A New Information Retrieval Model Based on Possibilistic Bayesian Networks

Kamel Garrouch[1], Mohamed Nazih Omri[1], Amira Kouzana[2]
[1]MARS Research Unit
[1,2] Department of Computer Science
Faculty of sciences of Monastir
Monastir, 5000, Tunisia
{kamelg_2001, akouzana}@yahoo.fr, MohamedNazih.Omri@fsm.rnu.tn

***ABSTRACT*:** *This paper proposes a new Information Retrieval Model based on possibilistic Bayesian network. This model encodes the most important dependence relationships existing between terms. It focuses on local dependencies between terms within each document. The relevance of a document to a query is interpreted by two degrees: the necessity and the possibility. The necessity degree evaluates the extent to which a given document is relevant to a query, whereas the possibility degree evaluates the reasons of eliminating irrelevant documents. These two measures are also used for quantifying terms-terms links and terms-documents links. Experiments carried out on three standard collections have proven the efficiency of the proposed model.*

Keywords: Information Retrieval Model, Bayesian Network, Possibility Theory

Received: 3 February 2012, Revised 24 April 2012, Accepted 30 April 2012

1. Introduction

The field of information retrieval (IR) has been defined by Salton [2] as the subject concerned with the representation, storage, organization, and accessing of information items. In an information retrieval system (IRS) the tasks of indexing and retrieval of documents are performed by a component called Information retrieval model. In this paper we mainly focus our attention on information retrieval model that breaks the independence assumption between terms used to index the documents. These models use Bayesian networks to make an explicit representation of these dependence relationships.

Generally most Bayesian network based information retrieval models suffer from two drawbacks. (1) They do not consider inside document terms dependencies relationships. In fact they use a formula that analyses term's co-occurrence between each pair of terms in the whole collection of documents to quantify the degree to which tow terms are considered as dependant. This leads to a great number of linked terms and to weak values of dependencies. To overcome this problem we propose to make within document terms dependency analyses. Our idea is to link tow term nodes only if they are dependant inside one or many documents. (2) They do not take into account uncertainty inherent to natural language. In fact they make use of statistical measures to select terms that are able to represent documents, to quantify the strength of term dependence relationships and to select document that matches user's queries. These measures cannot express to witch extent one term can represent one document, to witch extent two terms

are dependant or to witch extent one document matches a user query. To overcome this problem we propose to use the possibility theory to quantify these three measures.

This paper provides a new Possibilistic Bayesian Network Retrieval Model (PBNRM). It uses an inside document analyse to extract term to term dependence relationships and uses possibility theory to quantify the importance of one term to one document, the strength of a link between two terms and the degree to which one document matches one user query.

The paper is organized as follows. In section 2 we briefly present different approaches to IR using Bayesian networks and possibility theory. In section 3 we describe in detail the proposed model. Section 4 shows experiments carried out on the TREC collection. The latter section 5 presents conclusions and future work.

2. Bayesian network models for IR

Bayesian networks are directed acyclic graphs (DAG) where nodes represent random variables and links qualitatively denote causality, relevance or dependency relationships between them. The dependence relationships between variables are described quantitatively by conditional probability tables (CPT) associated with each node. A CPT describes the conditional probability distribution of its corresponding variable given the possible combinations of states of all its parents.

Bayesian network can be considered as an efficient representation of a joint probability distribution that takes into account the set of independence relationships represented in the graphical component of the model. In general terms, given a set of variables $\{X_1 \ldots X_n\}$ and a Bayesian network G, the joint probability distribution in terms of local conditional probabilities is obtained as follows:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P(X_i \mid Part(X_i)) \qquad (1)$$

Where $part(X_i)$ is any combination of the values of parent set of X_i, in graph [4].

Many information retrieval models based on Bayesian network have been proposed in the literature [3], [5], [6], [7], [9], [11], [13], [14]. Two of these models take into account dependence relationships between term nodes. The first one was proposed by [7]. It is composed of two layers: the term layer and the document layer. In the first layer, term to term dependence relationships are represented by means of a polytree. The second layer stores all the documents from the collection as in figure 1. The query is considered as evidence that should be introduced into the system.

