Structured Multimedia Document Classification

Ludovic Denoyer
Jean-Noël Vittaut
Patrick Gallinari
LIP6 – University of Paris 6
Paris – France
{denoyer,vittaut,gallinari}@ia.lip6.fr

Sylvie Brunessaux
Stephan Brunessaux
EADS S&DE
Val de Reuil – France
sylvie.brunessaux@sysde.eads.net
stephan.brunessaux@sysde.eads.net

ABSTRACT

We propose a new statistical model for the classification of structured documents and consider its use for multimedia document classification. Its main originality is its ability to simultaneously take into account the structural and the content information present in a structured document, and also to cope with different types of content (text, image, etc). We present experiments on the classification of multilingual pornographic HTML pages using text and image data. The system accurately classifies porn sites from 8 European languages. This corpus has been developed by EADS company in the context of a large Web site filtering application.

Keywords

Categorization, Structured Document, Multimedia Document, Bayesian Networks, Generative Model, Statistical Machine, Web Page Filtering

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; I.7 [**Document and Text Processing**]: Miscellaneous

General Terms

Algorithms

1. INTRODUCTION

The development of the Web and the growing number of documents available electronically has been paralleled by the emergence of semi-structured data models for representing textual or multimedia documents. These models allow to encode the document content and its logical structure i.e. relations between document elements – denoted *doxels* in the following –, they also allow to enrich the document description with different types of meta-data. These representations are also useful for efficiently storing and accessing this type of data. Description languages for structured documents such as HTML or XML have gained popularity and are now widely used. Given the growing amount of structured document collections, it is important to develop tools able to take into account the increased complexity of these representations and the diversity of doxel types inside the document, to address document parts and the relations between doxels. Up to now, Information Retrieval –*IR*– has mainly developed for handling flat documents and IR methods should now adapt to these new types of documents. We focus here on the particular task of document classification, this is a generic problem with many different applications like document indexing, e-mail or spam filtering, document ranking, document categorization, etc. Although classification has been considered in IR for a long time, it is mainly since the nineties that it has gained popularity and has developed as a sub-branch of the IR domain. Much progress in this area has been obtained through recent machine learning classification techniques. Most classification models for text or image have been developed before the emergence of structured documents and are devoted only to flat representations. Recently, some attempts have been made to adapt these techniques for the classification of complex documents, e.g. XML textual documents or multimedia documents. This is usually done in a crude way, by combining basic classifiers trained independently on different components of a document.

The work described here is an attempt to develop a more principled approach to the problem of structured multimedia document classification. We propose a new model which allows to take simultaneously into account the structure and the content information of electronic documents. This model offers a natural framework for the integration of different information sources. It is based on a statistical framework: Bayesian networks are used to model the documents and to combine the information present in the doxels. We present tests for the problem of Web pages filtering on a large database gathered in the context of a European project "NetProtect".

The paper is organized as follows: we first review in part 2 existing work on information classification for structured document. We then describe our model for classifying multimedia documents in part 3. Finally, we describe the corpus NetProtectII used for our experiments and present a series of experiments.

DocEng'03, November 20–22, 2003, Grenoble, France.

2. PREVIOUS WORK

A large amount of work has been devoted over the last few years on flat document categorization either text or image. In the information retrieval community, it is often considered that structure should play an important role: for example a word can have different meanings according to its location in the document and a document can be considered relevant for a class, if only one of its parts is relevant. The rapid growth of structured electronic documents has recently motivated the emergence of a new line of research for accessing these documents. We briefly review below recent work on structured and multimedia document classification.

