International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 11, Suppl. (September 2003) 101–116

World Scientific
www.worldscientific.com

IMPROVING THE EFFICIENCY OF THE BAYESIAN NETWORK RETRIEVAL MODEL BY REDUCING RELATIONSHIPS BETWEEN TERMS

DE CAMPOS, LUIS M., FERNÁNDEZ-LUNA, JUAN M., AND HUETE, JUAN F.
Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática. Universidad de Granada
18071, Granada. Spain
{lci,jmfluna,jhg}@decsai.ugr.es

Received October 2002
Revised January 2003

The Bayesian Network Retrieval Model is able to represent the main (in)dependence relationships between the terms from a document collection by means of a specific type of Bayesian network, namely a polytree. However, although the learning and propagation algorithms designed for this topology are very efficient, in collections with a very large number of terms, these two tasks might be very time-consuming. This paper shows how by reducing the size of the polytree, which will only comprise one subset of terms which are selected according to their retrieval quality, the performance of the model is maintained, whereas the efforts needed to learn and later propagate in the model are considerably reduced. A method for selecting the best terms, based on their inverse document frequency and term discrimination value, is also presented.

Keywords: Bayesian Networks, Information Retrieval, Learning, Dependence, Clustering

1. Introduction

A good definition of *Information Retrieval* (IR) is given by Salton and McGill [1], who define it as the representation, storage, organization, and accessing of information items. The software that performs these tasks is known as an *Information Retrieval System* (IRS). In our case, the information items will be *documents*, i.e. the textual representations of any item.

In this paper, we mainly focus our attention on that part of an IRS which is concerned with accessing information items, i.e. the identification of documents in a collection that are relevant to a particular information need: a user interacts with the IRS by formulating a query, which is a description of his/her information needs, and consequently obtains a set of documents which are supposedly the most suitable for his/her request.

In order to solve the IR problem, a great number of retrieval models have been developed. One of the main classical IR models is the Probabilistic model [2]. This represents the documents and queries as vectors containing a probabilistic weight for each term, and expresses the degree of importance for that term. This model computes the relevance probability given a document and a query (the probability that a document satisfies a query), and is based on the 'Probability ranking principle'. This principle states that the best overall retrieval effectiveness will be achieved when documents are ranked in decreasing order of their probability of relevance [3].

Bayesian Networks (BNs) [4], which are also based on probabilistic methods, have proved to be good models for managing uncertainty, even in the field of IR, where they have already been successfully applied as an extension of probabilistic IR models [5,6,7,8,9,10,11,12]. Intuitively, as stated in [13], "Bayesian networks are complex diagrams that organize the body of knowledge in a given area by mapping out cause-and-effect relationships between key variables and encoding them with numbers that represent the extent to which one variable is likely to affect another". More formally, a Bayesian network $G = (V, E)$ is a *Directed Acyclic Graph* (DAG), where the nodes in V represent the variables from the problem which we want to solve, and the arcs in E represent the dependence relationships between these variables. In this kind of graph, the knowledge is represented in two ways [4]: *(a)* qualitatively, showing the (in)dependences between the variables, and *(b)* quantitatively, by means of conditional probability distributions which shape the relationships. BNs can perform reasoning tasks efficiently: the independences represented in the graph reduce changes in the state of knowledge to local computations.

When a retrieval model is being designed, one of the assumptions that can usually be made is to consider the terms from the collection independent of each other, i.e. there are no relationships between the terms. From a computational point of view, this simplification makes all the posterior developments easier, but removes a certain accuracy from the model. Several examples of models have been published

which have considered term relationships, for instance by using probability theory and dependence trees [14,15], from a Logical Imaging point of view [16], with Neural Networks [17], or considering BNs [18,19].

The *Bayesian Network Retrieval* (BNR) Model [10,12] also includes term relationships by means of the (in)dependences encoded in a polytree (a DAG in which there is no more than one directed path connecting each pair of nodes), comprising nodes which represent the terms. Although for the topology chosen, there is a set of efficient learning and propagation algorithms, if the number of terms in the collection is very large, these two stages may be very time-consuming. The first is not so critical, since the graph needs only be constructed once, but the second, performed in retrieval time, must be very fast since a user is involved.

