



HAL
open science

Weighted Naive Bayes Model for Semi-Structured Document Categorization

Pierre-François Marteau, Gilbas Ménier, Eugen Popovici

► **To cite this version:**

Pierre-François Marteau, Gilbas Ménier, Eugen Popovici. Weighted Naive Bayes Model for Semi-Structured Document Categorization. 1st International Conference on Multidisciplinary Information Sciences and Technologies InSciT2006, Oct 2006, Merida, Spain. <http://www.instac.es/inscit2006/hal-00349731>

HAL Id: hal-00349731

<https://hal.science/hal-00349731>

Submitted on 3 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Weighted Naïve Bayes Model for Semi-Structured Document Categorization

Pierre-François Marteau^a, Gildas Ménie^a, Eugen Popovic^a

^a VALORIA Laboratory, University of South-Britany, Campus de Tohannic, 56000 Vannes, France

The aim of this paper is the supervised classification of semi-structured data. A formal model based on bayesian classification is developed while addressing the integration of the document structure into classification tasks. We define what we call the structural context of occurrence for unstructured data, and we derive a recursive formulation in which parameters are used to weight the contribution of structural element relatively to the others. A simplified version of this formal model is implemented to carry out textual documents classification experiments. First results show, for a adhoc weighting strategy, that the structural context of word occurrences has a significant impact on classification results comparing to the performance of a simple multinomial naïve Bayes classifier. The proposed SCANB implementation competes on the Reuters-21578 data with the SVM classifier associated or not with the splitting of structural components. These results encourage exploring the learning of acceptable weighting strategies for this model, in particular boosting strategies.

Keywords: I Bayesian Classification, Semi-Structured Text Categorization, XML, Weighting Fusion Heuristics.

1 INTRODUCTION

Since 1990, text classification has become a very active area (cf. [17] for a review) in the Information Retrieval community. A non exhaustive list of approaches based on machine learning includes, Naïve Bayes decision [6], [8], [11], [12], [16], k-nearest neighbours [9], Support Vector Machine (SVM) [7], [10], [21]. Decision Trees and decision rules [1] and some Meta approaches such as boosting [18] and stacking [19], [22]. More recently, growing volume of semi-structured documents available either on the World Wide Web or within traditional data bases increases drastically the needs for specific information retrieval algorithms able to cope with the composite nature of these documents.

This paper contributes to the attempt to exploit the structural knowledge when categorizing semi-structured text documents. From a naïve bayes classification perspective and from the definition of XML structural context of word occurrences, we first develop a formal model. We then concentrate on a simplified implementation of this model to address experimentation on the Reuters database [15]. The results obtained are compared with other approaches such as naïve bayes classification on flat text or SVM based classification with or without the integration of structural knowledge of semi-structured documents.

2 XML CONTEXT MODELING

A semi-structured document d is well represented by a tree structure T_d , containing a set of vertices S_d and a set of edges A_d . Within the DOM tree structure associated to a well formed XML document, each node n (in particular each leaf) is connected to the root of the tree by mean of a unique path that we will refer as $c(n)$. Following earlier developments initiated in the field of approximate searching in semi-structured XML data [13], this path is an ordered sequence of XML elements that determines the occurring context of node n inside the document. In case where a leaf l of the DOM tree can be decomposed into a set of sub-elements $\{v_i\}$ we will consider that each sub-element v_i is attached to the XML context $c(l)$. In particular, if a leaf is identified as a textual element, each word v_i (lemma, stem, string, etc.) is attached to the XML context $c(l)$.

More precisely, $c(n)$ is identifiable to an ordered sequence of XML elements attached to the nodes of the path connecting node n to the root of the DOM tree:

$$c(n) = \langle e(n_0), a(n_0) \rangle \langle e(n_1), a(n_1) \rangle \dots \langle e(n_p), a(n_p) \rangle \quad \text{with :}$$

- n_0 the root node,
- n_p is the ancestor node of n_{p-1} ,
- $p+1$ is the length (number of nodes) of the path $c(n)$.
- $e(n_i)$ is the XML element attached to node n_i (« TITLE », « BODY », « PCDATA », etc.)
- $a(n_i)$ is the set of <attribute, value> pairs potentially attached to node n_i .

