Bayesian Networks and INEX

Benjamin Piwowarski
LIP6
8 rue du capitaine Scott
75015 Paris, France
Benjamin.Piwowarski@lip6.fr

Georges-Etienne Faure
LIP6
8 rue du capitaine Scott
75015 Paris, France
Georges-Etienne.Faure@poleia.lip6.fr

Patrick Gallinari
LIP6
8 rue du capitaine Scott
75015 Paris, France
Patrick.Gallinari@lip6.fr

ABSTRACT

We present a bayesian framework for XML document retrieval. This framework allows us to consider content only and structure and content queries. We perform the retrieval task using inference in our network. Our model can adapt to a specific corpora through parameter learning.

Categories and Subject Descriptors

H.3.3 [**Information Storage and retrieval**]: Information Research and Retrieval—*Retrieval models, Relevance feedback, Search process*; I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*

Keywords

Bayesian networks, INEX, XML, Focused retrieval, Structured retrieval

1. STRUCTURED DOCUMENTS AND INFORMATION RETRIEVAL

The goal of our model is to provide a new generic system for performing different IR tasks on collections of structured documents. We take an IR approach to this problem. We want to retrieve specific relevant elements from the collection as an answer to a query. The elements may be any document or document part (full document, section(s), paragraph(s), ...) indexed from the structural description of the collection. We consider *content only* (CO) queries and *content and structure* queries (CAS). We use a probabilistic model based on bayesian networks (BN), whose parameters are learnt so that the model may adapt to different corpora. For CO queries, we consider the task is a *focused retrieval*, first described in [5, 13].

The organization of this paper is as follow. We introduce bayesian network in section 2 and our model in section 3. We desribe the three modes in which our model can be used: retrieval with CO and CAS queries and learning. Finally, in section 4 we describe related works.

2. BAYESIAN NETWORKS FOR STRUCTURED DOCUMENT RETRIEVAL

Bayesian networks [9, 10, 15, 18] are a probabilistic framework where conditional independence relationships between random variables are exploited, in order to simplify or/and to model decision problems. For textual data, the seminal work of Turtle & Croft [20] raised interest in this framework, and since that, simple BN have been used for IR tasks (see section 2). A bayesian network is a directed acyclic graph (DAG) whose nodes represent variables of the problem and arcs (in)dependance relations between variables. Let $\{x_i\}$, $i = 1$ to N, denote the variables of a BN. Their joint probability is given by:

$$P(\{x_i\}) = \prod_i P(x_i | x_i \text{ parents})$$

where x_i parents denotes the parents of x_i in the DAG. Each random variable can take values from a set $\mathcal{S}$. For simplicity, in this section we consider binary variables. Note that strong simplifying assumptions on the structure of the BNs are needed for modeling textual data, since documents are represented in very large feature spaces.

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

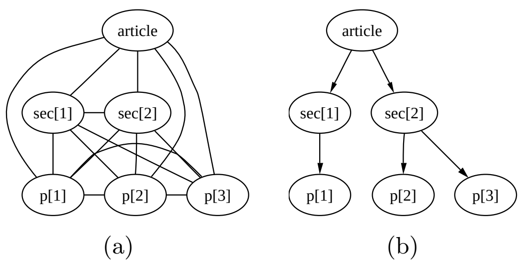

Figure 1: Two different modelizations of a same document. (a) All parts are dependent, (b) a simple BN inspired from the hierarchical document structure.

Consider the simple document of figure 1-a, composed of two sections and three paragraphs. A simple way to take into account the structure of d is to decompose the score

$P(d|q)$ as follows:

$$P(d|q) = \sum_{s_1,s_2,s_3,p_1,p_2,p_3} P(s_1, s_2, s_3, p_{1,2}, p_3|q)$$

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

In our model, BN are used to represent documents, one specific BN being associated to each document. Each node of a BN document model is a boolean variable which indicates whether or not the information associated to this node is relevant to the query. Different BN may be considered for modeling a document. Figure 1 shows a simple model for document 1-a, where the dependences go from parts to subparts so that section relevance depends on document relevance and paragraph relevance depends on section relevance.

