Ranking Structured Documents Using Utility Theory in the Bayesian Network Retrieval Model

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. In this paper a new method based on Utility and Decision theory is presented to deal with structured documents. The aim of the application of these methodologies is to refine a first ranking of structural units, generated by means of an Information Retrieval Model based on Bayesian Networks. Units are newly arranged in the new ranking by combining their posterior probabilities, obtained in the first stage, with the expected utility of retrieving them. The experimental work has been developed using the Shakespeare structured collection and the results show an improvement of the effectiveness of this new approach.

1 Introduction and Motivations

Information Retrieval (IR) systems are powerful and effective tools for accessing documents by content [2]. A user specifies the required content using a query, often consisting of a natural language expression. Documents estimated to be relevant to the user information need expressed by the query are presented to the user through an interface. New standards in document representation require 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 they were atomic entities, modern IR needs to be able to deal with more elaborate document representations, like for example documents written in SGML or XML, for instance. 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 that enables to represent the semantics of complex and long documents [5]. Examples of these documents are books and textbooks, scientific articles, technical manuals, educational videos, news broadcast, etc. This means that documents should no longer be considered as atomic entities, but as aggregates of interrelated semantic 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, operationally, given a query an IR system must retrieve the set of document components that are most relevant to this query, not just entire documents. An example of a task that required the identification of specific structural elements is the search of a long educational video on Art Noveau for parts describing the work of Charles Rennie Macintosh. In this case, is likely that the user is not interested in the entire video or in the few frames in which Macintosh appears (identified by image analysis or word spotting on the soundtrack), but on a set of video segments (the user might not care if these are frames, scenes or large elements) which describe the work of Macintosh. In structured document retrieval this is made possible by searching with appropriate models the structured description of the video to identify the structural elements that contain the information sought.

However, the above example enables to highlight one of the problems of structured document retrieval that has not been well studied yet. Faced with a query on Charles Rennie Macintosh, a structured document retrieval system will retrieve from the educational video on Art Noveau only those structural elements (frames, scenes, etc, depending on the indexing level used) that are found to be relevant to the query. In modern IR this is achieved by ranking the structural elements based on some model that uses the weights assigned to the word (or words) "Charles Rennie Macintosh". In this way structural elements assumed to be "about" Macintosh because the query words appear in them will be ranked at the top and presented first to the user. But this might not be the best way to present the sought information to the user. In fact, using this approach the user will only see the structural elements of the video that are found to be about Charles Rennie Macintosh without their *context*.

Context is very important in structured document retrieval, but it has rarely been studied. It is easy to recognise that the context in which some information is presented is an integral component of the understanding of the information itself. In the above example, it would be of little use to the user to present him with a ranked list of frames found to be about Macintosh. Similarly, it would be of little use to retrieve the entire video or large parts of it containing much irrelevant information. What the user would like to see, we believe, is some structural elements of the video that are about Macintosh, where information about Macintosh is presented within some context, that is it is accompanied by sufficient information to enable the user to fully understand what is conveyed by the structural elements found to be relevant. This might require the retrieval of larger structural elements of the video (e.g. scenes) containing a combination of smaller structural elements (e.g. frames), some of which are highly relevant and some others being retrieved only to provide the context for the information contained in the relevant elements.

The above problem is very difficult for standard structured document retrieval and can only be tackled effectively using models that enable to fully represent the complex relationships present in a structured document among the different structural elements that compose it. This is particularly true for hierarchically structured documents where the inclusion relation between structural

elements can be considered together with the proximity relation (one section following or preceding another) and the semantic similarity relation (two sections about the same topic) to fully capture the context.

Bayesian Networks (BN) are powerful tools to represent and quantify the strength of relationships between objects. As such, they are also being applied to structured document retrieval (see for example [13, 8, 14]). In [6] we proposed a retrieval model for structured document retrieval based on a multi-layered BN that is an extension of a previously developed model to manage standard (non-structured) documents [1, 7]. However, though these models can tackle structured document retrieval (with various degrees of success), they cannot tackle the context problem explained above.

The overall objective of our work is to design a system that will enable to retrieve from a collection of structured documents elements of varying structure containing relevant information within some meaningful context, so that these structural elements can be considered self-contained informative objects that can be used on their own without reference to their documents of origin.

