Improving the Context-Based Influence Diagram Model for Structured Document Retrieval: Removing Topological Restrictions and Adding New Evaluation Methods

Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete

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 we present the theoretical developments necessary to extend the existing Context-based Influence Diagram Model for Structured Documents (CID model), in order to improve its retrieval performance and expressiveness. Firstly, we make it more flexible and general by removing the original restrictions on the type of structured documents that CID represents. This extension requires the design of a new algorithm to compute the posterior probabilities of relevance. Another contribution is related to the evaluation of the influence diagram. The computation of the expected utilities in the original CID model was approximated by applying an independence criterion. We present another approximation that does not assume independence, as well as an exact evaluation method.

1 Introduction

Document collections in the Information Retrieval (IR) field have been considered as composed of only textual information for a long time [1]. Information Retrieval Systems (IRS) represented their contents by means of index terms, and they were mostly the only tool to retrieve the relevant documents given the users' information needs. Nowadays, the internal structure of these documents is taking more and more importance, basically due to the development of new formalisms, like SGML and XML, that contribute with features to easily represent a well defined structure, in order to organize the document contents comprehensibly and also to facilitate the reading to the user. Therefore, the aim of new IRSs has changed: by also using the document organization, instead of returning a relevant document as a whole, these applications will retrieve the set of document components (structural units) more relevant to a query (chapters, sections, or paragraphs in a book, for example), giving as a result a new research subarea on structured documents [2].

Classical probabilistic IRSs [4] rank the documents by considering their probability of relevance to a given query. In these systems, the action of retrieving (or not) a document is independent on the action of retrieving (or not) any other

D.E. Losada and J.M. Fernández-Luna (Eds.): ECIR 2005, LNCS 3408, pp. 215–229, 2005.

document. However, this is no longer true when dealing with structured documents, where the decision about retrieving a document component clearly may affect the retrieval of other components (for example, it makes no sense to retrieve two sections of a chapter and also the complete chapter itself). Therefore, it is clear that not only the probability of relevance has to be used to retrieve the document components, but we could also use the *usefulness* of these components for the user, taking into account the context where they are placed and what has been previously retrieved.

Following this direction, the Context-based Influence Diagram model for Structured Documents (CID model) [6] was born with the capability of making decisions about which document components should be retrieved, not only depending on their probability of relevance, but also on their *utility* for the user and the influences provided by the context in which each structural component is located. This is carried out by means of an *Influence Diagram* (ID) [11], a generalization of the well founded Bayesian network formalism [13] in the context of Decision Theory [8]. Examples in the specialised literature about the application of Bayesian networks to Structured Information Retrieval are [3, 9, 12, 14], although the CID model is the only one, as far as we know, that applies IDs.

However, the CID model presents an important restriction on the structure of the documents that it can represent: the documents have to be composed of a strict set of structural layers. So, the structural units from the j-th layer (all of them being of the same type) must be included in broader units from the $(j-1)$-th layer and so on (except for the units from the first layer). The last layer would contain the smallest structural units, composed only of text and not of other units. The CID model was endowed with an efficient propagation algorithm to compute the posterior probability of relevance of each unit given a query, which was specifically designed to deal with this strict structure. In this paper we extend this model to work with a general document organization, where the rule of strict layers is broken and textual units can be placed anywhere as well, reformulating the original propagation algorithm.

A second contribution is the development of two new mechanisms to evaluate the underlying influence diagram of the CID model. Solving an ID means to determine the expected utility of each one of the possible decisions, for those situations of interest, with the aim of making decisions which maximize the expected utility [15]. The expected utility in the CID model depends on the bi-dimensional posterior probabilities, corresponding to each structural unit and the unit where it is contained. In [6], and in order to simplify the computations, it was assumed that the two involved units were independent given the query, so the bi-dimensional distributions could be approximated just multiplying the unidimensional posterior probabilities of each unit given the query. In this paper we present, on the one hand, an exact evaluation method that computes the bi-dimensional distributions and, on the other hand, another efficient and more precise approximated evaluation method.

In order to describe precisely these ideas and formalize them, this paper is organised in the following way: In Section 2 we briefly introduce some preliminary

concepts: we assume a basic knowledge about Bayesian networks to the reader and only provide some background about influence diagrams. Section 3 describes the type of structured documents being considered. The next two sections introduce the model: Section 4 presents the Bayesian network that graphically represents the structure of the documents, and the corresponding influence diagram is described in Section 5. Section 6 explains how to use the model for retrieval purposes by computing the expected utilities of the document components. Formulas for the posterior probabilities which are necessary to carry out this computation are described in Section 7. Section 8 gives an illustrative example. Finally, Section 9 contains the concluding remarks.

2 Background: Influence Diagrams

Influence Diagrams [11, 16] are probabilistic graphical models that provide a simple notation for designing decision models by clarifying the qualitative issues of what factors need to be considered and how they are related, i.e. an intuitive representation of the model. They have also associated an underlying quantitative representation in order to measure the strength of the relationships: we can quantify uncertain interactions among random variables and also the decision maker's options and preferences. The model is used to determine the optimal decision policy. IDs contain three types of nodes: (a) *Decision nodes* (drawn as rectangles) represent variables that the decision maker controls directly, and model the decision alternatives available for the decision maker. (b) *Chance nodes* (drawn as circles) represent random variables, i.e. uncertain quantities that are relevant to the decision problem and can not be controlled directly, quantified by means of conditional probability distributions. (c) *Utility nodes* (drawn as diamonds) represent utility, i.e. express the profit or the preference degree of the consequences derived of the decision process, and are quantified by the utility of each of the possible combinations of outcomes of their parent nodes.

