Document Instantiation for Relevance Feedback in the Bayesian Network Retrieval Model

Luis M. de Campos, Juan F. Huete

Dpto. de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática. Universidad de Granada, 18071 Granada, Spain.

Juan M. Fernández-Luna

Dpto. de Informática, Escuela Politécnica Superior. Universidad de Jaén, 23071 Jaén, Spain.

Abstract

Relevance Feedback consists on formulating automatically a new query, according to the relevance judgements provided by the user after evaluating the set of retrieved documents. In this paper we introduce a new relevance feedback method for the Bayesian Network Retrieval Model. This method is based on the instantiation of the observed documents as relevant or non-relevant in the Bayesian Network. We explain the theoretical bases of the model and propose different schemes for carrying out this task. The quality of the method is tested using a preliminary experimentation with different collections.

1 Introduction

Relevance Feedback is one of the most useful Query Modification Techniques in the field of Information Retrieval (IR). This method is put into practice when the user needs to improve the query that he/she has formulated to the *Information Retrieval System (IRS)*, because the documents initially retrieved do not fulfil the user's information need. This technique works as follows: the user submits a query to the IRS, which generates a ranking of documents arranged by their corresponding relevance degrees according to the query. The user inspects this sorted list, and determines which documents are relevant and non-relevant. With this information, the IRS modifies the initial query, giving more importance to the terms appearing in relevant documents, and weakening the strength of those that belong to non-relevant documents. This process is repeated until the user is completely satisfied with the set of retrieved relevant documents. In [16] the reader can find a good review of this technique and how it has been applied to different retrieval models.

Bayesian Networks-based IRSs can be used to deal with the intrinsic uncertainty with which IR is pervaded [3,7] and might be considered as an extension of the probabilistic IR model. There are several Belief Network-based IRSs [4,8,9,13,17,18] differing in both, the structure and the probability distributions used. Among these models we can distinguish two main groups: those models that obtain a ranking of documents based on the probability that a document is relevant to a given query [4,8,9,13] and the Inference Network model [17,18], that ranks the documents based on the probability that a document satisfies the user's information need. In this paper we are going to present an approach for Relevance Feedback in the *Bayesian Network Retrieval* (BNR) model [4,8]. This approach is based on the instantiation of the judged documents to relevant or non-relevant. This is the main difference with other relevance feedback methods based on the addition of terms to the query and/or recalculating the probability matrices stored in the network [5,10].

In an experimental IRS, the performance of the retrieval must be determined, task that is usually carried out by computing the *recall (R)* - the proportion of relevant documents retrieved- and *precision (P)* - the proportion of retrieved documents that are relevant- measures. When Relevance Feedback is applied, the evaluation is usually done by means of the *Residual Collection* method [1], which removes from the collection all the documents that the user has evaluated in the first relevance judgement step.

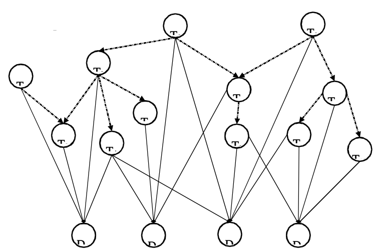
This paper is organised as follows: In the second section we briefly describe the basis of the Bayesian Network

Retrieval (BNR) model, showing its most important features. The third section deals with a theoretical overview of the proposed relevance feedback methods. Later, in sections four and five, we will explain how we have put into practice the ideas showed in the previous section. The sixth section of this paper shows the experimental results obtained with four classical collections. Finally, the last section deals with the conclusions of this work and introduces new research lines in this area.

