

A belief networks-based generative model for structured documents.

An application to the XML categorization

Ludovic Denoyer¹, Patrick Gallinari¹

Laboratoire d'Informatique de Paris VI

LIP6

France

{[ludovic.denoyer](mailto:ludovic.denoyer@lip6.fr), [patrick.gallinari](mailto:patrick.gallinari@lip6.fr)}@lip6.fr

<http://www-connex.lip6.fr>

Abstract. We present a generative Bayesian model for the modeling of structured (e.g. XML) documents. This model allows us to simultaneously take into account structure and content information. It is used here for classifying XML documents. We adopt a machine learning approach and the model parameters are learned from a labeled training set of representative documents. We discuss the role of structural information for classification and describe experiments on a small collection of class labeled structured documents. We also present preliminary results showing how this model could classify documents with DTDs not represented in the training set.

1 Introduction

The development of large electronic document collections and Web resources has been paralleled by the emergence of different structured format proposals, aimed at encoding content information in a suitable form, for a variety of information needs. In addition to providing standard representations, these formats allow us to enrich the document content with additional information (e.g. metadata, comments etc...) and allow to store and access this content in a more efficient way. Some proposals (e.g. RDF for Web documents) have gained some popularity. At the same time, description languages like XML have become standards and are already widely used by different communities. For text documents, these representations encode both structural and content information. Flat document collections will probably be superseded in the near future by structured collections.

There is an important need to adapt existing information access methods to these new document representations so that they take all the benefit of these richer representations and also answer new information access challenges and new user needs. Current Information Retrieval (IR) methods have mainly been developed for handling flat document representations and cannot be easily adapted to deal with structured representations. In this paper, we focus on the particular task of structured document categorization.

Intuitively, like for other IR tasks, structure might seem to play an important role in categorization. A word will not have the same meaning or the same significance depending on its position into the document (title, metadata, keyword, etc). Also, a large and complex document might be relevant to a specific class even when only one of its subparts is relevant to the class. This information is hardly exploited at all in classical document representations.

In this article, we examine the role of structural information for document categorization and propose methods for exploiting both the content and the structure information for categorization tasks. We describe a generative categorization model based on belief networks. This work offers a natural framework for encoding structured representations and allows us to perform inference both on the whole document and on document subparts.

2 Previous works

Text categorization is a classical information retrieval task which has motivated a large amount of work over the last few years. Most categorization models have been designed for handling bag of words representations and do not consider word ordering or document structure. Generally speaking, classifiers fall into two categories: generative models which estimate class conditional densities $P(\text{document}/\text{Class})$ and discriminant models which directly estimate the posterior probabilities $P(\text{Class}/\text{document})$. The naive Bayes model [11] for example is a popular generative categorization model whereas among discriminative techniques support vector machines [8] have been widely used over the last few years. [19] makes a complete review of flat document categorization methods. Note that more recently, models which take into account sequence information have been proposed [4].

The expansion of the Web has motivated a series of works on Web page categorization - viz. the last two Trec competitions [20]. Web pages are built from different type of information (title, links, text, etc) which play different roles. There has been several attempts to combine these information sources in order to increase page categorization scores. There is not yet a clear conclusion on the relevance of combining these different types of information, however, this is work in progress. Many authors propose combining different classifiers each of them being trained on a specific information source. For example [17] compares three HTML page classifiers: one operates on the flat textual content of the pages, a second on page and section titles and a third on hyperlinks text. Experiments performed on one of the few available labeled HTML dataset, WebKB [2], show that indeed titles and hyperlinks information is relevant for the task and allow to increase performance compared to a purely flat textual representation. [12] maps a structured document onto a vector by encoding different structural elements (title, links, text) into different parts of the vector. For each part, he makes use of a frequential term representation (tf-idf), where the frequencies are computed on this specific type of structural element. He then uses classical classifiers on

these vectors. However, this does not bring any improvement on WebKB. Dumais and Chen [6], make use of the HTML structure (tags) in order to select the more relevant document parts for the categorization problem. [21] classify hypertexts by combining 3 classifiers which operate on the different parts of the document (linked pages, HTML tags, metadata). All these approaches deal only with HTML, they propose simple schemes either for encoding the page structure or for exploiting the different types of information by combining basic classifiers. They represent an initial attempt to take into account HTML structure and do not allow us to exploit more complex structures. These models exploit a priori knowledge about the particular semantics of HTML tags, and as such cannot be extended to more complex languages like XML where tags may be defined by the user. We will see that our model does not exploit this type of semantics and is able to learn from data the importance of tag information.