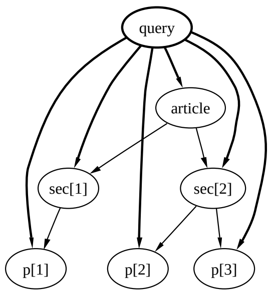

Figure 2: Retrieval network (model 1-b)

For retrieval, when a new query Q has to be answered, we add the evidence (the query) to the network as shown in figure 2. Relevance thus depends upon both the query and the englobing structural part.

The relevance of a document or document part is computed using the conditional independence assumptions encoded in the BN. As an example, the probability of relevance of section 1 with the model 2 is given by:

$$P(s_1|q) = \sum_d P(d|q)P(s_1|d, q)$$

Where the summation is over the values random variable d (document) can take. With such a model, complexity drops from $O(K^N)$ where N is the number of random variables to $O(NK^{N_{max}})$ where N_{max}[1] is the maximal number of parents for a given random variable in the bayesian network and K is the number of values the random variables can take in our BN.

[1]For type (b), complexity is thus $O(KN)$.

3. MODEL

Our work is an attempt to develop a formal model for structured document access. Our model relies on bayesian networks instead of evidence theory in [11] or probabilistic datalog in [7] and thus provides an alternative approach to the problem. We believe that this approach allows casting different access information tasks into a unique formalism, and that these models allow performing sophisticated inferences, e.g. they allow to compute the relevance of different document parts in the presence of missing or uncertain information. Compared to other approaches based on BN, we propose a general framework which should adapt to different types of structured documents or collections. Another original aspect of our work is that model parameters are learnt from data, whereas none of the other approaches relies on machine learning. This allows to rapidly adapt the model to different document collections and IR tasks.

The BN structure directly reflects the document hierarchy, i.e. we consider that each random variable is associated to a structural part within that hierarchy. The root of the BN is thus a variable "corpus", its childs the "journal collection" variables, etc. In this model, due to the conditional independence property of the BN variables, relevance is a local property in the following sense: if we know that the journal is (not) relevant, the relevance value of the journal collection will not bring any new information on the relevance of one article of this journal.

Three different models were considered, they correspond to three different sets of values $\mathcal{S}$ for the BN variables:

Model I Relevant (R) or not relevant (I). This was the only model where each structural element relevance is independant of any other[2]. It is a simple model that was used in models II and III;

Model II Relevant (R), too generic (G), not relevant (I);

Model III Relevant (R), too generic (G), too specific(S) or not relevant (I)

This definition of relevance is related to several definitions of what should be information retrieval with free text queries on structured documents, as proposed by Chiaramella et al. [5] and Lalmas [13].

In order perform the inference steps in the BN, needed for retrieval or learning, we need to compute $P(e|p, q)$ where e is the element, p its parent and q the query. For discrete variables, conditional probabilities in BNs are usually stored in tables. Here, these conditional probabilities should be computed for each Q, and they have to be learnt. For a given Q, we first compute a score $F_{e,a,b}$ for each structural element e. In this instance of the model, this score will depend on the element e type (a tag in the XML document) and on the value a (among R, G, S, I according to model I, II or III) of the element e and the value b of its parent:

$$F_{e,a,b}(q) = \alpha_{e,a,b}F^{\alpha}_{rel}(e) + \beta_{e,a,b}F^{\beta}_{rel}(e) + \gamma_{e,a,b}F^{\gamma}_{rel}(e, a, b)$$

