A Machine Learning Model for Information Retrieval with Structured Documents

Benjamin Piwowarski and Patrick Gallinari

LIP6 – Université Paris 6, 8, rue du capitaine Scott, 75015 Paris, France
{bpiwowar,gallinar}@poleia.lip6.fr

Abstract. Most recent document standards rely on structured representations. On the other hand, current information retrieval systems have been developed for flat document representations and cannot be easily extended to cope with more complex document types. Only a few models have been proposed for handling structured documents, and the design of such systems is still an open problem. We present here a new model for structured document retrieval which allows to compute and to combine the scores of document parts. It is based on bayesian networks and allows for learning the model parameters in the presence of incomplete data. We present an application of this model for ad-hoc retrieval and evaluate its performances on a small structured collection. The model can also be extended to cope with other tasks such as interactive navigation in structured documents or corpus.

1 Introduction

With the expansion of the Web and of large textual resources like e.g. electronic libraries, appeared the need for new textual representations allowing interoperability and providing rich document descriptions. Several structured document representations and formats were then proposed during the last few years together with description languages like e.g. XML. For electronic libraries, Web documents, and other textual resources[1], structured representations are now becoming a standard. This allows for richer descriptions with the incorporation of metadata, annotations, multimedia information, etc. Document structure is an important source of evidence, and in the IR community some authors have argued that it should be considered together with textual content for information access tasks [1]. This is a natural intuitive idea since human understanding of documents heavily relies on their structure. Structured representations allow capturing relations between document parts as it is the case for books or scientific papers. Information retrieval engines should be able to cope with the complexity of new document standards so as to fully exploit the potential of these representations and to provide new functionalities for information access. For example, users may need to access some specific document part, navigate through complex documents or structured collections; queries may address both metadata and textual content. On the other side, most current information retrieval systems still rely on

[1] See for example the DocBook standard [18]

P. Perner and A. Rosenfeld (Eds.): MLDM 2003, LNAI 2734, pp. 425–438, 2003.

simple document representations like e.g. bag of words and completely ignore the richer information allowed by structured representations.

Extending information retrieval systems so that they can handle structured documents is not trivial. Many questions for designing such systems are still open, e.g. there is no consensus on how to index these documents, nor on the design of efficient algorithms or models for performing information access tasks. Furthermore, this need being quite recent there is a lack of textual resources for testing and comparing existing systems and prototypes. The goal of this paper is to propose a new generic system for performing different IR tasks on collections of structured documents. Our model is based on bayesian networks (BN), probabilistic inference is used for performing IR tasks, BN parameters are learned so that the model may adapt to different corpora. In the paper, we consider ad-hoc retrieval and focus on the BN model. For simplification, we will only consider the case of hierarchical document structures, i.e. we make the hypothesis that documents may be represented as trees. This encompasses many different types of structured documents. For all other cases (e.g. Web sites), this will be an approximation of the reality which allows to keep inference model complexity down to a reasonable level.

The paper is organized as follows: Sect. 2 makes a review of the literature on structured documents and IR, Sect. 3 describes a general BN model for information retrieval and a particular instance of this model we have developed for document part retrieval in a web site, the last section discusses experiments on a test collection which has been built using the Hermitage museum web site.

2 State of the Art

One of the pioneer work on document structure and IR, is that of Wilkinson [24] who attempted to use the document division into sections of different types (abstract, purpose, title, misc., ...) in order to improve the performances of IR engines. For that he proposed several heuristic for weighting the relative importance of document parts and aggregating their contributions in the computation of the similarity score between a query and a document. Doing this way, he was then able to improve a baseline IR system.

A more recent and more principled approach is the one followed by Lalmas and co-workers [10]–[13]. Their work is based on the theory of evidence which provides a formal framework for handling uncertain information and aggregating scores from different sources. In this approach, when retrieving documents for a given query, evidence about documents is computed by aggregating evidence of sub-document elements. Paragraph evidence is aggregated to compute section evidence which in turn will allow computing the document relevance. They also make use of confidence measures which come with the evidence framework in order to weight the importance of document part score in the global aggregated score. The more confident the system is in a document element, the more important this element will be in the global score. In [12], tests were performed on a small home made collection.

Another important contribution is the HySpirit system developed by Fuhr et al. [5]. There model is based on a probabilistic version of datalog. When complex objects like structured documents are to be retrieved, they use rules modeling how a document part is accessible from another part. The more accessible this part is, the more will it will influence the relevance of the other part.