Bearing in mind that the use of term relationships is important when it comes to improving the performance of an Information Retrieval System, but also taking efficiency reasons into account, in this paper a modification of the BNR model is presented: in order to code term relationships, instead of considering a polytree containing the nodes representing all the terms in the collection, a new polytree will be learned which comprises a reduced set of terms and their relationships. The remaining nodes which do not belong to that network will be completely isolated from the others. The advantage of this new topology is that not only does it represent the best relationships, but it also reduces the time needed to build them and later perform inference. Another important aspect is that the retrieval performance with respect to the polytree containing all the terms is maintained almost at the same level. But a determining factor for achieving this aim is the way in which those terms which will belong to the new polytree are selected. Firstly, we will show how, by selecting those terms which are supposedly the best in terms of retrieval performance, even when "ad hoc" methods are used, the performance of the new BNR model is almost the same, with the advantage of a reduction in run time. Secondly, the selection method will be refined, and an automatic selection method presented which is based on a combination of two well-known term characteristics: inverse document frequency and term discrimination value.

In order to put these ideas into practice, this paper has been organized in the following way: the next section describes the BNR model; section 3 deals with the problem of reducing the set of term-to-term relationships; section 4 presents an "ad hoc" frequency-based method and the results of our experimentation; section 5 describes the method designed to automatically select the best set of terms and the empirical results; and the final section contains the concluding remarks and some proposals for future research.

2. The Bayesian Network Retrieval Model

We shall describe the retrieval model introduced in this section in the following way: firstly, we shall state its topology (types of variables and how they are related); secondly, we shall describe how the required probability distributions of each node are estimated; and thirdly, once the network has been built, we shall present the

inference mechanism that allows the model to retrieve documents.

2.1. *Topology of the model*

Our model comprises a DAG, where two different sets of nodes can be found:

- The set of term nodes, $\mathcal{T}$. A variable T_i associated to a term takes its values from 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'.

- The set of document nodes, $\mathcal{D}$. 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', and 'the document D_j is relevant' for a given query.

A document is relevant for a given query if it satisfies the user's information need expressed using this query. A term is considered relevant if the user believes that it will appear in relevant documents (hence s/he will explicitly use it when formulating the query). Similarly, a term is not relevant when the user believes that the relevant documents do not contain it: s/he is not interested in documents containing this term.

Our approach to building a BN from a document collection uses a combination of domain specific knowledge and machine learning methods: a set of guidelines that the model must preserve, which partially determine the network structure, together with the capacity to automatically infer relationships between the variables, thereby completing the network topology.

The following guidelines have been considered:

- There is a link joining each term node $T_i \in \mathcal{T}$ and each document node $D_j \in \mathcal{D}$ whenever T_i belongs to D_j. This simply reflects the dependence between the relevance values of a document and those of the terms used to index it.

- There are no links joining document nodes D_j and D_k. In other words, the dependence relationships between documents are not direct: they always depend on the terms included in these documents.

- Given a query, the degree of relevance of a document D_j can be completely determined by knowing the relevance status of all the terms indexing D_j. In the absence of this information, knowledge about the relevance or irrelevance for the same query of some other document, D_k, could have an influence on D_j. This means that any document D_j is conditionally independent of any other document D_k when we are sure of the relevance values for all the terms indexing D_j.

These assumptions also imply that the links joining term and document nodes must be directed from terms to documents; moreover, the parent set of a document node

D_j, $Pa(D_j)$, only comprises the set of term nodes that have been used to index D_j, i.e.

$$Pa(D_j) = \{T_i \in \mathcal{T} \mid \text{term } T_i \text{ is used to index document } D_j\}.$$

As we would like to give our model the ability to represent dependences between terms, we decided to apply an automatic learning algorithm that takes the set of documents as the input and generates a ***polytree*** of terms as the output. It uses term co-occurrence criteria in the collection in order to measure the strength of the dependences. The main reason for restricting the structure of the term subnetwork to a polytree is the existence of exact and efficient inference algorithms [4], specific for this topology, which run in a time which is proportional to the number of nodes.* An additional advantage is that the algorithms for learning polytrees are quite efficient, in comparison with those for learning general BNs. The specific polytree learning algorithm used within the BNR model is composed of three main parts: i) computation of the dependency degrees between all pairs of nodes using Kullback-Leibler's cross entropy; ii) the construction of the skeleton of the Bayesian network using a greedy approach to obtain the Maximum Weight Spanning Tree; and iii) the orientation of the tree edges, obtaining a polytree as a result. In all these steps, there are specific features adapted to the field of I.R. This algorithm is described in detail in [20]. Figure 1 shows an example of the BN that we have just described, where dashed arcs represent the learned polytree using the information stored in the document collection.

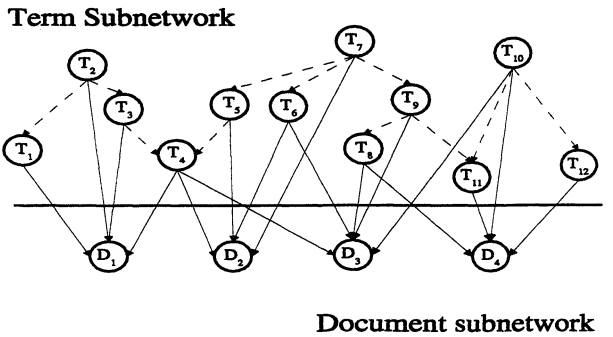

Fig. 1. The topology of the Bayesian Network Retrieval Model.