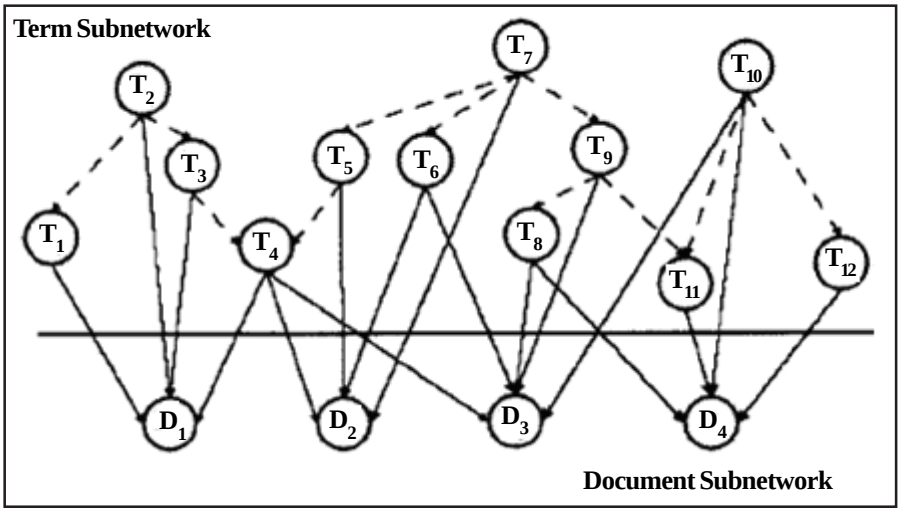

Figure 1. The Bayesian network retrieval model

The second one was proposed by [14]. This model is also composed of two layers but all the edges between terms are undirected as in figure 2. This is the main difference between this model and the first one.

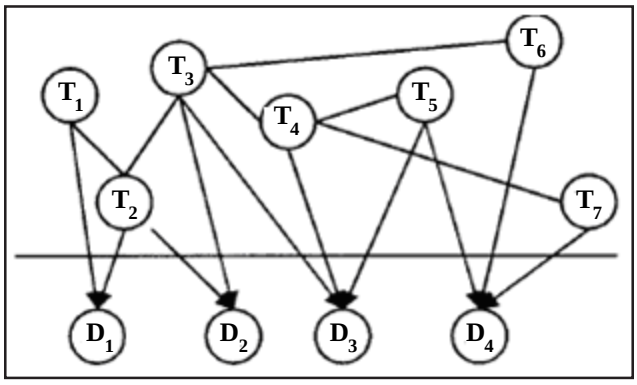

Figure 2. Probabilistic network retrieval model (PNRM)

These two models don't look for terms dependence relationships within documents and they ignore uncertainty inherent to natural language.

Another different IRM was recently proposed by [8]. Although it uses a possibilistic quantification of relevance of documents to user's query, it is based on the assumption of independence between term nodes.

3. Possibilistic Bayesian Information Retrieval model (PBNIRM)

In this model we propose a possibilistic network, where term to term/document dependence relationships are quantified by possibility and necessity measures. We propose also to use the possibilistic approach proposed by [8] to compute the degree of relevance of one document d_j to a user query Q. This model should be able to infer propositions like:

• It is more or less plausible (to a certain degree) that the document is relevant for the user need which is quantified by a degree of conditional possibility denoted by $\prod (d_j \mid Q)$.

• It is almost certain (in possibilistic sense) that the document is relevant to the query; which is quantified by a degree of conditional necessity denoted by $N (d_j \mid Q)$.

A low value of $\prod (d_j \mid Q)$ is meant to eliminate irrelevant documents (weak plausibility). If $\prod (d_j \mid Q) = 0$ it is certain that document d_j is not relevant to query Q. However $\prod(d_j \mid Q) = 1$does not imply that the document is relevant. It means only that nothing prevents the document from being relevant. The second evaluation focuses attention on what looks very relevant. Under a possibilistic approach, given the query, we are thus interested in retrieving necessarily relevant documents; or at least possibly relevant ones if there is none of the first kind.