A series of papers describing on-going research on different aspects of structured document storage and access, ranging from database problems to query languages and IR algorithms is available in the special issue of JASIST [1] and in two SIGIR XML-IR workshops[2]. There is also the recent INEX initiative for the development and the evaluation of XML IR systems. The first meeting of INEX was held in December 2002 and proceedings are available on line[3].

Since Inquery [2],[22], bayesian networks have been shown to be a theoretically sounded IR model, which allows to reach state of the art performances and encompasses different classical IR models. The simple network presented by Croft, Callan and Turtle computes the probability that a query is satisfied by a document[4]. This model has been derived and used for flat documents. Ribeiro and Muntz [20] and Indrawan et al. [6] proposed slightly different approaches also based on belief networks, with flat documents in minds. An extension of the Inquery model, designed for incorporating structural and textual information has been recently proposed by Myaeng et al. [16]. In this approach, a document is represented by a tree. Each node of the tree represents a structural entity of this document (a chapter, a section, a paragraph and so on). This network is thus a tree representation of the internal structure of the document with the whole document as the root and the terms as leaves. The relevance information goes from the document node down to the term nodes. When a new query is processed by this model, the probability that each query term represents the document is computed. In order to obtain this probability, one has to compute the probability that a section represents well the document, then the probability that a term represents well this section and finally the probability that a query represents well this term. In order to keep computations feasible, the authors make several simplifying assumptions. Other approaches consider the use of structural queries (*i.e.* queries that specifies constraints on the document structure). Textual information in those models is boolean (term presence or absence). Such a well known approach is the Proximal Nodes model [17]. The main purpose of these models is to cope with structure in databases. Results here are boolean: a document match or doesn't match the query.

Corpus structure has also been used for categorization, mainly for improving performance when dealing with small quantities of positive examples in the training sets. Some authors make use of specialized classifiers for each category [3],[8], others introduce constraints between different sets of parameters [14]. These investigations have shown that taking into account some type of structure present in the dataset may prove beneficial for the retrieval performances.

Our work is an attempt to develop a formal modeling of documents and of inferences for structured IR. In this sense, our goal is similar to that of Lalmas et al. [10]. Our formal modeling relies on bayesian networks instead of evidence theory in [10] and thus provides an alternative approach to the problem. We believe that this approach allows casting different access information tasks into a unique formalism, and that these models allow performing sophisticated inferences, e.g. they allow to compute the relevance of different document parts in the presence of missing or uncertain

[2] http://www.haifa.il.ibm.com/sigir00-xml/ and http://www.info.uta.fi/sigir2002/html/ws6.htm.

[3] See http://qmir.dcs.qmw.ac.uk/XMLEval.html for more details.

[4] More precisely, the probability that the document represents the query

information. Compared to other approaches based on BN, we propose a general framework which should allow adapting to different types of structured documents or collections. Another original aspect of our work is that model parameters are learned from data, whereas none of the other approaches relies on machine learning. This allows adapting the model to different document collections and IR tasks.

3 A Model for Structured Information Retrieval

We first describe below (Sect. 3.1) how Bayesian networks can be used to model and retrieve documents or document parts, we then present the general lines of our model (Sect. 3.2) and describe in details the particular implementation we have been using for our experiments (section 3.3).

3.1 Bayesian Networks for Structured Documents Retrieval

Bayesian networks [7],[9],[15],[19] are a *probabilistic framework* where conditional independence relationships between random variables are exploited, in order to simplify or/and to model decision problems. They have been used in different contexts, with many real world applications with an emphasis on diagnosis problems. For textual data, the seminal work of Turtle & Croft [22] raised interest in this framework, and since that, simple BN have been used for IR tasks (see Sect. 2). Bayesian networks provide a formal framework which allows representing the relations between document parts as conditional dependence (or independence). They also allow performing sophisticated inferences on the relevance of document parts for a given query and allowing to model different combinations of evidence. Note that strong simplifying assumptions are needed with textual data, since documents are represented in very large characteristic spaces.

Let us now present using a simple illustrative case how BN could be used to model and perform inference on structured documents. We will suppose that for retrieving documents, *P(d/q)* is used as the relevance score of document d with respect to query *q*.