[2]It can't therefore truly be considered as a BN model

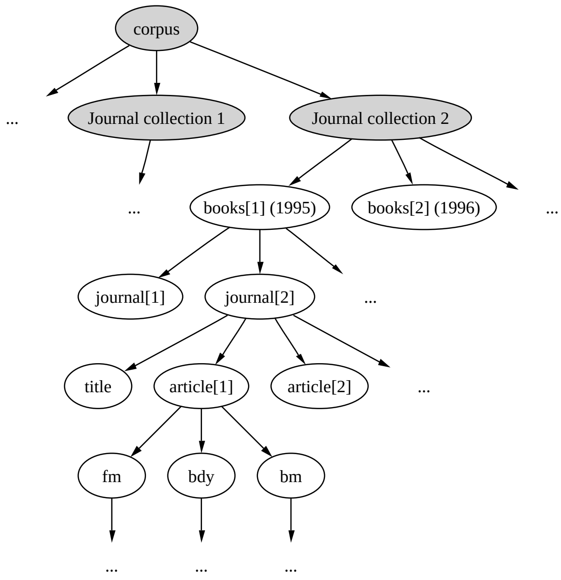

Figure 3: The document collection: each structured document is located in a specific part of the hierarchically organized collection. Here, each document is a collection of journals, each journal contains structured articles (on the figure are indicated the tags for ....). The query q is added to this network while retrieving or learning as in figure 2. Below article[1], we have indicated some tags used in the INEX collection. fm, bdy and bm respectively hold for "front matter", "body" and "back matter", each being composed of sub-elements not represented on the figure.

where $F_{rel}^{\diamondsuit}$ is the relevance of e content measured by a given flat retrieval model - in the experiments presented here, we have used a slightly modified version of OKAPI [21] as well as two other simple models. The first one gives a score that is the ratio between the number of occurences of the query terms into the element and into its parent. The second one is similar but takes into account the size of the element. The peculiar form of $F(e, a, b)$ has been chosen empirically and the two models have been chosen and tuned empirically.

This score is then used for computing the conditional probabilities $P(e = a|p = b, q)$ using a softmax function that gives values between 0 and 1.

$$P(e = a|p = b, q) \propto \frac{1}{1 + e^{F_{e,a,b}(q)}}$$

For each possible value a of e, we then get a score which is interpreted as a probability. α and β are to be learnt by the BN. β is the threshold while α scales the score given by the model.

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

3.1 Retrieval with CO queries

Answering CO queries was considered as *focused retrieval.* Focused retrieval consists in retrieving the most relevant structural elements in a document for a given query. Retrieval should focus on the smallest units that fulfill the query [5]. This unit should be the most relevant and should have a higher score than more generic or more specific units in the document.

When a new query Q has to be answered, we first compute $F_{e,a,b}(q)$ score for each element e and values a and b. The tree structure of this BN allows to use a fast and simple inference algorithm. We compute the relevance $P(e_i = R|q)$ for each element by marginalizing the conditional probability:

$$P(e_i = R|q) = \sum_{\{e_k\}_{k \neq i}} P(e_1, \dots, e_i, \dots, e_N|q)$$

where N is number of structural elements in the corpus and where the summation is taken over all combinations of values in the set $\mathcal{S}$ for all variables except e_i. This formula factorizes according to the conditional independance structure of the network:

$$P(e_i = R|q) = \sum_{\{e_k\}_{k \neq i}} \prod_{j=1}^{N} P(e_j|e_j\text{'s parent}, q)$$

In the absence of evidence in the network (i.e. when there is no external indication on the relevance value of any node), this formula can be furthermore simplified: the summation will be only over all combinations for the set of e_i *ancestors.* Elements with highest values are then presented to the user.

3.2 Retrieval with CAS queries

For CAS retrieval, we extend our bayesian network to handle multiple subqueries and use on sub-network for each one. Those networks are then connected to form one big network that represents the whole CAS query.

In order to describe CAS query processing, we make use of an example (figure 4). Each CAS query is first decomposed into different subqueries (here Q_0, Q_1 and Q_2). Each of these refers to a structural entity and an information need. Each information need is modeled by a BN constructed as for CO queries.

