}$$

where $S_j = \sum_{D_k \in Pa(D'_j)} p(d_j|d_k)$, and the values $p(d_j|d_i)$ are obtained, using the network with two layers, during the building process of the network with three layers.

To compute $p(d'_j|Q)$, we can again take advantage of both the layered topology and eq. (8) to replace the propagation process in the whole network by the following evaluation [2]:

$$p(d'_j|Q) = \frac{1}{S_j} \sum_{D_i \in Pa(D'_j)} p(d_j|d_i) p(d_i|Q) \tag{9}$$

where the probabilities $p(d_i|Q)$ are computed according to equations (6) or (7). Note that the value $p(d'_j|Q)$ measures the relevance of a document by combining the contribution of the query, $p(d_i|Q)$ $\forall D_i \in R_c(D_j)$ (using the probabilities computed for the two layered network), and the document relationships, $p(d_j|d_i)$.

To completely specify the Bayesian network model with three layers, we need to explain how to calculate the values $p(d_j|d_i)$ and how to select the documents D_i that will be the parents of D'_j. Although the derivation is somewhat more involved, it can be proven that $p(d_j|d_i)$ can be calculated (without propagation) by means of

$$p(d_j|d_i) = \frac{1}{M} \left(\sum_{T_k \in D_j} w_{kj} \right) + \frac{M-1}{M} \left(\frac{\sum_{T_k \in D_j \cap D_i} w_{kj} w_{ki}}{\sum_{T_h \in D_i} w_{hi}} \right) \tag{10}$$

Using eq. (10), given a document D_j, to select the c parents of its copy D'_j in the second document layer, we only have to select the c documents D_i with the greatest values of

$$\frac{\sum_{T_k \in D_j \cap D_i} w_{kj} w_{ki}}{\sum_{T_h \in D_i} w_{hi}} \tag{11}$$

Eq. (11) says that the more terms have the documents D_j and D_i in common, the more related (similar) is D_i with D_j, which seems us quite natural. At the same time, the more terms D_i contains (which may indicate that D_i is a document related with many topics), the degree of similarity of D_i with any document D_j decreases.

5 Related Work

In this section we will briefly describe the two main retrieval models based on Bayesian networks, comparing them with our model and establishing the main differences.

The first model was developed by Croft and Turtle [13,15,16], the *Inference Network Model*, which is composed, in its simplified form, by two networks: the document and query networks. The former represents the document collection and contains two kinds of nodes: the document nodes, representing the documents, and the concept nodes, symbolizing the index terms contained in the documents. The arcs go from each document node to each concept node used to index it. The document network is fixed for a given collection. However, the query network is dynamic, in the sense that it is specific for each query, and is composed by three types of nodes: The Information Need node (inn), that represents the user's generic information need; a set of intermediate query nodes, used in case of having multiple query representations, and, finally, the query concept nodes (in the simplified form, they are just the concept nodes in the document network, and represent the connection between the two networks). The arcs in the query network go from query concept nodes to query nodes, and from query nodes to the Information Need node.

Each type of node stores a probability matrix, called link matrix in their notation, that in certain cases, depends on the type of query being formulated (boolean or probabilistic).

The retrieval is carried out by instantiating a single document node D_j each time, and computing the probability that the Information Need is satisfied given that this document has been observed, $p(inn|d_j)$. Actually, Turtle and Croft precompute the intermediate probabilities $p(t_i|d_j)$ in the document network, and, later, use closed-form expressions to evaluate $p(inn|d_j)$ as a function of the probabilities $p(t_i|d_j)$, for those terms T_i appearing in the query submitted by the user.

A first difference with our approach is that we do not have a query network. A second distinction, also topological, is that the arcs in our model are directed in the opposite way (from term nodes to document nodes). We think that is more intuitive to speak about the probability that a document is relevant given a query than the opposite. Therefore, our choice implies to instantiate the query, or specifically, the terms that it contains, and propagate towards document nodes. This fact means that we only have to propagate once, unlike Turtle and Croft's

model in which they have to run one propagation per document. With respect to propagation, our inference method allows us to propagate in the whole network by only estimating prior probabilities and evaluating probability functions.

The Ribeiro and Reis' model [7,6,11] is designed to simulate the Vector Space, Boolean and Probabilistic models. Their network is composed of three types of nodes: document nodes, concept nodes, and the query node. The arcs go from concept nodes to the document nodes where they occur, and from the concept nodes (appearing in the query) to the query node. In this model, the probabilities of interest are $p(d_j|Q)$, which could be computed as

$$p(d_j|Q) = \alpha^{-1} \sum_{\tau} p(d_j|\tau)p(Q|\tau)p(\tau), \tag{12}$$

where τ represents any of the 2^M assignments of values to all the terms in the collection. This computation is obviously unfeasible. So, depending on the model to be simulated, the probabilities $p(Q|\tau)$ and $p(\tau)$ are defined in such a way that all the terms in the previous addition except one (corresponding to a given configuration τ_Q) are always equal to zero. Thus, the computation in eq. (12) becomes straightforward: the inference is reduced to evaluate a function $(p(d_j|\tau_Q))$ in the only non-zero configuration.