Consider the simple document of Fig. 1a, composed of two sections and three paragraphs. A simple way to take into account the structure of *d* is to decompose the score *P(d/q)* as follows:

$$P(d/q) = \sum_{s_1,s_2,p_1,p_2,p_3} P(d,s_1,s_2,p_1,p_2,p_3/q)$$

Where *s* and *p* are random variables associated respectively to sections and paragraphs. Suppose now that each random variable (node) in this network can take two values (R = relevant/ ¬R = irrelevant with respect to a given query). To compute the joint probability values $P(d,s_1,s_2,p_1,p_2,p_3)$. We need 2^6-1 values for this simple document, and summations with up to 2^5 terms in order to compute $P(d/q)$, $P(s_1/\mathrm{q})$, ... This is clearly infeasible with documents with many structural entities.

BN make use of conditional independence assumptions in order to simplify these computations. Let us proceed with our example.

In our model, BN are used to represent documents, one specific BN being associated to each document. Each node of a BN document model is a boolean variable which indicates whether or not the information associated to this node is relevant to the query. The structure of the BN is directly derived from the document structure. Different BN may be considered for modeling a document. Figures 1b,c show two different models for the simple document of Fig. 1a.

Let us first focus on the *d*, *s* and *p* nodes. Figure 1b represents a model where the relevance of a part is conditioned on the relevance of its subparts, section relevance is computed from the relevance of its paragraphs and document relevance from its sections. Figure 1c represents a model where the dependences are in the reverse order, section relevance depends on document relevance and paragraph relevance depends on section relevance. Both models are valid, but have different semantics.

Variables t_i represent relevance information on textual data, i.e. this is where the text comes into the model. They can be connected different nodes in the BN, examples are given in Figs. 1b,c. In Fig. 1b, textual evidence has been inserted at the paragraph level, whereas in Fig. 1c, it has been considered that textual information is present at any node in the BN. The choice of a particular model depends on the targeted task and on practical considerations (e.g. the complexity of the computation)[5].

The relevance of a document or document part is computed using the conditional independence assumptions encoded in the BN. As an example, the probability of relevance of Sect. 1 with the model 1c is given by:

$$P(s_1)=\sum_{d,t_1,t_2} P(d|t_1)P(t_1)P(s_1|d,t_2)P(t_2)\ ,$$

where $P(t_i)$s are prior probabilities and the summation is over the R, ¬R values of the *d* and t_i variables. With such a model, complexity drops from $O(2^N)$ where N is the number of random variables to $O(N2^{N\max})$ where $N_{\max}$ is the maximal number of parents for a given random variable in the Bayesian network.

3.2 General Model

We will now describe our BN model and in the next section, detail the particular implementation we have used for our experiments. In our bayesian network, there are two different kinds of variables, those associated with the relevance of a structural part and those associated with the relevance of textual data. Both are binary and take values from the set $\{R$ = Relevant to the query, $\neg R$ = Irrelevant to the query $\}$. The former are computed using inference in the bayesian network, and the latter may be computed by any probabilistic model as described below and are *a priori* probabilities for our BN. The BN thus propagates relevance probabilities from one node to its descendants. Although the binary scale may appear restrictive, it is used in most information retrieval models since it allows for a limited computational cost.

[5] The model in Fig. 1c can be used for passage retrieval or page retrieval in a web site as it will be shown in our experiments, whereas the other one (b) is more directed towards document retrieval where information about relevance of paragraphs is used to compute the document relevance