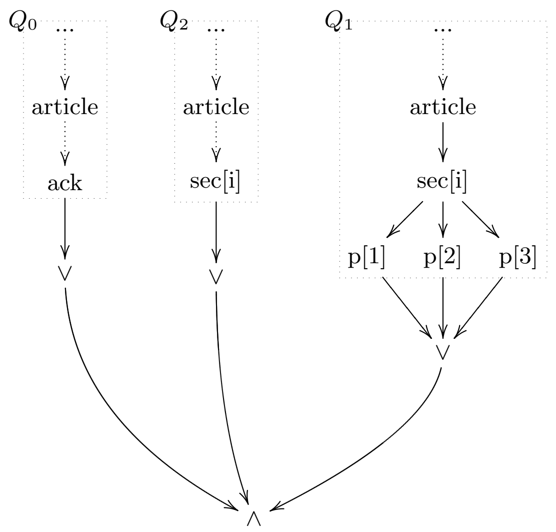

Figure 4: An example of BN for a CAS query: retrieval of sections of article. The article must contain an acknowledgment relevant to query Q_0. The section relevant to query Q_1. The acknowledgment must be relevant to query Q_3. Here only the part network involved in the relevance scoring of one `ack` element is shown.

Those BN are then connected for each target element in order to give this element a global score. Two different subquery type were distinguished:

1. sub-queries that were relative to the target element (Q_0);
2. sub-queries that were relative to the article element (Q_1 and Q_2).

Type 2 sub-queries networks are always constructed after finding a target element.

Two different types of inference are used to connect bayesian networks between them, namely "or" ($\vee$) and "and" ($\wedge$) functions. For $\wedge$ nodes we have:

$$P(\wedge = R|\text{parents}) = \begin{cases} 1 & \text{if one parent is } R \\ 0 & \text{otherwise} \end{cases}$$

and for $\vee$ nodes we have:

$$P(\vee = R|\text{parents}) = \begin{cases} 0 & \text{if one parent is } \neq R \\ 1 & \text{otherwise} \end{cases}$$

3.3 Training

In order to fit a specific corpus, parameters are learnt from observations using the Estimation Maximization (EM) algorithm. An observation $O^{(i)}$ is a query with its associated relevance assessments (document/part is relevant or not relevant to the query). EM [6] optimizes the model parameters Θ with respect to the likelihood $\mathcal{L}$ of the observed data :

$$\mathcal{L}(O, \Theta) = \log P(O|\Theta)$$

where $O = \left\{O^{(1)}, \ldots, O^{(|O|)}\right\}$ are the N observations.

Observations may or may not be *complete*, *i.e.* relevance assessments need not to be known for each structural element in the BN in order to learn the parameters. Each observation $O^{(i)}$ can be decomposed in $E^{(i)}$ and $H^{(i)}$ where $E^{(i)}$ corresponds to structural entities for which we know whether they are relevant or not, i.e. structural parts for which we have a relevance assessment. $E^{(i)}$ is called the evidence. $H^{(i)}$ corresponds to hidden observations, i.e. all other nodes of the BN.

In our experiment, we used for learning more or less 200 assessments from CO queries that were obtained by taking only the browse keywords of CAS queries.

4. RELATED WORKS

In this section, we make a short review of previous works in IR related structured retrieval and emphasis on BN information retrieval systems.

One of the pioneer work on document structure and IR, is that of Wilkinson [22] who attempted to use the document division into sections of different types (abstract, purpose, title, misc., ...) in order to improve the performances of IR engines. For that he proposed several heuristics for weighting the relative importance of document parts and aggregating their contributions in the computation of the similarity score between a query and a document. He was then able to improve a baseline IR system.

A more recent and more principled approach is the one followed by Lalmas and co-workers [11, 12, 13, 14]. Their work is based on the theory of evidence which provides a formal framework for handling uncertain information and aggregating scores from different relevance scores. In this approach, when retrieving documents for a given query, evidence about documents is computed by aggregating evidence of sub-document elements.

Another important contribution is the HySpirit system developed by Fuhr and colleagues which was described in a series of papers, see e.g. [7]. Their model is based on a probabilistic version of datalog. When complex objects like structured documents are to be retrieved, they use rules modeling how a document part is accessible from another part. The more accessible this part is, the more it will influence the relevance of the other part.