There are also different types of arcs in an influence diagram: arcs between chance nodes represent probabilistic dependences (note that the subgraph containing only chance nodes and the related arcs is a Bayesian network). Arcs from a decision node to a chance node or to a utility node establish that the future decision will influence the value of the chance node or in the profit obtained, respectively. Finally, arcs from a chance node to a utility node will express that the profit will depend on the value that this chance node takes.

3 Type of Structured Documents

We start with a document collection containing M *documents*, $\mathcal{D}=\{D_1, \ldots, D_M\}$, and the set of the *terms* used to index these documents (the glossary of the collection). We assume that each document D_i is organized hierarchically, representing structural associations of elements in D_i, which will be called *structural units*. Each structural unit is composed of other smaller structural units, except some

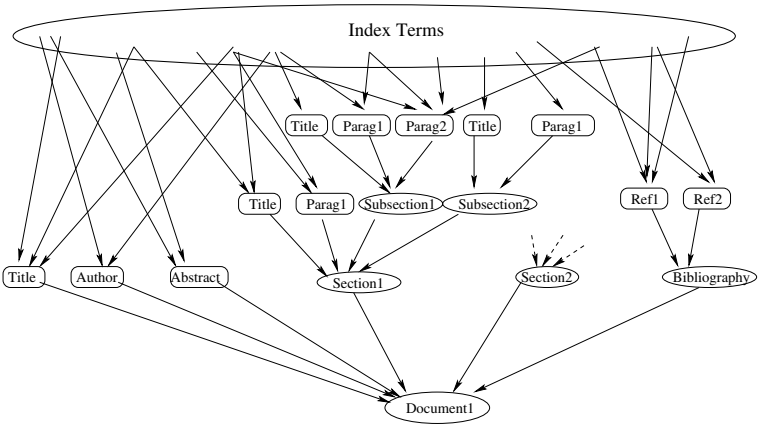

Fig. 1. Example of the structure of a scientific article

'terminal' or 'minimal' units which are indivisible, they do not contain any other unit. Instead, these are composed of terms: each term used to index the complete document D_i will be assigned to all the terminal units containing it. Conversely, each structural unit, except the one corresponding to the complete document, is included in only one structural unit. Therefore, the structural units associated to a document D_i form a (inverted) tree. There is not any restriction about this tree structure, which contrasts with the rigid structure considered in [6], where all the paths from the root to the leaves have the same length.

For instance, a scientific article may contain a title, authors, abstract, sections and bibliography; sections may contain a title, subsections and paragraphs; in turn subsections contain paragraphs and perhaps also a title; the bibliography contains references; titles, authors, paragraphs, abstract and references would be in this case the terminal structural units (see Figure 1).

4 The Underlying Bayesian Network

The Bayesian network will contain two kinds of nodes, representing the terms and the structural units. The former will be represented by the set $\mathcal{T} = \{T_1, T_2, \ldots, T_l\}$. There are two types of structural units: *basic structural units*, those which only contain terms, and *complex structural units*, that are composed of other basic or complex units. The notation for these nodes is $\mathcal{U}_b = \{B_1, B_2, \ldots, B_m\}$ and $\mathcal{U}_c = \{S_1, S_2, \ldots, S_n\}$, respectively. Therefore, the set of all structural units is $\mathcal{U} = \mathcal{U}_b \cup \mathcal{U}_c$. In this paper, T or T_k will represent a term; B or B_i a basic structural unit, and S or S_j a complex structural unit. Generic structural units (either basic or complex) will be denoted as U_i or U. Each node T, B or S has associated a binary random variable[1], which can take its values from the

[1] In this paper, the random variable and its associated node in the graph will be noted identically.

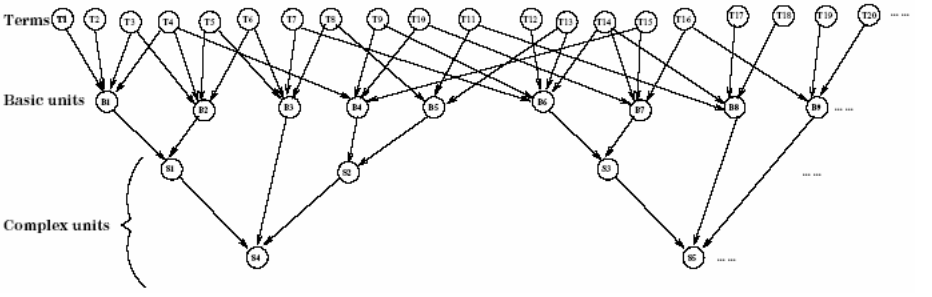

Fig. 2. Bayesian network representing a structured document collection

sets $\{t^-, t^+\}$, $\{b^-, b^+\}$ or $\{s^-, s^+\}$ (the term/unit is not relevant or is relevant), respectively. A unit is relevant for a given query if it satisfies the user's information need expressed by this query. A term is relevant in the sense that the user believes that it will appear in relevant units/documents.

Regarding the arcs of the model, there is an arc from a given node (either term or structural unit) to the particular structural unit node it belongs to, expressing the fact that the relevance of a given structural unit to the user will depend on the relevance values of the different elements (units or terms) that comprise it. It should be noted that with this criteria, terms nodes have no parents. Formally, the network is characterized by the following parent sets, $Pa(.)$:

- $\forall T \in \mathcal{T}$, $Pa(T) = \emptyset$.
- $\forall B \in \mathcal{U}_b$, $\emptyset \neq Pa(B) \subseteq \mathcal{T}$.
- $\forall S \in \mathcal{U}_c$, $\emptyset \neq Pa(S) \subseteq \mathcal{U}_b \cup \mathcal{U}_c$, with $Pa(S_1) \cap Pa(S_2) = \emptyset$, $\forall S_1 \neq S_2 \in \mathcal{U}_c$.