2 Description of the Bayesian network retrieval model

In this section we are going to present briefly the Bayesian Network Retrieval model [4,8]. When we are interested in representing our knowledge by means of Bayesian networks, the first task is to select those variables relevant to the problem we are tackling. In IR problems we can distinguish two different sets of variables (nodes in the graph): The set of terms in the glossary from a given collection and the set of documents, grouped in the *term* and *document* subnetworks, respectively. Each term, T_i, consists of a binary random variable taking values in the set $\{\bar{t}_i, t_i\}$, representing "the term T_i is not relevant" and "the term T_i is relevant", respectively. Similarly, a variable referring to a document, D_j, has its domain in the set $\{\bar{d}_j, d_j\}$ representing, respectively, "the document D_j is not relevant to a given query", and "the document D_j is relevant to a given query". In the paper, we will denote terms and documents with a lower-case letters to indicate that we are considering their particular instantiation to relevant or non-relevant.

Focusing on the structure of the Bayesian network (see Figure) we consider two main subsets of links: (1) The first one, that includes links between term and document nodes in the graph. For each term indexing a document, there is a directed link from the term node to the node associated with the document it belongs to. (2) The second one, that only includes links connecting pairs of term nodes. These links represent some dependence relationships between terms. The inclusion of these links will improve the performance of the IRS [8]. In order to capture the relationships between terms in the collection, a Bayesian network learning algorithm is used to construct the term sub-network. Due to efficiency considerations, we restrict the topology of the learned graph to a Polytree structure (a graph without cycles). The algorithm [6] is based on maximum weight spanning tree (the weights associated to the links are obtained using a dependence measure between terms) adding some additional features to include the directions of the links, increasing the expressive power of the term sub-network.



Once we know the structure of the graph, the final step to completely specify a Bayesian network is to estimate the probability distributions stored in each node. Three different cases have to be considered:

- Term nodes having no parents: In this case we store marginal distributions, estimated as follows: $p(t_i) = 1/M$ and $p(\bar{t}_i) = 1 - p(t_i)$, being M the number of terms in a given collection.
- Term nodes with parents: For each node, we need to store a set of conditional probability distributions, one for each possible configuration of the parent set. These distributions are estimated using frequencies of cooccurrence of terms in the collection.
- Document nodes: In this case, the estimation of the conditional probabilities is more problematic because of the huge number of parents that a document node has. For example, if a document has been indexed with 30 terms, we need to estimate and store 2^{30} (approx. 1.07 e10^9) probabilities. Therefore, instead of explicitly computing these probabilities, the BNR model uses a *probability function* [4,8], that returns a conditional probability value when it is called during the propagation stage, each time that a conditional probability is required. In this paper, we are

going to use a probability function based on the cosine measure [15]. Given a configuration $\pi(D_j)$ of the set of terms in document D_j, for example $\pi(D_j) = (t_1, t_2, \bar{t}_3, t_4, \bar{t}_5, \ldots)$, the probability function computes the probability of relevance of document D_j by means of $p\left(d_j \middle| \pi\left(D_j\right)\right) = \alpha_j \sum_{T_i \in D_j, T_i = t_i} tf_{ji} idf_i^2$ where tf_{ji} is the frequency of the term T_i in the document D_j, idf_i is its inverse document frequency and α_j is a normalising constant computed as $1/\sqrt{\sum_{T_i \in D_j} tf_{ji} idf_i^2}$.

Therefore, once the entire network is built and given a query submitted to the system, the retrieval process starts by placing the evidences, i.e., the terms belonging to the query, in the term subnetwork by setting their states to *"the term is relevant"*. The propagation process is run, obtaining for each document its probability of relevance given that the terms in the query are also relevant. Then, the documents are presented to the user decreasingly sorted by their posterior probability. This ranking of documents is also used to carry out the performance evaluation process.