In this model most important term to term dependence relationships are extracted by a within document investigation. This fact shall represent the first difference between our model and previous ones.

In order to present the PBNIRM, we shall first describe how we can determine the structure of the model; then we will present the assessment of the conditional distributions; and finally, we shall consider how the inference process is carried out.

3.1 Structure of the model

The structure of the model is composed of two layers: term layer and document layer. The first layer contains the set of terms $\boldsymbol{T} = \{T_i, i = 1 \ldots \boldsymbol{M}\}$ with M being the number of terms used to index the collection. The domain of an index term node T_i, is $\{t_i, \bar{t}_i\}$. $T_i = t_i$ refers to the presence of a term in a document and thus it is representative of the document to a certain degree. A non representative term, denoted by $\bar{t}_i$ is a term absent from (or not important in) the object.

Document layer contains the set of documents $D = \{D_j, j = 1 \ldots N\}$, N being the total number of documents. The domain of a

document node D_j is $\{d_i, \bar{d_i}\}$. $D_j = d_j$ (resp $\bar{d_j}$) means that a document is relevant query or not.

Arcs are directed from index term nodes to document node defining dependence relationships existing between index terms and documents. The query terms plays the role of evidence to propagate in the network. Terms appearing in a given user query are activated in the term layer and then propagated in the Bayesian network.

In this paper we propose to represent term to term dependence relationship by means of a polytree. The proposed network architecture, whose topology borrows from [7] model, appears in figure 1.

3.2 Structure construction

In our model, the term layer will be created using a polytree learning algorithm composed of three main steps detailed in the following.

3.2.1 Construction of the list of the most important dependencies

Considering the large number of terms involved in a given document collection, mining all dependence relationships is infeasible. We propose to keep in our model only the most important terms dependence relationships. The basic idea is to consider two terms as dependent if they exist together within one or many documents. To put this idea in practice we propose an algorithm (figure 3) that investigate the collection of documents document by document, create a list of dependence relationships form each document and keep only the dependence relationships that are frequent in one or in many documents.

To quantify the dependence relationship between two terms t_i and t_j within one document d_k we propose to use the following formula:

$$Dep_{d_k}(t_i,t_j) = \frac{tf_{ijk}}{\max\limits_{dl \in D}(tf_{ijl})} \quad (2)$$

Here

$$tf_{ijk} = \min_{dk}(tf_i, tf_j) \quad (3)$$

is the measure of the frequency value of the co-occurrence of the terms t_i and t_j in the document d_k.

$\max\limits_{dl \in D}(tf_{ijl})$ is the max between the term frequency (tf_j) of the terme t_i and the term frequency (tf_j) of the term t_j in the document collection.

Two complementary possibilistic measures are used to quantify the dependence relationships in the whole document collection: the possibility of dependence and the necessity of dependence. The possibility of dependence $Dep_{poss}(t_i, t_j)$ describes to witch extent two index terms t_i and t_j are possibly dependent in the document collection. We assume that two terms are possibly dependent if they have a height value of co-occurrence in many documents.

$$Dep_{poss}(t_i,t_j) = \begin{cases} Dep_{d_k}(t_i,t_j) \text{ if } Dep_{poss_{n-1}}(t_i,t_j) = 0 \\ \dfrac{(n-1) \times Dep_{poss_{n-1}}(t_i,t_j) + Dep_{d_k}(t_i,t_j)}{n} \ otherwise \end{cases} \quad (4)$$

$Dep_{poss}(t_i,t_j)$ is the final dependence measure of two terms t_i and t_j.