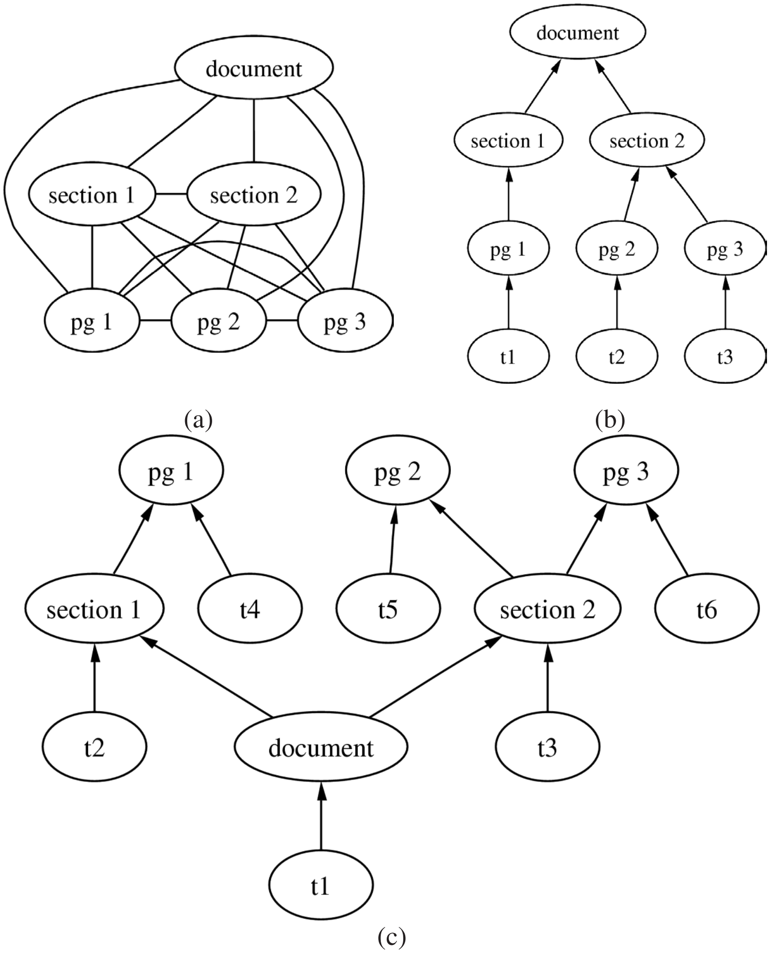

Fig. 1. Three different models of the same document. (a) All parts are dependent, (b) and (c) are two different models for conditional independences in the document network

Let T be a variable associated with textual data. In our experiments, text relevance *prior* probabilities are computed by the Okapi model [23] as follows:

$$P(T) = cosine(T,q)$$

Okapi gives scores between 0 and 1 which are used here *as* probabilities. Okapi has been used for simplicity since it is also our baseline model in the experiments, but other models could be used as well for computing priors. Note that the same BN framework could also be used for multimedia documents provided the relevance probability of content elements is computed by a model adapted to the media type of this element.

For random variables associated with structural parts, we do not use *prior* probabilities but conditional probabilities such as:

$$P(A \text{ relevance}|B_1,...,B_n \text{ relevance}),$$

where the B_i are the parents of A in the bayesian network (Fig. 2). In the model used in our experiments, these conditional probabilities are stored in a simple table for each node. They are learned from data and the probability estimates are the parameters of our BN model, they will be denoted Θ in the following.

This model operates in two modes, training and retrieval, which we now describe.

Training

In order to fit a specific corpus, parameters are learnt from observations using the Estimation Maximization (EM) algorithm. An observation $O^{(i)}$ is a query with its associated relevance assessments (document/part is relevant or not relevant to the query). EM [4] optimizes the model parameters Θ with respect to the likelihood L of the observed data $L(O,\in) = \log P(O/\in)$ where $O = (O^{(1)}, \ldots, O^{(N)})$ are the N observations.

Observations may or may not be *complete*, *i.e.* relevance assessments need not to be known for each document part in the BN in order to learn the parameters. Each observation $O^{(i)}$ can be decomposed in two sets of variables $O^{(i)} = (E^{(i)}, H^{(i)})$ where

- $E^{(i)}$ corresponds to structural entities for which we know whether they are relevant or not, *i.e.* structural parts for which we have a relevance assessment. $E^{(i)}$ is called the evidence and is a vector of 0/1 in our model.
- $H^{(i)}$ corresponds to hidden observations, i.e. all other nodes of the BN. Note that variables T associated with textual relevance (Okapi *a priori*) are in this set.

Instead of optimizing directly L, EM optimizes the auxiliary function

$$L' = \sum_{i=1}^{N} \sum_{H^{(i)}} Q(H^{(i)}) \log P(E^{(i)}, H^{(i)} / \Theta)$$

with

$$Q(H^{(i)}) = P(H^{(i)} / E^{(i)}, \Theta)$$

EM attempts to find the parameters maximizing the probability to observe the relevance judgments given in the training set. Optimizing L' is performed in two steps. The first one is the *Expectation* step in which we optimize L' with respect to Q -i.e. Q is estimated while Θ is kept fixed. This corresponds to a simple inference step in our bayesian network. The second one is the *Maximization* step where L' is optimized with respect to Θ. This step is performed by constraint optimization. In the first section, we gave the update formula used for our specific application.

Retrieval

For retrieval, when a new query Q has to be answered, *a priori* probabilities are first computed. For textual variables T, this is done using baseline models as described above; for non textual variables, ad-hoc priors will be used. After that, joint probabilities needed for scoring the document can be computed using the learned conditional probabilities and the priors. This is done using an inference algorithm suited to our bayesian network. Documents with highest scores are then presented to the user. If we are interested into retrieving document parts which correspond to BN nodes, instead of whole documents, we can proceed in the same way.