The previous models use *a priori* information about the nature of the tag, i.e this model uses the information that the title of an HTML page is described between the tag called `< title >` and `< /title >`. **We will see that our model does not use any information about the significance of the tags of an XML document.**

Some authors have proposed more principled approaches to deal with the general problem of structured document categorization. These models are not specific to HTML even when they are tested on HTML databases due to the lack of a reference XML corpus. [5] for example proposes the Hidden Tree Markov Model (HTMM) which is an extension of HMMs to a structured representation. They consider tree structured documents where in each node (structural element), terms are generated by a node specific HMM. [16] have proposed a Bayesian network for classifying structured documents. This is a discriminative model which computes directly the posterior probability corresponding to the document relevance for each class.

From a categorization model perspective, Bayesian networks (BN) have been used for information retrieval for some time, mainly for ad-hoc tasks. Inquiry [1], is a well known retrieval engine based on BNs which operates on flat text. Other authors have also used BNs for the retrieval of structured documents, e.g. [14], [15]. Outside the field of information retrieval, some models have been proposed to handle structured data. The hierarchical HMM (HHMM) [7] is also a generalization of HMMs to structured data, it has been tested on handwriting recognition and on the analysis of English sentences, similar HMM extensions have been used for multi-agent modeling [13]. However, inference and learning algorithms in these models are too computationally demanding for handling large IR tasks. The inference complexity for HHMM is $O(NT^3)$ where N is the number of states in their HMM and T the length of the text in words, for comparison our model is more like $O(N + T)$ as will be seen later.

The model we propose is a generative model which has been developed for the categorization of any tree like document structure (typically XML documents). This model bears some similarities with the one in [5], however, their description being very general, we cannot further compare the models. Their

model is adapted to the semantic of HTML documents and considers only the inclusion relation between two document parts. Ours is generic and can be used for any type of structured document, even when tags do not convey semantic information, it allows considering different types of relations between structured elements: inclusion, depth in the hierarchical document, etc. This model could be considered as a special case of the HHMM [7] since it is simpler and since HHMM can be represented as particular BNs [13]. It is computationally much less demanding and has been designed for handling large document collections.

[22] presents an extension of the Naive Bayes model to semi-structured documents where essentially global word frequencies estimators are replaced with local estimators computed for each path elements.

3 Document structure

A structured document d will be represented as a Directed Acyclic Graph (DAG). Each node of the graph represents a structural entity of the document, and each edge represents a hierarchical relation between two entities (for example, a paragraph is included in a section, two paragraphs are on the same level of the hierarchy, etc). For keeping inference complexity to a reasonable level, we do not consider circular relations which might appear in some documents (e.g. Web sites), this is a simplifying assumption which is not too severe since this definition already encompasses many different types of structured documents.

Each node of the DAG is composed of :

- **a label** : for example, labels can be *section*, *paragraph*, *title* and represent the structural semantic of a document.
- **a textual information** which is the textual content associated to this node if any.

A structured document then contains three types of information:

1. the logical structure information represented by the arcs of the DAG. (the position of the tag in an XML document)
2. the label information (the name of the tag in an XML document)
3. the textual information

Figure 1 is a simple example of structured document.

4 The generative model

We now present our BN model which allows to handle these 3 types of information. This model can be used with any XML document without using *a priori* informations about the semantic of the structure (i.e : we do ignore the significance of the tags describe in the DTD). We first briefly introduce BNs and then describe the different elements of the model.