The network topologies of this model and ours are very similar, except by the fact that we do not consider a query node. The main differences appear in the conditional probability distributions considered, in our case these distributions are not 'degenerated' and do not depend on the query, and we truly perform probabilities propagation.

Another important difference between these two models ans ours is that we include direct relationships between documents, thus obtaining a more expressive model.

6 Experimental Results

To test the performance of the two retrieval models explained in the previous sections, we have used four well-known document collections: ADI, CISI, CRANFIELD and MEDLARS. The main characteristics of these collections with respect to number of documents, terms and queries are (in this ordering): ADI (82, 828, 35), CISI (1460, 4985, 76), CRANFIELD (1398, 3857, 225) and MEDLARS (1033, 7170, 30). The results obtained by our models will be compared with the ones obtained by two different IR systems: SMART [10][4] and the Inference Network model[5]. The performance measure that we have used is the average precision for the *eleven* standard values of recall (denoted AP-11).

[4] We used the implementation of this IR system available at the Computer Science Department of Cornell University, using the *ntc* weighting scheme.

[5] In this case we have built our own implementation, and we used the configuration parameters proposed by Turtle in [13]: $p(t_i|d_j = \text{true}) = 0.4 + 0.6 * \text{tf} * \text{idf}$ and $p(t_i|\text{all parents false}) = 0.3$.

In the experiments, the specific weights w_{ij}, for each document D_j and each term $T_i \in D_j$, used by our models (see eq. 3) are:

$$w_{ij} = \alpha^{-1} \frac{\mathrm{tf}_{ij} \cdot \mathrm{idf}_i^2}{\sqrt{\sum_{T_k \in D_j} \mathrm{tf}_{kj} \cdot \mathrm{idf}_k^2}} \tag{13}$$

where α is a normalizing constant (to assure that $\sum_{T_i \in D_j} w_{ij} \leq 1 \ \ \forall D_j \in \mathcal{D}$).

The AP-11 values obtained by SMART and the Inference Network are shown in Table 1.

Table 1. AP-11 values for SMART and Inference Network.

	ADI	CISI	CRAN	MED
SMART	0.4706	0.2459	0.4294	0.5446
Inf. Network	0.4612	0.2498	0.4367	0.5534

The results for the experiments with the Bayesian network with two (BN-2) and three (BN-3) layers are displayed in Tables 2 and 3, respectively. The columns of the experiments that use eq. (6) are labeled with '1' and the ones that use eq. (7) are labeled with 'qf'. The rows labeled with '%SM' and '%IN' show the percentage of change of the performance measure obtained by our methods with respect to SMART and the Inference Network, respectively. For the Bayesian network with three layers, we have carried out experiments with three different values for the number, c, of document nodes in the first document layer that are parents of the document nodes in the second document layer ($c = 5, 10, 15$).

Table 2. Experiments with the two-layered Bayesian network.

	ADI		CISI		CRAN		MED	
	BN-2 1	BN-2 qf	BN-2 1	BN-2 qf	BN-2 1	BN-2 qf	BN-2 1	BN-2 qf
AP-11	0.4707	0.4709	0.2206	0.2642	0.4323	0.4309	0.5552	0.5458
%SM	+0.02	+0.06	-10.29	+7.44	+0.68	+0.35	+1.95	+0.22
%IN	+2.06	+2.10	-11.69	+5.76	-1.01	-1.33	+0.33	-1.37

Several conclusions may be drawn from these experiments: First, with respect to the use of eq. (7), i.e., the frequency qf of the terms in the query, instead of eq. (6), none of the two methods is clearly preferable to the other: For two collections (CRANFIELD and MEDLARS), the best results are obtained without using qf, whereas for the other two collections (ADI and CISI), the use of qf improves the results. Anyway, the differences between the two methods are rather

Table 3. Experiments with the three-layered Bayesian network.

