A Multi-Layered Bayesian Network Model for Structured Document Retrieval

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

[1] Department of Computer and Information Sciences.
University of Strathclyde, Glasgow, Scotland, UK.
Fabio.Crestani@cis.strath.ac.uk
[2] Departamento de Ciencias de la Computación e Inteligencia Artificial,
E.T.S.I. Informática. Universidad de Granada, 18071 – Granada, Spain.
{lci,jmfluna,jhg}@decsai.ugr.es

Abstract. New standards in document representation, like for example SGML, XML, and MPEG-7, compel Information Retrieval to design and implement models and tools to index, retrieve and present documents according to the given document structure. The paper presents the design of an Information Retrieval system for multimedia structured documents, like for example journal articles, e-books, and MPEG-7 videos. The system is based on Bayesian Networks, since this class of mathematical models enable to represent and quantify the relations between the structural components of the document. Some preliminary results on the system implementation are also presented.

1 Introduction

Information Retrieval (IR) systems are powerful and effective tools for accessing documents by content. A user specifies the required content using a query, often consisting of a natural language expression. Documents estimated to be relevant to the user query are presented to the user through an interface. New standards in multimedia document representation compel IR to design and implement models and tools to index, retrieve and present documents according to the given document structure. In fact, while standard IR treats documents as if they were atomic entities, modern IR needs to be able to deal with more elaborate document representations, like for example documents written in SGML, HTML, XML or MPEG-7. These document representation formalisms enable to represent and describe documents said to be *structured*, that is documents whose content is organised around a well defined structure. Examples of these documents are books and textbooks, scientific articles, technical manuals, educational videos, etc. This means that documents should no longer be considered as atomic entities, but as aggregates of interrelated objects that need to be indexed, retrieved, and presented both as a whole and separately, in relation to the user's needs. In other words, given a query, an IR system must retrieve the set of document components that are most relevant to this query, not just entire documents.

In order to enable querying both content and structure an IR system needs to posses the necessary primitives to model effectively the document's content and structure. Taking into account that Bayesian Networks (BNs) have been already successfully applied to build standard IR systems, we believe that they are also an appropriate tool to model both in a qualitative and quantitative way the content and structural relations of multimedia structured documents.

In this paper we propose a BN model for structured document retrieval, which can be considered as an extension of a previously developed model to manage standard (non-structured) documents [1, 6]. The rest of the paper is organized as follows: we begin in Section 2 with the preliminaries. In Section 3 we introduce the Bayesian network model for structured document retrieval, the assumptions that determine the network topology being considered, the details about probability distributions stored in the network, and the way in which we can efficiently use the network model for retrieval, by performing probabilistic inference. Section 4 shows preliminary experimental results obtained with the model, using a structured document test collection [9]. Finally, Section 5 contains the concluding remarks and some proposals for future research.

2 Preliminaries

Probabilistic models constitute an important kind of IR models, which have been widely used for a long time [5], because they offer a principled way to manage the uncertainty that naturally appears in many elements within this field. These models (and others, as the Vector Space model [15]) usually represent documents and queries by means of vectors of *terms* or *keywords*, which try to characterize their information content. Because these terms are not equally important, they are usually weighted to highlight their importance in the documents they belong to, as well as in the whole collection. The most common weighting schemes are the *term frequency*, tf_{ij}, i.e, the number of times that the i^{th} term appears in the j^{th} document, and the *inverse document frequency*, idf_i, of the i^{th} term in the collection, $idf_i = \lg(N/n_i) + 1$, where N is the number of documents in the collection, and n_i is the number of documents that contain the i^{th} term. The combination of both weights, $tf_{ij} \cdot idf_i$, is also a common weighting scheme.

2.1 Information Retrieval and Bayesian Networks: The Bayesian Network Model with Two Layers

Bayesian networks have also been successfully applied in a variety of ways within the IR environment, as an extension/modification of probabilistic IR models [6, 13, 16]. We shall focus on a specific BN-based retrieval model, the Bayesian Network Retrieval Model with two layers (BNR-2) [1, 6], because it will be the starting point of our proposal to deal with structured documents.

The set of variables V in the BNR-2 model is composed of two different sets[1], $V = \mathcal{T} \cup \mathcal{D}$: The set $\mathcal{T} = \{T_1, \ldots, T_M\}$, containing binary random variables representing the M terms in the glossary from a given collection, and the set

[1] We will use the notation T_i (D_j, respectively) to refer to the term (document, respectively) and also to its associated variable and node.