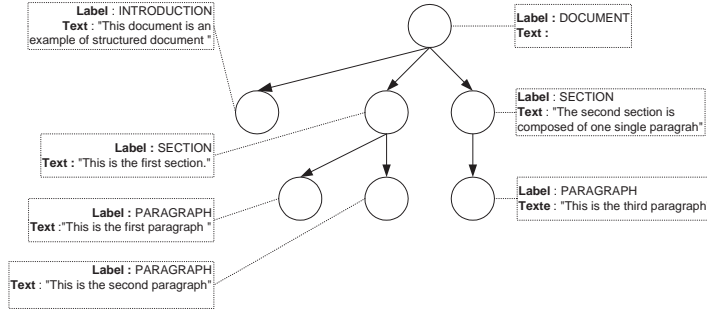


Fig. 1. An example of structured document represented as a Direct Acyclic Graph. This document is composed of an introduction and two sections. The first section has two paragraphs and the second one. Each part of the document is represented by a node with a **label** and a **textual information**

4.1 Belief networks

Belief networks [9] are stochastic models for computing the joint probability distribution over a set of random variable. They are DAGs whose nodes are the random variables and edges correspond to probabilistic dependence relations between 2 variables. The structure of the DAG reflects conditional independence properties between variables, the joint probability of a set of variables writes:

$$P(x_1, \dots, x_n) = \prod_{i=1..n} P(x_i | pa(x_i)) \text{ where } (pa(x_i)) \text{ denotes the parents of } x_i \text{ in the DAG.}$$

4.2 Model components

Let $d = (w_d, s_d)$ denote a document, with w_d the textual content and s_d the structural organization of the document. We will construct a generative model with parameters θ for computing the probability of a document:

$$P(d|\theta) = P((w_d, s_d)|\theta) = P(s_d|\theta)P(w_d|s_d, \theta) \quad (1)$$

In the following, we will successively detail the model components for the structural $P(s_d|\theta)$ and content $P(w_d|s_d, \theta)$ terms.

The structural probability : $P(s_d|\theta)$

We encode the structural information of a document into a belief network. This information s_d is the realization of a set of random variables denoted $s_d = \{s_d^i, i \in [1..|s_d|]\}$ (where $|s_d|$ is the number of structured nodes for document d), with $s_d^i \in A$ where A is the set of all the possible labels for the nodes of the DAG representing document d . Note that A depends on the DTD of the training XML documents. The corresponding BN structural parameters are then

the quantities $\{P(s_d^i|pa(s_d^i))\}$ which are the probabilities to observe s_d^i given its parents $pa(s_d^i)$ in the BN. In our model, we will construct one BN for each document. This BN can be thought of as a model of the structured document generation, where the generation process goes as follows: someone who wants to create a document about a specific topic will sequentially and recursively create the document organization and then fill the corresponding nodes with text. For example he first creates sections after what, for each section, he creates subsections etc... recursively. At the end, in each "terminal" node, he will create the textual information of this part as a succession of words. This is a typical generative approach which extends to structured information the classical HMM approach for modeling sequences. The corpus will then be represented as a series of BN models, 1 per document. Each will compute its structural density as:

$$P(s_d|\theta) = \prod_{i=1}^{|s_d|} P(s_d^i|pa(s_d^i)) \quad (2)$$

In order to have a robust estimation of the BN parameters, we will share sets of parameters among all the collection BNs. For the structural part, we make the hypothesis that the $\{P(s_d^i|pa(s_d^i))\}$ depend only on the labels of nodes s_d^i and $pa(s_d^i)$. i.e. two nodes in two different BNs which share the same label and whose parents also share the same labels will have the same transition probability.

Let us assume for now that our documents are XML documents which follow a specific DTD. Our "structural model" is based on the following ideas:

- A the set of values for the s_d^i , corresponds to the set of values for the tags in the DTD.
- We want to be able to take into account two types of structural information:
 1. The inclusion information. We want to represent the fact that a part (for example, a paragraph) is included into an other part (for example, a section).
 2. A sequential information which indicates how the different parts do appear sequentially in the document. e.g. a paragraph is followed by an other paragraph, or the first section follows the introduction, etc.
- Model complexity should remain low enough for the classification of large corpus: we will then use only first order dependencies between document parts.

