Multi-label text classification using multinomial models*

David Vilar[1], María José Castro[2], and Emilio Sanchis[2]

[1] Lehrstuhl für Informatik VI,
Computer Science Department,
RWTH Aachen University,
D-52056 Aachen (Germany),
email: vilar@cs.rwth-aachen.de
[2] Departament de Sistemes Informàtics i Computació,
Universitat Politècnica de València,
E-46022 València, Spain,
email: {mcastro,esanchis}@dsic.upv.es

Abstract. Traditional approaches to pattern recognition tasks normally consider only the unilabel classification problem, that is, each observation (both in the training and test sets) has one unique class label associated to it. Yet in many real-world tasks this is only a rough approximation, as one sample can be labeled with a set of classes and thus techniques for the more general multi-label problem have to be explored. In this paper we review the techniques presented in our previous work and discuss its application to the field of text classification, using the multinomial (Naive Bayes) classifier. Results are presented on the Reuters-21578 dataset, and our proposed approach obtains satisfying results.

1 Introduction

Traditional approaches to pattern recognition tasks normally consider only the unilabel classification problem, that is, each observation (both in the training and test sets) has one unique class label associated to it. Yet in many real-world tasks this is only a rough approximation, as one sample can be labeled with a set of classes and thus techniques for the more general multi-label problem have to be explored. In particular, for multi-labeled documents, text classification is the problem of assigning a text document into one or more topic categories or classes [1]. There are many ways to deal with this problem. Most of them involve learning a number of different binary classifiers and use the outputs of those classifiers to determine the label or labels of a new sample[2]. We explore this approach using a multinomial (Naive Bayes) classifier and results are presented on the Reuters-21578 dataset. Furthermore, we explore the result that using an accumulated posterior probability approach to multi-label text classification performs favorably compared to the more standard binary approach to multi-label classification.

The methods we discuss in this paper were applied to the classification phase of a dialogue system using neural networks [3], but the simplicity of the methods allows

* This work has been partially supported by the Spanish CICYT under contracts TIC2002-04103-C03-03 and TIC2003-07158-C04-03

us to easily extend the same ideas to other application areas and other types of classifiers, such as the multinomial Naive Bayes classifier considered in this work for text classification.

2 Unilabel and Multi-Label Classification Problems

Unilabel classification problems involve finding a definition for an unknown function $k^\star(\boldsymbol{x})$ whose range is a discrete set containing $|\mathcal{C}|$ values (i.e., $|\mathcal{C}|$ "classes" of the set of classes $\mathcal{C} = \{c^{(1)}, c^{(2)}, \ldots, c^{(|\mathcal{C}|)}\}$). The definition is acquired by studying collections of training samples of the form

$$\{(\boldsymbol{x}_n, c_n)\}_{n=1}^{N}, \quad c_n \in \mathcal{C}, \tag{1}$$

where $\boldsymbol{x}_n$ is the n-th sample and c_n is its corresponding class label.

For example, in handwritten digit recognition, the function $k^\star$ maps each handwritten digit to one of $|\mathcal{C}| = 10$ classes. The Bayes decision rule for minimizing the probability of error is to assign the class with maximum a posteriori probability to the sample $\boldsymbol{x}$:

$$k^\star(\boldsymbol{x}) = \underset{k \in \mathcal{C}}{\operatorname{argmax}} \Pr(k|\boldsymbol{x}). \tag{2}$$

In contrast to the unilabel classification problem, in other real-world learning tasks the unknown function $k^\star$ can take more than one value from the set of classes $\mathcal{C}$. For example, in many important document classification tasks, like the Reuters-21578 corpus we will consider in Section 4, documents may each be associated with multiple class labels [1, 4]. In this case, the training set is composed of pairs of the form

$$\{(\boldsymbol{x}_n, C_n)\}_{n=1}^{N}, \quad C_n \subseteq \mathcal{C}. \tag{3}$$

Note that the unilabel classification problem is a special case in which $|C_n| = 1$ for all samples.