For a decomposable leaf l , each sub-element v_i belonging to the decomposition of l is considered as a terminal node without attribute. In such case, the occurring context of sub-element v_i will be:

$$c(v_i) = \langle e(n_0), a(n_0) \rangle \langle e(n_1), a(n_1) \rangle \dots \langle e(n_p), a(n_p) \rangle \langle e(l), a(l) \rangle$$

```

<?xml version="1.0" encoding="ISO-8859-1"?> <FILE>
<REUTERSTOPICS="NO" LEWISSPLIT="TEST" CGISPLIT="TRAINING-SET"
  OLDID="24" NEWID="21">
  <DATE>19-OCT-1987 15:27:23.12</DATE>
  <TOPICS><D>earn</D></TOPICS>
  <PLACES><D>usa</D></PLACES>
  <PEOPLE/><ORGS/><EXCHANGES/><COMPANIES/>
  <UNKNOWN>5;5;5;F22;22;1;f283231;reuter BC-LANE-TELECOMMUNICATION 10-19 0080 </UNKNOWN>
  <TEXT> 2;
    <TITLE>LANE_TELECOMMUNICATIONS PRESIDENT RESIGNS</TITLE>
    <DATELINE>HOUSTON, Oct 19 </DATELINE>
    <BODY> Lane Telecommunications Inc said Richard Lane, its president and chief operating
officer, resigned effective Oct 23. Lane founded the company in 1976 and has been its president
since its inception, ...;
  </BODY>
  </TEXT>
</REUTERS>
</FILE>

```

Fig. 1. XML file Example, extracted from the Reuters corpus ([15]).

We can distinguish several cases for the decomposition of leaf l , e.g:

- l is identified as a set of sub-element $\{v_i\}$, in this case we will write: $e(l)=\{v_i\}$.
- l identified as a sequence of sub-elements $v_1 v_2 \dots v_k$, in this case we will write: $e(l)=v_1 v_2 \dots v_k$.

An example of XML document is given in Fig. 1 while the corresponding DOM tree is given in Fig. 2. From now on to the rest of the paper, we will focus on the first case. Based on this definition of the structural context for node occurrences we develop hereinafter a model of Structural Context Augmented Naïve Bayesian Classification (SCANB).

3 SCANB MODEL

In the context of a bayesian classification of documents, *a posteriori* probabilities to choose a class ω given a test document d are related to the conditional probabilities $P(d|\omega)$ according to the Bayes's law:

$$P(\omega | d) = \frac{P(d | \omega) * P(\omega)}{P(d)} \quad (1)$$

If we accept the tree representation (see Fig. 2.) for semi-structured documents, we are lead to assimilate $P(d|\omega)$ to a conditional probability $P(T_d|\omega)$ where T_d is the tree associated to document d . We have to consider two difficulties at this level:

- some assumptions need to be formulated to decompose and simplify the estimation of these probabilities,
- given the heterogeneity of semi-structured documents, the number of parameters to cope with increases rapidly with the size of the covered application domain. The training task could require a large volume of data.

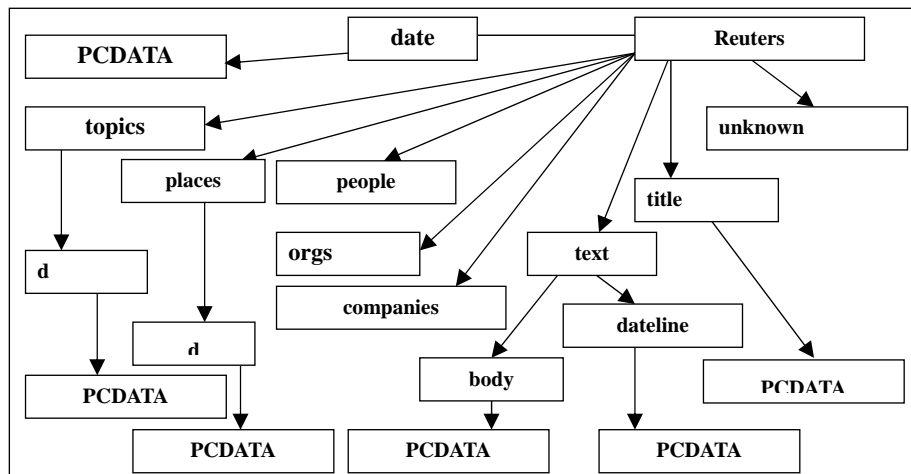


Fig. 2. DOM tree for the file presented in Figure 1. The PCDATA elements correspond to leaves (textual nodes).