Within this framework, several BN models may be associated to a document d . Figure 2 illustrates two of the models we have been working with. The DAG structure of Model 2 is copied from the tree structure of the document and reflects only the inclusion relation. The same type of relation is used in [5]. Model 1 contains both inclusion information (vertical edges) and sequence information (horizontal edges). Both models are an overly simplified representation of the real dependencies between document parts. This allows to keep the complexity of learning and inference algorithms low and to have robust inference models. Statistical models that work best are often very simple compared to the underlying phenomenon (e.g. naive Bayes in text classification or Hidden Markov

Models in speech recognition), practitioners of BNs have experienced the same phenomenon. In our tests on the WebKB collection, the two models behave similarly, in the experiments we show results for model 1. Note that other instances of our generic model could have also been used here.

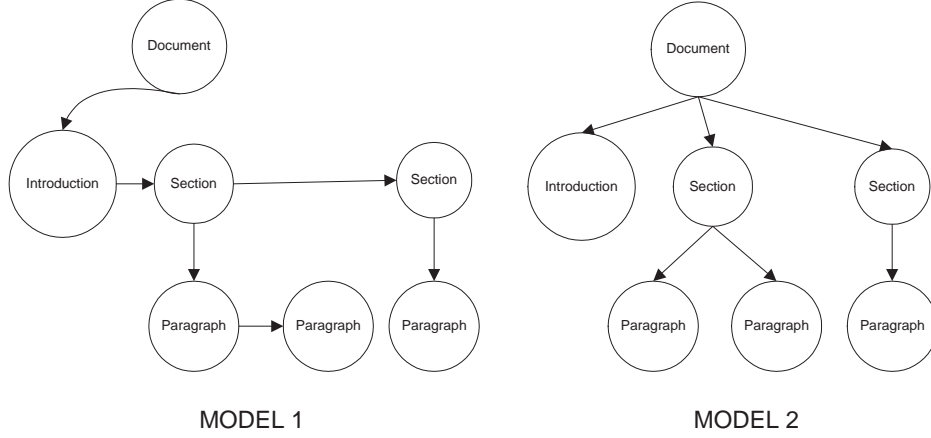


Fig. 2. Two possible structural belief networks constructed for the document presented in figure 1. For example in MODEL 1, we can compute the probability: $P(s_d) = P(intro|titre)P(sec|intro)P(sec|sec)P(par|sec)P(par|par)P(par|sec)$ with $\Lambda = \{intro, document, sec, par\}$ representing the label *document*, *introduction*, *section* or *paragraph*

Textual probability : $P(w_d|s_d, \theta)$ Let $w_d = w_d^i, i \in [1..|w_d|]$ be the set of all word instances of the document d , $w_d^i \in V$ where V represents the space of all the possible terms (the vocabulary). We make the following hypothesis :

- **H1** : the probability of a word w_d^i depends only on the label of the node that contains this word, i.e. $P(w_d^i/s_d^i)$ only depends on the value of s_d^i and not on the place of the node in the tree.
- **H2** : in a node, words are independent (Naive Bayes assumption)

Let $sel(w_d^i) = s_d^j$ (sel = structural element) be the function which indicates that word w_d^i is in the node labeled s_d^j , we then have:

$$P(w_d|s_d, \theta) = \prod_{i=1}^{|w_d|} P(w_d^i|sel(w_d^i), \theta) \quad (3)$$

As for the structure, both hypothesis H1 and H2 have been made to keep computation feasible. Hypothesis H1 means that word generation does not depend on the father label of the node it belongs to. For the generative model this

means that the document creator generates words by considering only the local context of the part he is currently writing. We could have considered a more realistic process where word occurrence depends on the whole document path which leads to this word at the price of more complex estimation models.

The naive Bayes hypothesis H2 is not mandatory here and any other term generative model (e.g. HMM) could be used instead, however this hypothesis allows for a robust density estimation and it is not clear that more sophisticated models could lead to any performance improvement.

Final belief network

Combining equations 2 and 3, we get:

$$P(d|\theta) = \prod_{i=1}^{|s_d|} P(s_d^i | pa(s_d^i), \theta) \prod_{i=1}^{|w_d|} P(w_d^i | sel(w_d^i), \theta) \quad (4)$$

The BN corresponding to the document in figure 1 is given in figure 3.