There are two common approaches to this problem of classification of objects associated with multiple class labels. The first is to use specialized solutions like the accumulated posterior probability approach described in the next section. The second is to build a binary classifier for each class as explained afterwards.

Note that in certain practical situations, the amount of possible multiple labels is limited due to the nature of the task and this can lead to a simplification of the problem. For instance, if we know that the only possible appearing multiple labels can be $\{c^{(i)}, c^{(j)}\}$ and $\{c^{(i)}, c^{(k)}\}$ we do not need to consider all the possible combinations of the initial labels. In such situations we can handle this task as an unilabel classification problem with the extended set of labels $\hat{\mathcal{C}}$ defined as a subset of $\mathcal{P}(\mathcal{C})$. The question whether this method can be reliably used is highly task-dependent.

2.1 Accumulated Posterior Probability

In a traditional (unilabel) classification system, given an estimation of the a posteriori probabilities $\Pr(k|\boldsymbol{x})$, we can think of a classification as "better estimated" if the probability of the destination class is above some threshold (i.e., the classification of a sample

$\boldsymbol{x}$ as belonging to class k is better estimated if $\Pr(k|\boldsymbol{x}) = 0.9$ than if it is only 0.4). A generalization of this principle can be applied to the multi-label classification problem.

We can consider that we have correctly classified a sample only if the *sum* of the a posteriori probabilities of the assigned classes is above some threshold $\mathcal{T}$. Let us define this concept more formally. Suppose we have an ordering (permutation) $\{k^{(1)}, k^{(2)}, \ldots, k^{(|\mathcal{C}|)}\}$ of the set $\mathcal{C}$ for a sample $\boldsymbol{x}$, such that

$$\Pr(k^{(i)}|\boldsymbol{x}) \geq \Pr(k^{(i+1)}|\boldsymbol{x}) \quad \forall 1 \leq i < |\mathcal{C}|\,. \tag{4}$$

We define the *accumulated posterior probability* for the sample $\boldsymbol{x}$ as

$$\Pr_j(\boldsymbol{x}) = \sum_{i=1}^{j} \Pr(k^{(i)}|\boldsymbol{x}) \qquad 1 \leq j \leq |\mathcal{C}|\,. \tag{5}$$

Using the above equation, we classify the sample $\boldsymbol{x}$ in n classes, being n the smallest number such that

$$\Pr_n(\boldsymbol{x}) \geq \mathcal{T}\,, \tag{6}$$

where the threshold $\mathcal{T}$ must also be learned automatically in the training process. Then, the set of classification labels for the sample $\boldsymbol{x}$ is simply

$$K^\star(\boldsymbol{x}) = \{k^{(1)}, \ldots, k^{(n)}\}\,. \tag{7}$$

2.2 Binary Classifiers

Another possibility is to treat each class as a separate binary classification problem (as in [5–7]). Each such problem answers the question, whether a sample should be assigned to a particular class or not.

For $C \subseteq \mathcal{C}$, let us define $C[c]$ for $c \in \mathcal{C}$ to be:

$$C[c] = \begin{cases} \mathsf{true}, & \text{if } c \in C\,; \\ \mathsf{false}, & \text{if } c \notin C\,. \end{cases} \tag{8}$$

A natural reduction of the multi-label classification problem is to map each multi-labeled sample $(\boldsymbol{x}, C)$ to $|\mathcal{C}|$ binary-labeled samples of the form $(\langle \boldsymbol{x}, c\rangle, C[c])$ for all $c \in \mathcal{C}$; that is, each sample is formally a pair, $\langle \boldsymbol{x}, c\rangle$, and the associated binary label, $C[c]$. In other words, we can think of each observed class set C as specifying $|\mathcal{C}|$ binary labels (depending on whether a class c is or not included in C), and we can then apply unilabel classification to this new problem. For instance, if a given training pair $(\boldsymbol{x}, C)$ is labeled with the classes $c^{(i)}$ and $c^{(j)}$, $(\boldsymbol{x}, \{c^{(i)}, c^{(j)}\})$, then $|\mathcal{C}|$ binary-labeled samples are defined as $(\langle \boldsymbol{x}, c^{(i)}\rangle, \mathsf{true})$, $(\langle \boldsymbol{x}, c^{(j)}\rangle, \mathsf{true})$ and $(\langle \boldsymbol{x}, c\rangle, \mathsf{false})$ for the rest of classes $c \in \mathcal{C}$.