2.2. *Estimating probability distributions*

Once the structure has been built, the next step is to estimate the probability distributions stored in each node of the network. All the root nodes, i.e. those with no parents, will therefore store marginal distributions. In our case, the only nodes

*It should be noted that inference in general unrestricted BNs is an NP-hard problem.

of this type are term nodes. For each root term node, we must assess $p(t_i)$ and $p(\bar{t}_i)$. We use the following estimator:†

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

where M is the number of terms in the collection. It should be noted that this estimator assigns the same prior probability to all the root nodes.

The nodes with parents (term and document nodes) will store a set of conditional probability distributions, one for each of the possible configurations that the parent sets $Pa(T_i)$ and $Pa(D_j)$ can take.

Before continuing, let us introduce some notation: given a subset of term variables, $S \subseteq \mathcal{T}$, a configuration s of S is an assignment of values to all the variables in S. For instance, for $S = \{T_1, T_2, T_3, T_4\}$, two possible configurations are $\{t_1, t_2, \bar{t}_3, t_4\}$ and $\{t_1, \bar{t}_2, t_3, \bar{t}_4\}$. Given S and a configuration s, let $R(s)$ and $NR(s)$ be the subsets of terms in S that are relevant and not relevant, respectively, in the configuration s, i.e.

$$R(s) = \{T_k \in S \,|\, t_k \in s\}, \quad NR(s) = \{T_k \in S \,|\, \bar{t}_k \in s\}\,. \tag{2}$$

Let $D(s)$ be the subset of documents that are indexed by all the terms that occur as relevant in s and are not indexed by those which are not relevant in s, i.e.

$$D(s) = \{D_j \in \mathcal{D} \,|\, R(s) \subseteq Pa(D_j) \text{ and } NR(s) \cap Pa(D_j) = \emptyset\}\,. \tag{3}$$

$n(s)$ will denote the number of elements in the set $D(s)$, and the configurations of the parent sets $Pa(T_i)$ and $Pa(D_j)$ will be denoted as $pa(T_i)$ and $pa(D_j)$, respectively.

For term nodes with parents, the required conditional probabilities are computed using an estimator based on the Jaccard coefficient [14] that measures the similarity between two sets as the ratio between the number of elements in the intersection and the union of these sets. This measure (also used by Savoy [26]) is adapted to our model using the following expression:

$$\begin{aligned} p(\bar{t}_i \,|\, pa(T_i)) &= \frac{n(\{\bar{t}_i\} \cup pa(T_i))}{n(\{\bar{t}_i\}) + n(pa(T_i)) - n(\{\bar{t}_i\} \cup pa(T_i))} \\ p(t_i \,|\, pa(T_i)) &= 1 - p(\bar{t}_i \,|\, pa(T_i))\,. \end{aligned} \tag{4}$$

Finally, we must estimate the conditional probabilities placed on the document nodes. This is more problematic as a result of the huge number of required probabilities. For example, if a document has been indexed with 30 terms, we need to estimate and store 2^{30} probabilities. Therefore, we use a specific canonical model to represent these conditional probabilities. For any configuration $pa(D_j)$ of $Pa(D_j)$, we define the conditional probability of relevance of D_j as follows:

$$\begin{aligned} p(d_j \,|\, pa(D_j)) &= \sum_{T_i \in R(pa(D_j))} w_{ij} \\ p(\bar{d}_j \,|\, pa(D_j)) &= 1 - p(d_j \,|\, pa(D_j))\,, \end{aligned} \tag{5}$$

†Other estimators of the prior probability distributions have been tested, but this is the one that exhibits the highest retrieval performance [21].

where each weight w_{ij} represents the importance of the term T_i in the document D_j, with $w_{ij} \geq 0$ and $\sum_{T_i \in Pa(D_j)} w_{ij} \leq 1$. Therefore, the greater the number of relevant terms in $pa(D_j)$, the greater the probability of relevance of D_j.

2.3. *The retrieval engine*

Once the complete network has been built, it can be used to obtain a relevance value for each document given a query submitted to the IRS by a user. In this case, we consider that the terms in the query are evidences for the propagation process. These terms are all instantiated to 'relevant'. After instantiation, propagation is performed, obtaining as a result the probability of relevance of each document, $p(d_j \mid Q)$, where Q is the set of terms belonging to the query. The documents are then sorted according to their corresponding posterior probability and shown to the user.