1. Prehistoric art
   a. Paleolithic art
      i. Female figurine
      ii. Anthropomorhpic figurine
      iii. ...
   b. Neolithic art
   c. ...
2. Antiquity
   a. Ancient Italy
   b. ...
3. ...

Fig. 2. A Web site viewed as a structured document

3.3 Instantiating the Model for Document Part Retrieval

We used an instance of this general model for retrieval on a hierarchically organized Web site: a part of the Hermitage museum web site in St Petersburg. This test collection was kindly given to us by M. Lalmas [12] and is one of the very few structured collection of documents where queries and corresponding relevance assessments are provided. This is a single Web site structured in 441 pages. Our goal here, similar to that of [12] is to retrieve pages relevant to a query, such pages are supposed to provide good entry points to relevant material on the site. For this particular application, we consider the Web site as a single document, hierarchically structured as shown in Fig. 3.

The structure of our network is directly inspired from the structure of the Web site. The relevance of each page depends on the relevance of its text and the relevance of the page that has a link to it. For example, on figure 4, "P_2 relevance" given "P_1 relevance" and "P_2 text relevance" (T_2) is independent of other variables. In other words, "P_2 relevance" is *determined* by its "text relevance" and "P_1 relevance"

As for the conditional probability tables associated to the nodes of this model, we will distinguish 2 cases.

For all nodes except the root P_1 the $P(P_i/T_i,P_i$ parent$)$ are learned from the data via EM as described below.

For the main page P_1 there is no other source of information than text to assess a relevance judgment for the main page. P_1 relevance is then:

$$P(P1) = P(P_1/T_1 = R)\ P(T_1 = R) + P(P_1/T_1 = \neg R)\ P(T_1 = \neg R)$$

With the conditional probabilities set as follows:

$$P(P_1/T_1 = R) = 1 \text{ and } P(P_1/T_1 = \neg R) = 0$$

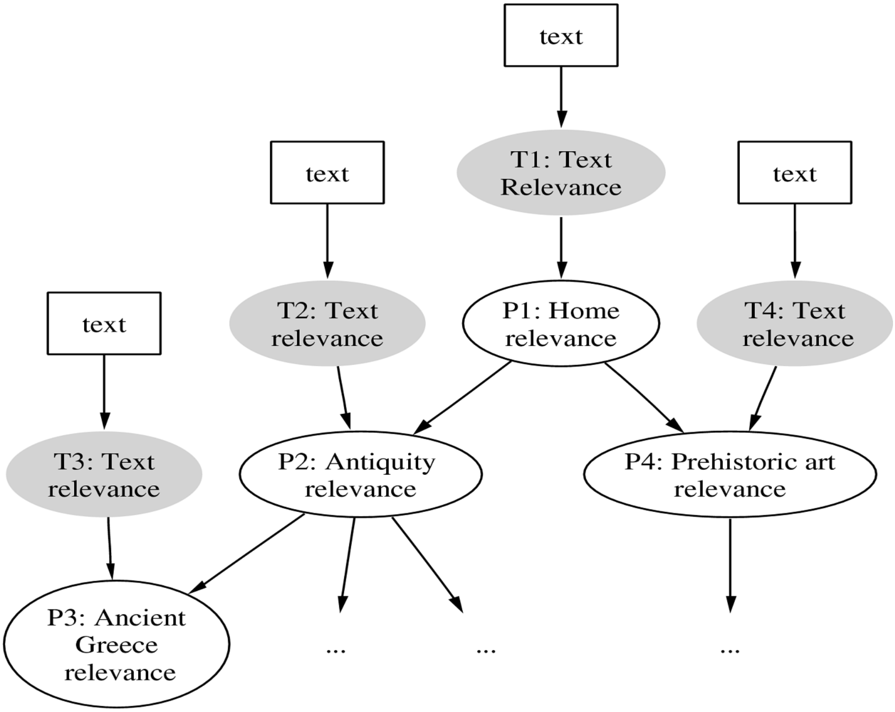

Fig. 3. A part of the network used for the Hermitage Web site. In this example, four different pages in three different levels are shown (home page, antiquity, ancient Greece, prehistoric art). The Hermitage Web site contains 441 pages.

In order to reduce the number of parameters to be learned, different nodes in the BN do share their parameters. For this application, all nodes within one layer do share the same conditional probability table. Let $\theta_{a,b,c}^{(l)}$ denote the conditional probability parameters for layer *l*.