2.1 Structured document classification

We consider stochastic classifier models which score any document class according to the value of the posterior probability $P(class|document)$. Generally speaking, classifiers fall into two categories: generative models which estimate class conditional densities $P(document|class)$ and discriminant models which directly estimate posterior probabilities $P(class|document)$. For instance, the Naive Bayes model [13] is a popular generative categorization model while Support Vector Machines [10] is a discriminant model. See [16] for an exhaustive review of categorization models for flat documents. In machine learning, most classifiers have been designed for coping with vector or sequence representations, only very few models allow to consider simultaneously content and structure information. The growing need for handling structured objects – documents, biological structures, chemical compounds, etc – has recently motivated some interest in this area, but the field is still new and widely open. Some years ago, in the IR community, the development of the Web has created a need for classifying HTML pages viz. the last two TREC competitions [17]. In a HTML page, the different parts of the text do not play the same role and do not have the same importance, e.g. titles, links, text can be considered as different sources of information. Most of the techniques proposed for HTML classification use prior knowledge about the meaning of HTML tags either to encode the page structure using very simple schemes or to combine basic flat classifiers [9, 18]. These first attempts show that combining the different types of information present in a web page may sometimes increase page categorization scores. This is not systematic and many of the early trials did not show any improvement at all. These ideas do not naturally extend to more general document representations. More recently, different techniques have been developed for general structured document classification. These models are not HTML specific and can be used in particular for XML documents. For example, [19] 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 element. A drawback of this technique is the dramatic index growth which leads to poor estimation of probabilities. The Hidden Tree Markov Model (HTMM) proposed by [8] extends classical HMM to semi-structured representations. Documents are represented by a tree and for each node, words are generated by a specific HMM. This model has been used for HTML classification. As for discriminative models, [15] proposed a model based on Bayesian Networks, which directly computes the posterior probability corresponding to document relevance for each class.

2.2 Multimedia documents

Many different methods have been proposed for the classification of multimedia documents. Most work in the multimedia area makes use of text information such as keywords as an auxiliary source of information for enhancing the performance of existing image, video or song classifiers. For example, [4] propose a system that combines textual and visual statistics into a single index vector. Textual statistics are captured using latent semantic indexing (LSI) and visual ones are color and orientation histograms. This approach allows improving performance in conducting content-based search. [2] present a generative hierarchical model, where the data is modeled as being generated by a fixed hierarchy of nodes. Each node in the tree has some probability of generating each word or image feature. This model could be useful for IR tasks such as database browsing and search for images based on text and image features. In [14], a document is represented as a collection of objects which themselves are represented as collections of features (words for text, color and texture features for images). Several similarity measures over text and images are then combined. More recently, a method has been proposed for using images for word sense disambiguation, which suggests that combining image features and text can outperform simple text classifiers [3]. In the field of Web filtering, [11] combine a naked people photo image detector with a standard text classifier using an "OR" operator. [5] present an algorithm to identify images that contain large areas of skin. For identifying pornographic pages, they combine this skin detector with a text classifier via a weighing scheme. They also claim that text based approaches generally give poor results, whereas our structured model performs well on text-only documents. Most of these attempts then rely on the combination of basic classifiers trained independently on the different information sources (e.g. text and images). They do not consider the global context of the document nor its logical organization, i.e. they ignore the relations between the different document parts.

The model we propose provides a general framework for the classification of structured multimedia documents. It can be used for any structural representation or language (e.g. HTML or XML). This model is an extension to multimedia data of the model proposed in [6] and [7] which operates only on textual structured document. Previous work has demonstrated the efficiency of the approach on large XML textual corpus. We show here how to combine different information sources in a natural way (text, image, sound etc...). The model was developed for large databases and we will see that its complexity is linear with the size of the documents.

3. MULTIMEDIA GENERATIVE MODEL

We present here the multimedia generative model for structured documents. We first explain the context of the classification task. We then give the different hypothesis used to model structured documents and show how this global model can be seen as a weighted mixture of local generative models. Finally, we describe the training algorithm.

Figure 1: An example of multimedia structured document

3.1 Context

Let $\mathcal{D}$ be the set of all documents and $\mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$ be the set of all classes where $|\mathcal{C}|$ is the number of classes.

We will consider that a document can be either *Relevant* or *Irrelevant* for a specific class c. We then transform our problem with $|\mathcal{C}|$ classes into $|\mathcal{C}|$ problems with two classes (Relevant R or Irrelevant I) so that in the following, we will discuss only the two classes problem. We adopt a machine learning approach and the model parameters are learned from a labeled training set of representative documents from each class.

Since we adopt a stochastic approach to the classification problem, a structured document d in $\mathcal{D}$ will be the realization of a random variable D. We use the generative approach to classification: the model will compute the probability of generating a specific document $P(D = d|\theta)$ where θ corresponds to the parameters of the generative model. $P(d|\theta)$ will be used as a shorthand for $P(D = d|\theta)$.