The necessity of dependence $Dep_{poss}(t_i, t_j)$ describe to witch extent two index terms t_i and t_j are necessarily (surely) dependent. We assume that two terms are necessarily dependent if they co-occur in many documents.

$$Dep_{poss}(t_i,t_j) = \frac{Ln(n_{ijk})}{\max(Ln(n_i), Ln(n_j))} \times Dep_{poss}(t_i, t_j) \quad (5)$$

Where

n_{ij} is the number of documents where t_i and t_j exist together.

n_i : is the document frequency of t_i.

n_j : is the document frequency of t_j.

```
1. Start with an empty List L of dependency relationships.
2. For each document in the docucument collection
   Compute dependence relationships value using Equation (2)
     If (it's the first document) then
        Add the most important dependence relationship to L
     else
        Update L using Equation (4)
     Return L.
```

Figure 3. Algorithm that extract the most important dependence relationships between terms

3.2. 2 Construction of the tree skeleton

If we assume that the computed dependency values are link weights in a graph, we can use the Prim's algorithm [10] to gets a maximum weight spanning tree (MWST), i.e. a tree where the sum of the weights of its links is maximum.

3.2.3 Orientation of the edges in the tree to make up a polytree

Once the skeleton is built, the last part of the learning algorithm deals with the orientation of the tree, getting as a result a polytree. In a head to head pattern $T_i \rightarrow T_k \leftarrow T_j$, the instantiation of the head to head node T_k should normally increase the degree of dependency between T_i and T_j, whereas in a non-head to head pattern such as $T_i \rightarrow T_k \leftarrow T_j$, the instantiation of the middle node T_k should produce the opposite effect, decreasing the degree of dependency between T_i and T_j.

So, we compare the degree of dependency between T_i and T_j after the instantiation of T_k, $Dep\ (T_i, T_j | T_k)$, with the degree of dependency between T_i and T_j before the instantiation of T_k, $Dep\ (T_i, T_j | \Phi)$, and direct the edges toward T_k if the former is greater than the latter. Finally, the algorithm directs the remaining edges without introducing new head to head connections.

To compute $Dep(\ T_i\ ,\ T_j \mid T_k\ )$, we also propose to use two possibilistic measures: the possibility of dependence $Dep_{poss}(t_i, t_j, t_k)$ and the necessity of dependence $Dep_{ness}(t_i, t_j, t_k)$ as follow :

$$Dep_{poss}(t_i, t_k, t_k) = \frac{Dep_{poss}(t_i, t_k) + Dep_{poss}(t_j, t_k)}{2} \tag{6}$$

$$Dep_{ness}(t_i, t_j, t_k) = \frac{Dep_{ness}(t_i, t_k) + Dep_{ness}(t_j, t_k)}{2} \tag{7}$$

Once the polytree has been learned, the last step to finish the retrieval model construction is to join each term node with its corresponding document node.

3.3 Parameters estimation

The next step after the creation of the structure of the bayesian network is to estimate the set of conditional possibility and necessity distributions. In our model we have three kinds of nodes: root term nodes, non root term nodes and leaf (document) nodes.

3.3.1 Root term nodes

Since a root node have no parents, we have to store the morginal possiblity of relevance $\prod(t_i)$ (respect the marginal necessity of relevance $N(t_i)$) and the marginal possiblity of being non-relevant, $\prod(\bar{t}_i)$ (respect the marginal necessity of being non-relevant $N(\bar{t}_i)$) defined by means of:

$$\prod(t_i) = 1 = \prod(\bar{t}_i) \tag{8}$$

$$N(t_i) = \frac{Ln\left(\frac{N}{n_i}\right)}{Ln(N)} \text{ and } N(\bar{t}_i) = 0 \tag{9}$$

With N being the number of document in the collection and n_i the number of document having t_i as index term.

3.3.2 Non-root term nodes

In this case, for each non-root term node T_i, with parents $Par(T_i)$ we need to estimate a set of conditional possibility distributions $\prod(T_i | Par(T_i))$ (respect conditional necessity distributions $N(T_i | Par(T_i))$), one for each possible combination of values that the parents of a node T_i can have. In our model this estimation is based on the Noisy-OR model, commonly used in probabilistic networks [1].