Taking into account the number of nodes in the BN and the fact that the entire network contains cycles in the underlying undirected graph, as well as nodes with a great number of parents, general purpose inference algorithms cannot be applied for reasons of efficiency, even for small document collections. In order to solve this problem, the BNR model uses a specific inference method, called *propagation + evaluation*, which takes advantage of both the topology of the network and the kind of canonical model used for document nodes, eq. (5). This method returns the same values as an exact propagation would [21]. It comprises two stages: 1) an exact propagation in the term subnetwork (using Pearl's propagation algorithm for polytrees [4]). The results of this first stage are the posterior probabilities of relevance for each term node, $p(t_i \mid Q)$, $\forall T_i$. 2) the evaluation of the following expression, using the information obtained in the previous propagation:

$$p(d_j \mid Q) = \sum_{T_i \in Pa(D_j)} w_{ij} \cdot p(t_i \mid Q) \,. \tag{6}$$

A simple modification of this model is to include the information about query term frequencies, qf_i, in order to give more importance to the terms most frequently used in the query Q (as is usual in other IR models). To obtain this performance, we propose to clone each term T_i in the query $qf_i - 1$ times. For example, if the query frequency of a term T_i is three ($qf_i = 3$), then two new fictitious nodes would be created in the network with the same information contained in the node T_i. Then, eq. (6) becomes:

$$p(d_j \mid Q) = \sum_{T_i \in Pa(D_j) \cap Q} w_{ij} \cdot p(t_i \mid Q) \cdot qf_i + \sum_{T_i \in Pa(D_j) \setminus Q} w_{ij} \cdot p(t_i \mid Q) \,. \tag{7}$$

3. Reducing Term to Term Dependence Relationships

From a computational point of view, the use of the proposed BNR model may present two main disadvantages when dealing with large document collections:

- The time needed to construct the term dependence structure, i.e. the polytree, might be large when we consider real databases. Nevertheless, we only learn the network once at the beginning of the process.

- The propagation of the evidence (given by the query) through the entire term subnetwork. Even though our model uses Pearl's polytree propagation algorithm (which is polynomial in the number of term nodes), it can be time-consuming for large collections with many terms.

In this section, the following question is considered: ***Is it possible to obtain a balance between the use of term relationships in a BN-based representation and the computational cost needed to build the model, and then, to retrieve documents with it?*** We believe that the answer is yes, and we propose to achieve such a balance by reducing the number of terms involved in the polytree, thereby reducing both the learning and propagation times. This reduction should be carried out without markedly decreasing the retrieval performance of the IRS.

In order to put this idea into practice, the entire set of terms $\mathcal{T}$ will be divided into two subsets, $\mathcal{T}_g$ and $\mathcal{T}_b$, which shall include those terms that can be considered *good* and *bad* for retrieval purposes, respectively. A different processing will then be carried out with these subsets (see Figure2):

- Terms in $\mathcal{T}_b$: we shall assume that they are marginally independent of the rest of the terms in the collection, i.e. $p(T_i \mid T_j) = p(T_i)$, $\forall T_i \in \mathcal{T}_b$, $\forall T_j \in \mathcal{T} \setminus \{T_i\}$. Therefore, there is no term to term relationship involving these terms. Taking into account the guidelines explained in Section 2 and the imposed independence relationships, we only need to connect each bad term with those documents it belongs to, leaving it isolated from all the other terms. It is therefore not necessary to apply any learning algorithm.

- Terms in $\mathcal{T}_g$: in this case, term to term dependence relationships are allowed, but they only involve terms in $\mathcal{T}_g$. These relationships are learned using exactly the same polytree algorithm [20] considered in the original BNR model, the only difference being that we restrict the set of variables considered by the algorithm to those terms in $\mathcal{T}_g$. Finally, and in order to complete the graph structure, we need to add an arc from each term in $\mathcal{T}_g$ to all the documents it belongs to.

The quantitative component (i.e. the probability distributions) of the new reduced model is the same as in the original BNR model (see Section 2.2). The effects of this new network topology on the inference process are obvious: in order to compute the posterior probabilities $p(t_i \mid Q)$, we only need to perform a real propagation in the reduced polytree associated to $\mathcal{T}_g$[‡] which is computationally more efficient.

[‡]It should be noted that for each term $T_j \in \mathcal{T}_b$, $p(t_j \mid Q) = 1$ if $T_j \in Q$ and $p(t_j \mid Q) = p(t_j)$ if $T_j \notin Q$.