To address these difficulties that relate to the complexity of the data, some simplifying approaches have been proposed while integrating document structure and content into classification paradigm. Among them, we find: the « splitting » approach [2], [3] which consists in exploiting as many classifier as there are different structural elements (or nodes) into the document data base. The final classification decision is taken according to a fusion function that combines the predictions of the classifiers associated to the structural elements. The Structured Vector Model proposed by Yi and Sundaresan [23] or the Structured Multimedia Document Classifier (SMDC) developed by Denoyer et al. [4] are examples of approaches that somehow generalize the splitting principle. The Structured Vector Model is a data structure dedicated to the representation of trees. The authors have proposed a probabilistic model that integrates local frequencies of terms that depend upon the precise localisation of the text content inside the document structure. When a large number of learning data is available, some experimentation tasks shows that the error rate is significantly lowered comparatively to a classical vector model classifier applied on the flat text. The SMDC is mainly based on two simplifying hypothesis that allow to ease the estimation of the parameters of the model, by relaxing the dependence assumption between the variables: these hypotheses states that the information contents attached to the nodes of the tree structure are independent from each other given the document structure, and furthermore, the information content depends solely on the node to which it is attached. The authors show that on a classification experimentation carried on web HTML data [14], the SMDC model improves significantly the performance of a naive bayes classifier performing on flat text.

The proposed SCANB model lies in between these previous approaches and relates also to some splitting principle. To decompose the probability $P(T_d/\omega)$ we consider the two following hypotheses:

$$(H1): \quad \begin{aligned} P(T_d/\omega) &= P(r, T_1, T_2, \dots, T_k/\omega) \\ &= P(T_1, T_2, \dots, T_k/r\omega) \cdot P(r/\omega), \end{aligned} \quad (2)$$

where r is the root node of tree T_d and $\{T_i\}$ is the set of sibling sub-trees accessible from r . To simplify the writing, r stands for the node itself or the $\langle e(r), a(r) \rangle$ pair composed with the XML element content $e(r)$ and the set of attributes $a(r)$ attached to the node r . This first hypothesis states that the order of occurrence of the sub-trees has no importance.

$$(H2): \quad \begin{aligned} P(T_d/\omega) &= K_{w,r} \cdot P(T_1/r\omega)^{w_1} \cdot P(T_2/r\omega)^{w_2} \dots P(T_k/r\omega)^{w_k} \cdot P(r/\omega) \\ &= K_{w,r} \cdot P(r/\omega) \cdot \prod_{i=1}^k P(T_i/r\omega)^{w_i} \end{aligned} \quad (3)$$

$\{w_i\}$ is a set of positive weighting factors that allow to balance the importance of the relative contribution of each sub-tree. K_w is a normalizing factor. The weightings act as a geometric mean on the sibling nodes. Note that for $w_i=1$ for all i , this hypothesis states the conditional independence of the sibling sub-trees given the root node r and the category ω . Given these two assumptions, we can recursively decompose the probability $P(T_d/\omega)$ since $P(T_i/r\omega)$ is decomposable into:

$$K_{w,r_i} \cdot P(r_i/r\omega) \cdot \prod_j P(T_{i,j}/r_i r_i \omega)^{w_j} \quad (4)$$

where r_i is the root of sub-tree T_i , $T_{i,j}$ are sub-trees accessible from r_i and $\{w_j\}$, K_{w,r_i} are the weightings and the normalizing constant attached to node r_i . Note that, by the end of the recursion, the weightings for node r_i will integrate the result of the multiplication of the weights positioned along the path from the root to node r_i .

As r_i is a sibling of the root r for tree T_d , r is identifiable to the XML context (path) of node r_i as previously defined, i.e. $c(r_i)$. In a similar way, ' r_i ' is the XML context of the root node of sub-tree $T_{i,j}$. Thus:

$$K_{w,r_i} \cdot P(r_i/r\omega) \cdot \prod_j P(T_{i,j}/r_i r_i \omega)^{w_j} = K_{w,r_i} \cdot P(r_i/c(r_i)\omega) \cdot \prod_j P(T_{i,j}/c(r_i)\omega)^{w_j} \quad (5)$$

By the end of the recursion, we obtain the simple formulation:

$$P(T_d/\omega) = \prod_{n_i \in S_d} K_{w,n_i} \cdot P(n_i/c(n_i)\omega)^{w_{n_i}} \quad (6)$$