In order to use our generative model for classification, we learn one model θ_R for the relevant class and one model θ_I for the irrelevant class. The score of a document will then be computed using Bayes-Rule:

$$P(R|d) = \frac{P(R)P(d|R)}{P(d|R)P(R) + P(d|I)P(I)} \equiv \frac{P(R)P(d|\theta_R)}{P(d|\theta_R)P(R) + P(d|\theta_I)P(I)} \quad (1)$$

In the following, we will denote the model parameters by θ corresponding either to θ_R or θ_I.

3.2 Description

A generative stochastic model for a document corresponds to specific hypotheses about the physical generation of this document. Different hypothesis should be considered and the choice of a particular model most often corresponds to a compromise between an accurate representation of the document generation process and practical constraints depending on the task the model will be used for, the difficulty for accurately estimating model parameters from data collections, the availability of labeled corpus, etc. Different hypotheses for structured textual document representations are discussed in [7] where it is shown that best performances for text classification are obtained with rather simple models of the dependencies between doxels. As it is often observed in document classification, more sophisticated models do not lead to increased performance. We adapt here one of these simple models to multimedia documents. The corresponding generative process is as follows: an author who wants to build a document about a specific topic (class) will first imagine the global logical structure of the document, once this structure is built he will then fill the content of each structural element. This content will depend on the type of the structural element: the process of writing a title will differ from the one for a paragraph, writing text is different from inserting an image or a piece of music, etc. This process is a simplified view of the reality and as will be seen below additional simplifying assumptions will be introduced in order to meet practical constraints. It embodies some crucial facts about structured documents: doxels are different depending on their type and the logical element they belong to, both logical structure and plain content of documents are essential for their descriptions, the topic of a document may influence both its logical structure and its content. The latter idea implies that the logical structure of a document may sometimes contain important information for characterizing the document class.

Let us now formally define the model.

We consider D as a random variable $D = (S, T)$ where S is the variable corresponding to the structure of a document and T corresponds to the content information. Let $d = (s_d, t_d)$ a realization of D where s_d represents the structure and t_d represents the content information of document d, we have:

$$P(d|\theta) = P(s_d|\theta)P(t_d|s_d, \theta) \quad (2)$$

In this equation, $P(s_d|\theta)$ is the probability of generating the **structural information** and $P(t_d|s_d,\theta)$ is the probability of the **content information**.

The structure of d consists of a set of nodes and their dependence relations. The set of nodes is denoted:

$$s_d = (s_d^1, ..., s_d^{|d|}) \quad (3)$$

where s_d^i is the i-th node of the document d and $|d|$ is the number of nodes of the document.

We will consider only tree like document, this is a reasonable simplification of real structured documents. Each node corresponds to a structural entity of the document (e.g. *paragraph, section*).

Let $pa(s_d^i)$ denote the parent of s_d^i. The structure of the document is described by the set $\{s_d^i, pa(s_d^i)\}$. Nodes take their values in Λ (i.e $s_d^i \in \Lambda$) which is the set of all possible node labels. Typically, for an XML document, this set is defined with the DTD and is the set of all possible tags. Figure (1) represents a structured multimedia document while figure (2) is the associated tree structure. For this example, we have Λ =(*title, paragraph, image, section*).

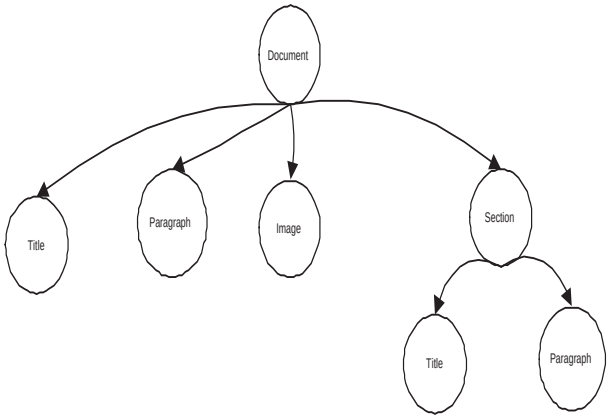

Figure 2: The structure graph corresponding to the previous example

Document content is denoted:

$$t_d = (t_d^1, ..., t_d^{|d|}) \quad (4)$$

where t_d^i represents the content information of the i-th node of the document.

We will make the hypothesis that each tag label contains only one type of information (text, image, sound ...). This hypothesis is not restrictive and it is often true in XML documents.