Until now, when the IR system decided to show a document, this decision was independent on showing any other document from the collection. But now, with structured documents, this is different because once it retrieves a piece of text, it may affect the retrieval of some others. To put into practice this previous idea, the best tool is *Decision Theory* [9], which is aimed to help making decisions, i.e., to choose an alternative among a set of them taking into account the possible consequences. In the context of this paper, the problem is to determine those parts of documents that will be shown to the users in response to a query, without showing any redundancy: if section 1.3 is more relevant than the whole chapter 1, then the IR system should only give this section to the user. But if the chapter contains more useful information, then the chapter is the text object returned and not the section, although it is also interesting. Specifically, our approach applies Utility Theory to solve this problem, i.e., the branch of Decision Theory concerned with measurement and representation of preferences. By means of *Utility Functions*, the preferences for the different decisions are described, and with them the *Expected Utility* for each alternative is computed. The alternative with the highest expected utility is considered the most preferable.

This paper is structured as follows. In section 2 we give some preliminaries to rest of the paper, including the description of the BN model for structured document retrieval. There, the assumptions that determine the network topology are considered, together with the details about the 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 3 presents how decision theory can be used to capture the contextual relations between structural elements on the BN model. In Section 4 we report on some preliminary experimental results obtained with the model, using a structured document test collection [10]. Finally, Section 5 contains the concluding remarks and some directions for future research.

2 Preliminaries

In this paper we present a model called **Sride$^{\mathbf{RB}}$**, which stands for *Information Retrieval System for Structured Documents based on Bayesian networks* (translated from the original name in Spanish and Italian). This model is composed of two parts: the retrieval model, which produces a ranking of all the structural units included in the documents according to the degree of relevance with respect a query, and a decision making model that will determine which units will be returned to the user in order to capture the relevant information in its context. The application of Decision Theory to Information Retrieval is a novel approach to this problem, which has been approached already with other technologies [15].

This paper addresses the issues related to the modeling of the retrieval of structured documents when the user does not explicitly specifies the structural element requested. In standard IR retrievable units are fixed, so only entire documents constitute retrievable units. The structure of documents, often quite complex, 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 [4], Dempster-Shafer's theory of evidence [12], probabilistic logic [3], and Bayesian inference [13]. A somewhat different approach has been presented in [15], 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 subsection contains a detailed presentation of our approach.

2.1 A Multilayered Bayesian Network Model for Structured Document Retrieval

Given a document collection composed of N documents, $\mathcal{D} = \{D_1, \ldots, D_N\}$, and the set $\mathcal{T} = \{T_1, \ldots, T_M\}$ of the M terms used to index these documents (the glossary of the collection), $A(D_i)$ will denote the subset of terms in $\mathcal{T}$ that are used to index the document D_i.

We shall assume that each document is composed of a hierarchical structure of l abstraction *levels* $\mathcal{L}_1, \ldots, \mathcal{L}_\ell$, 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 ($\mathcal{L}_1$), and the more specific level as $\mathcal{L}_\ell$.

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 $U_{i,j}$, where i is the identifier of that unit in the level j. The number of structural units contained in each level $\mathcal{L}_j$ is represented by $|\mathcal{L}_j|$. Therefore, $\mathcal{L}_j = \{U_{1,j}, \ldots, U_{|\mathcal{L}_j|,j}\}$. The units are organised according to the actual structure of the document: Every unit $U_{i,j}$ at level j, except the unit at level $j = 1$ (i.e., the complete document $D_i = U_{i,1}$), is related to only one unit $U_{z(i,j),j-1}$ of the lower level $j - 1$[1]. As the text (the whole set of terms) associated to $U_{i,j}$ is part of the text associated to $U_{z(i,j),j-1}$, abusing of the notation, we shall note this relation as $U_{i,j} \subseteq U_{z(i,j),j-1}$. Therefore, each structured document may be represented as a tree (Figure 1 shows an example).

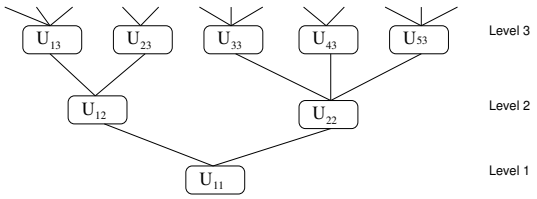

Fig. 1. A structured document.

Each term $T_k \in A(D_i)$, originally indexing a document D_i, will be assigned to those units in level $\mathcal{L}_\ell$ containing it which are associated with D_i. Therefore, only the units in level $\mathcal{L}_\ell$ will be indexed, having associated several terms describing their content (see Figure 2).