Formula (6) states that, given hypotheses (H1) and (H2), the estimation of the probability for tree T_d (associated to semi-structured document d) conditionally to category ω is reduced to the weighted products evaluated on the set of nodes S_d of the occurring probability for node n_i given the category ω and the occurring context of node n_i inside tree T_d . To simplify the writing without loss of generality, we consider that weights w_{n_i} integrates the weights product associated to n_i 's ancestor nodes. The following example in

Fig. 3 leads to the decomposition given in formula (7), in which we have only consider a weighting for node 'REUTERS' that possesses two sibling nodes. For all other nodes, the weights are set to unity.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<FILE>
  <REUTERS ID="21">
    <TITLE>
      texte1...
    </TITLE>
    <BODY>
      texte2...
    </BODY>
  </REUTERS>
</FILE>

```

Fig. 3. Simplified example of a Reuters-21578 document.

The example in Fig. 3 leads to the decomposition given in formula (7), in which we have only considered a weighting for node 'REUTERS' that possesses two sibling nodes. For all other nodes, the weights are set to unity.

$$\begin{aligned}
P(T_d / \omega) &= P(\langle FILE, \emptyset \rangle | \omega). \\
K_{w, \text{reuters}} \cdot P(\langle \text{REUTERS}, \{\text{ID} = "21"\} \rangle | \langle FILE, \emptyset \rangle, \omega) & \quad (7) \\
P(\langle \text{TITLE}, \emptyset \rangle | \langle FILE, \emptyset \rangle \times \langle \text{REUTERS}, \{\text{ID} = "21"\} \rangle, \omega)^{w_1} & \\
P(\langle \text{"texte1..."}, \emptyset \rangle | \langle FILE, \emptyset \rangle \times \langle \text{REUTERS}, \{\text{ID} = "21"\} \rangle \times \langle \text{TITLE}, \emptyset \rangle, \omega)^{w_1} & \\
P(\langle \text{BODY}, \emptyset \rangle | \langle FILE, \emptyset \rangle \times \langle \text{REUTERS}, \{\text{ID} = "21"\} \rangle, \omega)^{w_2} & \\
P(\langle \text{"texte2..."}, \emptyset \rangle | \langle FILE, \emptyset \rangle \times \langle \text{REUTERS}, \{\text{ID} = "21"\} \rangle \times \langle \text{BODY}, \emptyset \rangle, \omega)^{w_2} &
\end{aligned}$$

In formula (6), $P(n_i / c(n_i) \omega)$ may characterize complex dependence relationships that must be furthermore detailed. For instance one can use a markov chain to limit the dependence path (for an order *one* dependence, sibling nodes depend only on the root node). This could drastically lower the number of required parameters. Moreover, considering a decomposable leaf l of tree T_d , we address here the case where the decomposition of l is represented by a set $\{v_i\}$ of sub-elements, i.e: $e(l) = \{v_i\}$ thus:

$$\begin{aligned}
P(l / c(l) \omega) &= P(\langle e(l), a(l) \rangle / c(l) \omega) = P(e(l) / a(l) c(l) \omega) \cdot P(a(l) / c(l) \omega) \\
&= P(\{v_i\} / a(l) c(l) \omega) \cdot P(a(l) / c(l) \omega) \quad (8)
\end{aligned}$$

The last hypothesis states the conditional independence of characteristics v_i :

$$\text{(H3)} \quad P(l / c(l) \omega) = P(a(l) / c(l) \omega) \prod_i P(v_i / a(l) c(l) \omega) . \quad (9)$$

5 PARAMETER ESTIMATION FOR TEXT CATEGORIZATION

From this point to the end of the paper, we will not consider attributes attached to node and will focus on textual elements. The SCANB model relies on the three hypotheses (H1), (H2) and (H3) and is defined by the set of equations:

$$\begin{aligned}
\omega^* &= \arg \max_{\omega \in \Omega} \{P(\omega / T_d)\} = \arg \max_{\omega \in \Omega} \{P(T_d / \omega) \cdot P(\omega)\} \\
P(T_d / \omega) &= \prod_{n_i \in \mathcal{S}_d} K_{w, n_i} \cdot P(n_i / c(n_i) \omega)^{w_{n_i}} \quad (10) \\
P(l / c(l) \omega) &= \prod_i P(v_i / c(l) \omega)
\end{aligned}$$

with: T_d the tree representing the semi-structured document d , n_i is a node in tree T_d , l is a leaf that decomposes into a set of independent characteristics v_i , $c(n_i)$ is the path from node n_i to the root of tree T_d , w_{n_i} is the weights for node n_i .

The estimation of probabilities $P(l/c(l)\omega)$ is handled according to the following equation:

$$\prod_i P(v_i / c(l) \omega) = \frac{N^{l,d}!}{\prod_i N_i^{l,d}!} \prod_i (\theta_i^{l,\omega})^{N_i^{l,d}}, \quad (11)$$