With these notations, (2) writes:

$$P(d|\theta) = P(s_d^1, ..., s_d^{|d|}|\theta) P(t_d^1, ..., t_d^{|d|}|s_d^1, ..., s_d^{|d|}, \theta) \quad (5)$$

We will now detail the content and structural parts of the document model.

3.2.1 Content probability

We will make the following hypothesis in order to compute the content probability $P(t_d|s_d,\theta)$:

Hypothesis 1. **Conditional Independence:** given the structural organization of the document, the content information of the different nodes are independent.

Hypothesis 2. **First order dependency:** the content information depends only on the structural node containing it and not on other structural nodes.

These hypothesis are simplifications of real dependencies between the different parts of a document. They are needed for keeping the complexity of the model reasonably low and for using the model on very large corpus. Returning to our generation process, this means that once the document organization has been decided, each content node is filled independently of the others, by considering only the type of the structural element it belongs to. All content elements with the same type (paragraph, etc) will share the same generative process. It could seem more natural to consider that content elements are filled in sequence, but early tests with such a model did not led to improved results at the price of an increased complexity and this was then left out. Note that such simplification are frequent in stochastic modeling and have led to very efficient models in different application areas.

Using hypothesis 1 and 2, we can rewrite the content probability as:

$$P(t_d|\theta) = \prod_{i=1}^{|d|} P(t_d^i|s_d^i, \theta) \quad (6)$$

According to hypothesis 2, each node type in the structure will have its own generative content model. Let $\theta_{s_d^i}$ be the parameters of the generative model associated with label s_d^i, we have:

$$P(t_d|\theta) = \prod_{i=1}^{|d|} P(t_d^i|s_d^i, \theta_{s_d^i}) \quad (7)$$

Models with parameters θ_l , $l \in \Lambda$, will be the **local generative models** associated to nodes with label l. As an example, for modeling the document in figure (1) and figure (2), we will use 3 local generative models:

- a textual generative model of parameters θ_{title} for the text contained in tags *title*
- a textual generative model of parameters $\theta_{paragraph}$ for tags *paragraph*
- an image generative model of parameters θ_{image} for the image contained in tag *image*

The content probability of the whole document is then computed using a mixture of these local generative models (see (7)).

3.2.2 Structural probability

In our document model, each node s_d^i has only one parent $pa(s_d^i)$. The structural probability is computed as:

$$P(s_d|\theta) = \prod_{i=1}^{|d|} P(s_d^i|pa(s_d^i), \theta) \quad (8)$$

The hypothesis here is that a structural doxel only depends on its parent. For our generating process, this means that starting at the root, the first level of structural elements is built, after that, the descendant of a structural node are build independently of the node brothers descendants. Here again, this can be viewed as a simplified process for defining the logical organization of a document.

Let $\theta^s_{s_d^i, pa(s_d^i)}$ be an estimation of $P(s_d^i|pa(s_d^i))$, we can rewrite equation (8) as:

$$P(s_d|\theta) = \prod_{i=1}^{|d|} \theta^s_{s_d^i, pa(s_d^i)} \quad (9)$$

3.2.3 Final probability

Using equations (7) and (9), we have the final probability:

$$P(d|\theta) = \left\{ \prod_{i=1}^{|d|} \theta^s_{s_d^i, pa(s_d^i)} \right\} \left\{ \prod_{i=1}^{|d|} P(t_d^i|s_d^i, \theta_{s_d^i}) \right\} \quad (10)$$

The parameters of our generative model is the vector θ where

$$\theta = \theta^s \bigcup_{l \in \Lambda} \theta_l$$

with θ^s the set of **structural parameters** ($P(s_d^i|pa(s_d^i))$) and θ_l the parameters for the local generative model of the nodes with labels l.

Equation (10) then writes:

$$P(d|\theta) = \prod_{i=1}^{|d|} \left\{ \theta^s_{s_d^i, pa(s_d^i)} P(t_d^i|s_d^i, \theta_{s_d^i}) \right\} \quad (11)$$

From equation (11), it can be seen that our global generative model corresponds to **a mixture of local generative models of the document content, weighted by transition probability models depending on the document structure**.