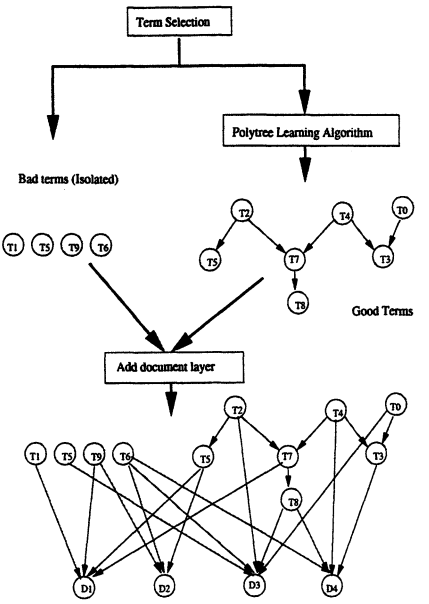

Fig. 2. Construction of the reduced BNR model.

The remaining problem is to decide how to partition the set of terms into the two subsets $\mathcal{T}_g$ and $\mathcal{T}_b$, in such a way that the relationships between terms in $\mathcal{T}_g$ are the most useful for retrieval purposes. This problem is very similar to feature selection methods in statistical learning of text categorization [22,23], in which the aim is to reduce dimensionality. The main difference is that, while feature selection attempts to remove a set of non-informative terms according to corpus statistics, we do not intend to remove terms (in fact we will use all the terms) but rather relationships between terms. Our task also involves selection, but the selection of the set of the best terms capable of expressing only the strongest relationships.

4. Frequency-based Terms Selection Methodology

In order to divide the complete set of terms into the subsets $\mathcal{T}_g$ and $\mathcal{T}_b$, we could measure the quality of a term in the collection by first using the document frequency of a term (which is defined as the number of documents in which that term occurs in the whole collection). A similar frequency-based approach for reducing the number of terms has been used in other IR problems, such as automatic query expansion [24], automatic thesaurus construction [25], as well as in text categorization [22]. By using this frequency, we can therefore divide the set of terms into three subsets:

1. *High frequency terms*: these types of terms are present in a large number of documents in the collection, and therefore they are not good discriminators for distinguishing between relevant and non-relevant documents. Focusing on our retrieval model, the learning algorithm could connect these terms with

others, even with very low frequency terms. This would imply that by instantiating one of these low frequency terms, the probability of relevance of the high frequency term would increase, as well as the probability of relevance of those documents it belongs to, thus retrieving many non-relevant documents. Therefore, frequently occurring terms will be classified as bad terms.

2. *Low frequency terms*: these terms are contained by a small number of documents in the collection. These terms will exhibit a high dependence when they occur in the same document, which might be due to random associations. As a result, our learning algorithm will include links between these terms when constructing the polytree. However, considering that we do not have enough knowledge about these relationships, the inclusion of these links will not help improve the retrieval effectiveness of the system. Therefore, low frequency terms will be classified as bad terms.

3. *Medium frequency terms*: considering that the dependence relationships that involve medium frequency terms will help to discriminate between relevant and non-relevant documents and to retrieve documents which are most similar to a given query, these terms will be considered as good terms.

4.1. *Experimentation*

In order to test the performance of the new reduced model, we have used five well-known test document collections, whose characteristics are shown in Table 1, all of which were obtained from the Computer Science Department ftp site at Cornell University (ftp.cs.cornell.edu). The reason why we chose them for our experiments is that they establish a good test bed to experiment with our model, preparing and tuning our algorithms to work with larger collections.

Colection	No. Documents	No. Terms	No. Queries
ADI	82	828	35
CACM	3204	7562	64
CISI	1460	4985	76
CRANFIELD	1398	3857	225
MEDLARS	1033	7170	30

Table 1: Main features of the five test collections.

The retrieval performance will be determined by computing the *recall* (the proportion of relevant documents retrieved) and *precision* (the proportion of retrieved documents that are relevant) measures, plotted in a recall-precision graph. Another way of measuring performance, which has finally been adopted in this paper due to questions of space, is the average precision for the *eleven* standard values of recall (denoted AP-11).

The specific weights w_{ij} used in our experiments (in eq. 5), based on the cosine

measure [1], are:

$$w_{ij} \propto \frac{tf_{ij} \cdot idf_i^2}{\sqrt{\sum_{T_i \in Pa(D_j)} tf_{ij}^2 \cdot idf_i^2}}, \tag{8}$$

where tf_{ij} is the frequency of the term T_i in the document D_j, and idf_i is the inverse document frequency of such a term in the collection.

Before discussing the influence of the term reduction in the performance of the model, we will display the results obtained by our original BNR model and two different IRSs for the five test collections, which may be useful for comparative purposes. These two additional systems are SMART [1] and the Inference Network (IN) model [5,6] § The AP-11 values obtained by SMART, the Inference Network, and the original BNR model are shown in Table 2. We present the results obtained with BNR considering query term frequencies (i.e. using eq. (7), labeled in the table as BNRq) and without considering query term frequencies (using eq. (6), labeled with BNR). From these results, we can conclude that none of these methods is clearly preferable to another.