$\mathcal{D} = \{D_1, \ldots, D_N\}$, corresponding also to binary random variables, representing the N documents that compose the collection. A variable D_j has its domain in the set $\{\bar{d}_j, d_j\}$, where $\bar{d}_j$ and d_j respectively mean 'the document D_j is not relevant', and 'the document D_j is relevant' for a given query[2]. A variable T_i takes its values from the set $\{\bar{t}_i, t_i\}$, where in this case $\bar{t}_i$ stands for 'the term T_i is not relevant', and t_i represents 'the term T_i is relevant'[3]. To denote a generic, unspecified value of a term variable T_i or a document variable D_j, we will use lower-case letters in bold type, $\boldsymbol{t_i}$ and $\boldsymbol{d_j}$.

With respect to the topology of the network (see Figure 1), there are arcs going from term nodes to those document nodes where these terms appear, and there are not arcs connecting pairs of either document nodes or term nodes. This means that the terms are marginally independent among each other, and the documents are conditionally independent given the terms that they contain. In this way, we get a network composed of two simple layers, the term and document subnetworks, with arcs only going from nodes in the first subnetwork to nodes in the second one.

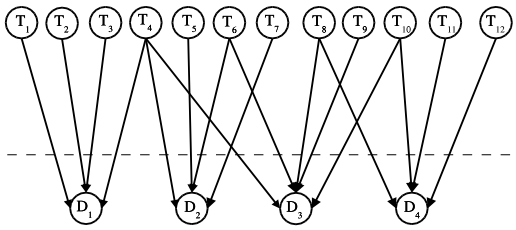

Fig. 1. Two-layered Bayesian network for the BNR-2 model.

The probability distributions stored in each node of the BNR-2 model are computed as follows: For each term node we need a marginal probability distribution, $p(\boldsymbol{t_i})$; we use $p(t_i) = \frac{1}{M}$ and $p(\bar{t}_i) = \frac{M-1}{M}$ (M being the number of terms in the collection). For the document nodes we have to estimate the conditional probability distribution $p(\boldsymbol{d_j}|pa(D_j))$ for any configuration $pa(D_j)$ of $Pa(D_j)$ (i.e., any assignment of values to all the variables in $Pa(D_j)$), where $Pa(D_j)$ is the parent set of D_j (which coincides with the set of terms indexing document D_j). As a document node may have a high number of parents, the number of conditional probabilities that we need to estimate and store may be huge. Therefore, the BNR-2 model uses a specific canonical model to represent these conditional probabilities:

$$p(d_j|pa(D_j)) = \sum_{T_i \in R(pa(D_j))} w_{ij}\,, \tag{1}$$

where $R(pa(D_j)) = \{T_i \in Pa(D_j) \,|\, t_i \in pa(D_j)\}$, i.e., $R(pa(D_j))$ is the set of terms in $Pa(D_j)$ that are instantiated as relevant in the configuration $pa(D_j)$;

[2] A document is relevant for a given query if it satisfies the user's information need expressed by means of this query.

[3] A term is relevant in the sense that the user believes that this term will appear in relevant documents.

w_{ij} are weights verifying $w_{ij} \geq 0$ and $\sum_{T_i \in Pa(D_j)} w_{ij} \leq 1$. So, the more terms are relevant in $pa(D_j)$ the greater the probability of relevance of D_j.

The BNR-2 model can be used to obtain a relevance value for each document given a query Q. Each term T_i in the query Q is considered as an evidence for the propagation process, and its value is fixed to t_i. Then, the propagation process is run, thus obtaining the posterior probability of relevance of each document given that the terms in the query are also relevant, $p(d_j|Q)$. Later, the documents are sorted according to their corresponding probability and shown to the user. Taking into account the number of nodes in the network $(N+M)$ and the fact that, although its topology seems relatively simple, there are multiple pathways connecting nodes as well as nodes with a great number of parents, general purpose inference algorithms cannot be applied due to efficiency considerations, even for small document collections. So, the BNR-2 model uses a tailored inference process, that computes the required probabilities very efficiently and ensures that the results are the same that those obtained using exact propagation in the entire network:

$$p(d_j|Q) = \sum_{T_i \in Pa(D_j)} w_{ij} \cdot p(t_i|Q)\,. \tag{2}$$

Taking into account the topology of the term subnetwork, $p(t_i|Q) = 1$ if $T_i \in Q$ and $p(t_i|Q) = \frac{1}{M}$ if $T_i \notin Q$, hence eq. (2) becomes

$$p(d_j|Q) = \sum_{T_i \in Pa(D_j) \cap Q} w_{ij} + \frac{1}{M} \sum_{T_i \in Pa(D_j) \setminus Q} w_{ij}\,. \tag{3}$$