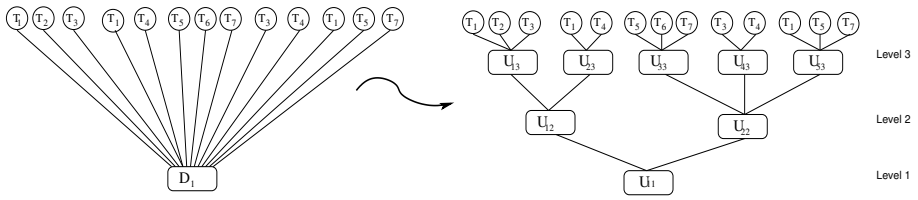

Fig. 2. From an indexed document to an indexed structured document.

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

[1] $z(i, j)$ 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.

terms, so that $V = \mathcal{T} \cup \mathcal{U}$, where $\mathcal{U} = \cup_{j=1}^{l} \mathcal{L}_j$. Each node represents a binary random variable: $U_{i,j}$ takes its values in the set $\{u_{i,j}^-, u_{i,j}^+\}$, representing that the unit is not relevant and is relevant, respectively.[2]; T_i takes its values from the set $\{t_i^-, t_i^+\}$, where in this case 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 unit variable $U_{i,j}$, we will use lower-case letters, t_i and $u_{i,j}$. Notice that we use the notation T_i ($U_{i,j}$, respectively) to refer to the term (unit, respectively) and also to its associated variable and node.

The Bayesian network representing the structured collection has a graph topology 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 $j-1$, $j = 2, \ldots, l$. More formally, the network is characterized by the following parent sets, $Pa(.)$, for each type of node:

- $\forall T_k \in \mathcal{T}$, $Pa(T_k) = \emptyset$.
- $\forall U_{i,l} \in \mathcal{L}_\ell$, $Pa(U_{i,l}) = \{T_k \in \mathcal{T} \,|\, U_{i,l}$ is indexed by $T_k\}$.
- $\forall j = 1, \ldots, l-1$, $\forall U_{i,j} \in \mathcal{L}_j$, $Pa(U_{i,j}) = \{U_{h,j+1} \in \mathcal{L}_{j+1} \,|\, U_{h,j+1} \subseteq U_{i,j}\}$.

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

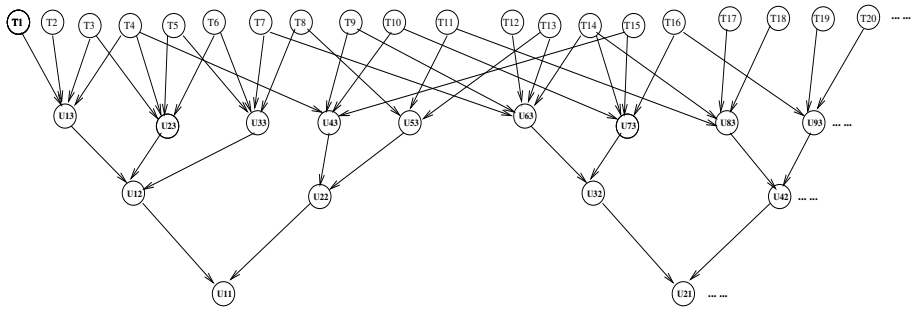

Fig. 3. Multi-layered Bayesian network for the BNR-SD model.

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

• Term nodes T_k: they store the following marginal probabilities: $p(t_i^+) = \frac{1}{M}$ and $p(t_i^-) = \frac{M-1}{M}$.

• Structural units $U_{i,j}$: we have to assess $p(u_{i,l}|pa(U_{i,l}))$ and $p(u_{i,j}|pa(U_{i,j}))$, $j \neq l$, where $pa(U)$ denotes any configuration of $Pa(U)$, i.e., any assignment of values to all the variables in $Pa(U)$. The following canonical model is considered:

$$p(u_{i,l}^+|pa(U_{i,l})) = \sum_{T_k \in R(pa(U_{i,l}))} w(T_k, U_{i,l})\,, \tag{1}$$

[2] A unit 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.

$$p(u^+_{i,j}|pa(U_{i,j})) = \sum_{U_{h,j+1} \in R(pa(U_{i,j}))} w(U_{h,j+1}, U_{i,j})\,, \tag{2}$$

where $w(T_k, U_{i,l})$ is a weight associated to each term T_k indexing the unit $U_{i,l}$, $w(U_{h,j+1}, U_{i,j})$ is a weight measuring the importance of the unit $U_{h,j+1}$ within $U_{i,j}$, with $w(T_k, U_{i,l}) \geq 0$ and $w(U_{h,j+1}, U_{i,j}) \geq 0$. In either case $R(pa(U))$ is the subset of parents of U (terms for $j = l$, units in level $j+1$ for $j \neq l$) that are instantiated as relevant in the configuration $pa(U)$, i.e., $R(pa(U_{i,l})) = \{T_k \in Pa(U_{i,l}) \,|\, t^+_k \in pa(U_{i,l})\}$ and $R(pa(U_{i,j})) = \{U_{h,j+1} \in Pa(U_{i,j}) \,|\, u^+_{h,j+1} \in pa(U_{i,j})\}$. So, the more parents of U are relevant the greater the probability of relevance of U