where $N_i^{l,d}$ is the number of time characteristic v_i occurs in the XML element attached to node l of tree T_d , $N^{l,d} = \sum_j N_j^{l,d}$ and $\theta_i^{l,\omega}$ the probability that characteristic v_i occurs in node l for class ω .

$$\text{We evaluate } \theta_i^{l,\omega} \text{ using Laplace's estimator: } \theta_i^{l,\omega} = \frac{N_i^{l,\omega} + 1}{N^{l,\omega} + |V_l|} \quad (12)$$

Where $N_i^{l,\omega}$ is the number of time characteristic v_i occurs in the set of XML elements attached to node l of training documents for class ω ,

$$N^{l,\omega} = \sum_i N_i^{l,\omega} \quad \text{and } |V_l| \text{ the size of the vocabulary for node } l.$$

In practice, we will use the following decision function: $\omega^* = \arg \max_{\omega \in \Omega} \{ \text{Log}(P(T_d / \omega) \cdot P(\omega)) \}$, i.e.:

$$\omega^* = \arg \max_{\omega \in \Omega} \left\{ \begin{array}{l} \text{Log}(P(\omega)) + LK_w + \\ \sum_{n_i \in S_d} w_{n_i} \cdot \text{Log}(P(n_i / c(n_i) \omega)) \end{array} \right\}, \quad (13).$$

where $LK_w = \sum_{n_i} \text{Log}(K_{w_{n_i}})$.

This formulation shows that the decision function is nothing but a linear combination (whose coefficients are the weights w_{n_i}) of the elementary decision rules attached to each node elements in S_d . Various learning policies or adhoc strategies can be proposed to set up these weight parameters.

6 EXPERIMENTATION

6.1 Experimental Dataset

The Reuters-21578 [25] database is widely exploited for the validation and comparison of text categorization models. In our experiments, we selected the ten most frequent categories from this corpus as our dataset for training and testing. The articles are distributed unevenly across the 10 categories with the largest category containing 3964 documents and the smallest only 286: $\Omega = \{Acq, Corn, Crude, Earn, Interest, Ship, Trade, Grain, Money-fx, Wheat\}$.

6.2 Pre-processing

We simplify the XML structure of the documents by retaining within the T_d tree the paths connecting the textual leaves to the root. Considering the Reuters data, we have retain $\langle TITLE \rangle$, $\langle BODY \rangle$ and $\langle DATELINE \rangle$ as textual elements, we segment the elements text value into words according to the following separator characters: « . ; , : ? < > = + } { () ' \ " ^ \$ \% # [] \ \ / \ n » . Reuters-21578 is a *multilabeled* dataset, so we train a binary *one-vs-rest* classifier for each category. This choice to retain the previous three XML elements and to reject the others has been governed by the concern to compare our results with other studies.

6.3 Weight Heuristic for the SCANB Model

To balance the node contributions according to a vocabulary coverage principle, we propose the following heuristic for the weight parameters of the SCANB model:

$$w_n = \begin{cases} \frac{|V_{n,\omega}|}{|d_n|}, & \text{if } n \text{ has a textual element attached} \\ 1, & \text{otherwise} \end{cases},$$

where $|V_{n,\omega}|$ is the cardinal of the vocabulary associated to node n for the category ω and $|d_n|$ stands for the size of the textual element attached to node n inside document d . As the weights w_n are independent

from the category, the LK_w term is the same for all categories and the decision rule can reduce to:

$$\omega^* = \arg \max_{\omega \in \Omega} \left\{ \text{Log}(P(\omega)) + \sum_{n \in S_d \cap T} \frac{|V_{n,\omega}|}{|d_n|} \cdot \text{Log}(P(n/c(n)\omega)) \right\}, \quad (14)$$

where $T = \{ \langle \text{TITLE} \rangle, \langle \text{BODY} \rangle, \langle \text{DATELINE} \rangle \}$.

The aim of this heuristic is, for a given document d , to increase the contribution of a node n for which the size of the attached textual element is high comparatively to the vocabulary size associated to n , and conversely, to penalize the contribution of a node n for which the size of the attached textual element is low comparatively to the vocabulary size associated to n . Thus, the weight associated to node n in document d is a kind of quality criteria that measures the vocabulary coverage of node n for document d .

6.4 Measures Used for Evaluation