		ADI		CISI		CRAN		MED	
		BN-3 1	BN-3 qf	BN-3 1	BN-3 qf	BN-3 1	BN-3 qf	BN-3 1	BN-3 qf
$c = 5$	AP-11	0.4724	0.4728	0.2211	0.2639	0.4331	0.4318	0.5651	0.5551
	%SM	+0.38	+0.47	-10.09	+7.32	+0.86	+0.56	+3.95	+1.93
	%IN	+2.43	+2.52	-11.49	+5.64	-0.82	-1.12	+2.29	+0.55
$c = 10$	AP-11	0.4717	0.4719	0.2221	0.2650	0.4333	0.4321	0.5687	0.5580
	%SM	+0.23	+0.28	-9.68	+7.77	+0.91	+0.63	+4.43	+2.46
	%IN	+2.28	+2.32	-11.09	+6.08	-0.78	-1.05	+2.76	+0.83
$c = 15$	AP-11	0.4715	0.4716	0.2223	0.2651	0.4332	0.4323	0.5708	0.5598
	%SM	+0.19	+0.21	-9.60	+7.81	+0.89	+0.68	+4.81	+2.79
	%IN	+2.23	+2.26	-11.01	+6.12	-0.80	-1.01	+3.14	+1.16

small, except in the case of CISI, where the results are remarkably better using qf (perhaps the explanation may be that the qf values for CISI are considerably larger than for the other collections).

Second, the results in Tables 1 and 2 show that BN-2 can compete with SMART and the Inference Network: in general, BN-2 obtains better AP-11 values, although the percentages of change are very low, except in the case of CISI.

Third, looking at Tables 2 and 3, we can see that the extended network BN-3 systematically improves the results of BN-2, showing that taking into account document interrelationships may be a good idea. It can also be observed that, except in the case of ADI, the AP-11 values obtained by BN-3 increase as the parameter c increases. However, the differences between BN-2 and BN-3 are so small, that it could be questioned the usefulness of increasing the complexity of the Bayesian network retrieval model by including the new document layer (which implies the necessity of precomputing the probabilities $p(d_j|d_i)$). After analysing, for each document D_j, the values $p(d_j|d_i)$, we realized that even the greatest values of $p(d_j|d_i)\ \forall i \neq j$ are extremely low compared with $p(d_j|d_j) = 1$ (typically $p(d_j|d_i) \approx 0.0025$). This fact may be the cause of the scarce improvement produced by BN-3 with respect to BN-2, since the value $p(d_j|Q) = p(d_j|d_j)p(d_j|Q)$ dominates completely the other components in eq. (9), $\sum_{D_i \in Pa(D'_j), D_i \neq D_j} p(d_j|d_i)p(d_i|Q)$, and therefore the ranking of documents obtained by using eq. (9) would be almost the same that the one obtained by the BN-2 model (which only uses $p(d_j|Q)$).

In order to test the truthfulness of this conjecture and, if possible, overcome the problem, we have modified the probability function defined in eq. (8) to reduce the importance of the term $p(d_j|Q)$ in the computation of $p(d'_j|Q)$. The new probability function $p(d'_j|pa(D'_j))$, also of the type defined in eq. (3), is the following:

$$p(d'_j|pa(D'_j)) = \begin{cases} \frac{1-\beta}{S_j-1} \sum_{\substack{D_i \in Pa(D'_j) \\ d_i \in pa(D'_j) \\ D_i \neq D_j}} p(d_j|d_i) & \text{if } d_j \notin pa(D'_j) \\ \\ \frac{1-\beta}{S_j-1} \sum_{\substack{D_i \in Pa(D'_j) \\ d_i \in pa(D'_j) \\ D_i \neq D_j}} p(d_j|d_i) + \beta & \text{if } d_j \in pa(D'_j) \end{cases} \tag{14}$$

where the parameter β will control the importance of the contribution of the document relationships being considered for document D_j to its final degree of relevance. Once again taking advantage of the layered topology, we can compute $p(d'_j|Q)$ as follows:

$$p(d'_j|Q) = \frac{1-\beta}{S_j-1} \sum_{\substack{D_i \in Pa(D'_j) \\ D_i \neq D_j}} p(d_j|d_i)p(d_i|Q) + \beta p(d_j|Q) \tag{15}$$

The results obtained by using eq. (15) instead of eq. (9), with a value $\beta = 0.3$, are displayed in Table 4.

Table 4. Experiments with the BN-3 model using eq. (15) and $\beta = 0.3$.

		ADI		CISI		CRAN		MED	
		BN-3 1	BN-3 qf	BN-3 1	BN-3 qf	BN-3 1	BN-3 qf	BN-3 1	BN-3 qf
$c = 5$	AP-11	0.4732	0.4787	0.2301	0.2575	0.4477	0.4453	0.6640	0.6480
	%SM	+0.55	+1.72	-6.43	+4.72	+4.26	+3.70	+21.92	+18.99
	%IN	+2.60	+3.79	-7.89	+3.08	+2.52	+1.97	+19.99	+17.09
$c = 10$	AP-11	0.4757	0.4822	0.2413	0.2754	0.4591	0.4554	0.6878	0.6734
	%SM	+1.08	+2.46	-1.87	+12.00	+6.92	+6.05	+26.29	+23.65
	%IN	+3.14	+4.55	-3.40	+10.25	+5.13	+4.28	+24.29	+21.68
$c = 15$	AP-11	0.4783	0.4825	0.2424	0.2787	0.4630	0.4577	0.6999	0.6847
	%SM	+1.64	+2.53	-1.42	+13.34	+7.82	+6.59	+28.52	+25.73
	%IN	+3.71	+4.62	-2.96	+11.57	+6.02	+4.81	+26.47	+23.73