Under the above shared parameters assumption, we have for the network of Fig. 4:

$$\theta_{a,b,c}^{(2)} = P(P_2{=}a|T_2{=}b,P_1{=}c) = P(P_4{=}a|T_4{=}b,P_1{=}c)$$

Where *a*, *b* and *c* may take the values R or ¬R respectively for the current node P_2 or P_4 , the text node T_2 or T_4 and the parent node P_1. Note that except for l = 1, $\theta^{(l)}$ is an array with 2^3 real values. Besides providing more robust estimators, parameter sharing allows to learn general relationships between structural entities of the Web site. We will then learn *how* the relevance of the homepage influences the relevance of department pages, and *how* relevance of department pages influences the relevance of a specific collection, and so on. Additional constraints may be imposed as described in the experiments below (Sect. 4).

Retrieving Pages: Inference

When retrieving web pages from the site, we compute the relevance P(P_i) for each page:

$$P(P_i) = \sum_{\{p_k, t_k\}_{k \neq i}} P(P_1, \dots, P_M, T_1 \dots, T_M)$$

where M is the number of nodes in the BN, and the summation is taken over all combinations of the binary values (R, ¬R) for all variables except P_i. This formula factorizes according to the conditional independence structure of the network:

$$P(P_i) = \sum_{\{p_k, t_k\}_{k \neq i}} \prod_{j=1..M} P(P_j / P_j \text{ parent}, \mathrm{T_j}) P(T_j)$$

It can be efficiently computed if the network structure is simple (inference cost is linear with respect to the number of web pages), as it is the case with our experiments.

Learning: EM Algorithm

For learning, EM algorithm leads to the following update rule for the model parameters:

$$\theta^{(l)}_{a,b,c} \leftarrow \frac{1}{K_{i,b,c}} \sum_{i=1}^{N} \sum_{P \text{ in level } l} \frac{P(E^{(i)}, P = a, P \text{ text} = b, P \text{ parent} = c)}{P(E^{(i)}, P \text{ text} = b, P \text{ parent} = c)}$$

where N is the number of observations, and the second sum is restricted to pages where P($E^{(i)}$,*P text= b, P parent=c*)≠0. K is a normalizing constant that insures that probabilities sum to 1:

$$K_{l,b,c} = \theta^{(i)}_{\text{relevant},b,c} + \theta^{(i)}_{\text{not relevant},b,c}$$

4 Experiments

The test collection contains 441 documents and 15 queries that were randomly split into a training and a test set. For comparison, we used as a baseline model Okapi [23] to compute the relevance of the web pages. Okapi is one of the best known, top ranking, IR model for ad-hoc retrieval on flat documents, with this model, corpus structure is ignored. We also used the model proposed by Lalmas and Moutogianni as described in [12], this model takes into account the corpus structure.

Each document is a single page of the Hermitage web site. The maximum depth (largest distance between the main page and any other page) of this site is 6 and there is an average number of children of 1 (ranging from 0 to 16).

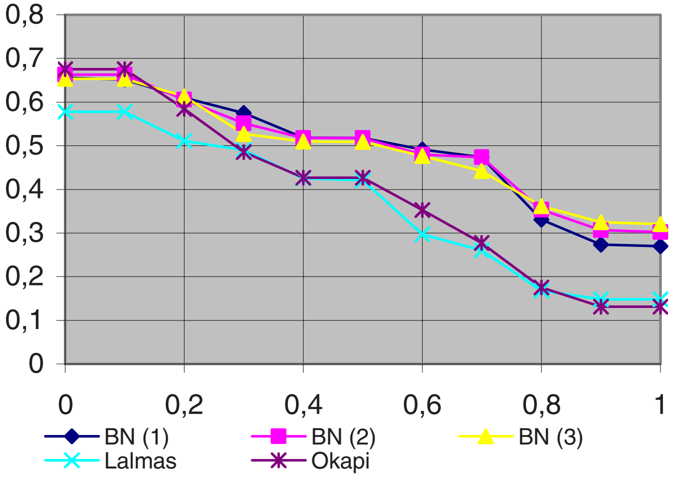

Fig. 4. Precision-recall curve with Okapi, Lalmas & Moutogianni model and our model (BN).

Different experiments were performed with different settings for the model parameters. Our model is denoted "BN (*depth*)" where d*epth* corresponds to the maximum number of different conditional probabilities tables we learn. For example, with depth 1, we constrain $\theta^{(i)}=\theta^{(j)}\ \forall\ i, j$. (*i.e.* only one set of parameters is learned for the whole network), with a depth of 2 we constrain $\theta^{(i)}=\theta^{(j)}$.for $i, j \geq 2$ and so on.