It should be noticed that the hierarchical structure of the model determines that each structural unit $U \in \mathcal{U}$ has only one structural unit as its child, the unique structural unit containing U (except for the leaf nodes, i.e. the complete documents, which have no child). We shall denote indistinctly by $Hi(U)$ or $U_{hi(U)}$ the single child node associated with node U (with $Hi(U) = null$ if U is a leaf node). Figure 2 displays an example of the proposed network topology.

The numerical values for the conditional probabilities have also to be assessed: $p(t^+)$, $p(b^+|pa(B))$, $p(s^+|pa(S))$, for every node in $\mathcal{T}$, $\mathcal{U}_b$ and $\mathcal{U}_c$, respectively, and every configuration of the corresponding parent sets ($pa(X)$ denotes a configuration or instantiation of the parent set of X, $Pa(X)$). Once specified, the network may be used to compute the posterior probabilities of relevance of all the structural units $U \in \mathcal{U}$ for a given query.

In our case, the number of terms and structural units considered may be quite large (thousands or even hundreds of thousands). Moreover, the topology of the Bayesian network contains multiple pathways connecting nodes (because the terms may be associated to different basic structural units) and possibly nodes with a great number of parents (so that it can be quite difficult to assess and store the required conditional probability tables). For these reasons we shall use the canonical model to represent the conditional probabilities proposed in [5] (as the

CID model does), which supports a very efficient inference procedure. We have to consider the conditional probabilities for the basic structural units, having a subset of terms as their parents, and for the complex structural units, having other structural units as their parents. These probabilities are defined as follows:

$$\forall B \in \mathcal{U}_b,\, p(b^+|pa(B)) = \sum_{T \in R(pa(B))} w(T,B)\,, \tag{1}$$

$$\forall S \in \mathcal{U}_c,\, p(s^+|pa(S)) = \sum_{U \in R(pa(S))} w(U,S)\,, \tag{2}$$

where $w(T,B)$ is a weight associated to each term T belonging to the basic unit B, $w(U,S)$ is a weight measuring the importance of the unit U within S. In any case $R(pa(U))$ is the subset of parents of U (terms for B, and either basic or complex units for S) relevant in the configuration $pa(U)$, i.e., $R(pa(B)) = \{T \in Pa(B) \,|\, t^+ \in pa(B)\}$ and $R(pa(S)) = \{U \in Pa(S) \,|\, u^+ \in pa(S)\}$. So, the more parents of U are relevant the greater the probability of relevance of U. These weights can be defined in any way, the only restrictions are that $w(T,B) \geq 0$, $w(U,S) \geq 0$, $\sum_{T \in Pa(B)} w(T,B) \leq 1$, and $\sum_{U \in Pa(S)} w(U,S) \leq 1$. For example, they can be defined using a normalized tf-idf scheme, as in [6], or we could also consider the relative importance of each type of unit (for example, the title or the abstract could be more representative of the content of a document than a section).

With respect to the prior probabilities of relevance of the terms, $p(t^+)$, they can also be defined in any reasonable way, for example an identical probability for all the terms, $p(t^+) = p_0$, $\forall T \in \mathcal{T}$, as proposed in [6].

5 The Influence Diagram Model

Once the Bayesian network has been constructed, it is enlarged by including decision and utility nodes, thus transforming it into an influence diagram. We use the same topology proposed in [6] for the CID model: a) *Decision nodes*: One decision node, R_i, for each structural unit $U_i \in \mathcal{U}$. R_i represents the decision variable related to whether or not to return the structural unit U_i to the user. The two different values for R_i are r_i^+ and r_i^-, meaning 'retrieve U_i' and 'do not retrieve U_i', respectively. b) *Utility nodes*: One of these, V_i, for each structural unit $U_i \in \mathcal{U}$, will measure the value of utility of the corresponding decision.

In addition to the arcs between chance nodes (already present in the Bayesian network), a set of arcs pointing to utility nodes are also included, employed to indicate which variables have a direct influence on the desirability of a given decision, i.e. the profit obtained will depend on the value of these variables. In order to represent that the utility function of V_i obviously depends on the decision made and the relevance value of the structural unit considered, we use arcs from each chance node U_i and decision node R_i to the utility node V_i. Another important set of arcs are those going from $Hi(U_i)$ to V_i, which represent that the utility of the decision about retrieving the unit U_i also depends on the relevance of the unit which contains it (obviously, for the units which are

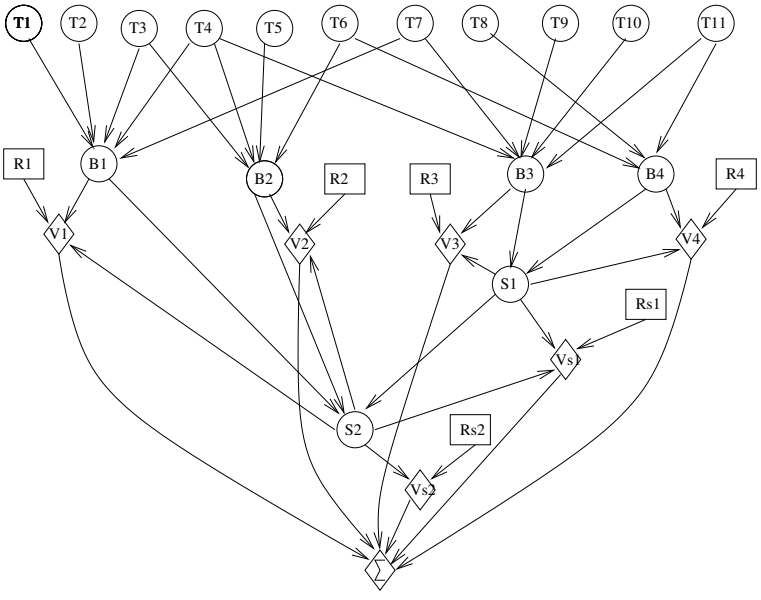

Fig. 3. Topology of the influence diagram

not contained in any other unit these arcs do not exist). This last kind of arc allows us to represent the context-based information and can avoid redundant information being shown to the user. For instance, we could express the fact that on the one hand, if U_i is relevant and $Hi(U_i)$ is not, then the utility of retrieving U_i should be large (and the one of not retrieving it almost null). On the other hand, if $Hi(U_i)$ is relevant, even if U_i were also relevant the utility of retrieving U_i should be small because, in this case, it would be preferable to retrieve the largest unit as a whole, instead of each of its components separately.