2.2 Structured Document Retrieval

In IR the area of research dealing with structured documents is known as *structured document retrieval*. A good survey of the state of the art of structured document retrieval can be found in [4]. The inclusion of the structure of a document in the indexing and retrieval process affects the design and implementation of the IR system in many ways. First of all, the indexing process must consider the structure in the appropriate way, so that users can search the collection both by content and structure. Secondly, the retrieval process should use both structure and content in the estimate of the relevance of documents. Finally, the interface and the whole interaction has to enable the user to make full use of the document structure. In fact, querying by content and structure can only be achieved if the user can specify in the query *what* he/she is looking for, and *where* this should be located in the required documents. The what involves the specification of the content, while the where is related to the structure of the documents.

It has been recognised that the best approach to querying structured documents is to let the user specify in the most natural way both the content and the structural requirements of the desired documents [4]. This can be achieved by letting the user specify the content requirement in a natural language query, while enabling the user to qualify the structural requirements through a graphical

user interface. A GUI is well suited to show and let the user indicate structural elements of documents in the collection [17].

This paper addresses the issues related to the modelling of the retrieval of structured documents when the user does not explicitly specifies the structural requirements. In standard IR retrievable units are fixed, so only the entire document, or, sometimes, some pre-defined parts such as chapters or paragraphs constitute retrievable units. The structure of documents, often quite complex and consisting of a varying numbers of chapters, sections, tables, formulae, bibliographic items, etc., is therefore "flattened" and not exploited. Classical retrieval methods lack the possibility to interactively determine the size and the type of retrievable units that best suit an actual retrieval task or user preferences. Some IR researchers are aiming at developing retrieval models that dynamically return document components of varying complexity. A retrieval result may then consist of several entry points to a same document, corresponding to structural elements, whereby each entry point is weighted according to how it satisfies the query. Models proposed so far exploit the content and the structure of documents to estimate the relevance of document components to queries, based on the aggregation of the estimated relevance of their related components. These models have been based on various theories, like for example fuzzy logic [3], Dempster-Shafer's theory of evidence [10], probabilistic logic [2], and Bayesian inference [11]. What these models have in common is that the basic components of their retrieval function are variants of the standard IR term weighting schema, which combines term frequency with inverse document frequency, often normalised keeping into account document length. Evidence associated with the document structure is often encoded into one or both of these dimensions. A somewhat different approach has been presented in [14], where evidence associated with the document structure is made explicit by introducing an "accessibility" dimension. This dimension measures the strength of the structural relationship between document components: the stronger the relationship, the more impact has the content of a component in describing the content of its related components. Our approach is based on a similar view of structured document retrieval. In fact, we use a BN to model the relations between structural elements of documents. A BN is a very powerful tool to capture these relations, with particular regards to hierarchically structured document. The next section contains a detailed presentation of our approach. Other approaches to structured document retrieval also based on BNs can be found in [8, 11, 12].

3 From Two-Layered to Multi-Layered Bayesian Networks for Structured Document Retrieval

To deal with structured document retrieval, we are going to assume that each document is composed of a hierarchical structure of l abstraction *levels* $L_1, \ldots, L_l$, each one representing a structural association of elements in the text. For instance, chapters, sections, subsections and paragraphs in the context of a general structured document collection, or scenes, shots, and frames in MPEG-7 videos. The level in which the document itself is included will be noted as level 1 (L_1), and the more specific level as L_l.

Each level contains *structural units*, i.e., single elements as Chapter 4, Subsection 4.5, Shot 54, and so on. Each one of these structural units will be noted as U_{ij}, where i is the identifier of that unit in the level j. The number of structural units contained in each level L_j is represented by $|L_j|$. Therefore, $L_j = \{U_{1j}, \ldots, U_{|L_j|j}\}$. The units are organised according to the actual structure of the document: Every unit U_{ij} at level j, except the unit at level $j = 1$ (i.e., the complete document D_i), is contained in only one unit $U_{z(i,j)j-1}$ of the lower level $j-1$[4], $U_{ij} \subseteq U_{z(i,j)j-1}$. Therefore, each structured document may be represented as a tree (an example is displayed in Figure 2).