Then a set of binary classifiers is trained, one for each class. The ith classifier is trained to discriminate between the ith class and the rest of the classes and the resulting classification rule is

$$K^\star(\boldsymbol{x}) = \{k \in \mathcal{C} \mid \Pr(k|\boldsymbol{x}) \geq \mathcal{T}\}\,, \tag{9}$$

being $\mathcal{T}$ a threshold which must also be learned. Note that in the standard binary classification problem $\mathcal{T} = 0.5$, but experiments have shown that better results are obtained if we allow the more general formulation of equation (9). We can also allow more generalization by estimating one different threshold $\mathcal{T}_c$ for each class, but this would mean an increased number of parameters to estimate and the approach with only one threshold often works well in practice.

3 The Multinomial Model

As application of the multi-label classification rules we will consider a text classification task, where each document will be assigned a W-dimensional vector of word counts, where W is the size of the vocabulary. This representation is known as "bag-of-words". As classification model we use the *Naive Bayes* text classifier in its *multinomial* event model instantiation [8]. In this model, we make the assumption that the probability of each event (word occurrence) is independent of the word's context and position in the document it appears, and thus the chosen representation is justified. Given the representation of a document by its counts $\boldsymbol{x} = (x_1, \ldots, x_W)^t$ the class-conditional probability is given by the multinomial distribution

$$p(\boldsymbol{x}|c) = p(x_+|c)p(\boldsymbol{x}|c, x_+) = p(x_+|c)\frac{x_+!}{\prod_w x_w!}\prod_w p(w|c, x_+)^{x_w}\,, \tag{10}$$

where $w = 1, \ldots, W$ denotes the word variable, $x_+ = \sum_w x_w$ is the length of document $\boldsymbol{x}$, and $p(w|c, x_+)$ are the parameters of the distribution, with the restriction

$$\sum_w p(w|c, x_+) = 1 \quad \forall c, x_+\,. \tag{11}$$

In order to reduce the number of parameters to estimate we assume that the distribution parameters are independent of the length x_+ and thus $p(w|c, x_+) = p(w|c)$, and that the length distribution is independent of the class c, so (10) becomes

$$p(\boldsymbol{x}|c) = p(x_+)\frac{x_+!}{\prod_w x_w!}\prod_w p(w|c)^{x_w}\,. \tag{12}$$

Applying Bayes rule we obtain the unilabel classification rule

$$\begin{aligned} k^\star(\boldsymbol{x}) &= \operatorname*{argmax}_{c\in\mathcal{C}}\{p(c|\boldsymbol{x})\} \\ &= \operatorname*{argmax}_{c\in\mathcal{C}}\{\log p(c)p(\boldsymbol{x}|c)\} \\ &= \operatorname*{argmax}_{c\in\mathcal{C}}\left\{\log p(c) + \sum_w x_w \log p(w|c)\right\}\,. \end{aligned} \tag{13}$$

The multi-label classification rules can be adapted accordingly.

To estimate the prior probabilities $p(c)$ of the class and the parameters $p(w|c)$ we apply the maximum-likelihood method. In the training phase we replicate the multi-labeled samples, that is, we transform the training set $\{(x_n, C_n)\}_{n=1}^{N}, C_n \subseteq \mathcal{C}$ into the the "unilabel" training set

$$\begin{aligned}\mathcal{M}\left(\{(\boldsymbol{x}_n, C_n)\}_{n=1}^{N}\right) &= \bigcup_{n=1}^{N} \bigcup_{c \in C_n} \{(\boldsymbol{x}_n, c)\} \\ &=: \{(\tilde{\boldsymbol{x}}_n, \tilde{c}_n)\}_{n=1}^{\tilde{N}}.\end{aligned} \tag{14}$$