Another utility node, denoted by Σ, that represents the joint utility of the whole model is also considered. It has all the utility nodes V_j as its parents. These arcs represent the fact that the joint utility of the model will depend (additively) on the values of the individual utilities of each structural unit. Figure 3 displays an example of the topology of the influence diagram being considered.

Moreover, the influence diagram requires numerical values for the utilities. For each utility node V_i we need eight numbers, one for each combination of values of the decision node R_i and the chance nodes U_i and $Hi(U_i)$ (except for the leaf nodes, which only require four values). These values are represented by $v(r_i, u_i, u_{hi(U_i)})$, with $r_i \in \{r_i^-, r_i^+\}$, $u_i \in \{u_i^-, u_i^+\}$, and $u_{hi(U_i)} \in \{u_{hi(U_i)}^-, u_{hi(U_i)}^+\}$.

6 Solving the Influence Diagram

To solve an influence diagram, the expected utility of each possible decision (for those situations of interest) has to be computed, thus making decisions which

maximize the expected utility. In our case, the situation of interest corresponds with the information provided by the user when he/she formulates a query. Let $\mathcal{Q} \subseteq \mathcal{T}$ be the set of terms used to express the query. Each term $T_i \in \mathcal{Q}$ will be instantiated to either t_i^+ or t_i^-; let q be the corresponding configuration of the variables in $\mathcal{Q}$. We wish to compute the expected utility of each decision given q. As we have assumed a global additive utility model, and the different decision variables R_i are not directly linked to each other, we can process eachone independently. The expected utilities for each U_i can be computed by means of

$$EU(r_i^+ \mid q) = \sum_{\substack{u_i \in \{u_i^-, u_i^+\} \\ u_{hi(U_i)} \in \{u_{hi(U_i)}^-, u_{hi(U_i)}^+\}}} v(r_i^+, u_i, u_{hi(U_i)})\, p(u_i, u_{hi(U_i)}|q)\,. \tag{3}$$

$$EU(r_i^- \mid q) = \sum_{\substack{u_i \in \{u_i^-, u_i^+\} \\ u_{hi(U_i)} \in \{u_{hi(U_i)}^-, u_{hi(U_i)}^+\}}} v(r_i^-, u_i, u_{hi(U_i)})\, p(u_i, u_{hi(U_i)}|q)\,. \tag{4}$$

In the context of a typical decision making problem, once the expected utilities are computed, the decision with greatest utility is chosen: this would mean to retrieve the structural unit U_i if $EU(r_i^+|q) \geq EU(r_i^-|q)$, and not to retrieve it otherwise. However, our purpose is not only to make decisions about what to retrieve but also to give a ranking of those units. The simplest way to do it is to show them in decreasing order of the utility of retrieving U_i, $EU(r_i^+|q)$[2]. In this case only four utility values have to be assessed, and only eq. (3) is required.

7 Computing Probabilities

In order to provide to the user an ordered list of structural units, we have to be able to compute the posterior probabilities of relevance of all the structural units $U \in \mathcal{U}$, $p(u^+|q)$, and also the bi-dimensional posterior probabilities, $p(u^+, u_{hi(U)}^+|q)$[3]. The specific characteristics of the canonical model used to define the conditional probabilities will allow us to efficiently compute the posterior probabilities[4].

7.1 Calculus of Unidimensional Posterior Probabilities

Proposition 1

$$\forall B \in \mathcal{U}_b,\ \ p(b^+|q) = \sum_{T \in Pa(B) \setminus Q} w(T,B)\, p(t^+) + \sum_{T \in Pa(B) \cap R(q)} w(T,B)\,. \tag{5}$$

[2] Other options would also be possible to generate a ranking, as for example to use the difference between both expected utilities, $EU(r_i^+|q) - EU(r_i^-|q)$.

[3] Notice that the other required bi-dimensional probabilities, $p(u^+, u_{hi(U)}^-|q)$, $p(u^-, u_{hi(U)}^+|q)$ and $p(u^-, u_{hi(U)}^-|q)$, can be easily computed from $p(u^+, u_{hi(U)}^+|q)$, $p(u^+|q)$ and $p(u_{hi(U)}^+|q)$.