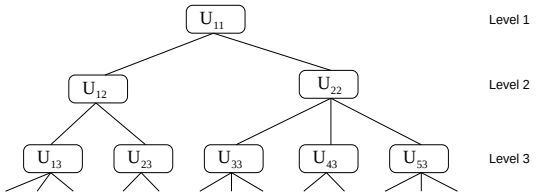

Fig. 2. A structured document.

Now, we shall describe the Bayesian network used by our Bayesian Network Retrieval model for Structured Documents (BNR-SD).

3.1 Network Topology

Taking into account the topology of the BNR-2 model for standard retrieval (see Figure 1), it seems to us that the natural extension to deal with structured documents is to connect the term nodes with the structural units $U_{1l}, \ldots, U_{|L_l|l}$ of the upper level L_l. Therefore, only the units in level L_l will be indexed, having associated several terms describing their content (see Figure 3).

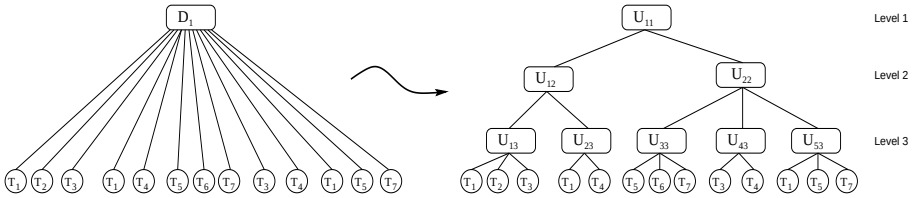

Fig. 3. From an indexed document to an indexed structured document.

From a graphical point of view, our Bayesian network will contain two different types of nodes: those associated to structural units, and those related to terms. As in the BNR-2 model, each node represents a binary random variable: U_{ij} takes its values in the set $\{\bar{u}_{ij}, u_{ij}\}$, representing that the unit is not relevant

[4] $z(i,j)$ is a function that returns the index of the unit in level $j-1$ where the unit with index i in level j belongs to.

and is relevant, respectively; a term variable T_i is treated exactly as in the BNR-2 model. The independence relationships that we assume in this case are of the same nature that those considered in the BNR-2 model: terms are marginally independent among each other, and the structural units are conditionally independent given the terms that they contain. These assumptions, together with the hierarchical structure of the documents, completely determine the topology of the Bayesian network with $l+1$ layers, where the arcs go from term nodes to structural units in level l, and from units in level j to units in level $j-1$, $j = 2, \ldots, l$. So, the network is characterized by the following parent sets for each type of node:

- $\forall T_k \in \mathcal{T},\ Pa(T_k) = \emptyset$.
- $\forall U_{il} \in L_l,\ Pa(U_{il}) = \{T_k \in \mathcal{T} \,|\, U_{il} \text{ is indexed by } T_k\}$.
- $\forall j = 1, \ldots, l-1,\ \forall U_{ij} \in L_j,\ Pa(U_{ij}) = \{U_{kj+1} \in L_{j+1} \,|\, U_{kj+1} \subseteq U_{ij}\}$.

An example of this multi-layer BN is depicted in Figure 4, for $l = 3$.

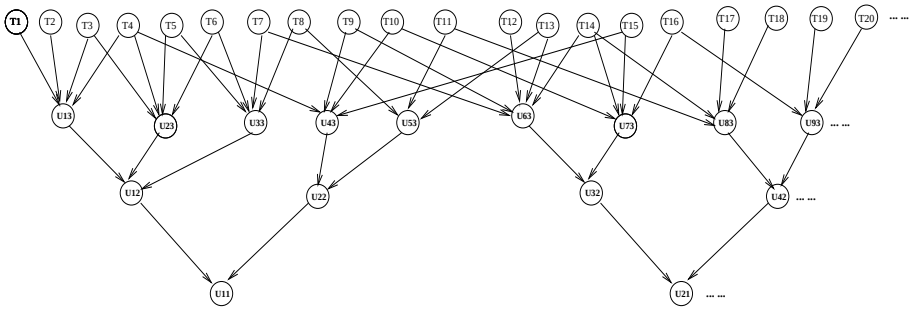

Fig. 4. Multi-layered Bayesian network for the BNR-SD model.