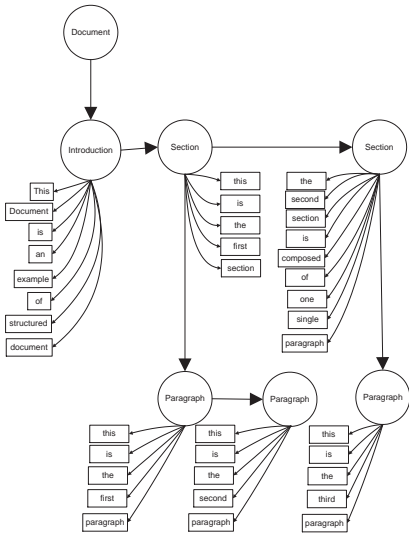


Fig. 3. The final belief network constructed to represent the document in 1. The random variables corresponding to the word variables are represented with rectangles whereas the random variables corresponding to the tags variables are represented with circles.

4.3 Learning

In this model, there are two sets of parameters to learn the transition and emission probabilities respectively denoted by $P(\hat{s}_i | s_j)$ and $P(\hat{w}_i | s_j)$.

$$\theta = \{P(\hat{s}_i|s_j)\}_{s_i, s_j \in A} \cup \{P(\hat{w}_i|s_j)\}_{w_i \in V, s_j \in A}$$

In order to learn the θ , we use the EM algorithm. In this network, since evidence is available for any variable, this amounts to a count of each possible values of the random variables. In our case, the EM algorithm is a maximum likelihood (ML) solving.

We want to maximize the log-likelihood for all the BNs. Using equation 4, we have :

$$L = \sum_{d \in D} \sum_{i=1}^{|s_d|} \log P(s_d^i | pa(s_d^i), \theta) + \sum_{i=1}^{|w_d|} \log P(w_d^i | sel(w_d^i), \theta) \quad (5)$$

For simplification, the model parameters $P(s_d^i | pa(s_d^i))$ and $P(w_d^i | sel(w_d^i))$ are denoted $\theta_{s_d^i, pa(s_d^i)}$ and $\theta_{w_d^i, sel(w_d^i)}$. Equation 5 then writes :

$$L = \sum_{d \in D} \sum_{i=1}^{|s_d|} \log \theta_{s_d^i, pa(s_d^i)} + \sum_{i=1}^{|w_d|} \log \theta_{w_d^i, sel(w_d^i)} \quad (6)$$

In the following, $\theta_{n,m}$ denotes either a textual probability or a structural probability. It corresponds to $P(\text{node with value } n / \text{his parent has the value } m)$.

The derivative of L is:

$$\begin{aligned} \frac{\partial L}{\partial \theta_{n,m}} &= \sum_{d \in D} \sum_{\left(s_d^i / s_d^i = n \text{ and } pa(s_d^i) = m \right)} \frac{1}{\theta_{n,m}} + \sum_{\left(w_d^i / w_d^i = n \text{ and } sel(w_d^i) = m \right)} \frac{1}{\theta_{n,m}} \\ &= \sum_{d \in D} \frac{\text{number of times a node with value } n \text{ has his parent with value } m}{\theta_{n,m}} \end{aligned} \quad (7)$$

The learning algorithm then solves $\frac{\partial L}{\partial \theta_{n,m}} = 0$ with the constraint $\sum_n \theta_{n,m} = 1$.

Using the Lagrange multipliers, we solve :

$$\frac{\partial(L - \lambda_m (\sum_n \theta_{n,m} - 1))}{\partial \theta_{n,m}} = 0 \quad (8)$$

Let $N_{n,m}$ the number of times a node with value n has his parent with value m for all the documents of the training set, we solve :

$$\frac{N_{n,m}}{\theta_{n,m}} = \lambda_m \quad (9)$$

So :

$$\theta_{n,m} = \frac{N_{n,m}}{\sum_i N_{i,m}} \quad (10)$$

The complexity of the algorithm is $O(\sum_{d \in D} |s_d| + |w_d|)$. In a classical structured document, the number of node of the structural network is smaller than

the number of words of the document. So the complexity is quasi-equivalent to $O(\sum_{d \in D} |w_d|)$ which is the classical learning complexity of the Naive Bayes algorithm.