To carry on comparative experimental studies, we will use the F_1 measure [20] defined as the harmonic mean of two complementary measures: precision (P) and recall (R):

$$F_1(P,R) = \frac{2 \cdot P}{P+R}, \text{ where } P = \frac{TP}{TP+FP} \text{ and } R = \frac{TP}{TP+FN}. \quad (15)$$

Precision and recall are two standard measures widely used in text categorization literature to evaluate an algorithm's effectiveness [20] on a given category ω . In the formulas above, for a category ω , the true positives (TP) is the number of documents belonging to category ω that are correctly classified as category ω ; the false positives (FP) is the number of documents not belonging to category ω that are incorrectly classified as category ω ; the false negative (FN) is the number of documents belonging to category ω that are incorrectly classified as non-category ω by a classifier. For the multilabeled Reuters-21578 dataset, we report on micro-averaged recall, (R) precision (P), and F_1 measures calculated as follows:

$$P = \frac{\sum_{\omega_i \in \Omega} TP(\omega_i)}{\sum_{\omega_i \in \Omega} TP(\omega_i) + \sum_{\omega_i \in \Omega} FP(\omega_i)}, \quad R = \frac{\sum_{\omega_i \in \Omega} TP(\omega_i)}{\sum_{\omega_i \in \Omega} TP(\omega_i) + \sum_{\omega_i \in \Omega} FN(\omega_i)} \quad (16)$$

In our experiments, we use the ten-fold cross validation method to evaluate the previous measures and to compare the classification models on the Reuters dataset.

6.5 Comparative Study with SVM Classifier as Reported by Bratko and Filipic

The above procedure has been applied for the 10 categories: $\Omega = \{Acq, Corn, Crude, Earn, Grain, Interest, money-fx, Ship, Trade, Wheat\}$, using $\langle \text{TITLE} \rangle$, $\langle \text{DATELINE} \rangle$ and $\langle \text{BODY} \rangle$ XML components as specified in the work reported by Bratko and Filipic [2][3]. We can thus compare the Naive Bayes on flat text (NB), the Naïve Bayes with Splitting (NBS), the SCANB model with the weightings given above, the SVM on flat text (SVM) and the SVM with splitting (SVMS) models. Our results on SCANB are given in table 2, while Bratko and Filipic's results are given in table 1. In our Experiments, a NBS' model close to the NBS model has been derived and evaluated from the SCANB model by setting all the weights equal to unity.

Table 1. Precision, recall and F_1 measure for models NB, NBS, SVM and SVMS from Bratko and Filipic work on Reuters-21578 database

Measure	NB	NBS	SVM	SVMS
Recall	0.9623	0.9548	0.9214	0.9660
Precision	0.8280	0.8485	0.9658	0.9178
F_1	0.8901	0.8985	0.9431	0.9413

Table 2. Precision, recall and F_1 measure for models NB, NBS and SCANB models on Reuters-21578 database, according to our tests

Measure	NB	NBS	SCANB
Recall	0.9185	0.9241	0.9540
Precision	0.8540	0.8586	0.9294
F_1	0.8851	0.8902	0.9415

On these two experiments, we notice that the NBS models perform slightly better than the NB model as previously shown by other studies [4][23]. This is corroborated by Bratko and Filipic [2][3] that find similar results when comparing the naive bayes classifier applied on flat text or in conjunction with a splitting method. On the other hand the SCANB model, with the proposed weightings heuristic shows

significant improvements since a 5.6% gain is achieved on the F_1 measure against the naïve bayes model applied on flat text and a 5.1% gain is achieved on the F_1 measure against the NBS' model. This experimentation shows that a proper weighting strategy, in the context of the Reuters-21578 data base, has a significant impact on the classification accuracy. The proposed weightings strategy performs quite comparatively to the SVM model that seems insensitive to the structural organization of documents.

7 CONCLUSION