3.3 Learning

The model parameters will be learned by maximizing the data likelihood. Model for class c will be trained on class c data, etc. The log-likelihood of our training data is $\mathcal{D}_{TRAIN}$:

$$\begin{aligned} L &= \sum_{d \in \mathcal{D}_{TRAIN}} \left\{ \sum_{i=1}^{|s_d|} \left(logP(s_d^i|pa(s_d^i), \theta^s) \right) \right. \\ &\quad \left. + \sum_{i=1}^{|s_d|} \left(logP(t_d^i|\theta_{s_d^i}) \right) \right\} \\ &= \left\{ \sum_{d \in \mathcal{D}_{TRAIN}} \sum_{i=1}^{|s_d|} \left(logP(s_d^i|pa(s_d^i), \theta^s) \right) \right\} + \\ &\quad \left\{ \sum_{d \in \mathcal{D}_{TRAIN}} \sum_{i=1}^{|s_d|} \left(logP(t_d^i|\theta_{s_d^i}) \right) \right\} \\ &= L_{structure} + L_{content} \end{aligned} \quad (12)$$

The maximization of L amounts at two separate maximizations on $L_{structure}$ and $L_{content}$. This is a classical optimization problem and the solution for the structural and content parameters is described in the appendix.

4. EXPERIMENTS

In the following, we demonstrate the potential of the model for classifying structured documents with image and text content. Tests are performed for the particular task of HTML page filtering. We first describe the corpus and then detail the local generative models used for this application for modeling text and image information. We finally detail performances on this corpus.

	porno	general	ambiguous	total not porno
French	830	2042	420	2462
English	3808	1827	640	2467
German	357	1428	290	1718
Dutch	349	1200	220	1420
Portuguese	63	200	93	293
Spanish	530	1448	641	2089
Greek	309	870	359	1229
Italian	368	1138	223	1361
total	6614	10153	2886	13039

Figure 4: The size of each class for the 8 languages

4.1 The Netprotect II Corpus

We used for the experiments a database of HTML documents provided by EADS company. These documents have been collected on the web in the context of the European project Netprotect [1]. The project is aimed at developing a library of software tools for Internet access filtering. There are 4 categories to be filtered – pornography, violence, bomb-making and drugs – and 8 European languages – Dutch, English, French, German, Greek, Italian, Portuguese, Spanish. The database has been manually labeled.

In the experiments below, we consider all 8 languages, but only the pornography category for which most web pages do have an image + text content. Previous experiments with text only content have shown similar performances for the different categories. The dataset consists of 6613 pornographic pages, 2886 non pornographic pages with ambiguous content or dealing with sexuality and 10153 general non-pornographic Web pages which were collected using Google search engine. Figure (4) presents the corpus size for the different languages for each group (porn, ambiguous, general).

In our experiments, we used half of each group for training, and the remaining half for testing.

4.2 Content classifiers: text and image

The model described in section 3.2 makes use of a different classifier for each structural element type. For all textual doxels, we used a Naive-Bayes model (NB)[12] and for all image doxels, an histogram generative model. For the text, we use one specific NB model for each HTML tag, i.e. all text doxels under the same tag do share the same model. All image doxels are considered of the same type and share the same NB generative model.

For classifying a document, the complexity of our model is $O(|s_d| + |t_d|)$ where s_d is the number of structural nodes and t_d is the number of words and image components. This complexity is dominated by t_d and is of the same order as Naive-Bayes.

4.2.1 Text model

The text model used here is the Naive Bayes model. This is a reference model which has already been used by many different authors ([12] for example) in different contexts. It