	ADI	CISI	CRANFIELD	CACM	MEDLARS
SMART	0.4706	0.2459	0.4294	0.3768	0.5446
IN	0.4612	0.2498	0.4367	0.3974	0.5534
BNR	0.4130	0.2007	0.4314	0.3759	0.6200
BNRq	0.4613	0.2301	0.4116	0.4046	0.5792

Table 2: AP-11 values for SMART, IN and BNR models.

In order to distinguish between good and bad terms, we need to define two bounds in the frequency of the terms of the collection: on the one hand, we say that a low frequency term is any term with a document frequency less than five. On the other, in order to discriminate high frequency terms, we will use a collection dependent criterion, such as in [24,25]. In particular, we will consider a term to be highly frequent if it appears in more than 10% of the documents in the collection. Therefore, good terms are those that have a document frequency in the interval $[5, N/10]$, with N being the number of documents in the collection. This way of selecting terms is closely related to Luhn's work on automatic text analysis based on Zipf's law [14].

Table 3 shows, for each collection, the percentage of terms that have been classified as good terms; the rows labeled AP-11 rBNR and AP-11 rBNRq display the performance measure of the reduced model using eq. (6) and (7), respectively. We also present the percentage of change ($\%C$) of the performance measure in the reduced models with respect to the original BNR and BNRq models.

§ We used the implementation of SMART available at the Computer Science Department of Cornell University, using the *ntc* weighting scheme. For Inference Network, we built our own implementation, and used the configuration parameters proposed by Turtle [5]: $p(t_i|d_j = \text{true}) = 0.4+0.6*\text{tf}*\text{idf}$ and $p(t_i \mid \text{all parents false}) = 0.3$.

	ADI	CISI	CRANFIELD	CACM	MEDLARS
% terms	9	30	35	22	26
AP-11 rBNR	0.4632	0.2104	0.4395	0.3692	0.6180
$\%C$	12.5	4.8	1.9	-1.8	-0.3
AP-11 rBNRq	0.4605	0.2454	0.4101	0.3983	0.5764
$\%C$	-0.2	6.6	-0.4	-1.5	-0.5

Table 3: Size and performance measures in the reduced BNR model using frequency-based terms selection.

In terms of efficiency, we could conclude that the reduction in the number of terms is quite significant, implying that the learning and propagating tasks are around 70% faster. Considering the performance measures, we can see that they are similar to or even better than those in the original model. Therefore, the idea of reducing the number of terms looks promising for the construction of a Bayesian Network-based retrieval model.

The problem presented by the method used is that of selecting the upper and lower frequency bounds. Although it is possible to tune these bounds up in order to obtain a better performance of the system in terms of efficiency and effectiveness [26], the process is clearly collection dependent. For this reason, in the next section we propose a method to automatically select the best terms in order to learn the polytree.

5. A Method to Automatically Select the Best Terms

Our approach will combine the information given by the *Discrimination Value* and the *Inverse Document Frequency* of each term in the collection. We shall first define both measures, and then explain how they have been used to select terms.

- *Term Discrimination Value (tdv)* [1]. This value attempts to measure the usefulness of a term so as to distinguish between documents within a given collection. It is based on a similarity measure between documents, $S(D_i, D_j)$. Let $\bar{S}$ be the average similarity across all the documents and $\bar{S}_i$ the average similarity across the same documents after the term T_i has been removed (i.e. not using T_i as an index term for any document). If a term T_i is discriminating, its removal will result in an increment of the average similarity $\bar{S}_i$; if removing T_i changes the average similarity very little, this term is less helpful. The discrimination value of a term T_i is therefore computed as the difference between $\bar{S}_i$ and $\bar{S}$, i.e. $tdv(T_i) = \bar{S}_i - \bar{S}$. One criterion for selecting the terms to be included in the polytree could be to select those terms with the highest tdv.

- *Inverse Document Frequency.* The second alternative consists in using the inverse document frequency of each term T_i, idf_i, which is an inversely proportional value to the number of occurrences of the 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 T_i. Thus, the more frequent a term, the smaller the *idf* value. If we sort the terms according to their *idf* values, the terms in which we are interested are those which are placed in the central positions of the ranking. It should be noted that this option is equivalent to the "ad hoc" approach using term frequencies, since *idf* is a monotonic decreasing transformation of the frequency of the terms in the collection.

Our aim is to select those terms which simultaneously have high discrimination values and medium-high inverse document frequencies. With this choice, the best terms are captured according to the two different measures.