In our experiments, we performed 9 EM steps in order to learn the parameters: since our database is small, the EM algorithm converged very fast.

The first experiment () compares three different models: Okapi, Lalmas' and ours. We performed cross-validation on the dataset. The query set was divided into 5 equal parts (3 queries each), we performed 5 different runs, each time using 3 queries for testing and 12 for training. Results were averaged over all runs. This allows our bayesian model to optimize its parameters with a sufficient number of queries while using all the queries for the evaluation.

In Fig. 4, *recall* is the ratio between the number of retrieved relevant documents and the total number of relevant documents. *Precision* is the ratio between the number of retrieved relevant documents and the number of retrieved documents.

All experiments show that for this dataset, the BN model does improve the baseline Okapi. This is a nice result since Okapi is a very efficient IR system which has been tuned over years. It also performed better than Lalmas & Moutogianni model in our experiments. The increase is significant as can be seen on the figure. For all experiments, the three BN variants do offer similar performances, the BN with a depth of 3 being slightly better. Overfitting was not observed even when using more parameters and performing more EM steps.

Table 1. Effect of feedback Table 1 gives mean precision, R-precision and break-even measures when using relevance information using the 5, 10 and 15 first documents returned by our BN model. For one query, *R-precision* is the precision at rank R where R is the total number of relevant document. *Mean Precision* is the mean of precisions over all retrieved documents. *Break-even point* is the point in the precision/recall curve where precision is equal to recall. All values in the table are averages over all test queries.

# Relevance assessments	5		10		15	
# queries	13		10		9	
Feedback	Yes	No	Yes	No	Yes	No
Mean precision	0.46	0.32	0.38	0.17	0.39	0.16
R-precision	0.43	0.25	0.37	0.10	0.36	0.13
Break-even point	0.47	0.34	0.39	0.19	0.40	0.16

In a second series of experiments, we introduced feedback in the BN model. We first use the BN model to rank documents with respect to a query *q*. We use the known relevance assessments for the top *n* retrieved documents as evidence for the BN. In a practical situation, this feedback will be provided by the user. Let $d'_1,\ldots,d'_n$ denote the top *n* ranked documents.

We then compute for any document not in the top *n* its relevance for *q*, knowing the relevance of the *d'*s. Stated otherwise, with feedback, we compute for any document *d* not in the top *n* $P(d/q, d'_1,\ldots,d'_n)$ instead of $P(d/q)$ without feedback. In the BN, this means that inference is performed with a known value (*R* or *I*) for the variables corresponding to $d'_1,\ldots,d'_n$. We then perform an evaluation[7] using cross-validation as above. $P(d/q, d'_1,\ldots,d'_n)$ represents the distribution of the probabilities of relevance *knowing that the user found d' relevant to his/her need.*

This experiment measures the potential of the model for incorporating evidence (feedback) during a session. It also measures in some way the ability of the system to help interactive navigation through the site: when the user provides feedback on some documents, the system takes this information into account and outputs a list of new documents.

[7] Note that we removed the query from the evaluation set when all relevant documents were in the top *n* documents, since looking for other documents had no sense. We thus indicate in Table 1 how many queries were used for each evaluation.

When we increase the value of n, fewer documents remain in the test set and he performance measures decrease. The values above should be compared for a given value of n. It shows a clear improvement when using feedback. This demonstrates the ability of the model to incorporate feedback in a natural way and to perform tasks such as interactive navigation in a structured corpus.

5 Conclusion

We have described a new model for performing IR on structured documents. It is based on BN whose conditional probability tables are learned from the data via EM. Experiments on a small structured document collection have shown that this model can significantly improve performance compared to a state of the art “flat” information retrieval system like Okapi. These results show that even simple structures like the one we have been dealing with are a valuable source of information for retrieving documents. Of course, further experiments are needed in order to assess this improvement on different types of corpora and on larger collections. The only corpus we are aware of for XML-IR is the one being built for the INEX initiative. We are currently participating to this task using a slightly different model than the one described here, but our results are still too preliminary to be presented here.

The model has still to be improved and developed in order to obtain an operational structured information retrieval system. Nevertheless results are already encouraging and findings are interesting enough to continue investigating this model. Bayesian networks can handle different sources of information and allows training which proves to be important for many IR applications.

Acknowledgement. Many thanks to M. Lalmas who gave us the structured collection we have been using in our experiments.