[4] Proofs of the results stated in the paper are not included due to space limitations. They can be found in [7].

$$\forall S \in \mathcal{U}_c,\ \ p(s^+|q) = \sum_{U \in Pa(S)} w(U,S)\, p(u^+|q)\,. \tag{6}$$

As we can see, the posterior probabilities of the basic units can be computed directly. The posterior probabilities of the complex units can be calculated in a top-down manner, starting from those for the basic units. However, it is possible to design a more direct inference method. We need some additional notation: $\forall S \in \mathcal{U}_c$, let $A_b(S) = \{B \in \mathcal{U}_b \,|\, B \text{ is an ancestor of } S\}$, $A_c(S) = \{S' \in \mathcal{U}_c \,|\, S' \text{ is an ancestor of } S\}$, and $\forall B \in \mathcal{U}_b$, let $D_c(B) = \{S \in \mathcal{U}_c \,|\, S \text{ is a descendant of } B\}$. Notice that, for each basic unit B in $A_b(S)$, there is only one path going from B to S. Let us define the weight $w(B,S)$ as the product of the weights of the arcs in the path from B to S, i.e. $w(B,S) = w(B, Hi(B)) \prod_{S' \in A_c(S) \cap D_c(B)} w(S', Hi(S'))$. Then, we get the following result:

Proposition 2

$$\forall S \in \mathcal{U}_c,\ p(s^+|q) = \sum_{B \in A_b(S)} w(B,S)\, p(b^+|q)\,. \tag{7}$$

Proposition 2 states that we can compute the posterior probability of a complex structural unit S by calculating the average of the posterior probabilities of all the basic structural units B contained in S, each probability being weighted by the product of the weights of the arcs in the (single) path going from B to S. This result is the basis to develop an inference process able to compute all the posterior probabilities of the structural units in a single traversal of the graph, starting only from the instantiated terms in $\mathcal{Q}$, provided that the prior probabilities of relevance have been calculated and stored within the structure:

Proposition 3

$$\forall B \in \mathcal{U}_b,\ p(b^+|q) = p(b^+) + \sum_{T \in Pa(B) \cap R(q)} w(T,B)\,\big(1 - p(t^+)\big) - \sum_{T \in Pa(B) \cap (Q \setminus R(q))} w(T,B)\, p(t^+) \tag{8}$$

$$\forall S \in \mathcal{U}_c,\ p(s^+|q) = p(s^+) + \sum_{\substack{B \in A_b(S) \\ Pa(B) \cap Q \neq \emptyset}} w(B,S)\ \big(p(b^+|q) - p(b^+)\big)\ . \tag{9}$$

This result indicates how we can compute the posterior probabilities from the prior probabilities traversing the nodes in the graph that will require updating. An algorithm that computes all the posterior probabilities $p(b^+|q)$ and $p(s^+|q)$, based on Proposition 3, starts from the terms in $\mathcal{Q}$ and carries out a width graph traversal until it reaches the basic units that require updating, computing $p(b^+|q)$ using eq. (8). Starting from these modified basic units, it carries out a depth graph traversal to compute $p(s^+|q)$, only for those complex units that require updating, using eq. (9). This algorithm needs the previous computation and storage of the nodes' prior probabilities. This can be done easily using Propositions 1 and 2 (with $q = \emptyset$).

7.2 Calculus of Bi-dimensional Posterior Probabilities

Now, the required probabilities are the posterior bi-dimensional probabilities $p(u^+, u^+_{hi(U)}|q)$, for any structural unit $U \in \mathcal{U}$ and its unique child $U_{hi(U)}$, provided that $U_{hi(U)} \neq null$. We have to distinguish two cases, depending on whether U is a basic unit ($U \in \mathcal{U}_b$) or a complex unit ($U \in \mathcal{U}_c$). The following two propositions provide formulas to compute these bi-dimensional probabilities.

Proposition 4. *$\forall S \in \mathcal{U}_c$, $\forall B \in \mathcal{U}_b$ such that $B \in Pa(S)$,*

$$p(s^+, b^+|q) = \sum_{\substack{B_i \in A_b(S) \\ B_i \neq B}} w(B_i, S)\, p(b_i^+, b^+|q) \;+\; w(B, S) p(b^+|q)\,. \tag{10}$$

Proposition 5. *$\forall S_1, S_2 \in \mathcal{U}_c$ such that $S_1 \in Pa(S_2)$,*

$$p(s_1^+, s_2^+|q) = \sum_{B_1 \in A_b(S_1)} \sum_{B_2 \in A_b(S_2) \setminus A_b(S_1)} w(B_1, S_1) w(B_2, S_2)\, p(b_1^+, b_2^+|q) + w(S_1, S_2) p(s_1^+|q). \tag{11}$$

These results, which are analogous to Proposition 2 in the unidimensional case, show that we can compute the required bi-dimensional probabilities as soon as we compute the bi-dimensional probabilities for pairs of basic structural units in $\mathcal{U}_b$ and the unidimensional probabilities of all the structural units in $\mathcal{U}_b \cup \mathcal{U}_c$. The following proposition shows how these bi-dimensional probabilities for pairs of basic structural units can be computed.

Proposition 6. *$\forall B_1, B_2 \in \mathcal{U}_b$, let us define*

$$\delta(B_1, B_2|q) \;=\; \sum_{T \in (Pa(B_1) \cap Pa(B_2)) \setminus \mathcal{Q}} w(T, B_1) w(T, B_2)\, p(t^+)(1 - p(t^+))\,. \tag{12}$$

Then

$$\begin{aligned} p(b_1^+, b_2^+|q) &= p(b_1^+|q)\, p(b_2^+|q) + \delta(B_1, B_2|q) \\ p(b_1^+, b_2^-|q) &= p(b_1^+|q)\, p(b_2^-|q) - \delta(B_1, B_2|q) \\ p(b_1^-, b_2^+|q) &= p(b_1^-|q)\, p(b_2^+|q) - \delta(B_1, B_2|q) \\ p(b_1^-, b_2^-|q) &= p(b_1^-|q)\, p(b_2^-|q) + \delta(B_1, B_2|q) \end{aligned} \tag{13}$$

The results in Proposition 6 state that the bi-dimensional probabilities can be expressed as the product of the unidimensional probabilities, plus a factor that outweighs the common relevance or irrelevance of the units and penalizes relevance of one unit and irrelevance of the other. This factor, $\delta(B_1, B_2|q)$, depends essentially of the number of common terms for B_1 and B_2 which are not instantiated. So, if two basic units do not share any term, or all the shared terms are instantiated, $\delta(B_1, B_2|q) = 0$ and the units are independent. On the other hand, the more uninstantiated terms share B_1 and B_2, the greater $\delta(B_1, B_2|q)$ and the more degree of dependence between these units exists.