Remark

Let us consider what happens to the model for a classical flat document. The corresponding belief network is presented in figure 4. The probability of the document d derived from equation 4 is :

$$P(d|\theta) = \prod_{i=1}^{|w_d|} P(w_d^i | sel(w_d^i) = document, \theta) \tag{11}$$

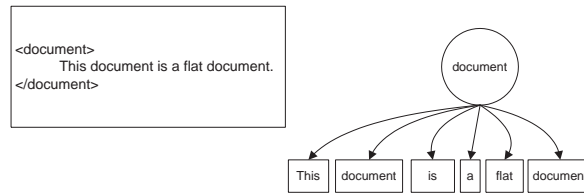


Fig. 4. A flat document (in XML format) and the associated belief network constructed using the previous hypothesis.

This is the equation of a Naive Bayes model, we can conclude that **for flat documents**, our model is **strictly equivalent to Naive Bayes**.

4.4 Extending the model to unknown DTDs

We now show that the model could be easily extended to more complex categorization situations. As an example, we consider the categorization of XML documents with unknown DTDs (we don't know the values of the tags).

Up to now, we have made the hypothesis of a unique DTD in the corpus. This is convenient for controlled collections like e.g. an editor scientific journals. In many situations (e.g. documents gathered from the web), documents may follow different DTDs so that with our approach, one will have to learn different structures. One may also have to classify documents with DTDs not represented in the training set. We will consider here a simple instance of the latter problem where it is supposed that all the document in the training set follow the same DTD and documents in the test set do have other DTDs. We make an additional -homogeneity- hypothesis: all DTDs carry similar structural information (documents are more or less of the same type), but that tags may have different names, some may be missing and some may be added. Within this framework,

we propose an extension of our model to enable the classification of documents with new DTDs.

Suppose that we have learned θ using a train set of documents with DTD1 exactly as part 4.3). We want to compute the probability of a document d which use an other unknown DTD2. We will not try to make an a priori correspondence between the two DTDs but we rather consider that the labeling information is missing in DTD2. This XML document contains the *textual information*, the *organization information* and no *label information* -label "Part" in (figure 5) (See part 3 for the definition of *textual information*, *label information* and *organization information*). We will score a document with DTD2 against the models learn with DTD1, i.e. we will try to classify this new document using the available knowledge on the document structure which is embodied in the BN models learned from DTD1. This makes sense under the homogeneity hypothesis. Alternatively, one could think of DTD1 as a reference DTD against which series of documents must be matched.

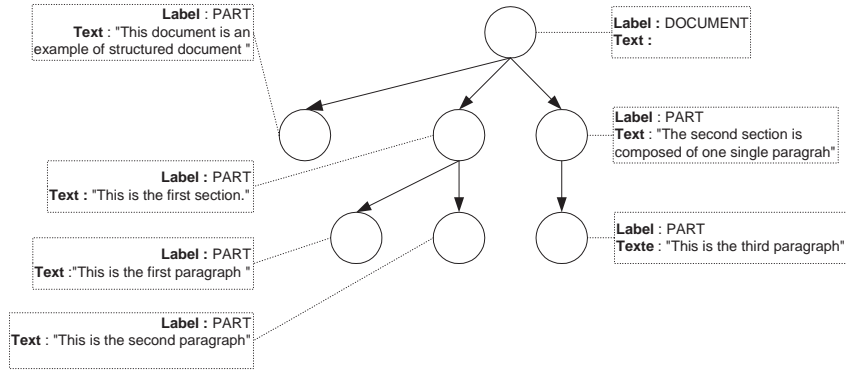


Fig. 5. The document presented in figure 1 with a different DTD. DTD2 corresponds to the tags $\{Document, PART\}$

The likelihood of a document with no label information can be computed by summing over all possible labellings of the document parts into the set of allowed labels in A :

$$P(d|\theta) = \sum_{s_d^1, s_d^2, \dots, s_d^{|s_d|} \in A} \prod_{i=1}^{|s_d|} P(s_d^i | pa(s_d^i), \theta) \prod_{i=1}^{|w_d|} P(w_d^i | sel(w_d^i), \theta) \quad (12)$$

This model has a higher complexity that the one presented in part 4.2. Alternatively, instead of summing over all allowed segmentations, one might compute the best segmentation of a document with unknown labels into the labels of a given DTD, i.e. compute the score of the most probable structure for this document according to the reference DTD. This can be done with a Viterbi

like algorithm as for HMMs except that we deal here with structures instead of sequences. This corresponds to computing:

$$P(d|\theta) = \max_{s_d^1, s_d^2, \dots, s_d^{|s_d|} \in \Lambda} \prod_{i=1}^{|s_d|} P(s_d^i | pa(s_d^i), \theta) \prod_{i=1}^{|w_d|} P(w_d^i | sel(w_d^i), \theta) \quad (13)$$

The most probable segmentation with respect to a reference DTD could also be used for transforming a DTD2 document into a DTD1 document. This might be useful for comparing series of XML documents. However, document segmentation involves further developments and has not been tested here.