A series of papers describing on-going research on different aspects of structured document storage and access, ranging from database problems to query languages and IR algorithms is available in the special issue of JASIST and in the proceedings of two SIGIR XML workshops[4, 1, 2].

Since Inquery [3, 20], bayesian networks have proved to be a theoretically sounded IR model, which allows to reach state of the art performances and encompasses different classical IR models. The simple network presented by Croft, Callan and Turtle computes the probability that a query is satisfied by a document More precisely, the probability that the document represents the query. This model has been derived and used for flat documents. Ribeiro and Muntz [19] and Indrawan et al. [8] proposed slightly different approaches also based on belief networks, with flat documents in minds. An extension of the Inquery model, designed for incorporating structural and textual information has been recently proposed by Myaeng et al. [16]. In this approach, a document is represented by a tree. Each node of the tree represents a structural entity of this document (a chapter, a section, a paragraph and so on). This network is thus a tree representation of the internal structure of the document with the whole document as the root and the terms as leaves. The relevance information goes from the document node down to the term nodes. When a new query is processed by this model, the probability that each query term represents the document is computed. In order to obtain this probability, one has to compute the probability that a section represents well the document, then the probability that a term represents well this section and finally the probability that a query represents well this term. In order to keep computations feasible, the authors make several simplifying assumptions. Other approaches consider the use of structural queries (i.e. queries that specifies constraints on the document structure). Textual information in those models is usually boolean (term presence or absence). Such a well known approach is the Proximal Nodes model [17]. The main purpose of these models is to cope with structure in databases. Results here are boolean: a document matches or doesn't match the query.

5. CONCLUSION

We have described a new model for performing IR on structured documents. It is based on BN whose conditional probability functions are learned from the data via EM.

The model has still to be improved, tuned and developed, and several limitations have still to be overcome in order to obtain an operational structured information retrieval system. For example, we chose to discard textual information from the bayesian network (we use external models). A wiser choice would be to include terms within the bayesian network in order to give more expression power to our model. Other limitations are more technical and are related to the model speed.

Nevertheless some aspects of this model are interesting enough to continue investigating this model. Bayesian networks can handle different sources of information. Multimedia data can be integrated in our model by the mean of their relevance to a specific user need. Interactive navigation is also permitted. Our model is also able to learn its parameters from a training set. Since the relevance relationship between structural elements may change with the database, this seems to be an important feature.

6. REFERENCES

[1] R. Baeza-Yates, D. Carmel, Y. Maarek, and A. Soffer, editors. *Journal of the American Society for Information Science and Technology (JASIST)*, number 53(6), Mar. 2002.

[2] R. Baeza-Yates, N. Fuhr, and Y. S. Maarek, editors. *ACM SIGIR 2002 Workshop on XML*, Aug. 2002.

[3] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY Retrieval System. In A. M. Tjoa and I. Ramos, editors, *Database and Expert Systems Applications, Proceedings of the International Conference*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.

[4] D. Carmel, Y. Maarek, and A. Soffer, editors. *ACM SIGIR 2000 Workshop on XML*, July 2000.

[5] Y. Chiaramella, P. Mulhem, and F. Fourel. A Model for Multimedia Information Retrieval. Technical report, IMAG, Grenoble, France, July 1996.

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

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

[8] M. Indrawan, D. Ghazfan, and B. Srinivasan. Using Bayesian Networks as Retrieval Engines. In *ACIS 5th Australasian Conference on Information Systems*, pages 259–271, Melbourne, Australia, 1994.

[9] F. V. Jensen. *An introduction to Bayesian Networks*. UCL Press, London, England, 1996.

[10] P. Krause. Learning Probabilistic Networks. 1998.

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

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

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

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

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

[16] S. H. Myaeng, D.-H. Jang, M.-S. Kim, and Z.-C. Zhoo. A Flexible Model for Retrieval of SGML documents. In W. B. Croft, A. Moffat, C. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–140, Melbourne, Australia, Aug. 1998. ACM Press, New York.

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

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

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

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

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

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