Taking into account the topology of the BNR model, and due to efficiency considerations, general-purpose inference algorithms can not be applied. To solve this problem, an inference process composed of two steps has been designed: *propagation+evaluation*. It is important to note that this process ensures that the results are the same that the ones obtained using exact propagation in the entire network [8]. First, we propagate the query just only in the term subnetwork by means of the Pearl's Exact Propagation Algorithm in Polytrees [12] computing for each term its probability of being relevant given the query submitted to the IRS, i.e., $p(t_i|Q)$. In the second step, we compute the probability that each document is relevant given the query by means of the evaluation of the following formula. In this way we are performing exact probabilistic inference in a complex network very efficiently.

$$p(d_j|Q) = \alpha_j \sum_{T_i \in D_j} tf_{ji} idf_i^2\, p\left(t_i \middle| Q\right) \qquad \textbf{[1]}$$

There are three main differences between BNR and the other Bayesian network based IRSs in the literature: i) The ability of the model to include relationships between terms, learned from the collection. ii) The way of computing and storing the probability values in the nodes of the network. iii) In the propagation process, considering a two-steps inference approach, that computes the a posteriori probability of relevance for a given document. A detailed study of this model and its performance can be found in [8].

3 Description of the methodology for relevance feedback

The methodology for Relevance feedback in the BNR model is based on the idea that, by evaluating a set of documents, the user obtains new pieces of evidence (about the relevance to our information need) for those variables relevant to the process. The particular implementation of the proposed methodology gives rise to two different approaches, that basically differ in the way that new evidences are included in the model: (1*) Term based relevance feedback,* that considers that, by evaluating the retrieved documents, the user obtains new evidences over the set of terms in the evaluated documents. A detailed study of this approach can be found in [5]. (2) *Document based relevance feedback,* where the set of evidences is focused on document nodes. The main advantage of this methodology is that we can include the evidences in the model without adding new nodes neither re-estimating the probability distributions stored in the network, as the Inference Network model does [10].

In this paper we will focus on the second approach. From our point of view, this new approach is closer to the user thinking, since, by evaluating a set of documents, the user gives a relevance judgement about the whole document and not on the particular terms indexing the document. Formally, in order to include the new evidences in the IRS, it would be sufficient to instantiate each judged document as relevant, d_i, or non-relevant, $\bar{d}_i$, in the network. Therefore, the new query becomes $Q_1 = (Q, d_1, d_2, \ldots, d_k, \bar{d}_{k+1}, \ldots, \bar{d}_{|R|})$, where |R| is the number of documents judged by the user. A direct implementation of this approach forbids the use of the designed *propagation+evaluation* algorithm. So that, and taking into account that the use of general-purpose exact inference algorithms is prohibitive, it was necessary the development of a new two-step method that approximate efficiently the desired probabilities: First, by computing a set of messages in document nodes and second, by combining these messages in term nodes. In the following sections we will detail this method.

Therefore, in the propagation process, and as a consequence of document instantiation, each evaluated document sends a message to the set of terms indexing it, encoding that the user has judged it as relevant or non-relevant. Particularly, each term T indexing an evaluated document will receive a message, denoted by $\lambda(T)$, encoding a pair of likelihood values, i.e., $\lambda(T) = \{p(Obs|\bar{t}), p(Obs|t)\}$. For example, the message $\lambda(T) = \{0,1\}$ encodes that "our belief supports that term T is relevant" and the message $\lambda(T) = \{1,2\}$ encodes the assertion "the belief in the relevance of T is two times more than the belief supporting its non-relevance". This message, in a normalized form, will be combined with the whole information that node T obtains from its parents and children in the network, in order to update the belief on term relevance. Then, this information will be distributed through the whole network.

At this point, we need to distinguish whether a term belongs to the original query or not. Particularly, non-query terms indexing the observed documents receive a set of messages, one from each judged document, modifying the belief about their relevance for our information needs. Therefore, it can be considered that they are playing the same role that classical query expansion when evaluating the probability $p(d_j \mid Q_1)$.