5 Experiments

We now present the experiments made with our model. Using a generative model for the task of categorization is easy. Let us consider $C = \{c_1, \dots, c_{|C|}\}$ the set of all classes with probabilities $P(c_i)$. For each c_i , we will learn the model parameters denoted θ_{c_i} over all the documents of the training set within topic c_i . We consider that each document may have only one label and assume that:

$$P(d|c_i) = P(d|\theta_{c_i}) \quad (14)$$

So, the class c^d of a document d will be :

$$c^d = \operatorname{argmax}_{c \in C} P(c)P(d|\theta_c) \quad (15)$$

5.1 The webKBXML corpus

There are still few labeled XML corpus available for text categorization and whose documents do have a non trivial structure like in ([10],[18]).

The webKB collection ([2]) became a reference corpus in the Machine Learning community for the classification of structured collections. It is composed of 8,282 pages which were manually classified into the categories : student, faculty, staff, department, course, project and other. For each class the data set contains pages from the four universities : Cornell, Texas, Washington, Wisconsin, and miscellaneous pages collected from other universities. In our work, we only use the 6 topics : student, faculty, staff, department, course, project.

We thus used the HTML webKB collection and transformed the pages into XML documents using a DTD with the following tags : *website*, *section*, *text*, *link*, *title*, *sectiontitle*, *highlighed*. Figure 6 represents the transformation of an HTML document to the corresponding XML document. This is more like a real XML corpus than the original WebKB, e.g., tags here do not specify the depth of a section like in HTML.

We call the constructed corpus webKBXML. This corpus is available at <http://www-connex.lip6.fr/denoyer/corpus/webxml.tar.gz> The corpus has been preprocessed with Porter Stemmer, and words that do not appear in at least 5

<pre> <html> <head> <title>CS414 Home Page</title> </head> <body> <h1> The first section </h1> The first secon is composed of two paragraphs <h2> first paragprah </h2> This is a link <h2>Second paragraph</h2> This is the second paragraph <h1>This is the second section</h1> This thext is in <i> italic</i> </pre>	<pre> <website> <title>CS414 Home Page</title> <section> <sectiontitle> The first section </sectiontitle> <text> The first secon is composed of <text> <highlighted> two <highlighted> <text> paragraphs<text> <section> <sectiontitle> first paragprah </sectiontitle> <link>This is a link </link> </section> <section> <sectiontitle>Second paragraph</sectiontitle> <text> This is the second paragraph</text> </section> </section> <section> <sectiontitle> This is the second section</sectiontitle> <text> This thext is in </text> <highlighted> italic</highlighted> </section> </pre>
---	--

Fig. 6. Left : The original HTML document (truncated) Right : The corresponding XML file (truncated)

documents are eliminated. The vocabulary size is about 8000. There are sites from 4 universities in WebKB, we used a leave one out methodology for training the models: training is performed on 3 sites and test on the 4th, this is repeated 4 times.

5.2 Results

We have used a Naive Bayes model as a baseline. Results appear in figure 7. The BN model achieves a mean 3 % improvement with regard to Naive Bayes for micro-average, the No Tag BN Model (the model from section 4.4) which handles a more difficult task achieves a 2 % improvement. The results are very encouraging even the gain is small, it is significant and superior to that obtained on similar evaluation [5]. Note that the small size of the database penalizes the BN models compared to Naive Bayes since they have more parameters. Superior improvements should be obtained with larger databases.