3.2 Conditional Probabilities

The following task is the assessment of the (conditional) probability distributions:

1. Term nodes T_k: they store the same marginal probabilities $p(\boldsymbol{t_k})$ as in the BNR-2 model.
2. Structural units U_{il} in level l: to compute $p(\boldsymbol{u_{il}}|pa(U_{il}))$ we use the same kind of canonical model considered for the relationships between terms and documents in the BNR-2 model (see eq. (1)):

$$p(u_{il}|pa(U_{il})) = \sum_{T_k \in R(pa(U_{il}))} w_{ki}\,, \tag{4}$$

where in this case w_{ki} is a weight associated to each term T_k indexing the unit U_{il}, with $w_{ki} \geq 0$ and $\sum_{T_k \in Pa(U_{il})} w_{ki} = 1$[5].

[5] Notice that we use here the symbol $=$ instead of $\leq$. The only reason for this restriction is to ease some implementation details of the model.

3. Structural units U_{ij} in level j, $j \neq l$: to estimate $p(\boldsymbol{u_{ij}}|pa(U_{ij}))$ we intend to use a similarity measure between two sets of terms, one associated to the whole unit U_{ij} and the other associated to the units contained in U_{ij} that are instantiated as relevant in the configuration $pa(U_{ij})$. More precisely, let $R(pa(U_{ij})) = \{U_{kj+1} \in Pa(U_{ij}) \,|\, u_{kj+1} \in pa(U_{ij})\}$, and let $A(U_{ij})$ and $A(R(pa(U_{ij})))$ be the sets of terms that have been used to index U_{ij} and the units in $R(pa(U_{ij}))$, respectively[6]. In graphical terms, $A(U_{ij}) = \{T_k \in \mathcal{T} \,|\, T_k \text{ is an ancestor of } U_{ij}\}$ and $A(R(pa(U_{ij}))) = \{T_k \in \mathcal{T} \,|\, T_k \text{ is an ancestor of some node in } R(pa(U_{ij}))\}$. Then, if $Sim(\mathcal{B}, \mathcal{C})$ denotes a similarity measure between two sets of terms $\mathcal{B}$ and $\mathcal{C}$, we define $p(u_{ij}|pa(U_{ij}))$ as follows:

$$p(u_{ij}|pa(U_{ij})) = Sim\left(A(U_{ij}), A(R(pa(U_{ij})))\right) . \tag{5}$$

The (asymmetrical) definition of similarity that we are going to use is

$$Sim(\mathcal{B}, \mathcal{C}) = \frac{\sum_{T_k \in \mathcal{C}} w_{k\mathcal{C}}}{\sum_{T_h \in \mathcal{B}} w_{h\mathcal{B}}} , \tag{6}$$

where $w_{h\mathcal{B}}$ and $w_{k\mathcal{C}}$ are the weights of the terms T_h and T_k in the sets $\mathcal{B}$ and $\mathcal{C}$ respectively.

To conclude the specification of the conditional probabilites in the network, we have to give values to the weights w_{ki}, $w_{kA(U_{ij})}$ and $w_{kA(R(pa(U_{ij})))}$ associated to a term T_k:

Let tf^l_{ki} be the *frequency* of the term T_k (number of times that T_k occurs) in the unit U_{il}, and idf_k be the *inverse document frequency* of T_k in the whole collection. Similarly, for $j \neq l$, let tf^j_{ki} be the frequency of T_k in $A(U_{ij})$ (we will refer to tf^j_{ki} as the *term frequency* of T_k in the unit U_{ij}) and $tf_{k\mathcal{C}}$ be the frequency of T_k in the set of terms $\mathcal{C}$. Then, we define:

$$\forall j = 1, \ldots, l,\ \forall U_{ij} \in L_j,\ \forall T_k \in \mathcal{T},\ \ w^j_{ki} = tf^j_{ki} \cdot idf_k . \tag{7}$$

$$\forall U_{il} \in L_l,\ \forall T_k \in Pa(U_{il}),\ \ w_{ki} = \frac{w^l_{ki}}{\sum_{T_h \in Pa(U_{il})} w^l_{hi}} . \tag{8}$$

$$\begin{aligned} &\forall j = 1, \ldots, l-1,\ \forall U_{ij} \in L_j, \\ &\qquad \forall T_k \in A(U_{ij}),\ \ w_{kA(U_{ij})} = w^j_{ki} \\ &\qquad \forall T_k \in A(R(pa(U_{ij}))),\ \ w_{kA(R(pa(U_{ij})))} = tf_{kA(R(pa(U_{ij})))} \cdot idf_k . \end{aligned} \tag{9}$$