This way of expressing the bi-dimensional probabilities of the basic units as a product of marginals plus an interaction factor, can be extended to the other cases, as the following two propositions show.

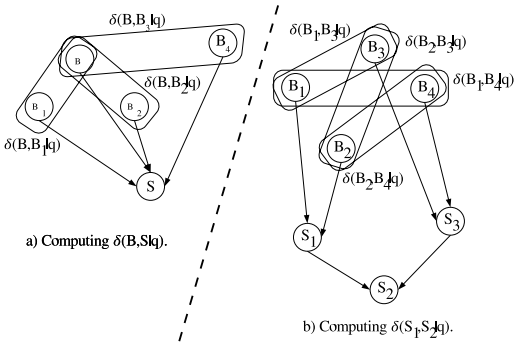

Fig. 4. Graphical representations of some δ interactions

Proposition 7. *$\forall S \in \mathcal{U}_c$, $\forall B \in \mathcal{U}_b$ such that $B \in Pa(S)$, let us define*

$$\delta(B, S|q) = \sum_{\substack{B_i \in A_b(S) \\ B_i \neq B}} w(B_i, S)\, \delta(B, B_i|q)\,. \tag{14}$$

Then

$$\begin{aligned}
p(s^+, b^+|q) &= p(s^+|q)\, p(b^+|q) \;+\; \delta(B, S|q) \;+\; w(B, S)\, p(b^+|q)\, (1 - p(b^+|q)) \\
p(s^+, b^-|q) &= p(s^+|q)\, p(b^-|q) \;-\; \delta(B, S|q) \;-\; w(B, S)\, p(b^+|q)\, (1 - p(b^+|q)) \\
p(s^-, b^+|q) &= p(s^-|q)\, p(b^+|q) \;-\; \delta(B, S|q) \;-\; w(B, S)\, p(b^+|q)\, (1 - p(b^+|q)) \\
p(s^-, b^-|q) &= p(s^-|q)\, p(b^-|q) \;+\; \delta(B, S|q) \;+\; w(B, S)\, p(b^+|q)\, (1 - p(b^+|q))
\end{aligned} \tag{15}$$

The interaction factor between a complex unit S and a basic unit B, $\delta(B, S|q)$, is a weighted average of the interaction factors between B and the basic units (different from B) included in S. The other value in eq. (15), $w(B, S)p(b^+|q)(1 - p(b^+|q))$, can be considered as a kind of interaction of B with itself. The first part of figure 4 shows the interactions computed according to eq. (15).

Proposition 8. *$\forall S_1, S_2 \in \mathcal{U}_c$ such that $S_1 \in Pa(S_2)$, let us define*

$$\delta(S_1, S_2|q) = \sum_{B_1 \in A_b(S_1)} \sum_{B_2 \in A_b(S_2) \setminus A_b(S_1)} w(B_1, S_1) w(B_2, S_2)\, \delta(B_1, B_2|q)\,. \tag{16}$$

Then

$$\begin{aligned}
p(s_1^+, s_2^+|q) &= p(s_1^+|q)\, p(s_2^+|q) + \delta(S_1, S_2|q) + w(S_1, S_2)\, p(s_1^+|q)\, (1 - p(s_1^+|q)) \\
p(s_1^+, s_2^-|q) &= p(s_1^+|q)\, p(s_2^-|q) - \delta(S_1, S_2|q) - w(S_1, S_2)\, p(s_1^+|q)\, (1 - p(s_1^+|q)) \\
p(s_1^-, s_2^+|q) &= p(s_1^-|q)\, p(s_2^+|q) - \delta(S_1, S_2|q) - w(S_1, S_2)\, p(s_1^+|q)\, (1 - p(s_1^+|q)) \\
p(s_1^-, s_2^-|q) &= p(s_1^-|q)\, p(s_2^-|q) + \delta(S_1, S_2|q) + w(S_1, S_2)\, p(s_1^+|q)\, (1 - p(s_1^+|q))
\end{aligned} \tag{17}$$

The interaction factor between two complex units S_1 ans S_2 is also a weighted average of the interaction factors between the basic units included in S_1 and those

included in S_2 but not in S_1. The other value in eq. (17), $w(S_1, S_2)p(s_1^+|q)(1-p(s_1^+|q))$, acts as a kind of interaction between the basic units included in S_1 with themselves. The second part of figure 4 shows the involved interactions.

Therefore, in the light of the results in Propositions 6, 7 and 8, to compute the bi-dimensional posterior probabilities, in addition to the calculus of the posterior unidimensional probabilities, we only need to compute the interactions $\delta(B_1, B_2|q)$, $\delta(B, S|q)$ and $\delta(S_1, S_2|q)$.