The automatic term selection method that we propose is based on a combined measure of the *tdv* and *idf* values of each term. Bearing in mind the aim of automating the process, we leave the responsibility of the selection to a ***non-supervised classification (clustering) algorithm.*** The classification algorithm we have used is the *k-Means* algorithm [27] with the Euclidean distance as similarity measure.¶ Thus, using this clustering algorithm we are able to group (according to some similarities between *tdv* and *idf* features) the set of terms into two fixed classes (*Good Terms*, $\mathcal{T}_g$; *Bad Terms*, $\mathcal{T}_b$). Then, *Good Terms* will be employed to learn the polytree (these terms will be interconnected), and *Bad Terms* will be included in the term subnetwork but isolated from any other term.

5.1. *Experimental results*

The specific similarity measure used to compute the term discrimination values in our experiments is the cosine [1]. After applying the *k-Means* algorithm and the polytree learning process to the five test collections, in order to build the corresponding reduced BNR model, and later run the retrieval process, we obtained the results shown in Table 4. We show the percentage of good terms (% terms), the percentage of reduction in the average propagation times for the reduced polytree with respect to the original (% time), as well as the AP-11 values, using and without using qf (AP-11 rBNRq and AP-11 rBNR, respectively), and the percentages of change with respect to the corresponding original models ($\%C$).

We can observe how the sizes of the sets of good terms obtained by the *k-Means* algorithm are smaller than those shown in Table 3, except for the ADI collection. As a result, it seems that the percentages of change with respect to the original BNR model are slightly worse (except in the case of CISI, where the opposite occurs). We could also say that the results shown in Table 4 are similar to those in Table 3, showing the appropriate behavior of the method that we have designed. We can therefore conclude that it is possible to automatically select the set of good terms without considerably changing the performance of the system, thereby avoiding the problem of setting the thresholds required by the frequency-based approach.

¶The implementation of the *k-Means* algorithm that we have used is the one included in the STATGRAPHICS statistical package.

	ADI	CISI	CRANFIELD	CACM	MEDLARS
% terms	27	14	13	11	14
% time	84.60	97.37	65.43	79.59	87.05
AP-11 rBNR	0.3985	0.2092	0.4257	0.3664	0.5911
$\%C$	-3.5	4.2	-1.3	-2.5	-4.7
AP-11 rBNRq	0.4459	0.2521	0.4055	0.3956	0.5669
$\%C$	-3.3	9.6	-1.5	-2.2	-2.1

Table 4: Size and performance measures in the reduced BNR model using automatic term selection by means of the k-Means algorithm.

6. Conclusions and Further Research

In this paper, we have presented an Information Retrieval model based on Bayesian Networks. It comprises a DAG divided into two parts: the term subnetwork, in which all the terms of the collection are represented by means of graph nodes, as well as the main relationships between them; and the document subnetwork, where no document relationships are considered. Document nodes are only linked to the term nodes by which they have been indexed. The main relationships between all the terms in the collection are captured using a polytree, a DAG with a simple structure that supports relatively fast learning and propagation algorithms.

The main problem of this approach is presented in the term subnetwork because even when an efficient algorithm is used to learn it and then propagate in it, these processes might be very time-consuming if the number of nodes is very high. We propose to tackle this problem by reducing the size of the polytree underlying the term subnetwork: instead of comprising all the terms, the polytree would be learned using only one subset of terms. The remaining terms would be included in the term subnetwork but isolated from all the other terms. The first aim of this work was to determine whether this variation of the original model would maintain the same level of performance or, at least, whether the loss was not so important with respect to the gain in efficiency. After designing a set of "ad hoc" experiments and analyzing the results, we concluded that the results were very similar and efficiency had been improved considerably.

The second aim of this work was to design a technique that, given the set of all the terms in a collection, allows the automatic selection of the best terms in order to learn the polytree. Our criteria of term quality could be measured by means of its term discrimination value and inverse document frequency, selecting those terms which present a high discrimination value and a medium-high inverse document frequency. In order to determine the best terms according to these criteria, we left the responsibility of selecting the terms to a non-supervised classification algorithm, which returns the classification of the terms into two classes as the output: the class of good terms, used to learn the polytree, and the class of bad terms. The results of the experiments carried out using the classification algorithm gave good results, maintaining similar results with respect to the "ad hoc" experiments and, therefore,

with respect to the original model.

We can therefore conclude from this study that it is important to reduce the size of the polytree, obtaining relevant savings in the learning and propagation stages, without notably worsening the quality of the results. This is a good technique to be employed with actual collections such as TREC databases, in which their size would impede an easy construction of the BNR model, and its subsequent use for retrieval purposes. With respect to future work, we shall apply this technique to the aforementioned TREC collections and evaluate their performance.