It is important to notice that, as $tf_{k\mathcal{B}\cup\mathcal{C}} = tf_{k\mathcal{B}} + tf_{k\mathcal{C}}$, then $w_{k\mathcal{B}\cup\mathcal{C}} = w_{k\mathcal{B}} + w_{k\mathcal{C}}$. Moreover, $A(R(pa(U_{ij}))) = \bigcup_{U_{hj+1} \in R(pa(U_{ij}))} A(U_{hj+1})$. Taking into account these facts, we can easily derive another expression for $p(u_{ij}|pa(U_{ij}))$:

$$p(u_{ij}|pa(U_{ij})) = \sum_{U_{hj+1} \in R(pa(U_{ij}))} p^j_{hi} , \tag{10}$$

[6] Strictly speaking, a unit in level $j \neq l$ is not indexed by any term; we refer to the terms indexing structural units in level l that are included either in unit U_{ij} or in some of the units in $R(pa(U_{ij}))$.

where the weight, p^j_{hi}, of the unit U_{hj+1} in the unit U_{ij} is defined as

$$p^j_{hi} = \frac{\sum_{T_k \in A(U_{hj+1})} w^{j+1}_{kh}}{\sum_{T_k \in A(U_{ij})} w^j_{ki}} \,. \tag{11}$$

As $p^j_{hi} \geq 0$ and $\sum_{U_{hj+1} \in Pa(U_{ij})} p^j_{hi} = 1$, then we obtain that the conditional probabilities $p(u_{ij}|pa(U_{ij}))$ for $j \neq l$ are also modeled by the same kind of canonical model used for $p(u_{il}|pa(U_{il}))$. This fact has important consequences for the inference process within the BNR-SD model.

3.3 Inference

The inference process that we have to carry out in order to use the BNR-SD model is, given a query Q, to compute the posterior probabilities of relevance of all the structural units, $p(u_{ij}|Q)$. Although this computation may be difficult in a general case, in our case all the conditional probabilities have been assessed using the canonical model in eq. (1) and only terms nodes are instantiated (so that only a top-down inference is required). In this context, the inference process can be carried out very efficiently, in the following way [7]:

– For the structural units in level L_l, the posterior probabilities are (as in the BNR-2 model):

$$P(u_{il}|Q) = \sum_{T_k \in Pa(U_{il}) \cap Q} w_{ki} + \frac{1}{M} \sum_{T_k \in Pa(U_{il}) \setminus Q} w_{ki} \,. \tag{12}$$

– For the structural units in level L_j, $j \neq l$:

$$P(u_{ij}|Q) = \sum_{U_{hj+1} \in Pa(U_{ij})} p^j_{hi} \cdot p(u_{hj+1}|Q) \,. \tag{13}$$

Therefore, we can compute the required probabilities on a level-by-level basis, starting from level l and going down to level 1.

3.4 Model Implementation

The BNR-SD model has been implemented using the *Lemur Toolkit*, a software written in C++ designed to develop new applications on Information Retrieval and Language Modelling. This package (available at http://www-2.cs.cmu.edu/~lemur/) offers a wide range of classes that cover almost all the tasks required in IR.

Our implementation uses an inverted file, i.e., a data structure containing, for each term in the collection, the structural units in level l where it occurs (the term's children in the network). The evaluation of units in level l is carried out by accumulating, for each unit U_{il}, the weights w_{ki} of those terms belonging to the query by which they have been indexed. To speed up the retrieval, all the weights w_{ki} (eq. 8) have been precomputed at indexing time and stored in a binary random access file. When the accumulation process is finished, for

each unit U_{il} sharing terms with the query (i.e., $Pa(U_{il}) \cap Q \neq \emptyset$) we have an accumulator $S_{il} = \sum_{T_k \in Pa(U_{il}) \cap Q} w_{ki}$; then we can compute the value $P(u_{il}|Q)$ in eq. (12) as $P(u_{il}|Q) = S_{il} + \frac{1}{M}(1 - S_{il})$. Notice that the units containing no query term do not need to be evaluated, and their posterior probability is the same as their prior, $P(u_{il}|Q) = \frac{1}{M}$.