Given a term T_i having a set of parent terms, let Θ be the set of possible configurations θ of parent nodes of T_i and θ_j is the instantiation of only one term variable T_j in configuration θ. For instance if the node T_i is related to nodes $\{T'_1$ and $T'_2\}$:

$$\Theta = \{(t'_1, t'_2), (t'_1, \bar{t}'_2), (\bar{t}'_1, t'_2), (\bar{t}'_1, \bar{t}'_2)\}$$

instence θ_j in the configuration $\theta = (t'_1, \bar{t}'_2)$, is $\theta_1 = (t'_1, t'_2)$

For a node term t_i having n parents, every cause T'_j has a possibility P_j and a necessity n_j to be good enough to produce the effect in the case of absence of other causes.

$$P_j = \prod(t_i | \bar{t}'_1, \bar{t}'_2, ..., t'_j, ... \bar{t}'_{n-1}, \bar{t}'_n) = \prod(t_i | \theta_j)$$

$$n_j = N(t_i | \bar{t}'_1, \bar{t}'_2, ..., t'_j, ... \bar{t}'_{n-1}, \bar{t}'_n) = N(t_i | \theta_j)$$

The conditional possibility of a node term t_i given a configuration θ of his parents is computed by

$$\prod(t_i | \theta) = 1 - \prod_{\theta_j \in \theta} (1 - P_j) \tag{10}$$

$$\prod(\bar{t}_i | \theta) = 1 - \prod_{\theta_j \in \theta} (1 - P_j) \tag{11}$$

The conditional necessity of a node term t_i given a configuration θ of his parents is computed by:

$$N(t_i | \theta) = 1 - \prod_{\theta_j \in \theta} (1 - n_j) \tag{12}$$

$$N(\bar{t}_i | \theta) = 1 - \prod_{\theta_j \in \theta} (1 - n_j) \tag{13}$$

3.3.3 Document nodes

In this case, for each document node d_i, with a set of parents $Par(d_i)$ we need to estimate a set of conditional possibility distributions $\prod(d_i | \text{Par}(d_i))$ (respect conditional necessity distributions $N(d_i | Par(d_i))$), one for each possible combination of values that the

parents can have. Here *Par* (d_i) is set of term nodes used to index the document d_j.

In order to avoid the complexity of estimation due to the large number of terms by witch a document can be indexed, we have adapted the probability functions proposed by [7] to a possibility and necessity functions applicable in our case.

In the inference process, the possibility and necessity functions will compute the required conditional possibility and necessity values just at the moment when they are needed.

3.3.4 The retrieval engine: inference in the Possibilistic Bayesian Network Retrieval Model

Once the Bayesian network has been built, it can be used to retrieve documents that are relevant to a user query using the inference process. The last aim is to obtain two kinds of relevance measure for each document in the collection given a query: the necessity of relevance $N(D_j | Q)$ and the possibility of relevance $\prod(D_j | Q)$. The terms in a user query plays the role of a new piece of evidence provided to the system. This information will be propagated toward the document nodes, finally obtaining relevant document. The retrieved documents ranked first are necessarily relevant documents and then possibly relevant document.

Taking into account the number of nodes in our Bayesian network and the fact that 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. Therefore, we ought to look for a solution to carry out the inference in an acceptable time. Our proposal for solving this problem has been proposed by [7] named Propagation + Evaluation, and consists of a two-stage approximate propagation:

3.3.4.1 Exact propagation in the term layer, obtaining

$$\prod(t_i | Q), \forall T_i \text{ and } N(t_i | Q), \forall T_i .$$

Bearing in mind that the evidences will always be term nodes composing the query, we could use Fonc's propagation algorithm [12] in order to obtain the posterior possibility and necessity of each term node.

3.3.4.2 Evaluation of a possibility and necessity functions in the document nodes

We have to compute $\prod(d_i | Q)$ and $N(d_i | Q)$ for every document d_i in the collection of documents using the posterior possibilities and necessities obtained in the previous stage.