We have considered the problem of supervised categorization for semi-structured data. A formal model named SCANB in between approaches proposed by Yi and Sundaresan [23], Denoyer an al. [4] or the splitting procedure described in Bratko and Filipic [2][3] has been developed. This model extends traditional naïve Bayesian classification while integrating document structure knowledge. Given some acceptable hypotheses, we get a recursive development for the model that is original to our knowledge. It allows approximating the tree structure of semi-structured data as a set of nodes from which the path to the root is attached. For each node, a weighting parameter is added to balance the importance of the node comparatively to the others according to some heuristics. First results performed on the reuters-21578 database shows that the structure of documents plays a crucial role as it improve the classification efficiency. Given an ad hoc heuristic independent from the categories, the SCANB model attain the same level of accuracy as the SVM model applied on flat text or after the splitting of textual components. The linear form of the decision function given in equation 13 suggests that a boosting approach [18] [24] can be easily experimented.

REFERENCES

- [1] Apté, C. d., Damerau, F. J., and Weiss, S. M.: Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems* 12, 3, (1994) 233–251.
- [2] Bratko, A. and Filipic, B.: Exploiting Structural Information in Semi-structured Document Classification. *Proc. 4th International 13th Electrotechnical and Computer Science Conference, ERK'2004*, September 2004.
- [3] Bratko, A. and Filipic, B.: A Study of Approaches to Semi-structured Document Classification. Technical Report IJS-DP-9015, Department of Intelligent Systems, Jozef Stefan Institute, November 2004.
- [4] Denoyer L., Vittaut J.N., Gallinari P., Brunessaux Sy., Brunessaux St.: Structured Multimedia Document Classification, in *DocEng'03*, November 20–22, 2003, Grenoble, France.
- [5] Domingos P. and Pazzani M.: Beyond independence: conditions for the optimality of the simple bayesian classi_er. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML)*, 1996.
- [6] Duda R.O. and Hart P.E. *Pattern Classification and Scene Analysis* John Wiley & S., 1973
- [7] Dumais S., Platt J., D.H., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: *CIKM*, ACM (1998) 148–155
- [8] Eyheramendy, S., Lewis, D., Madigan, D. (2003) On the Naive Bayes Model for Text Categorization. *Artificial Intelligence and Statistics* 2003.
- [9] Guo G, Wang H., Bell D., Bi Y., and Greer K.. “Using kNN Model-based Approach for Automatic Text Categorization”. Submitted to *Soft Computing Journal*, October, 2003,
- [10] Joachims, T.: Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning* (1999)
- [11] Lewis, D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: *SDAIR*. (1994) 81–93
- [12] McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In *Proceedings of AAAI-98 Workshop on “learning for Text Categorization”*, AAAI Press, (1998).
- [13] Ménier G, Marteau P.F., Information retrieval in heterogeneous XML knowledge bases, *The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IEEE, 1-5 July, (2002), Annecy, France.
- [14] Netproject, 2001, <http://www.netproject.org>
- [15] <http://www.daviddlewis.com/resources/testcollections/reuters21578/>
- [16] Rish, I.: An Empirical Study of the Naive Bayes Classifier. In *Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, (2001).
- [17] Sebastiani, F.: *Machine Learning in Automated Text Categorization*. *ACM Computing Surveys*, Vol.34, No.1, March 2002, (2002) p. 1-47.
- [18] Schapire, R., Singer, Y.: BoosTexter: A boosting-based system for text categorization. *Machine Learning* 39 (2000) 135–168
- [19] Ting K. M. and Witten I. H.: Stacked generalization: When does it work? In *Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence*, p. 866–873, (1997).
- [20] van Rijsbergen C. J. : *Information Retrieval*. Butterworths, London, second edition, (1979).
- [21] Vapnik, V.: *Support Vector Networks*. Statistical Learning Theory. Wiley-Interscience, (1995)
- [22] Wolpert D. H.: Stacked generalization. *Neural Networks*, 5:241–259, (1992).
- [23] Yi J. and Dundaresan N.: A classifier for semi-structured documents. in *Proc. of the 6th International Conference on Knowledge Discovery and Data mining (KDD)*, (2000).
- [24] Schapire R. E.: A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, (1999).
- [25] Reuters-21578, ModApte version: <http://www.daviddlewis.com/resources/testcollections/reuters21578>