Before defining the weights $w(T_k, U_{i,l})$ and $w(U_{h,j+1}, U_{i,j})$ in equations (1) and (2), let us introduce some additional notation: for any unit $U_{i,j} \in \mathcal{U}$, let $A(U_{i,j}) = \{T_k \in \mathcal{T} \,|\, T_k \text{ is an ancestor of } U_{i,j}\}$, i.e., $A(U_{i,j})$ is the set of terms that are included in the unit $U_{i,j}$[4]. Let $tf_{k,C}$ be the *frequency* of the term T_k (number of times that T_k occurs) in the set of terms C and idf_k be the *inverse document frequency* of T_k in the whole collection. We shall use the weighting scheme $\rho(T_k, C) = tf_{k,C} \cdot idf_k$. We define

$$\forall U_{i,l} \in \mathcal{L}_\ell,\ \forall T_k \in Pa(U_{i,l}),\ \ w(T_k, U_{i,l}) = \frac{\rho(T_k, A(U_{i,l}))}{\sum_{T_h \in A(U_{i,l})} \rho(T_h, A(U_{i,l}))}\,. \tag{3}$$

$$\forall j = 1, \ldots, l-1,\ \forall U_{i,j} \in \mathcal{L}_j,\ \forall U_{h,j+1} \in Pa(U_{i,j}),$$
$$w(U_{h,j+1}, U_{i,j}) = \frac{\sum_{T_k \in A(U_{h,j+1})} \rho(T_k, A(U_{h,j+1}))}{\sum_{T_k \in A(U_{i,j})} \rho(T_k, A(U_{i,j}))} \tag{4}$$

Observe that the weights in eq. (3) are only the classical tf-idf weights, normalized to sum up one. The weights $w(U_{h,j+1}, U_{i,j})$ in eq. (4) measure, in some sense, the proportion of the content of the unit $U_{i,j}$ which can be attributed to each one of its components.

The inference process that we have to carry out with this model is to obtain a relevance value for each structural unit, 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 probabilities of relevance of all the structural units, given that the terms in the query are also relevant, $p(u^+_{i,j}|Q)$. Later, the documents are sorted according to their corresponding probability and shown to the user. Although this computation may be difficult in a general case, in our case all the conditional probabilities have been assessed using a specific canonical model 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:

[4] Notice that, although a unit $U_{i,j}$ in level $j \neq l$ is not connected directly to any term, it contains all the terms indexing structural units in level l that are included in $U_{i,j}$. Notice also that $A(U_{i,l}) = Pa(U_{i,l})$.

– For the structural units in level $\mathcal{L}_\ell$:

$$P(u^+_{i,l}|Q) = \sum_{T_k \in Pa(U_{i,l}) \cap Q} w(T_k, U_{i,l}) + \frac{1}{M} \sum_{T_k \in Pa(U_{i,l}) \setminus Q} w(T_k, U_{i,l})\,. \quad (5)$$

– For the structural units in level $\mathcal{L}_j$, $j \neq l$:

$$P(u^+_{i,j}|Q) = \sum_{U_{h,j+1} \in Pa(U_{i,j})} w(U_{h,j+1}, U_{i,j}) \cdot p(u^+_{h,j+1}|Q)\,. \quad (6)$$

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

3 Document Re-Ranking using Utility Theory

Once the probability of relevance has been computed for each structural unit, a ranking with all of them is generated. However, this ranking could show the user redundant information. Let us suppose that the top of the list of units is composed of the three subsections of a section from the same article, and the fourth item is that section. In this case, the system should detect this situation and decide to show either these three subsections or only the section, but not the four units. The problem, therefore, is to make a decision about what to retrieve, not only depending on the probability of relevance of the units but also in terms of the *usefulness* of these units for the user. One way to put into practice that idea is to use the *Decision Theory*, which would help making that decision which maximises the *Expected Utility*.

In a first step to achieve this goal, instead of deciding what to show to the user, i.e. to determine the *Best Entry Points* for a given query, the approach in this paper will be to modify the relevance value associated to each structural unit, taking into account the information involved in that decision. Taking up again the previous example, and as a consequence of this new relevance value, the section could change its position in the ranking overtaking its components.