The log-likelihood function of this training set is then

$$\begin{aligned}\log \mathcal{L}(\{p(c)\}, \{p(w|c)\}) = \sum_{n=1}^{\tilde{N}} \Bigg(&\log p(c_n) + \sum_{w} \tilde{x}_{nw} \log p(w|\tilde{c}_n) \\ &+ \operatorname{const}(\{p(c)\}, \{p(w|c)\}) \Bigg) .\end{aligned} \tag{15}$$

Using Lagrange multipliers we maximize this function under the constrains

$$\sum_{c} p(c) = 1 \quad \text{and} \quad \sum_{w} p(w|c) = 1, \quad \forall 1 \leq c \leq |\mathcal{C}| . \tag{16}$$

The resulting estimators[3] are the relative frequencies

$$\hat{p}(c) = \frac{N_c}{\tilde{N}} \tag{17}$$

and

$$\hat{p}(w|c) = \frac{N_{cw}}{\sum_{w'} N_{cw'}} , \tag{18}$$

where $N_c = \sum_n \delta(\tilde{c}_n, c)$ is the number of documents of class c and similarly $N_{cw} = \sum_n \delta(\tilde{c}_n, c)\tilde{x}_{nw}$ is the total number of occurrences of word w in all the documents of class c. In this equations $\delta(\cdot, \cdot)$ denotes the Kronecker delta function, which is equal to one if its both arguments are equal and zero otherwise.

3.1 Smoothing

Parameter smoothing is required to counteract the effect of statistical variability of the training data, particularly when the number of parameters to estimate is relatively large in comparison with the amount of available data. As smoothing method we will use *unigram interpolation* [9].

The base of this method is known as *absolute discounting* and it consists of gaining "free" probability mass from the seen events by discounting a small constant b to every (positive) word count. The idea behind this model is to leave the high counts virtually

[3] We will denote parameter estimations with the hat (ˆ) symbol.

unchanged, with the justification that for a corpus of approximately the same size, the counts will not differ much, and we can consider the "average" value, using a non-integer discounting. The gained probability mass for each class c is

$$M_c = \frac{b \cdot \left|\{w' : N_{cw'} > 0\}\right|}{\sum_{w'} N_{cw'}}\,, \tag{19}$$

and is distributed in accordance to a *generalized distribution*, in our case, the *unigram distribution*

$$p(w) = \frac{\sum_c N_{cw}}{\sum_{w'} \sum_c N_{cw'}}\,. \tag{20}$$

The final estimation thus becomes

$$\hat{p}(w|c) = \max\left\{0, \frac{N_{cw} - b}{\sum_{w'} N_{cw'}}\right\} + p(w) M_c\,. \tag{21}$$

The selection of the discounting parameter b is crucial for the performance of the classifier. A possible way to estimate it is using the so called *leaving-one-out* technique. This can be considered as an extension of the cross-validation method [10, 11]. The main idea is to split the N observations (documents) of the training corpus into $N-1$ observations that serve as training part and only 1 observation, the so called hold-out part, that will constitute the simulated training test. This process is repeated N times in such a way that every observation eventually constitutes the hold-out set. The main advantage of this method is that each observation is used for both the training and the hold-out part and thus we achieve and efficient exploitation of the given data. For the actual parameter estimation we again use maximum likelihood. For further details the reader is referred to [12].

No closed form solution for the estimation of b using leaving-one-out can be given. Nevertheless, an interval for the value of this parameter can be explicitly calculated as

$$\frac{n_1}{n_1 + 2n_2 + \sum_{r\geq 3} n_r} < b < \frac{n_1}{n_1 + 2n_2}\,. \tag{22}$$

where $n_r = \sum_w \delta(\sum_c N_{cw}, r)$ is the number of words that have been seen exactly r times in the training set. Since in general leaving-one-out tends to underestimate the effect of unseen events we choose to use the upper bound as the leaving-one-out estimate

$$\hat{b}_{llo} \cong \frac{n_1}{n_1 + n_2}\,. \tag{23}$$

3.2 A Note About Implementation

On the actual implementation of the multinomial classifier we can not directly compute the probabilities as given in equation (12) due to underflows in the computation of the exponentiation of the multinomial parameters[4]. In the unilabel classification tasks (and therefore in the extension to binary classifiers) we avoid this problem by using the joint