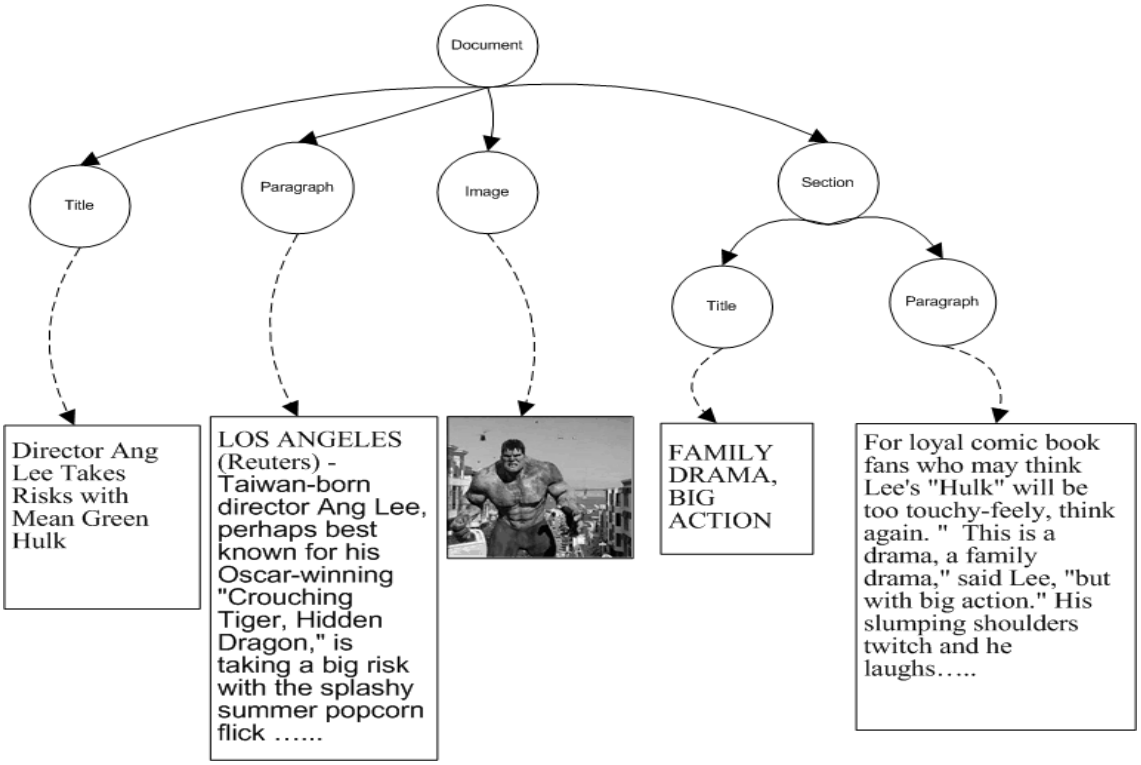

Figure 3: The belief network corresponding to the previous example

is known to be very robust when data belong to very high dimensional spaces. Since we simultaneously consider 8 languages, with different alphabets, the dictionary size is rather large and is about 20 000 for all languages (see part 4.2.3 for more details).

Let $t_d^i = (w_{d,1}^i, ..., w_{d,|t_d^i|}^i)$ be the textual content of node i in d, where $w_{d,k}^i$ represents the k-th word of node i in d. $|t_d^i|$ is the number of words in node i.

NB computes the probability $P(t_d^i|\theta_{s_d^i})$ as:

$$P(t_d^i|s_d^i, \theta_{s_d^i}) = \prod_{k=1}^{|t_d|} P(w_{d,k}^i|\theta_{s_d^i}) \tag{13}$$

Our model will use one Naive Bayes model for each label $l \in \Lambda$. The model with parameters θ_l will be learned using the flat textual representation of all nodes with label l in the training set.

4.2.2 Image model

Before deciding on the image modeling, we made extensive preliminary experiments on the classification of pornographic images. As in [5], the conclusion was that the best workspace to detect pornographic images was the RGB color space and that additional components like texture or shape did not improved performance. We then decided to represent images with a color histogram in a normalized space.

Let t_d^i be an image, its histogram representation will be:

$$t_d^i = (p_{d,1}^i, ..., p_{d,N_c}^i) \tag{14}$$

where $p_{d,k}^i$ is the number of pixels in the image with color k. N_c represents the number of colors in the histogram. In order to keep image scores comparable, image size has been normalized to N_p pixels before computing the histogram. Under the independence hypothesis, we have:

$$P(t_d^i|\theta_{s_d^i}) = \prod_{k=1}^{N_c} P(P_k = p_{d,k}^i|\theta_{s_d^i}) \tag{15}$$

where $P(P_k = p_{d,k}^i|\theta_{s_d^i})$ is the probability that there are $p_{d,k}^i$ pixels with color k in image t_d^i.

This model is learned using a simple pixel count over all the images in the training set.