Acknowledgments

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

References

1. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, 1983).
2. F. Crestani and M. Lalmas and C. J. van Rijsbergen and L. Campbell, "Is this document relevant?... probably. A survey of probabilistic models in Information Retrieval", *ACM Computing Survey* **30:4** (1991) 528 – 552.
3. S. E. Robertson, "The probability ranking principle in IR", *Journal of Documentation* **33:4** (1977) 294 – 304.
4. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan and Kaufmann, 1988).
5. H. R. Turtle, *Inference Networks for Document Retrieval*, Ph.D. Thesis, University of Massachusetts, 1990.
6. H. R. Turtle and W. B. Croft, "Inference Networks for document retrieval", *Proc. of the* 13^{th} *ACM–SIGIR Conference on Research and Development in Information Retrieval (SIGIR'90).* 1990, pp. 159 – 168.
7. B. A. Ribeiro–Neto and R. R. Muntz, "A Belief network model for IR", *Proc. of the* 19^{th} *ACM–SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96).* 1996, pp. 253 – 260.
8. I. Reis Silva, *Bayesian Networks for Information Retrieval Systems*, Ph.D. Thesis, Federal University of Minas Gerais, 2000.
9. D. Ghazfan and M. Indrawan and B. Srinivasan, "Toward meaningful Bayesian networks for information retrieval systems", *Proc. of the* 6^{th} *Inf. Proc. Management of Uncertainty in Knowledge-based Systems Conference (IPMU'96).* 1996, pp. 841 – 846.
10. L.M. de Campos and J.M. Fernández-Luna and J.F. Huete, "Building Bayesian network-based information retrieval systems", *Proc. of the* 2^{nd} *Workshop on Logical and Uncertainty Models for Information Systems (LUMIS).* 2000, pp. 543 – 552.
11. L. M. de Campos and J. M. Fernández-Luna and J. F. Huete, "A layered Bayesian network model for document retrieval", *Lecture Notes in Computer Science* **2291** (2002) 169 – 182.
12. L.M. de Campos and J.M. Fernández-Luna and J.F. Huete, "The Bayesian network retrieval model: Foundations and performance", *International Journal of Approximate Reasoning.* Submitted.
13. L. Helm, "Improbable Inspiration", *Los Angeles Times.* October, 28. 1996.
14. C. J. van Rijsbergen, *Information Retrieval*, Second Edition (Butter Worths, 1979).
15. C. T. Yu and C. Buckley and K. Lam and G. Salton, "A generalized term dependence model in Information Retrieval", *Information Technology: Research and Development* **2** (1983) 129 – 154.

16. F. Crestani and C.J. van Rijsbergen, "Information Retrieval by Logical Imaging", *Journal of Documentation* **51:1** (1995) 1 - 15.
17. S. K. M. Wong and Y. J. Cai and Y. Y. Yao, "Computation of term associations by a neural network" *Proc. of the* 16^{th} *ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR-93).* 1993, pp. 107 -115.
18. Y. Park and K. Choi, "Automatic thesaurus construction using Bayesian networks", *Information Processing & Management* **32:5** (1999) 543 - 553.
19. J. Savoy, "Bayesian Inference Networks and Spreading Activation in Hypertext Systems", *Information Processing & Management* **28:3** (1992) 389 - 406.
20. L. M. de Campos and J. M. Fernández-Luna and J. F. Huete, "Query expansion in Information Retrieval systems using a Bayesian network-based thesaurus", *Proc. of the* 14^{th} *Uncertainty in Artificial Intelligence Conference.* 1998, pp. 53 - 60.
21. J. M. Fernández-Luna, *Modelos de Recuperación de Información basados en Redes de Creencia* (In Spanish), Ph.D. Thesis, University of Granada, 2001.
22. Y. Yang and J.O. Pedersen, "A comparative study on feature selection in text categorization", *Proceeding of the ICML'97 Conference.* 1997, pp. 412 - 420.
23. M. Sahami, *Using Machine Learning to Improve Information Access*, Ph.D. Thesis, University of Standford, 1998.
24. Y. Qiu and H. Frei, "Concept-based query expansion", *Proc. of* 16^{th} *ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93).* 1993, pp. 160 - 169.
25. C. Crounch and B. Yang, "Experiments in automatic statistical thesaurus construction", *Proc. of* 15^{th} *ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92).* 1992, pp. 77 - 88.
26. L. M. de Campos and J. M. Fernández-Luna and J. F. Huete, "Reducing term to term relationships in an extended Bayesian network retrieval model", *Proc. of the* 9^{th} *Inf. Proc. Management of Uncertainty in Knowledge-based Systems Conference (IPMU'02).* 2002, pp. 1195 - 1202.
27. A. K. Jain and M. Murty and P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys* **31:3** (1999) 264 - 323.