[4] Note that the multinomial coefficient cancels when applying Bayes rule.

probability in the maximization (see eq. (13)), but for the accumulated posterior probability approach we have to work with real posterior probabilities in order to handle the threshold in a correct way. A possibility to compute this probabilities in a numerically stable way is to introduce a maximum operation in Bayes rule

$$p(c|\boldsymbol{x}) = \frac{\dfrac{p(\boldsymbol{x},c)}{\max_{c''} p(\boldsymbol{x},c'')}}{\displaystyle\sum_{c'} \frac{p(\boldsymbol{x},c')}{\max_{c''} p(\boldsymbol{x},c'')}}\,, \tag{24}$$

and then introduce a logarithm and an exponentiation function that allow us to compute the probabilities in a reliable way

$$p(c|\boldsymbol{x}) = \frac{\exp\left(\log p(\boldsymbol{x},c) - \max_{c''} \log p(\boldsymbol{x},c'')\right)}{\displaystyle\sum_{c'} \exp\left(\log p(\boldsymbol{x},c') - \max_{c''} \log p(\boldsymbol{x},c'')\right)}\,. \tag{25}$$

4 Experimental Results

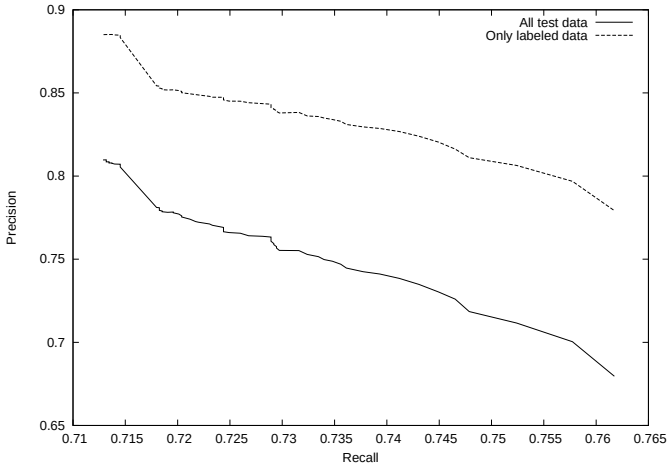
4.1 The Dataset

As corpus for our experiments we use the Reuters-21578, a collection of articles appeared in the Reuters newswire in 1987. More precisely we use the Modified Apte Split as described in the original corpus, consisting of a training set of $9\,603$ documents and a test set of $3\,299$ documents (the remaining $8\,676$ are not used). Although this partition originally intended to restrict the set of used documents to those with one or more well defined class labels (topics as they are called in the documentation), problems with an exact definition of what was exactly meant with 'topic' results in documents without associated class labels appearing both in the training and the test set. Statistics of the corpus are shown in Table 1.

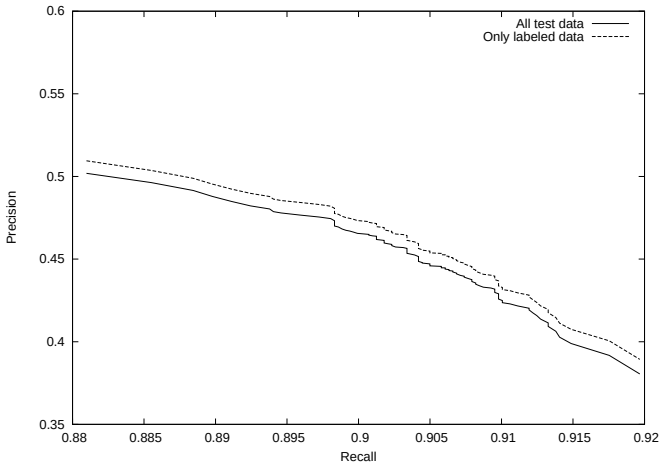
Table 1. Statistics for the Reuters-21578 dataset.