4.2.3 Preprocessing

Textual parts of HTML documents have been cleaned by deleting figures, words smaller than three letters and all non-alphabetical symbols. We have kept all accents and have not stemmed any word. In order to reduce the size of the vocabulary, we have suppressed the words appearing in less than 20 documents. The final vocabulary was composed of 20834 terms. These choices satisfy different constraints. First, the model has to be simple enough to be used for example as an add-on to a navigator. An additional constraint is that new languages could be easily added for filtering. Within this context, it is thus unfeasible to use more sophisticated preprocessing since they may considerably differ from one language to the other and that the corresponding system would be to slow.

Images have been converted into features using color histograms of size 100. All images have been projected in a 216 color space. The resulting set of parameters for the image generative model has a size of 21600 features ($216 * 100$).

Each doxel – text or image – is then represented in a high

model	porno	not porno	micro average	macro average
NB	92.4	87.3	88.4	89.9
WPT	91.8	93.1	92.9	92.5
IMAGE	88.4	77.6	82.7	83.0
WPT-IMAGE	**91.6**	**95.4**	**94.7**	**93.6**

Figure 5: The recall values for the 4 models

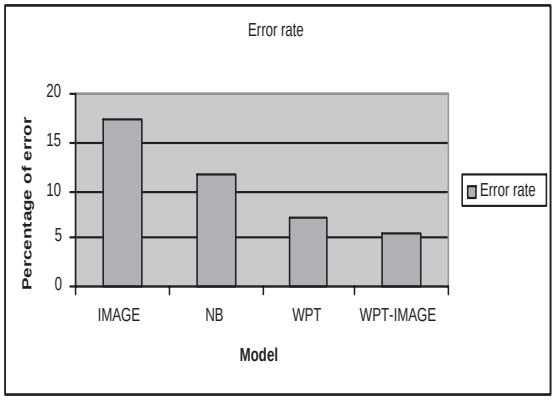

Figure 6: The error rate for the 4 models

dimensional space of about 20 K dimensions.

4.3 Evaluation

We present results obtained on the NetProtect corpus. We used 4 document models for this comparison:

- The Naive-Bayes model is the reference baseline model on the textual information
- The IMAGE model is our structured model with only images and no text.
- The WordPerTag (WPT) model is our structured model with only textual data.
- The WPT-IMAGE model is the multimedia model text and image.

In order to evaluate our model, we use the recall obtained on the test corpus for each class (porn, non porn). This is the percentage of correctly classified documents for each class on the test set. A document will be considered pornographic if:

$$P(d|\theta_{pornographic}) > P(d|\theta_{notpornographic}) \qquad (16)$$

In order to give a synthetic recall value for each classifier, we compute for each of them their micro-average and macro-average recall. *Macro-average recall* is obtained by averaging recall values for the porn and non porn classes. *Micro-average recall* is obtained by weighting the average by the relative size of each class. These results are presented in figure (5). Figure (6) present the error rate ($100 - microaverage$) for the 4 models.

The baseline Naive-Bayes model yields reasonably high micro-average and macro-average recall (88.4 % and 89.9 %). The IMAGE model is lower and achieves only 82.7 % on micro-average and 83% an macro-average. Our structured textual model WPT has good results and is 4.5% better than NB for the micro-average and 2.6% for the macro-average. The multimedia model is even better than the WPT model and achieves respectively 94.7% and 93.6% for the micro-average and the macro-average recall. These results are very encouraging and show that our structured approach, which proved to be efficient for textual data [7], can be extended to take into account different information sources. This model is able to combine efficiently the information of simple generative models. Error rate is divided by 2 compared to NB and by 3 compared to IMAGE.

5. CONCLUSION

We have described a general model for the classification of structured multimedia documents. It offers a principled approach for considering simultaneously the relations between document parts embodied in the logical structure and for integrating different sources of information. It can be used with any type of structured document and information sources, provided we have the possibility to compute local scores for all document components. We have tested a particular instance of the model on the task of Web page filtering by considering two information sources: text and image. The model has been compared to baseline flat text and image classifiers. The experiments show that taking the structure into account increases the performance compared to a flat text classifier and that the integration of textual and image information via this structured document model still increases the performance. The global classifier divides by a factor of 2 to 3 the error rate of individual classifiers.

6. REFERENCES

[1] Netprotect project page, 2001. Available as `http://www.netprotect.org/`.

[2] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *Proc. 8th Int. Conference on Computer Vision*, volume 2, pages 408–415, 2001.