Focusing on query terms, *Tq*, since they are findings (they belong to the original query *Q*), we are sure about their relevance to our information need. In this case, we have that by also instantiating the evaluated documents (to relevant or non-relevant) our belief on the relevance for query terms can not be modified. Nevertheless, query term reweighting has been considered as a valuable tool in order to improve the performance of relevance feedback in many IRSs. Therefore, and in order to perform query term reweighting in the BNR model, query terms should be handled in a different way. Particularly, we will retract the evidence and insert soft (likelihood values) evidence instead. Therefore, these terms will receive only one message (from an imaginary node) representing our belief on the relevance of the term after judging the retrieved documents. Thus, we can still consider query terms as relevant or, otherwise, we need to decrease the belief supporting the relevance of the terms, i.e., penalise them. Empirical results [5] show that a better performance is obtained if we only penalise those terms that exclusively occur in non-relevant documents. Therefore, for those terms considered completely relevant we use the λ message $\lambda(T_q) = \{0,1\}$ and for penalised terms we use $\lambda(T_q) = \{1 - (1/(n_{\bar{r}t} + 1)),1\}$ with $n_{\bar{r}t}$ representing the number of times that this term occurs in non-relevant retrieved documents.

From now on, we will focus on non-query terms. We can distinguish two main sub-problems: How an observed document computes the λ values sent to the terms indexing the document and how a term node combines all the received messages in order to obtain a global value representing the evidence on the term.

4 Measuring λ-messages

Each term T_i indexing an evaluated document D_j receives a message from this document encoding a pair of likelihood values, i.e., $\lambda_{Dj}(T_i) = \{p(Obs|\bar{t}_i), p(Obs|t_i)\}$. Nevertheless, assessing these probabilities is problematic due to the huge number of parents that documents have. Therefore, the solution that we propose is to approximate the instantiation of documents by an explicit computation of an approximation of these likelihood messages. These values will be computed using the probabilities stored in the network. In order to determine the particular values of the λ vectors, we will distinguish between relevant and non relevant documents:

Non Relevant documents: In this case, since the document D_j has been considered as non relevant, it will send the message $\lambda_{Dj}(T_i) = \{1,0\}$ meaning that the term it contains is not useful to retrieve relevant documents, because it might introduce in the ranking documents that could be completely non-relevant[1].

Relevant documents: Different approaches will be considered:

d1: It is a naive approach that considers all the terms indexing a relevant document to be also completely relevant. In this case, we send the pair of likelihood values $\lambda_{Dj}(T_i) = \{0,1\}$. Experimental results [8] demonstrate that this is not a good choice, because terms indexing relevant documents are treated as query terms, thus changing the original sense of the query.

d2: The second approach is based on the λ messages that a node sends to its parents in Pearl's polytree propagation algorithm [12]. In this case, the likelihood values are encoded using $\lambda_{Dj}(T_i) = \{p(d_j|\bar{t}_i), p(d_j|t_i)\}$. In order to calculate $p(d_j|t_i)$ we can use Equation 1, i.e., $p(d_j|t_i) = \alpha_j \sum_{T_k \in D_j} tf_{jk} idf_k^2 p(t_k|t_i)$.

To calculate the different values $p(t_k|t_i)$ it would be necessary to perform a large number of propagations in the network, one for each term indexing the set of observed documents, being a time-consuming process. Therefore, we approximate these probabilities assuming independence between terms. Thus, we have $p(t_k|t_i) = p(t_k)$ if $i \neq k$ and $p(t_k|t_i) = 1$ if $i = k$. A similar reasoning is used to calculate $p(d_j|\bar{t}_i)$

After analysing the experimental results, we realised that λ vectors are very close to {1,1}, because $p(d_j|t_i)$ is quite similar to $p(d_j|\bar{t}_i)$. Nevertheless, they seem to discriminate properly the importance of the terms. To solve this problem, we thought that it could be interesting to change the scale. Particularly, we could estimate $\lambda_{Dj}(T_i) = \lambda_{Dj}(T_i)^\delta$ with $\delta > 1$. This method will be denoted by $d2^\delta$.