	Number of documents			
	Total	**No label**	**Unilabel**	**Multi-Label**
Training	9 603	1 828 (19.0%)	6 552 (68.3%)	1223 (12.7%)
Test	3 299	280 (8.5%)	2 581 (78.2%)	438 (13.3%)

In spite of the explanation given in the "README" file accompanying the dataset, we feel that the presence of unlabeled documents in the corpus is not adequate, as they seem to be the result of an incorrect labelling, and therefore should be eliminated of the test set. We report results with the whole set, however, in order to better compare our results with other researches. On the other hand, the presence of such documents in the training set does provide some useful information and can be considered as a "real life"



(a) Accumulated Posterior Probability



(b) Binary Classifiers

Fig. 1. Precision and recall curves for the Reuters-21578 dataset. Note the different scaling of the axis.

situation, where only a subset of the available data has been labeled. In our case we use the unlabeled documents as an aid to better estimate the smoothing parameters, but can also be used in a more powerful way [7]. This will be the subject of further research.

For the accumulated posterior probability approach, the presence of unlabeled samples in the test set represents immediately a classification error, as the definition of the approach requires that at least one label to be detected. One possibility to avoid this problem could be to include a "<no_class>" label, trained with the unlabeled samples in the training set and being mutually exclusive with the other classes. This seems however an ad hoc solution that does not generalize well so we decided not to apply it. On the other hand, the binary classifiers can handle the case of unlabeled samples in a natural way, if none of the posterior probabilities lies above the predefined threshold.[5]

4.2 Results

We will present present several figures as a measures of the effectiveness of our methods in order of increasing difficulty of the task. First we consider the simple unilabel classification problem, that is, only the samples with one unique class-label are considered. We obtain an error rate of 8.56% in this case. If we include the non-labeled samples for a better estimation of the smoothing parameters we do not get any improvement in the error rate.

In addition to the error rate, in the multi-label classification problem we also consider the precision/recall measure. It is worth noting that in most previous work, the error rate is not considered as an appropiate measure of the effectiveness of a multi-label classification system, as it does not take into consideration "near misses", that is for example the case when all the detected labels are correct but there is still one label missing. This is clearly an important issue, but for some applications, specially when the classification system is only a part of a much bigger system (see for example [3]) such a "small" error does have a great influence on the output of the whole system, as it propagates into the subsequent constituents. Therefore we feel that the true error rate should also be included in such a study.

In the case of multi-label classification, precision is defined as

$$\text{precision} = \frac{\text{\# of correct detected labels}}{\text{\# of detected labels}} \tag{26}$$

and recall as

$$\text{recall} = \frac{\text{\# of correct detected labels}}{\text{\# of reference labels}} \tag{27}$$

where "detected labels" corresponds to the labels in the output of the classifier.

The curves shown in Figure 1 are obtained modifying the threshold $\mathcal{T}$ in the range $(0, 1)$. Note that because of this method for generating the curves the axis ranges are quite different. We can observe that the accumulated posterior probability approach has a much higher precision rate than the binary classifiers, which, in turn, have a higher recall rate. That means that the accumulated posterior probability approach does a "safe"

[5] In the "normal" case where each sample should be labeled, we could choose the class with highest probability as the one unique label if no probability is higher than the threshold.

classification, where the output labels have a high probability to be right, but it does not find all the reference class labels. The binary classifiers, on the other hand, do find most of the correct labels but at the expense of also outputting a big amount of incorrect labels. The effect of (not) including the non labeled test samples can also be seen in the curves. As expected, the performance of the accumulated posterior probability approach increases when leaving this samples out. In the case of the binary classifiers, the difference is not as big, but better results are still obtained when using only the labeled data.

It is also interesting to observe the evolution of the error rate when varying the threshold value. For the multi-label problem, for a sample to be correctly classified, the whole set of reference labels must be detected. That is, the number of detected labels must be the same in the reference and the output of the classifier (and obviously the labels also have to be the same). This a rather strict measure, but one must consider that in many systems the classification is only one step in a chain of processes and we are interested in the exact performance of the classifier [3]. The error rate is showed in Figure 2. Note that this curves show the error rate on the *test set* in order to analyze the behavior of the classification methods. For a real-world classification system we should choose an appropriate threshold value (for example by using a validation set) and then use this value in order to obtain a figure for the error rate.