[3] K. Barnard, M. Johnson, and D. Forsyth. Word sense disambiguation with pictures. In *Workshop on learning word meaning from non-linguistic data*, 2003.

[4] M. L. Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, June 1998.

[5] Y. Chan, R. Harvey, and D. Smith. Building systems to block pornography. In *Challenge of Image Retrieval*, 1999.

[6] L. Denoyer and P. Gallinari. A belief networks-based generative model for structured documents. An application to the XML categorization. In *MLDM 2003*, 2003.

[7] L. Denoyer and P. Gallinari. Using Belief Networks and Fisher Kernels for structured document classification. In *PKDD 2003*, 2003.

[8] M. Diligenti, M. Gori, M. Maggini, and F. Scarselli. Classification of HTML documents by Hidden Tree-Markov Models. In *6th International Conference on Document Analysis and Recognition*, Seattle, WA, USA, Aug. 2001.

[9] S. T. Dumais and H. Chen. Hierarchical classification of Web content. In N. J. Belkin, P. Ingwersen, and

M.-K. 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.

[10] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[11] M. J. Jones and J. M. Rehg. Detecting adult images. Technical report, 2002.

[12] D. D. Lewis. *Representation and learning in information retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.

[13] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[14] M. Ortega, K. Porkaew, and S. Mehrotra. Information retrieval over multimedia documents. In *the SIGIR Post-Conference Workshop on Multimedia Indexing and Retrieval (ACM SIGIR)*, 1999.

[15] B. Piwowarski, L. Denoyer, and P. Gallinari. Un modèle pour la recherche d'information sur des documents structurés. In *6èmes Journées internationales d'Analyse statistique des Données Textuelles (JADT 2002)*, Saint-Malo, France, Mar. 2002.

[16] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.

[17] Trec. Text REtrieval Conference (trec 2001), National Institute of Standards and Technology (NIST).

[18] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.

[19] J. Yi and N. Sundaresan. A classifier for semi-structured documents. In *Proc. Conf. Knowledge Discovery in Data*, pages 190–197, 2000.

APPENDIX

A. MAXIMIZATION OF $L_{STRUCTURE}$

We want to maximize:

$$\begin{aligned} L_{structure} &= \sum_{d \in \mathcal{D}_{TRAIN}} \sum_{i=1}^{|s_d|} logP(s_d^i | pa(s_d^i), \theta^s) \\ &= \sum_{d \in \mathcal{D}_{TRAIN}} \sum_{i=1}^{|s_d|} log\theta^s_{s_d^i, pa(s_d^i)} \end{aligned} \tag{17}$$

under the constraint $\forall m \in \Lambda, \sum_{l \in \Lambda} \theta^s_{l,m} = 1$.

Using the Lagrange multipliers, for each $(n, m) \in \Lambda \times \Lambda$, we have:

$$\frac{\partial (L_{structure} - \lambda_m (\sum_n \theta^s_{n,m} - 1)}{\partial \theta^s_{n,m}} = 0 \tag{18}$$

Let $N^d_{n,m}$ be the number of times a node of label n has his parent with label m in the document d, we solve:

$$\frac{\sum_{d \in \mathcal{D}_{TRAIN}} N^d_{n,m}}{\theta^s_{n,m}} = \lambda_m \tag{19}$$

The solution is:

$$\theta^s_{n,m} = \frac{\sum_{d \in \mathcal{D}_{TRAIN}} N^d_{n,m}}{\sum_i \sum_{d \in \mathcal{D}_{TRAIN}} N^d_{i,m}} \tag{20}$$

B. MAXIMIZATION OF $L_{CONTENT}$

We want to maximize:

$$\begin{aligned} L_{content} &= \sum_{d \in \mathcal{D}_{TRAIN}} \sum_{i=1}^{|s_d|} \left(logP(t_d^i | \theta_{s_d^i}) \right) \\ &= \sum_{l \in \Lambda} \left(\sum_{d \in \mathcal{D}_{TRAIN}} \sum_{i=1/s_d^i = l}^{|s_d|} logP(t_d^i | \theta_l) \right) \\ &= \sum_{l \in \Lambda} L^l_{content} \end{aligned} \tag{21}$$

This maximization is performed by learning each local generative model on its own data.