d3: This approach takes into account the influence of the original query, Q, in the computations of the λ vector, breaking the first restriction imposed in the previous method. Therefore, we will try to estimate the λ messages as $\lambda_{Dj}(T_i) = \{p(d_j|\bar{t}_i Q), p(d_j|t_i Q)\}$, which consider how the posterior probability of relevance of a document is affected by the addition of a new term to the query. Using Equation 1, the probability $p(d_j|t_i, Q)$ can be computed by means of

$$p(d_j|t_i, Q) = \alpha_j \sum_{T_k \in D_j} tf_{jk} idf_k^2 p(t_k|t_i Q) = \alpha_j \left[\sum_{T_k \in D_j, k \neq i} tf_{jk} idf_k^2 p(t_k|t_i Q) + tf_{ji} idf_i^2 \right]$$

The next target should be to estimate $p(t_k|t_i, Q)$ for each term T_k belonging to D_j. These computations would imply to propagate in the network considering both, the query and the new term t_i, as evidences, although this possibility is discarded because it is also a very time-consuming task. Therefore, we will try to find an approximation with the information that we own. In this case, since both terms, T_i and T_k, index the document D_j that has been judged as relevant, we could consider T_i and T_k being positively correlated, although very weakly, and that adding T_i to the original query will increase the evidence on the relevance of D_j. These two assumptions can be combined using an or-gate in the following way: $p(t_k|t_i, Q) = p(t_k|t_i, \bar{Q}) + (1 - p(t_k|t_i, \bar{Q}))p(t_k|\bar{t}_i, Q)$.

On the one hand, as $\bar{t}_i$ is the most probable state that T_i can take on, it is very reasonable to suppose that it does not add more information to the information given by Q, so we can make the approximation: $p(t_k|\bar{t}_i, Q) \approx p(t_k|Q)$. On the other hand, we also approximate $p(t_k|t_i, \bar{Q}) \approx \varepsilon$, ε being a very low value.

[1] Some other approaches have been studied but empirical results support this one [8].

The reason is that $p\left(t_k \middle| t_i, \overline{Q}\right)$ should be very small because when $\overline{Q}$ is instantiated, the posterior probability of t_k should also be very small, and the influence of instantiating t_i will not contribute to increase it. Taking into account these assumptions, the new expression for $p\left(t_k \middle| t_i, Q\right)$ becomes $p\left(t_k \middle| t_i, Q\right) = \varepsilon + (1-\varepsilon)\, p\left(t_i \middle| Q\right)$. Analogously, $p\left(d_j \middle| \bar{t}_i, Q\right)$ is computed using equation [1] and assuming that $p\left(t_k \middle| \bar{t}_i, Q\right) = p\left(t_k \middle| Q\right)$. The result is $p\left(d_j \middle| \bar{t}_i, Q\right) = p\left(d_j \middle| Q\right) - tf_{ji}\, idf_i^{\,2}\, p\left(t_i \middle| Q\right)$.

d4: In this method the query Q plays the role of a fictitious document that has also been observed as relevant. Therefore, in the set of observed components we include both, the document D_j and the query Q and the likelihood vector becomes $\lambda_{Dj}(T_i) = \left\{ p\left(d_j, Q \middle| \bar{t}_i\right), p\left(d_j, Q \middle| t_i\right) \right\}$. It is interesting to note that this likelihood vector is equivalent (differing only in the normalising constant) to $\lambda_{Dj}(T_i) = \left\{ \frac{p\left(\bar{t}_i \middle| d_j, Q\right)}{p(\bar{t}_i)}, \frac{p\left(t_i \middle| d_j, Q\right)}{p(t_i)} \right\}$, that measures how considering the document and the query as relevant the belief on term relevance changes with respect to the original prior probability. The numerators of the previous quotients can be computed as follows (similarly for $p\left(\bar{t}_i \middle| d_j, Q\right)$): $p\left(t_i \middle| d_j, Q\right) = p\left(d_j \middle| t_i, Q\right) p\left(t_i \middle| Q\right) / p\left(d_j \middle| Q\right)$. The values $p\left(t_i \middle| Q\right)$ and $p\left(d_j \middle| Q\right)$ were calculated in the original query. So, we only need to compute $p\left(d_j \middle| t_i, Q\right)$. In this case, we use two different methods to calculate this probability: The first one, denoted by *d4.1*, uses the approach in *d3*. We are going to explain the second one, denoted by *d4.2*. In this case, we also consider Equation 1, $p(d_j | t_i, Q) = \alpha_j \sum_{T_k \in D_j} tf_{jk}\, idf_k^{\,2}\, p\left(t_k \middle| t_i Q\right)$, but assuming that T_i and T_k are almost independent given Q, i.e., $p\left(t_k \middle| t_i, Q\right) \approx p\left(t_k \middle| Q\right)$ and will be computed as $p\left(t_k \middle| t_i, Q\right) = \beta + p\left(t_k \middle| Q\right)$ being β a small value.