References

[1] ACM SIGIR 2000 Workshop on XML and Information Retrieval. Athens, Greece. July 28, 2000 – also published in JASIST, Vol 53, n° 6, 2002, special topic issue : XML.

[2] Jamie P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY Retrieval System. In A. Min Tjoa and Isidro Ramos, editors, *Database and Expert Systems Applications, Proceedings of the International Conference*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.

[3] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In *23rd International Conference on Very Large Data Bases*, Athens, Greece, 1997.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from incomplete data via de EM algorithm. *The Journal of Royal Statistical Society*, 39:1–37, 1977.

[5] Fuhr, N. and Rölleke, T. HySpirit – a Probabilistic Inference Engine for Hypermedia Retrieval in Large Databases. In: Schek, H.-J.; Saltor, F.;Ramos, I.; Alonso, G. (eds.). *Proceedings of the 6th International Conference on Extending Database Technology (EDBT)*, Valencia, Spain, pages 24–38. Springer, Berlin, 1998.

[6] Maria Indrawan, Desra Ghazfan, and Bala Srinivasan. Using Bayesian Networks as Retrieval Engines. In *ACIS 5th Australasian Conference on Information Systems*, pages 259–271, Melbourne, Australia, 1994.

[7] Finn Verner Jensen. *An introduction to Bayesian Networks*. UCL Press, London, England, 1996.

[8] Daphne Koller and Mehran Sahami. Hierarchically Classifying Documents Using Very Few Words. In *ICML-97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 435–443, San Francisco, CA, USA, 1997. Morgan Kaufmann.

[9] Paul Krause. Learning Probabilistic Networks. 1998.

[10] Mounia Lalmas. Dempster-Shafer's Theory of Evidence Applied to Structured Documents: Modelling Uncertainty. In *Proceedings of the 20th Annual International ACM SIGIR*, pages 110–118, Philadelphia, PA, USA, July 1997. ACM.

[11] Mounia Lalmas. Uniform representation of content and structure for structured document retrieval. Technical report, Queen Mary & Westfield College, University of London, London, England, 2000.

[12] Mounia Lalmas and Ekaterini Moutogianni. A Dempster-Shafer indexing for the focussed retrieval of a hierarchically structured document space: Implementation and experiments on a web museum collection. In *6th RIAO Conference, Content-Based Multimedia Information Access*, Paris, France, April 2000.

[13] Mounia Lalmas, I. Ruthven, and M. Theophylactou. Structured document retrieval using Dempster-Shafer's Theory of Evidence: Implementation and evaluation. Technical report, University of Glasgow, UK, August 1997.

[14] Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Y. Ng. Improving Text Classification by Shrinkage in a Hierarchy of Classes. In Ivan Brasko and Saso Dzeroski, editors, *International Conference on Machine Learning (ICML 98)*, pages 359–367. Morgan Kaufmann, 1998.

[15] Kevin Patrick Murphy. A Brief Introduction to Graphical Models and Bayesian Networks. web: http://www.cs.berkeley.edu/~murphyk/Bayes/bayes.html, October 2000.

[16] Sung Hyon Myaeng, Dong-Hyun Jang, Mun-Seok Kim, and Zong-Cheol Zhoo. A Flexible Model for Retrieval of SGML documents. In W. Bruce Croft, Alistair Moffat, C.J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–140, Melbourne, Australia, August 1998. ACM Press, New York.

[17] Gonzalo Navarro and Ricardo Baeza-Yates. Proximal Nodes: A Model to Query Document Databases by Content and Structure. *ACM TOIS*, 15(4):401–435, October 1997.

[18] OASIS. Docbook standard. http://www.oasis-open.org/specs/docbook.shtml, 2 2001.

[19] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[20] Berthier A.N. Ribeiro and Richard Muntz. A Belief Network Model for IR. In *Proceedings of the 19th ACM-SIGIR conference*, pages 253–260, 1996.

[21] Stephen E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[22] Howard R. Turtle and W. Bruce Croft. Evaluation of an Inference Network-Based Retrieval Model. *ACM Transactions On Information Systems*, 9(3):187–222, 1991.

[23] S. Walker and Stephen E. Robertson. Okapi/Keenbow at TREC-8. In E. M. Voorhees and D.K. Harman, editors, *NIST Special Publication 500–246: The Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, USA, November 1999.

[24] Ross Wilkinson. Effective retrieval of structured documents. In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, pages 311–317, Dublin, Ireland, July 1994. Springer-Verlag.