In order to design an algorithm to compute the interactions $\delta(B_1, B_2|q)$ between pairs of basic units, and considering that the number of terms instantiated in $\mathcal{Q}$ will usually be much lesser than the total number of terms in $\mathcal{T}$, it may be interesting to compute (only once) and store the 'prior interactions' $\delta(B_1, B_2|\emptyset)$ and then to derive the values of $\delta(B_1, B_2|q)$ from those of $\delta(B_1, B_2|\emptyset)$, by traversing the graph starting only from the terms in $\mathcal{Q}$. This can be done easily because from eq. (12) we obtain:

$$\delta(B_1, B_2|q) = \delta(B_1, B_2|\emptyset) - \sum_{T \in Pa(B_1) \cap Pa(B_2) \cap \mathcal{Q}} w(T, B_1) w(T, B_2)\, p(t^+)(1 - p(t^+)) . \quad (18)$$

In order to design an algorithm to compute $\delta(B, S|q)$ and $\delta(S_1, S_2|q)$, it is important to notice that we need these values only when $B \in Pa(S)$ and $S_1 \in Pa(S_2)$; as each unit has only one child, the required values are $\delta(B, Hi(B)|q)$ and $\delta(S, Hi(S)|q)$, which only depend on B and S respectively. Therefore, we shall use a variable delta[U] to store the value $\delta(U, Hi(U)|q)$ for each unit $U \in \mathcal{U}_b \cup \mathcal{U}_c$ such that $Hi(U) \neq null$. As in the case of $\delta(B_1, B_2|q)$, it is also convenient to compute (only once) and store the values $\delta(U, Hi(U)|\emptyset)$. This will allow us to compute all the values $\delta(U, Hi(U)|q)$ by only traversing the nodes in the graph that require updating, starting from the terms instantiated in $\mathcal{Q}$. This is possible because from of eqs. (14) and (16), we easily obtain:

$$\delta(B, Hi(B)|q) = \delta(B, Hi(B)|\emptyset) + \sum_{\substack{B_i \in A_b(Hi(B)) \\ B_i \neq B}} w(B_i, Hi(B)) \left(\delta(B, B_i|q) - \delta(B, B_i|\emptyset)\right) .$$

$$\delta(S, Hi(S)|q) = \delta(S, Hi(S)|\emptyset) + \sum_{B \in A_b(S)} \sum_{B' \in A_b(Hi(S)) \setminus A_b(S)} w(B, S) w(B', Hi(S)) \left(\delta(B, B'|q) - \delta(B, B'|\emptyset)\right) .$$

From eq. (14) and (16), it can also be noticed that $\delta(U, Hi(U)|q)$ is the weighted sum, over all the pairs of basic units B_1 and B_2, B_1 included in U and B_2 included in $Hi(U)$ but not in U, of the values $\delta(B_1, B_2|q)$, the weighting values being the products of the weights of the arcs in the single path joining B_1 and B_2 and passing through U and $Hi(U)$, except the weight $w(U, Hi(U))$ of the arc from U to $Hi(U)$. This simple observation is the basis of the algorithm to compute all the values $\delta(U, Hi(U)|q)$. Starting from each pair of basic units that had required updating (i.e. $\delta(B_1, B_2|q) \neq \delta(B_1, B_2|\emptyset)$), we traverse the graph from parents to children (and computing the product of the weights of the arcs we encounter), until we identify the single node S (if it exists) where the two paths that started at B_1 and B_2 converge. If U_1 and U_2 are the nodes

in these two paths nearest to S (i.e. arcs $U_1 \rightarrow S \leftarrow U_2$ exist in the graph), then we can update the values $\delta(U_1, S|q)$ and $\delta(U_2, S|q)$ by adding to delta[U_1] the computed product of weights of arcs times the difference between $\delta(B_1, B_2|q)$ and $\delta(B_1, B_2|\emptyset)$, divided by the weight $w(U_1, Hi(U_1))$ (and performing the same kind of updating for delta[U_2]).

7.3 Approximating the Bi-dimensional Posterior Probabilities

The previous results show how we can compute exactly the bi-dimensional probabilities involved in the computation of the expected utilities. But this process could be expensive in terms of memory and time for very large document collections. This reason leads us to propose another approximation, finer than the one presented in [6], which assumed the independence between each structural unit and the one which contains it, i.e. $p(u^+, u^+_{hi(U)}|q) = p(u^+|q)p(u^+_{hi(U)}|q)$. In the light of the results in the previous section, this approximation assumes that $\delta(U, U_{hi(U)}|q) \approx 0$ and $w(U, U_{hi(U)})\, p(u^+|q)\, (1 - p(u^+|q)) \approx 0$. While the first equality may be justified at some extend, the second one clearly can not. The proposed approximation is therefore

$$p(u^+, u^+_{hi(U)}|q) = p(u^+|q)p(u^+_{hi(U)}|q) + w(U, U_{hi(U)})\, p(u^+|q)\, (1 - p(u^+|q)) \quad (19)$$

which can be computed as efficiently as the previous one.

8 Example

To illustrate the behaviour of the generalized CID model, let us consider a simple example, where there is a single document, composed of the Sections 6 and 7 of this paper. Moreover, we use as indexing terms only the words appearing in the titles of these sections and the corresponding subsections. The Bayesian network representing this document is displayed in Figure 5. This 'collection' contains ten terms, five basic and two complex structural units. We shall use the same normalized tf-idf weighting scheme proposed in [6] (the resulting weights of the arcs are also displayed in Figure 5), and the prior probability of all the terms has been set to 0.1. The utility values are $v(r_i^+, u_i^+, u^+_{hi(U_i)}) = 0.5$, $v(r_i^+, u_i^+, u^-_{hi(U_i)}) = 1$, $v(r_i^+, u_i^-, u^+_{hi(U_i)}) = -1$, $v(r_i^+, u_i^-, u^-_{hi(U_i)}) = 0$ for all the structural units, except for the complete document, where $v(r_i^+, u_i^+) = 1$ and $v(r_i^+, u_i^-) = -0.5$.