It is important to note that for all the methods, as the calculus of the messages is carried out using previously computed values, this technique for relevance feedback can be implemented in a very efficient way.

5 Combining λ-messages

When several documents, $D_1, \ldots, D_s$, have a certain term T_i in common, the term node will receive a set of messages, one from each observed document, $\lambda_{Dj}(T_i), i = 1, \ldots, s$. These λ vectors must be combined to obtain a unique measure, $\lambda(T_i)$, that reflects the belief on the relevance or non-relevance of that term. In this paper we study to different methods to combine the information:

- The direct one, without using extra information. In this case, multiplying all the λ vectors received by a particular term carries out the combination $\lambda(T_i) = \prod_{j=1}^{s} \lambda_{Dj}(T_i)$
- The second approach, more informed, takes into account the query quality (measured as the number of relevant documents retrieved). Basically, we try to capture that when the number of relevant documents retrieved in the first query is high, then this initial query is doing a good work. Thus, it could be interesting to add new terms but in such a way that these terms do not have a strong impact in the original query. Similarly, if we retrieve a small number of relevant documents it will be helpful to favour the belief on the relevance of those terms indexing these documents. The way of implementing this idea could be to use the convex combination $\lambda'(T_i) = \alpha + (1-\alpha)\lambda(T_i)$, where $\alpha = n_r / |R|$, being n_r the number of relevant document retrieved and $\lambda(T_i)$ is the value obtained using the direct approach.

Finally, we need to distinguish between three different groups of terms indexing the observed documents: those terms that only occur in relevant documents (*positive terms*), those that only occur in non-relevant documents (*negative terms*), and those that occur in both types of documents (*neutral terms*).

For positive or negative terms we apply directly the equations of the direct or convex combination method. For neutral terms, because a non-relevant document sends the λ vector $\lambda_D(T_i) = \{1,0\}$, i.e., instantiating the term to non-relevant, we discard the messages, which is equivalent to not instantiating the term at all. Therefore, in this case we use the message $\lambda(T_i) = \{1,1\}$.

6 Empirical Results

To test these methods, we carried out the experimentation[2] with four collections: ADI, Cranfield, CISI and CACM. The main characteristics of these collections, presented with the format "Collection (number of documents, terms and queries: average precision at the three intermediate points of recall obtained by the BNR model)", are: ADI (82, 828, 35: 0.36), CACM (3204, 7562, 52: 0.34), CISI (1460, 4985, 76: 0.17), Cranfield (1398, 3857, 225: 0.42)[3].

We evaluate the feedback performance using the residual collection method [1] by considering the percentage of change of the average precision for the three points of recall with respect to the results obtained after submitting to the system the original queries. In the experimentation, the number of documents that the IRS gives back to the user is fifteen, and we take also into account those initial queries in which no relevant document has been retrieved.

Some initial remarks have to be done: The δ parameter in the $d2^{\delta}$ method has been set to 5, the ε parameter used to approximate $p(t_k | t_i, \overline{Q})$ in *d3* is set to 0.0075 and the β value in method *d4.2* has been set to $1/|R|$. All these values were obtained in a detailed previous experimentation stage with CACM. From this experimentation, we also concluded that the combination of both, query term reweighting and document instantiation, improves the performance of relevance feedback. Particularly, by considering only query term re-weighting, i.e., without instantiating document nodes, the feedback performance is 29.90, and when we instantiate the evaluated documents without modifying the belief on query terms the maximum feedback performance reached is 51.73. The best performance combining both techniques increases to 70.58.