The conclusion is that structure does indeed contain information, even for informal documents like those from WebKB. The proposed models allow us to take advantage of this structural information at a low increase in the complexity compared to flat classification models like Naive Bayes. The experiments show that even when the DTD tags are unknown, one can take advantage of the structure of the document in separate parts. Our generative models improve a baseline generative model (Naive Bayes) and similar ideas could be used to improve baseline discriminant models like SVM.

	<i>course</i>	<i>department</i>	<i>staff</i>	<i>faculty</i>	<i>student</i>	<i>project</i>		<i>Micro</i>	<i>Macro</i>
Naive Bayes	88.11%	100.00%	2.17%	76.32%	82.08%	66.67%		78.09%	69.22%
Model BN	89.34%	100.00%	0.00%	80.92%	86.02%	68.97%		81.12%	70.88%
Model BN No Tags	90.16%	100.00%	0.00%	76.32%	85.66%	66.67%		80.29%	69.80%

Fig. 7. The micro and macro average classification rate

6 Conclusion and perspectives

We have presented new models for the classification of structured documents. These models are generic and can be used to classify any XML document provided they have been trained on a representative corpus. Tests have been performed on a small but representative collection and encouraging results have been obtained. Further developments are needed in order to build models able to handle the whole variety of classification tasks needed for general XML collections. We are currently working on such extensions and on the evaluation on larger databases.

7 Bibliography

References

1. Jamie P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY Retrieval System. In A. Min Tjoa and Isidro Ramos, editors, *Database and Expert Systems Applications, Proceedings of the International Conference*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.
2. Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pages 509–516, Madison, US, 1998. AAAI Press, Menlo Park, US. An extended version appears as [3].
3. Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1/2):69–113, 2000.
4. Ludovic Denoyer, Hugo Zaragoza, and Patrick Gallinari. HMM-based passage models for document classification and ranking. In *Proceedings of ECTR-01, 23rd European Colloquium on Information Retrieval Research*, pages 126–135, Darmstadt, DE, 2001.
5. M. Dilegenti, M. Gori, M. Maggini, and F. Scarselli. Classification of html documents by hidden tree-markov models. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 849–853, Seattle, 2001. WA (USA).
6. Susan T. Dumais and Hao Chen. Hierarchical classification of Web content. In Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors, *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, GR, 2000. ACM Press, New York, US.

7. Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.
8. Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the “Lecture Notes in Computer Science” series, number 1398.
9. Jin H. Kim and Judea Pearl. A Computational Model for Causal and Diagnostic Reasoning in Inference Systems. In Alan Bundy, editor, *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, August 1983. William Kaufmann.
10. David D. Lewis. *Reuters-21578 text categorization test collection*. AT&T Labs - Research, September 1997.
11. David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the “Lecture Notes in Computer Science” series, number 1398.
12. Cline M. Utilizing HTML structure and linked pages to improve learning for text categorization. In *Undergraduate Honors Thesis, Department of Computer Science, University of Texas*.
13. K. Murphy and M. Paskin. Linear time inference in hierarchical hmms, 2001.
14. Sung Hyon Myaeng, Dong-Hyun Jang, Mun-Seok Kim, and Zong-Cheol Zhoo. A Flexible Model for Retrieval of SGML documents. In W. Bruce Croft, Alistair Moffat, C.J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–140, Melbourne, Australia, August 1998. ACM Press, New York.
15. Benjamin Piwowarky and Patrick Gallinari. A Bayesian Network Model for Page Retrieval in a Hierarchically Structured Collection. In *XML Workshop of the 25th ACM SIGIR Conference*, Tampere, Finland, 2002.
16. B. Piwowarski, L. Denoyer, and P. Gallinari. Un modele pour la recherche d’informations sur les documents structures. In *Proceedings of the 6emes journees Internationales d’Analyse Statistique des Donnees Textuelles (JADT2002)*.
17. CH. Y. Quek. Classification of world wide web documents, 1997.
18. Reuters. The reuters corpus volume 1 english language 1996-08-20 to 1997-08-19.
19. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
20. Trec. Text REtrieval Conference (trec 2001), National Institute of Standards and Technology (NIST).
21. Yiming Yang, Seán Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2/3):219–241, 2002. Special Issue on Automated Text Categorization.
22. Jeonghee Yi and Neel Sundaresan. A classifier for semi-structured documents. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 340–344. ACM Press, 2000.