The basis of the approach is the decision of retrieving (meaning showing it directly to the user) a structural unit, $U_{i,j}$, or not. This will be represented by introducing a decision variable, $R_{i,j}$, with possible values $r^+_{i,j}$ (retrieve $U_{i,j}$) and $r^-_{i,j}$ (do not retrieve $U_{i,j}$). The information that we shall use to make the decision is the relevance value of the own unit $U_{i,j}$ and that of the single unit, $U_{k,j-1}$, containing it (the $U_{i,j}$'s child in the network). An important element to make the corresponding decision is a *utility function*, $V(r_{ij})$, which assigns a value of utility to each possible decision $r_{i,j}$.

In our problem, the utility function $V(r_{ij})$ may be represented by means of a table that expresses the utility of making a decision, taking into account the values that both random variables, $U_{i,j}$ and $U_{k,j-1}$, could take. Therefore, for each different combination of possible units' values as well as decision's values, the values in table 1 express the corresponding user's utilities:

	$r^+_{i,j}$	$r^-_{i,j}$
$u^+_{i,j}\ u^+_{k,j-1}$	$v(r^+_{i,j} \mid u^+_{i,j}, u^+_{k,j-1}) \equiv v^+_{++}$	$v(r^-_{i,j} \mid u^+_{i,j}, u^+_{k,j-1}) \equiv v^-_{++}$
$u^+_{i,j}\ u^-_{k,j-1}$	$v(r^+_{i,j} \mid u^+_{i,j}, u^-_{k,j-1}) \equiv v^+_{+-}$	$v(r^-_{i,j} \mid u^+_{i,j}, u^-_{k,j-1}) \equiv v^-_{+-}$
$u^-_{i,j}\ u^+_{k,j-1}$	$v(r^+_{i,j} \mid u^-_{i,j}, u^+_{k,j-1}) \equiv v^+_{-+}$	$v(r^-_{i,j} \mid u^-_{i,j}, u^+_{k,j-1}) \equiv v^-_{-+}$
$u^-_{i,j}\ u^-_{k,j-1}$	$v(r^+_{i,j} \mid u^-_{i,j}, u^-_{k,j-1}) \equiv v^+_{--}$	$v(r^-_{i,j} \mid u^-_{i,j}, u^-_{k,j-1}) \equiv v^-_{--}$

Table 1. Utility function V for the decision node $R_{i,j}$, with $j \neq 1$.

For instance, $v(r^+_{i,j} \mid u^+_{i,j}, u^+_{k,j-1})$ is a value that represents the utility of showing unit $U_{i,j}$ to the user once that it is known that this unit is relevant and its child also is; $v(r^-_{i,j} \mid u^+_{i,j}, u^-_{k,j-1})$ is the utility of not retrieving $U_{i,j}$ when $U_{i,j}$ is relevant and $U_{k,j-1}$ is not, and so on. To simplify the notation, the utility values will be noted as v with a $+$ or $-$ superscript depending on the semantic of the decision, and two subscripts representing the meaning of the two unit variables. We are assuming that the utility values are uniform, i.e., they do not depend on the specific unit being considered, although this restriction is not necessary.

For the structural units in level 1, that are not contained in any other, the utility function is expressed with a different table because the decision of retrieving it only depends on itself. In this case, the table is composed of two rows, one for each value that this variable may take, and two columns, representing the two possible decisions, as may be noticed in table 2. The same notation as previously explained, but only with one subscript, is used for values in that table.

	$r^+_{i,1}$	$r^-_{i,1}$
$u^+_{i,1}$	$v(r^+_{i,1} \mid u^+_{i,1}) \equiv v^+_{+}$	$v(r^-_{i,1} \mid u^+_{i,1}) \equiv v^-_{+}$
$u^-_{i,1}$	$v(r^+_{i,1} \mid u^-_{i,1}) \equiv v^+_{-}$	$v(r^-_{i,1} \mid u^-_{i,1}) \equiv v^-_{-}$

Table 2. Utility function V for the structural units in level $j = 1$.