In Table 1 we present the global results obtained considering both, query term reweighting and document instantiation, with the studied collections. Columns labelled with DC contain the percentage of change when the λ vectors have been combined using the direct combination and those columns labelled with CC show the results obtained using the convex combination approach. These results allow us to conclude that relevance feedback using document instantiation is effective in the BNR model, being its performance collection-dependent.

		ADI		Cranfield		CISI		CACM	
Exp.	*d*	DC	CC	DC	CC	DC	CC	DC	CC
1	*d2*	67.75	70.73	67.55	67.54	44.17	44.15	54.85	53.68
2	$d2^5$	137.70	64.88	87.81	70.32	43.64	44.53	53.86	55.31
3	*d3*	85.86	82.98	82.30	90.53	2.99	48.49	-11.60	69.46
4	*d4.1*	85.85	78.76	82.30	93.46	2.98	45.76	-11.66	70.58
5	*d4.2*	102.07	105.77	91.37	101.87	-7.72	42.51	-48.74	67.93

Table 1. Percentages of change using query term re-weighting and document instantiation

Finally, we would like to compare these results with the ones presented in the literature using other IRSs. Some

---

[2] We do not include all the results obtained in this experimentation due to the lack of space.

[3] We want to remark that some important preprocessing steps to obtain a vectorial representation of documents and queries, such as indexing and stemming, have been carried out using the facilities provided by the SMART system.

previous remarks have to be done: First, we must note that relevance feedback performance is highly dependent on the results obtained after submitting the original query to the system and that these results could vary from one IRS to another. Second, the number and the set of collections used are different from one experimentation to another and even when working with the same collection the number of queries used is not necessarily the same.

Therefore, in order to compare the results, we will consider the mean performance values obtained with different collections. The selected IRSs and relevance feedback methodology are: Ide (dec hi) expanding all terms for vector-based IRS and the probabilistic adjusted revised derivation for Probabilistic IRS. These two methods obtain the best results in Salton and Buckley's [14] experimentation. We also include the results obtained with two Bayesian Network based approaches, the first one presents the results obtained by Haines and Croft [10] with the Inference Network and the second one shows the best results obtained using term-based relevance feedback in BNR model [5].

Table 2 displays these results, where columns labelled with '#' indicate the number of used collections, columns labelled with 'm.p.' represent the mean performance over the used collections and columns labelled with 'std' represent the standard deviation. From this table we could conclude that the performance of the proposed methodology is competitive with other models, presenting similar mean values and smaller standard deviations. It is interesting the comparison with relevance feedback using term-based relevance feedback in the BNR model since they have the same experimental conditions. In this case, we can say that *d4.2* has a similar performance.

Relevance feedback Method	#	m.p.	std.	Relevance feedback method	#	m.p.	std.
Ide (dec hi) Expands all terms	5	87.4	39.72	d2 (Convex Combin.)	4	59.03	10.72
Probabilistic (adj. revised derivation)	5	68.8	54.82	$d2^5$ (Direct Combin.)	4	80.75	36.72
Inference Network	2	79.3	38.90	d3 (Convex Combin.)	4	72.87	15.97
Term-based method in the BNR	4	80.95	27.98	d4.1 (Convex Combin.)	4	72.14	17.30
				d4.2 (Convex Combin.)	4	79.52	25.94

Table 2. Mean percentages of change using different relevance feedback methods in different IRSs.

7 Concluding remarks

In this paper we have introduced a relevance feedback method for the Bayesian Network Retrieval model based on both, query term reweighting and the instantiation of the judged documents. We have presented the theoretical framework over it is based on. Summing up, each node representing a judged document computes a pair of likelihood values that will be sent to the terms indexing the document. Then, term nodes combine these values with the rest of messages received in order to obtain a global value representing the evidence on the term relevance.