With respect to the structural units from the rest of layers, the only information needed is also stored in a binary random access file, containing, for each unit U_{hj+1}, that one where it is contained (its unique child in the network), U_{ij}, and the corresponding weight p^j_{hi} (eq. 11), which are also precomputed at indexing time. In order to evaluate the units in level $j \neq l$, those units in level $j+1$ evaluated in a previous stage will play the same role as query terms do in the evaluation of units in level l: for each unit U_{ij} containing units U_{hj+1} previously evaluated (this happens if $A(U_{hj+1}) \cap Q \neq \emptyset$), we use two accumulators, one for the weights p^j_{hi} and the other for the products $p^j_{hi} \cdot p(u_{hj+1}|Q)$. At the end of the accumulation process, for each one of these units U_{ij} we have two accumulators, $S_{ij} = \sum_{U_{hj+1} \in Q_{ij}} p^j_{hi}$ and $SP_{ij} = \sum_{U_{hj+1} \in Q_{ij}} p^j_{hi} \cdot p(u_{hj+1}|Q)$, where $Q_{ij} = \{U_{hj+1} \in Pa(U_{ij}) \mid A(U_{hj+1}) \cap Q \neq \emptyset\}$. Then we can compute the value $P(u_{ij}|Q)$ in eq. (13) as $P(u_{ij}|Q) = SP_{ij} + \frac{1}{M}(1 - S_{ij})$.

4 Preliminary Experiments

Our BNR-SD model has been tested using a collection of structured documents, marked up in XML, containing 37 William Shakespeare's plays [9]. A play has been considered structured in acts, scenes and speeches (so that $l = 4$), and may contain also epilogues and prologues. Speeches have been the only structural units indexed using Lemur. The total number of unique terms contained in these units is 14019, and the total number of structural units taken into account is 32022. With respect to the queries, the collection is distributed with 43 queries, with their corresponding relevance judgements. From these 43 queries, the 35 which are content-only queries were selected for our experiments.

As a way of showing the new potential of retrieving structured documents, several experiments have been designed. Let us suppose that a user is interested in the structural units of a specific type that are relevant for each query (i.e., s/he selects a given granularity level). Therefore, four retrievals have been run for the set of queries: only retrieving plays, only acts, only scenes, prologues and epilogues, and finally, speeches. A last experiment tries to return to the user, in only one ranking, all the structural units ranked according to their relevance. Table 1 shows the average recall-precision values (using the 11 standard recall values) for the five experiments. The row *AVP-11p* shows the average precision for the 11 values of recall. The maximum number of units retrieved for each experiment has been fixed to 1000.

An important fact to notice is that when the system offers a ranking with all the structural units, the performance is not very good. This behaviour is due to the fact that, according to the expressions used to compute the relevance of the units, the posterior probability of a play, for instance, is very small compared to that assigned to a speech. This implies that the lower level units, like plays

or acts, for example, are located in the furthest positions in the ranking and therefore, never retrieved. After observing the ranking produced in the last two experiments, we noticed that there are a number of units, in this case speeches, that have a posterior probability equal to 1.0. The reason is that they are very short, perhaps one or two terms, occurring all of them in the query. As the weights are normalised to 1.0, the final relevance is very high and these units are placed on the top of the ranking but introducing some noise. This is other cause of the poor behaviour of the retrieval considering only speeches and all types of units as well. These facts suggests the convenience of including in our model a decision procedure to select the appropriate units to be retrieved.

On the other hand, the effectiveness of the system is quite good for the first three experiments, where the objective is to retrieve larger units, containing more terms, as acts and scenes. However, it should be noticed that the effectiveness decreases as the number of units involved in the retrieval increases and the number of terms per unit decreases.

Recall	AP-PLAY	AP-ACT	AP-SCENE	AP-SPEECH	AP-All
0	0.9207	0.7797	0.5092	0.1957	0.2018
0.1	0.9207	0.7797	0.5065	0.1368	0.0821
0.2	0.9207	0.7797	0.4600	0.1100	0.0245
0.3	0.9207	0.7738	0.4279	0.0846	0.0055
0.4	0.9207	0.7518	0.4088	0.0721	0.0054
0.5	0.9207	0.7318	0.3982	0.0434	0.0004
0.6	0.9207	0.6755	0.3663	0.0362	0.0
0.7	0.9207	0.6512	0.3220	0.0201	0.0
0.8	0.9207	0.6453	0.3054	0.0138	0.0
0.9	0.9207	0.6253	0.2580	0.0079	0.0
1	0.9207	0.6253	0.2484	0.0025	0.0
AVP-11p	0.9207	0.7108	0.3828	0.0657	0.0290

Table 1. Average precision values for the experiments with the BNR-SD Model.