The *Expected Utility* of retrieving a structural unit $U_{i,j}$ in level $j \neq 1$, given the query submitted to the IR system is computed according to the following expression:

$$EU(r^+_{i,j} \mid Q) = \sum_{\substack{u_{i,j} \in \{u^+_{ij}, u^-_{ij}\} \\ u_{k,j-1} \in \{u^+_{k,j-1}, u^-_{k,j-1}\}}} v(r^+_{i,j} \mid u_{i,j}, u_{k,j-1}) \cdot p(u_{i,j}, u_{k,j-1} \mid Q) \tag{7}$$

Alternatively, the expected utility of not retrieving this same unit is the following:

$$EU(r^-_{i,j} \mid Q) = \sum_{\substack{u_{i,j} \in \{u^+_{ij}, u^-_{ij}\} \\ u_{k,j-1} \in \{u^+_{k,j-1}, u^-_{k,j-1}\}}} v(r^-_{i,j} \mid u_{i,j}, u_{k,j-1}) \cdot p(u_{i,j}, u_{k,j-1} \mid Q) \tag{8}$$

Analogously, the two expected utilities for units in level 1 are the following:

$$\begin{array}{l} EU(r^+_{i,1} \mid Q) = \sum_{u_{i,1} \in \{u^+_{i,1}, u^-_{i,1}\}} v(r^+_{i,1} \mid u_{i,1}) \cdot p(u_{i,1} \mid Q) \\ EU(r^-_{i,1} \mid Q) = \sum_{u_{i,1} \in \{u^+_{i,1}, u^-_{i,1}\}} v(r^-_{i,1} \mid u_{i,1}) \cdot p(u_{i,1} \mid Q) \end{array} \quad (9)$$

From a computational point of view, obtaining the joint probability of a structural unit and its child conditioned to the query, i.e., $p(u_{i,j}, u_{k,j-1} \mid Q)$, may be a time consuming process, because of the great amount of calculations required on retrieval time. Taking into account this drawback, in this paper and as a first stage to cope with the problem, it has been considered the simplifying assumption that both units are conditionally independent given the query. Therefore, this probability distribution is computed applying the following expression:

$$p(u_{i,j}, u_{k,j-1} \mid Q) = p(u_{i,j} \mid Q) \cdot p(u_{k,j-1} \mid Q) \quad (10)$$

4 Experimentation

The model has been tested using a collection of structured documents, marked up in XML, containing 37 William Shakespeare's plays [10]. 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 Retrieval Toolkit (available at http://www-2.cs.cmu.edu/~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 judgments. From these 43 queries, the 35 which are content-only queries were selected for our experiments. The system evaluation has been carried out using the average precision for the eleven standard recall values.

The experimental design carried out with this model tries to determine the contribution of the use of the expected utility on the final ranking of structural units. Therefore, once the first stage in which the posterior probability of each structural unit, $p(u^+_{i,j} \mid Q)$, has been computed, the second stage obtains the expected utility of each variable, combining these posterior probabilities and the utility function, achieving finally a second ranking.

With respect to the values contained in tables 1 and 2, and before giving values to them, it seemed to us interesting to sort them according to the utility of each value for the user. Therefore, the following ordering has been obtained using a small previous experimentation in which three users were asked to sort the values according to what they think it was more useful. All of them agreed in this ordering, being the one that has been applied in the first experiments:

$$v^-_{+-} \leq v^+_{--} \leq v^+_{-+} \leq v^+_{++} \leq v^-_{++} \leq v^-_{-+} \leq v^-_{--} \leq v^+_{+-} \quad (11)$$

From there, we could say that, for instance, to show to a user a non-relevant section in a relevant chapter (v^+_{-+}) is less useful than retrieving a relevant section in a relevant chapter (v^+_{++}), which in turns is less useful than not to present a relevant section within a relevant chapter (v^-_{++}, because in this case we would

prefer to present the complete chapter and not the section). The less useful decision would be not to show a relevant section in a non-relevant chapter (v_{+-}), because we would definitively lose relevant information (note that the chapter would not be presented either, since it also is not relevant). The most useful decision is to retrieve a relevant section in a non-relevant chapter (v^{+}_{+-}). As the utility function is usually normalised in the interval [0.0, 1.0], then the limits have been assigned to $v^{-}_{+-} = 0.0$, and $v^{+}_{+-} = 1.0$.

In the context of a usual decision problem, once the expected utilities have been calculated, we make the decision with greatest expected utility. In our case this would mean, for each structural unit $U_{i,j}$, to retrieve $U_{i,j}$ if $EU(r^{+}_{i,j} \mid Q) \geq EU(r^{-}_{i,j} \mid Q)$ and not to retrieve $U_{i,j}$ otherwise. However, we do not only wish to decide what units to retrieve but also to give a ranking of these units. Therefore, a second important design aspect is what technique we have to use to sort these units. The first natural approach is to rank them according to the expected utility of showing a unit, $EU(r^{+}_{ij} \mid Q)$. But there are two more natural options that also involve the expected utility of not showing the corresponding unit, $EU(r^{-}_{ij} \mid Q)$: the quotient between both expected utilities, $EU(r^{+}_{i,j} \mid Q)/EU(r^{-}_{i,j} \mid Q)$ and the difference $EU(r^{+}_{i,j} \mid Q) - EU(r^{-}_{i,j} \mid Q)$. These measures will be generically called Re-ranking Utility Measures (RUM) and denoted, respectively, RUM-u, RUM-q and RUM-d.