The computation of $\prod(d_i | Q)$ *and* $N(d_i | Q)$ can be carried out as follows:

$$\prod(d_i | Q) = \frac{\min_{\forall d_k \in D | v_{kQ} > 0} (v_{kQ})}{v_{jQ}} \tag{14}$$

$$N(d_i | Q) = \frac{\min_{\forall d_k \in D | w_{kQ} > 0} (w_{kQ})}{w_{jQ}} \tag{15}$$

Here

$$v_{jQ} = \frac{\prod_{t_i \in d_j | \pi(t_i | Q) > 0} ntf_{ij} . \pi(t_i | Q)}{n_{jQ}} \tag{16}$$

$$w_{jQ} = \frac{\prod_{t_i \in d_j | N(t_i | Q) > 0} \Phi_{ij} . N(t_i | Q)}{n_{jQ}} \tag{17}$$

where

$$nft_{ij} = \frac{tf_{ij}}{\max_{t_k \in d_j} (tf_{ik})} \tag{18}$$

And

$$\Phi_{ij} = \frac{\text{In}\left(\frac{N}{n_i}\right)}{\text{In}(N)} \times ntf_{ij} \qquad (19)$$

Here, n_{jQ} is the number of terms shared by the document d_j and the query Q, N is the number of document in the collection, n_i is the number of document having the term t_i as index term and tf_{ij} is the frequency of the term t_i in the document d_j.

Φ_{ij} is the necessity measure of relevance of a given term t_i to a document d_j and ntf_{ij} is the possibility measure of relevance of given term t_i to a document d_j.

4. Measuring the performance of the model: experiments and results

Several experiments have been conducted to evaluate the performance of the proposed model.

We have applied the PBNIRM to three well-known test document collections: ADI, CACM, and CRANFIELD. The main characteristics of these collections with respect to the number of documents, terms and queries are shown in Table 1.

Collection	Documents	Terms	Queries
ADI	82	828	35
CRANFIELDS	1398	3857	225
CACM	3204	7562	64

Table 1. characteristics of document collections used for implementation of The Proposed model

For the ADI collection, our model is compared with two other models for information retrieval, including BNRM model [7] and PNRM model [8]. For the two other collections, our model is compared only to PNRM.

The model's comparison is based on recall and precision estimates [2]. The first measures the ability of the IR system to present all the relevant documents (recall = number of relevant documents retrieved/number of relevant documents). The second,

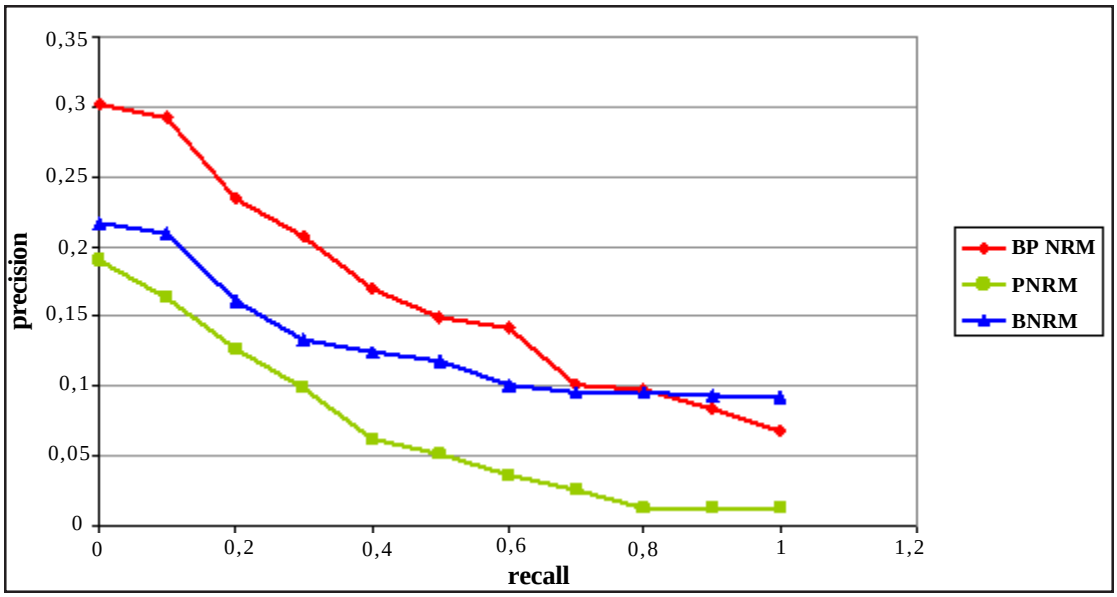

Figure 4. 11-point average precision curves for ADI document collection

precision, measures its ability to present only the relevant documents (precision = number of relevant documents retrieved/ number of documents retrieved). In our case, models are compared using 11-point average precision [15]. For the ADI document collection, comparison result is depicted on figure 4. It is clear that our model outperform the two other models from the beginning until the 0.8 point of recall. After that the BNRM model shows a better performance.