Empirically, we have shown that this method has a robust behaviour with the test collections used. We can conclude that the performance of the relevance feedback is highly dependent on the collection. But, in general, we can say that the use of the convex combination in term nodes will help to improve the results. Also, the more sophisticated techniques (*d3* and *d4*) can be considered better than the simpler ones (*d2* and $d2^5$). The best results of these experiments are similar to those obtained by other models [5,10,11,14]. Anyway, these values are not totally comparable, because the experiments have been carried out with different models and under different experimental conditions.

The future works will be centred in the development of new relevance feedback methods for the BNR model based on the underlying concept of partial evidences, trying to improve the performance obtained with the methods introduced here and testing these method with bigger document collections like TREC.

Acknowledgements: The Spanish Ministry of Science and Technology under the Project TIC2000-1351 has supported this work.

Bibliography

1 Y.K. Chang, C. Cirillo, and J. Razon. *Evaluation of feedback retrieval using modified freezing, residual collection and test and control groups*, pages 355--370. Prentice Hall, Inc., Englewood Cliffs, NJ, 1971.

2 C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transaction on Information Theory*, 14:462--467, 1968.

3 W.B. Croft and H. R.Turtle *Text Retrieval and Inference*, pages 127--156. Paul Jacobs Ed., 1992.

4 L.M.de Campos, J. M. Fernández and J. Huete. Building Bayesian network-based information retrieval systems. In *11 Int. Works. on Database and Expert Systems Applications*: *2º Works. on Logical and Uncertainty Models for Information Systems (LUMIS)*, pages 543--552. Database and Expert Systems Applications, 2000.

5 L.M.de Campos, J.M. Fernández, and J. Huete. Relevance feedback in the Bayesian Network Retrieval model: An approach based on term instantiation. In. *Proc. of the Intelligent Data Analysis Confer*. 2001. (to appear).

6 L.M. de Campos, J.M. Fernández, and J.Huete. Query expansion in information retrieval systems using a Bayesian network-based thesaurus. In *Proc. of the 14 Uncertainty in Artificial Intelligence Conf.*, pages 53--60, 1998.

7 R. Fung and B.D. Favero. Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(2):42--57, 1995.

8. J.M. Fernández. Modelos de Recuperación de Información basados en Redes de Creencia. (In spanish) PhD thesis, University of Granada, 2001.

9 D.Ghazfan, M.Indrawan, and B.Srinivasan. Towards meaningful Bayesian networks for information retrieval systems. In *Proceedings of the IPMU'96 Conference*, pages 841--846, 1996.

10 D.Haines and W.B. Croft. Relevance feedback and inference networks. In *Proceedings of the 16 ACM -- SIGIR Conference on Research and development in information retrieval*, pages 2--11, 1993.

11 D.Harman. Relevance feedback revisited. In *Proceedings of the 15 ACM--SIGIR Conference on Research and development in information retrieval*, pages 1--10, 1992.

12 J.Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, San Mateo, California, 1988.

13 B.A. Ribeiro-Neto and R.R. Muntz. A belief network model for IR. *In Proceedings of the 19 ACM--SIGIR Conference on Research and Development in Information Retrieval*, pages 253--260. 1996.

14 G.Salton and C.Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science (JASIS)*, 41:288--297, 1990.

15 G.Salton and M.J. McGill. *Introduction to modern Information Retrieval*. McGraw-Hill, Inc., 1983.

16 A.Spink and R.M. Losee. Feedback in information retrieval. *Annual Review of Information Science and Technology (ARIST)*, 31:33--78, 1996.

17 H.R. Turtle. *Inference Networks for Document Retrieval*. PhD thesis, University of Massachusetts, 1990.

18 H.R. Turtle and W.B. Croft. Inference networks for document retrieval. In *Proceedings of the 13 ACM--SIGIR Conf. on Research and Development in Information Retrieval*, pages 1--24. 1990.