Let us study the output provided by the model for two queries Q_1 and Q_2, where Q_1 is "posterior probabilities" and Q_2 is "approximating posterior probabilities". After instantiating to relevant these terms, we propagate this evidence through the network. The posterior probabilities of the structural units are displayed in Table 1. For Q_1, all the three subsections of section 7 appear more relevant than the section itself, whereas for Q_2 subsection 7.3 is clearly the most relevant structural unit and section 7 is the second one. However, for Q_1, it seems us that retrieving section 7 would be better for the user than retrieving its subsections (section 7 speaks about posterior probabilities as a whole). If we

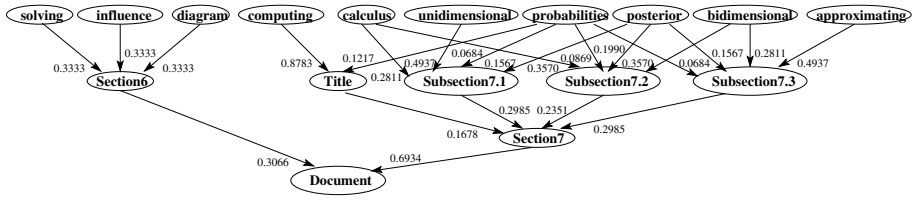

Fig. 5. The Bayesian network representing part of this document

Table 1. Posterior probabilities and expected utilities for queries Q_1 and Q_2. EU_{ex}, EU_{ap} and EU_{in} represent the utilities computed using the exact method, the approximation proposed in eq. (19) and the approximation using independence, respectively

	section 6	section 7	title (sect.7)	subsect. 7.1	subsect. 7.2	subsect. 7.3	document
$p(.\|Q_1)$	0.100	0.300	0.210	0.303	**0.357**	0.303	0.239
$EU_{ex}(.\|Q_1)$	-0.113	**0.170**	-0.045	0.081	0.141	0.081	-0.142
$EU_{ap}(.\|Q_1)$	-0.113	**0.170**	-0.045	0.080	0.138	0.080	-0.142
$EU_{in}(.\|Q_1)$	-0.127	0.097	-0.059	0.048	**0.111**	0.048	-0.142
$p(.\|Q_2)$	0.100	0.432	0.210	0.303	0.357	**0.747**	0.331
$EU_{ex}(.\|Q_2)$	-0.180	0.258	-0.164	-0.032	0.032	**0.505**	-0.004
$EU_{ap}(.\|Q_2)$	-0.180	0.258	-0.164	-0.033	0.029	**0.504**	-0.004
$EU_{in}(.\|Q_2)$	-0.214	0.173	-0.178	-0.064	0.002	**0.476**	-0.004

compute the expected utilities (Table 1 also displays all the utility values) using either the exact method or the proposed approximation, we can see that section 7 gets the highest value for Q_1 (and subsection 7.3 maintains the highest value for Q_2), as desired. Notice also that the approximation that assumes independence behaves differently.

9 Concluding Remarks

In this paper we have presented the theoretical developments concerning two extensions of the CID model for structured document retrieval. First, we have generalised the type of structure of documents that the CID model can deal with. In the new approach, the structural units containing text can be placed anywhere, and the organization of document components is general, in the sense that they do not have to be included in homogeneous layers. The change of structure has required the design of a new propagation algorithm that supports it.

We have also proposed two new methods of computing the bi-dimensional probability distributions needed for the calculus of the expected utilities of retrieving document components. The CID model assumed independence between each structural unit and the unit containing it, given the query. This is a very strong assumption, reason by which we have designed a method to compute exactly these distributions, based on interactions among units. We have also developed a new approximation in order to alleviate the possible computational

cost of the exact method in very large collections, which also considers interactions but approximates them without complex calculations.

At present, we are in the implementation stage. We intend to test our model with the INEX structured collection [10], in order to determine the quality and efficiency of each evaluation method. Also, as future work, we want to perform some experiments oriented towards the detection of best entry points, since this structured IRS has been specifically designed to find them.

Acknowledgments. This work has been supported by the Spanish Fondo de Investigación Sanitaria, under Project PI021147.

References

1. R. Baeza-Yates and B. Ribeiro-Nieto. *Modern Information Retrieval*. Addison-Wesley, Harlow, UK, 1999.
2. Y. Chiaramella. Information retrieval and structured documents. *Lectures Notes in Computer Science*, 1980:291–314, 2001.
3. F. Crestani, L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. A multi-layered Bayesian network model for structured document retrieval. *Lectures Notes in Artificial Intelligence*, 2711:74–86, 2003.
4. 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.
5. L.M. de Campos, J.M Fernández-Luna, and J.F. Huete. The BNR model: foundations and performance of a Bayesian network-based retrieval model. *International Journal of Approximate Reasoning*, 34:265–285, 2003.
6. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. Using context information in structured document retrieval: An approach using influence diagrams. *Information Processing & Management*, 40(5):829–847, 2004.
7. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. Bayesian networks and influence diagrams: Models and algorithms useful for structured retrieval. *DECSAI Technical Report*, 2004. Available at ftp://decsai.ugr.es/pub/utai/tech_rep/lci
8. S. French. *Decision Theory. An introduction to the Mathematics of Rationality*. Ellis Horwood Limited, Wiley, 1986.
9. A. Graves and M. Lalmas. Video retrieval using an MPEG-7 based inference network. In *Proc. of the* 25^{th} *ACM–SIGIR Conference*, 339–346, 2002.
10. INitiative for the Evaluation of XML Retrieval. http://inex.is.informatik.uni-duisburg.de:2004/
11. F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer Verlag, 2001.
12. S.H. Myaeng, D.H. Jang, M.S. Kim, and Z.C. Zhoo. A flexible model for retrieval of SGML documents. In *Proc. of the* 21^{th} *ACM–SIGIR Conference*, 138–145, 1998.
13. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, San Mateo, 1988.
14. B. Piwowarski, G.E. Faure, and P. Gallinari. Bayesian networks and INEX. In *Proc. of the INEX Workshop*, 7–12, 2002.
15. R. Shachter. Evaluating influence diagrams. *Operations Research*, 34:871–882, 1986.
16. R. Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36(5):527–550, 1988.