For the CACM and CRANFIELDS document collections, the performance of our model is only compared to the PNRM performance. Figure 5 and Figure 6 shows the results of these comparisons. For the both collections, the curves shows that our model outperform PNRM model.

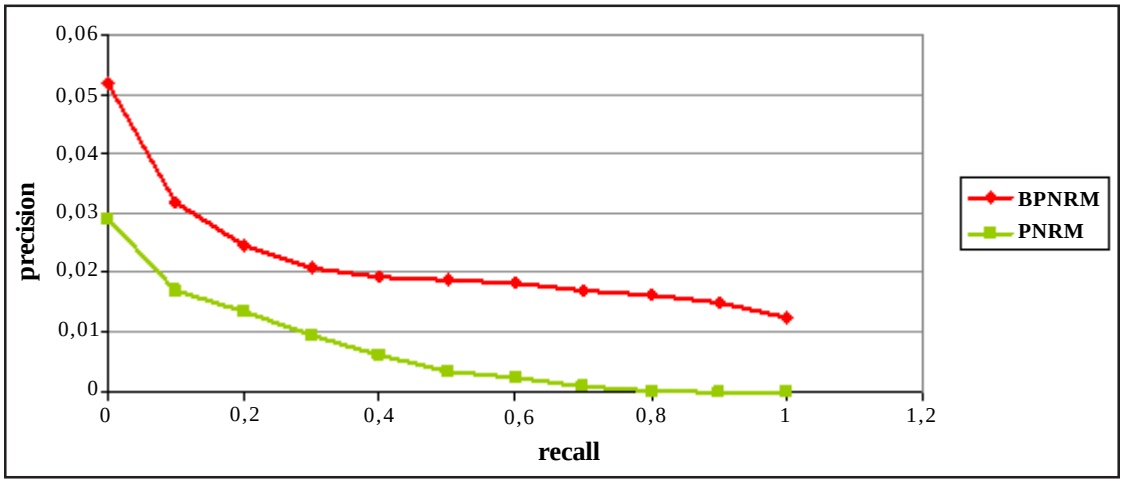

Figure 5 . 11-point average precision curves for CACM document collection

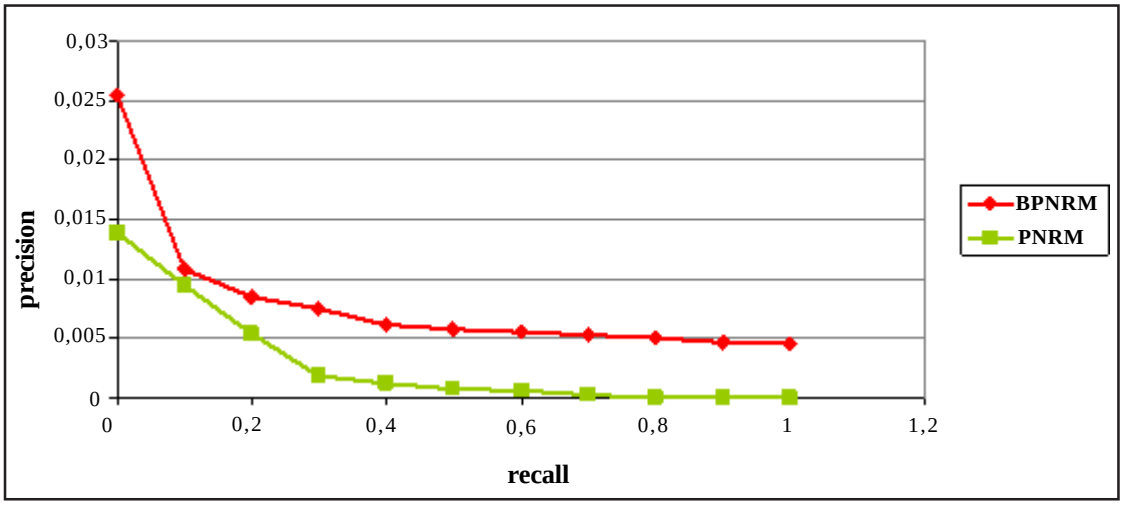

Figure 6. 1 : 11-point average precision curves for CRANFIELDS document collection

5. Conclusions and future work

In this paper, we have presented a new information retrieval model that mix the Bayesian network representation of document collection with a possibilistic quantification of both relevance and dependence relationships measures.

The first objective of this model was to focus only on the most important dependence relationships between terms. For that we have developed an algorithm that looks for dependence relationships within documents. Our second objective was to take into account fuzziness inherent to natural language. For that we have proposed a set of possibilistic formula to compute the necessity and possibility measures for both relevance of documents to user's queries and dependence relationships between index terms.

The performance of the proposed model was compared to the performance of two existing models. Primary experimental results showed that it outperform other models on three document collections. Despite these fact, we steel have to do more experiments on other document collections to be able to conclude about the performance of our model.

Another of our future lines of research that we are considering is to develop new mechanisms to extract term dependence relationships based on semantic analysis of documents.

References

[1] Onisko, A., Druzdzel, M. J., Wasyluk, H. (2001). Learning Bayesian network parameters form small data sets: application of Noisy-OR gates, *International Journal of Approximate Reasoning*, p. 165-182.

[2] Salton, G., McGill, M. J. (1983). Introduction to Modern Information Retrieval, McGraw-Hill, New York.