The results obtained in Table 4 clearly represent a remarkable improvement with respect to the ones in Table 3, thus showing that the use of the document relationships is quite useful, provided that the weights measuring the strength of these relationships are set appropriately. In this case the best results are always obtained using $c = 15$ parents for each document node in the second document layer. We have also carried out some other experiments with different values for the parameter β and, in general, the results are quite similar to the ones displayed in Table 4, for values $\beta \leq 0.5$, whereas the performance decreases for higher values of β.

7 Concluding Remarks

In this paper we have presented two new IR models based on Bayesian networks. The first model, BN-2, is composed of a layer of term nodes and a layer of document nodes, joining each term node to the document nodes representing the documents indexed by this term. This model has been endowed with an inference mechanism that allows us performing exact propagation in the whole network efficiently. The experimental results obtained with four collections show that this model is competitive with respect to SMART and the Inference Network Model.

This initial model has been enriched, establishing the most important relationships among documents, thus increasing the expressiveness of BN-2 and giving rise to the model with three layers, BN-3. In this second approach, we have shown the mechanism by which the document relationships are captured. The new inference method, also exact, is composed of two stages: a propagation in the original network, and the combination of this information with that one stored in the second document layer, updating the probability of relevance of each document given a query with the strength of the relationships among the documents. The empirical results show an improvement of the performance of the BN-3 model, revealing the suitability of the document layer extension.

As future works, we plan to design another method to establish the parents of each document node by using, instead of the values $p(d_j|d_i)$, the values $p(d_j|pa(D_i))$ (i.e., instantiating, instead of each document D_i, the individual terms it contains). We want to test whether there is any difference in the results, and determine which method would perform better. A second research line will be the development of new probability functions in the second document layer, to more accurately combine the information about the relevance of the documents given the query and the strength of the document relationships. We are also planning to extend our model to cope with boolean queries.

On the other hand, we have tested our models with some standard test collections, whose sizes are smaller than actual collections. Our objective has been just to determine the validity of the proposed models for IR, focusing our attention only in modelling aspects. Experimentation with TREC collections will be one of the most important points in which we are going to center our future research. The basic next objective will be to determine the efficiency and effectiveness of our models with these collections. This task could suggest some modifications or refinements in our models, related to the propagation and construction of the second document layer.

Acknowledgments. This work has been supported by the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT), under Project TIC2000-1351.

References

1. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. Building Bayesian network-based information retrieval systems. In 2^{nd} *Workshop on Logical and Uncertainty Models for Information Systems (LUMIS)*, 543–552, 2000.
2. J.M. Fernández-Luna. Modelos de Recuperación de Información Basados en Redes de Creencia (in Spanish). Ph.D. Thesis, Universidad de Granada, 2001.
3. R. Fung and B.D. Favero. Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(2):42–57, 1995.
4. M.E. Maron and J.L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7:216–244, 1960.
5. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, San Mateo, 1988.
6. I. Reis Silva. Bayesian Networks for Information Retrieval Systems. Ph.D. Thesis, Universidad Federal de Minas Gerais, 2000.
7. B.A. Ribeiro-Neto and R.R. Muntz. A belief network model for IR. In *Proceedings of the* 19^{th} *ACM–SIGIR Conference*, H. Frei, D. Harman, P. Schäble and R. Wilkinson, eds., 253–260, 1996.
8. C.J. van Rijsbergen. *Information Retrieval. Second Edition*. Butter Worths, London, 1979.
9. S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.
10. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1983.
11. I. Silva, B. Ribeiro-Neto, P. Calado, E. Moura, and N. Ziviani. Link-based and content-based evidential information in a belief network model. In *Proceedings of the* 23^{th} *ACM–SIGIR Conference*, 96–103, 2000.
12. K. Sparck Jones, S. Walker, and S.E. Robertson. A probabilistic model of information retrieval: development and comparative experiments Part 1. *Information Processing and Management*, 36:779–808, 2000.
13. H.R. Turtle, Inference Networks for Document Retrieval, Ph.D. Thesis, Computer and Information Science Dpt., University of Massachusetts, 1990.
14. H.R. Turtle and W.B. Croft. Inference networks for document retrieval. In *Proceedings of the* 13^{th} *ACM–SIGIR Conference*, J.-L. Vidick, ed., 1–24, 1990.
15. H.R. Turtle and W.B. Croft. Efficient probabilistic inference for text retrieval. In *Proceedings of the RIA0'91 Conference*, 644–661, 1991.
16. H. R. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *Information Systems*, 9(3):187–222, 1991.