Therefore, the behaviour of this utility model depends on the utility function applied, as well as the expression of the expected utility used to rank the structural units. The aim of this experimentation has been to determine, if possible, a pattern that guarantees a good performance, by varying these two parameters.

All the re-ranking experiments with utilities are carried out using the same initial ranking of all the previously cited structural units from the test collection, obtained after performing the inference process described in subsection 2.1. The average precision for the 11 standard recall points of this running is 0.0653 [6].

Ex.	Measure	v^{-}_{+-}	v^{+}_{--}	v^{+}_{-+}	v^{+}_{++}	v^{-}_{++}	v^{-}_{-+}	v^{-}_{--}	v^{+}_{+-}	AVP-11	%C
1	RUM-u	0.0	0.1	0.2	0.3	0.6	0.7	0.9	1.0	0.0674	3.21%
2	RUM-q	0.0	0.1	0.2	0.3	0.6	0.7	0.9	1.0	0.0684	4.75%
3	RUM-d	0.0	0.1	0.2	0.3	0.6	0.7	0.9	1.0	0.0687	5.20%
4	RUM-u	0.0	0.0	0.955	0.960	1.0	1.0	1.0	1.0	0.0735	12.57%
5	RUM-q	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0726	11.17 %
6	RUM-d	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0705	7.96%
7	RUM-u	0.0	0.0	0.955	0.960	1.0	1.0	1.0	1.0	0.0735	12.57%
8	RUM-q	0.0	0.0	1.0	1.0	0.5	0.0	0.4	1.0	0.0740	13.32 %
9	RUM-d	0.0	0.0	0.85	1.0	0.05	0.0	1.0	1.0	0.0733	12.25%

Table 3. Summary of experimental results.

Table 3 shows some representative experiments numbered in the first column (Ex.), from the great amount of tests run. In it, the measure to re-rank documents (RUM), the different utility values, as well as the average precision (AVP-11) and its corresponding percentage of change with respect to the baseline (%C) are included.

In the first three experiments we used an increasing series of utility values according to eq. (11) with the three different RUM measures. It's noticeable a slight improvement in the performance of the system, fact that lead us to look for better combinations of utility values. By means of an intensive experimentation, where we tried many utility values, all of them satisfying the ordering restrictions imposed by eq. (11), the best results found correspond with the three next rows in table 3.

Our next objective concerned with the ordering of the utilities suggested by the users: Is the users' supposition correct? Could other different combinations of utility values that do not verify the ordering restriction obtain good or even better average precision? To find the answer other series of experiments were run, but in this case without imposing any ordering restriction. The best values obtained are shown in the last three rows of table 3.

Studying the results and the different combinations of utility values in all the experiments, it could be noted that the best performance for RUM-u is obtained when the utilities involved in the expected utility of retrieving a unit are sorted increasingly ($v^+_{--} \leq v^+_{-+} \leq v^+_{++} \leq v^+_{+-}$), and are close to 1.0, except the first, which is not useful at all. The rest of utility values are not taken into account by RUM-u, and therefore their values do not matter. This behaviour of RUM-u seems to point toward a conservative strategy (probably recall-enhancing), where it is very useful to retrieve a relevant unit irrespective of the relevance of its context (v^+_{++} and v^+_{+-} values) and it is also quite useful to retrieve non-relevant units if their context is relevant (v^+_{-+} value).

The other two RUM measures, RUM-q and RUM-d, also exhibit the same pattern for the utility values of retrieving a unit. Now, if we focus our attention on the other values, corresponding to the utilities of not retrieving units, v^-_{++}, v^-_{-+} and v^-_{--} (v^-_{+-} is always set to 0) for RUM-d and RUM-q, data usually shows crossed values for v^-_{++} and v^-_{--}. When in RUM-q the former is greater than the latter, in RUM-d the opposite situation occurs. Moreover, in RUM-q both values tend to be quite similar, whereas in RUM-d they are usually close to the extremes, i.e., $v^-_{++} \simeq 0.0$ and $v^-_{--} \simeq 1.0$. A surprising fact is that for both RUM measures, the utility of not retrieving a unit which is not relevant, contained in a relevant one, v^-_{-+}, is null, when our first hypothesis considered that it should be a rather high number.