[3] Turtle, H., Croft, W. (1996). Inference networks for document retrieval, *In*: Proc. of the International ACM-SIGIR Conference, p. 1–24.

[4] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann.

[5] Cai, K., Chen, C., Bu, J. (2009). Exploration of term relationship for Bayesian network based sentence retrieval, *Pattern Recognition Letters*, p. 805–811.

[6] de Campos, L. M., Fernandez, J. M., Huete, J. F. (1998). Query expansion in information retrieval systems using a Bayesian network-based thesaurus, *In*: Proceedings of the 14th Uncertainty in Artificial Intelligence Conference, p. 53–60.

[7] de Campos, L. M., Fernandez, J. M., Huete, J. F. (2003). The BNR model: Foundations and Performance of a Bayesian network-based Retrieval Model, *International Journal of Approximate Reasoning***,** p. 265-285.

[8] Boughanem, M., Brini, A., Dubois, D. (2009). Possibilistic networks for information retrieval, *International Journal of Approximate Reasoning*, p. 957-968.

[9] Maragoudakis, M. (2007). A Bayesian network Model for Information Retrieval from Greek Texts,19th IEEE International Conference on Tools with Artificial Intelligence.

[10] Christofides, N. (1975). Graph Theory, an Algorithmic Approach, Academic Press, New York. (prim)

[11] Calado, P., da Silva, A.S., Laender, A., Ribeiro-Neto, B.A., Vieira, R.C. (2004). A Bayesian network approach to searching Web databases through keyword-based queries, Information Processing and Management, p. 773–790.

[12] Fonck, P. (1994). Réseau d'Inférence pour Raisonnement Possibiliste. Thèse de de Doctorat en Informatique. Université de Liège - Faculté des sciences. Liège.

[13] Fung, R., Favero, B. D. (1995). Applying bayésien Network to information retrieval, *Communications of the ACM*.

[14] Dongyu , S., Zhengwei, O., Cheng, F., Jinyuan, Y. (2004). An information retrieval model Based on probabilistic network, *In*: Proceedings of the IEEE International Conference on Services Computing.

[15] Yang, Y. (1999). An Evaluation of Satatistical Approaches to Text Categorization. Information retrieval, p. 69-90.