We see that when considering the error rate, the accumulated posterior probability approach performs much better than the binary classifiers. For this approach the threshold does not have a great influence on the error rate unless we use high values, where an increase of the number of class labels the classifier has to include for reaching the threshold produces an increase of the error rate. Somehow surprisingly, for binary classifiers, the best results are obtained for low threshold values. This is probably due to the unclean division between the classes defined in every binary subproblem, that leads to an incorrect parameter estimation. Taking into account the correlation between the classes may help to alleviate the problem.

5 Conclusions

In this paper we have discussed some possibilities to handle the multi-label classification problem. The methods are quite general and can be applied to a wide range of statistical classifiers. Results on text classification with the Reuters-21578 corpus have been presented, where the accumulated posterior probability approach performs better that the most widely used binary classifiers.

However, in these approaches we did not take the relation between the different classes into account. Modeling this information may provide a better estimation of the parameters and better results can be expected. For the Reuters-21578 corpus in particular, a better exploitation of the unlabeled data can also lead to an improvement in performance.

References

1. McCallum, A.K.: Multi-Label Text Classification with a Mixture Model Trained by EM. In: NIPS99. (1999)

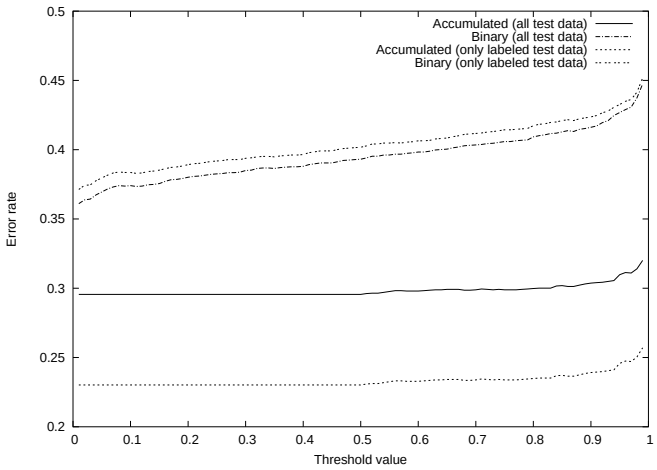


Fig. 2. Error rate for the two multi-label methods.

2. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34** (2002) 1–47
3. Castro, M.J., Vilar, D., Sanchis, E., Aibar, P.: Uniclass and Multiclass Connectionist Classification of Dialogue Acts. In Sanfeliu, A., Ruiz-Shulcloper, J., eds.: Progress in Pattern Recognition, Speech and Image Analysis. 8 th Iberoamerican Congress on Pattern Recognition (CIARP 2003). Volume 2905 of Lecture Notes in Computer Science. Springer (2003) 266–273
4. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning **39** (2000) 135–168
5. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval **1** (1999) 69–90
6. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: Proceedings of ECML-98, 10th European Conference on Machine Learning. Number 1398, Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142
7. Nigam, K., McCalum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning **39** (2000) 103–134
8. McCallum, A., Nigam, K.: A comparison of event models for naive Bayes text classification. In: AAAI/ICML-98 Workshop on Learning for Text Categorization, AAAI Press (1998) 41–48
9. Juan, A., Ney, H.: Reversing and Smoothing the Multinomial Naive Bayes Text Classifier. In: Proc. of the 2nd Int. Workshop on Pattern Recognition in Information Systems (PRIS 2002), Alacant (Spain) (2002) 200–212
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, New York, NY, USA (2001)
11. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York, NY, USA (1993)
12. Ney, H., Martin, S., Wessel, F.: Satistical Language Modeling Using Leaving-One-Out. In: Corpus-based Methods in Language and Speech Proceesing. Kluwer Academic Publishers, Dordrecht, the Netherlands (1997) 174–207