5 Concluding Remarks

In this paper a Bayesian network-based model for structured document retrieval, BNR-SD, has been presented, together with some promising preliminary experiments with the structured test collection of Shakespeare's plays. Our model can be extended/improved in several ways, and we plan to pursue some of them in the near future:

- To incorporate to our network model a decision module, in order to select the appropriate structural units (the *best entry points*) that will be shown to the users, depending on their own preferences.
- To allow that structural units in levels different from l have associated specific textual information (for example the title of a chapter or a section).
- To include in our network model specific term relationships (as those in [7]) and/or document relationships (as those in [1]). Alternatively, we could also use *Ontologies* to model concepts and their relationships in a given domain of knowledge.

– To permit our model to deal, not only with content-only queries, but also with structure-only and content-and-structure queries.
– To apply our model, in combination with techniques for image analysis, to multimedia retrieval, particularly MPEG-7 videos.

Acknowledgments: This work has been supported by the Spanish CICYT and FIS, under Projects TIC2000-1351 and PI021147 respectively, and by the European Commission under the IST Project MIND (IST-2000-26061).

References

1. S. Acid, L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. An information retrieval model based on simple Bayesian networks. *International Journal of Intelligent Systems*, 18:251–265, 2003.
2. C. Baumgarten. A probabilistic model for distributed information retrieval. In *Proceedings of the* 20^{th} *ACM SIGIR Conference*, 258–266, 1997.
3. G. Bordogna and G. Pasi. Flexible representation and querying of heterogeneous structured documents. *Kibernetika*, 36(6):617–633, 2000.
4. Y. Chiaramella. Information retrieval and structured documents. *Lectures Notes in Computer Science*, 1980:291–314, 2001.
5. F. Crestani, M. Lalmas, C.J. van Rijsbergen, and L. Campbell. Is this document relevant?... probably. A survey of probabilistic models in information retrieval. *ACM Computing Survey*, 30(4):528–552, 1998.
6. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. A layered Bayesian network model for document retrieval. *Lecture Notes in Computer Science*, 2291:169–182, 2002.
7. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. The Bayesian network retrieval model: Foundations and performance. Submitted to the *International Journal of Approximate Reasoning*.
8. A. Graves and M. Lalmas. Video retrieval using an MPEG-7 based inference network. In *Proceedings of the* 25^{th} *ACM–SIGIR Conference*, 339–346, 2002.
9. G. Kazai, M. Lalmas, and J. Reid. The Shakespeare test collection. Available at http://qmir.dcs.qmw.ac.uk/Focus/collbuilding.htm
10. M. Lalmas and I. Ruthven. Representing and retrieving structured documents with Dempster-Shafer's theory of evidence: Modelling and evaluation. *Journal of Documentation*, 54(5):529–565, 1998.
11. S.H. Myaeng, D.H. Jang, M.S. Kim, and Z.C. Zhoo. A flexible model for retrieval of SGML documents. In *Proceedings of the* 21^{th} *ACM–SIGIR Conference*, 138–145, 1998.
12. B. Piwowarski, G.E. Faure, and P. Gallinari. Bayesian networks and INEX. In *Proceedings of the INEX Workshop*, 7–12, 2002.
13. B.A. Ribeiro-Neto and R.R. Muntz. A belief network model for IR. In *Proceedings of the* 19^{th} *ACM–SIGIR Conference*, 253–260, 1996.
14. T. Roelleke, M. Lalmas, G. Kazai, I. Ruthven, and S. Quicker. The accessibility dimension for structured document retrieval. *Lecture Notes in Computer Science*, 2291:284–302, 2002.
15. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., 1983.
16. H. R. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *Information Systems*, 9(3):187–222, 1991.
17. J. Vegas, P. de la Fuente, and F. Crestani. A graphical user interface for structured document retrieval. *Lecture Notes in Computer Science*, 2291:268–283, 2002.