Summing up, a good pattern when the RUM-u measure is being used is to follow the ordering in eq. (11), with high values for those utility values involved in the expected utility of retrieving a unit, except v^+_{--} that is assigned to 0.0. For RUM-q and RUM-d, it is more or less the same pattern for those utility values, and for those which are used in the computation of the expected utility of not retrieving a unit, v^-_{-+} should be very low, almost 0.0. v^-_{++} and v^-_{--} should be very

similar and around 0.5 for RUM-q and extreme for RUM-d. In all these cases, the performance improvement with respect to the baseline ranking obtained by using only the posterior probabilities of relevance computed from the Bayesian network, is above 12%[5].

5 Conclusions

This paper is framed as a first approach to solve a decision making problem, in which the IR system has to decide whether to retrieve or not a structural unit from a structured document collection, given a query submitted by a user. Instead of making this decision, this work presents a new way of re-ranking the structural units according to the expected utility of showing each unit, or by means of a variation in which the expected utility of not retrieving the corresponding unit is also involved.

Taking into account the experimental collection used to test the model, its performance could be described as rather good although could be clearly improved. The utility theory applied to re-rank structured documents seems to be promising. The main purpose of the experimentation has been to find patterns for the utility functions that present a good performance with the different RUM measures.

Of course we are conscious that the conclusions of this experimentation are completely related to the collection with which it has been carried out, and specially the relevance judgments, being able to change if the test bed is different. As a future work, the BN model with the utility module will be applied to other structured collections, as INEX, to test if the same patterns of utility are fulfilled.

To improve the results, one action to be taken could be to remove the simplifying assumption about the independence of a unit and the unit where it is included, given the query (eq. 10). To put it into practice, the probabilities $p(u_{ij}, u_{kj-1} \mid Q)$ have to be computed, preferably in an exact and efficient way. With this assumption, the utility model could be completely represented by means of an influence diagram [16], providing a clear semantics and a solid frame.

Only one utility function has been considered for all the layers in the model, although another approach could be to use a different one for each type of structural unit or layer, thus giving the possibility of assigning particularised utility values to them, modeling user's preferences.

The next stage is to use the model to determine the best entry points for a query. This task means to put into practice the whole decision making process, determining what to show to the user, and not only providing a ranking as it has been done in this paper.

Regarding the Bayesian network topology, other tasks to be done are to represent the specific textual information assigned to structural units in levels different

[5] We have not carried out a comparison of our results with other systems. The reason is that we only are aware of a paper containing empirical results with the same test collection [11], and there the results are obtained from a (unknown) subset containing only 25 queries from the 35 Shakespeare collection's content-only queries.

from l (for example the title of a chapter or a section) and to allow direct relationships between units in non-consecutive levels of the hierarchy (e.g. paragraphs and chapters). Also, to permit our model to deal, not only with content-only queries, but also with structure-only and content-and-structure queries and let the queries to include, in addition to terms, also structural units.

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. R. Baeza-Yates and B. Ribeiro-Nieto. *Modern Information Retrieval*. Addison-Wesley, Harlow, UK, 1999.
3. C. Baumgarten. A probabilistic model for distributed information retrieval. In *Proceedings of ACM–SIGIR Conference*, 258–266, 1997.
4. G. Bordogna and G. Pasi. Flexible representation and querying of heterogeneous structured documents. *Kibernetika*, 36(6):617–633, 2000.
5. Y. Chiaramella. Information retrieval and structured documents. *Lectures Notes in Computer Science*, 1980:291–314, 2001.
6. F. Crestani, L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. A multilayered Bayesian network model for structured document retrieval. *Lecture Notes in Computer Science*, 2711:74-86,2003.
7. 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.
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. S. French Decision Theory. An introduction to the Mathematics of Rationality. Ellis Horwood Limited, Wiley, 1986.
10. G. Kazai, M. Lalmas, and J. Reid. The Shakespeare test collection. Available at http://qmir.dcs.qmul.ac.uk/Focus/resources2.htm
11. G. Kazai, M. Lalmas, and T. Roelleke. Focussed structured document retrieval. *Lecture Notes in Computer Science*, 2476:241–247, 2002.
12. 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.
13. 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.
14. B. Piwowarski, G.E. Faure, and P. Gallinari. Bayesian networks and INEX. In *Proceedings of the INEX Workshop*, 7–12, 2002.
15. 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.
16. R. D. Shachter. Probabilistic Inference and Influence Diagrams. *Operations Research*, 36(5):527–550, 1988.