International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
ISSN: 2150-7988 Vol.1 (2009), pp.133-144
http://www.mirlabs.org/ijcisim

# Multi-Label Text Categorization Using a Probabilistic Neural Network

**Patrick Marques Ciarelli[1], Elias Oliveira[2], Claudine Badue[3] and Alberto Ferreira De Souza[3]**

[1]Department of Electrical Engineering
*pciarelli@lcad.inf.ufes.br*

[2]Department of Information Science
*elias@lcad.inf.ufes.br*

[3]Department of Computer Science
{*claudine, alberto*}*@lcad.inf.ufes.br*

Universidade Federal do Espírito Santo
Campus de Goiabeiras, Av. Fernando Ferrari, s/n, Cx Postal 5011, 29060-970 – Brazil.

*Abstract*: **Techniques for categorization and clustering, range from support vector machines, neural networks to Bayesian inference and algebraic methods. The k-Nearest Neighbor Algorithm (kNN) is a popular example of the latter class of these algorithms. Recently, slightly modified versions of support vector machines, kNN and decision trees have been proposed to deal better with multi-label classification problems. In this paper, we also proposed a new version of a Probabilistic Neural Network (PNN) to tackle these kind of problems. This PNN was proposed aiming at executing automatic classification of economic activities, which is the focus of this article. Nevertheless, we compared the PNN algorithm against other classifiers. In addition to economic activities database, we applied our algorithm to some other databases found in the literature. In general, our approach surpassed the other algorithms in many metrics typically well known in the literature for the multi-label categorization problems.**

***Keywords***: Multi-Label Categorization Problems, Machine Learning, Business Activities Classification, Probabilistic Neural Network.

## I. Introduction

Automatic text classification and clustering are still very challenging computational problems to the information retrieval (IR) communities, both in academic and industrial contexts. Currently, a great effort of work on IR, one can find in the literature, is focused on classification and clustering of generic content of text documents. However, there are still many other important applications to which little attention has hitherto been paid, which are as well very difficult to deal with. One example of these applications is the classification of companies based on the descriptions of their economic activities, also called mission statements, which represent the business context of the companies' activities, in other words, the business economic activities from free text description by the company's founders.

The categorization of companies according to their economic activities constitutes a very important step towards building tools for obtaining information for performing statistical analysis of the economic activities within a city or country. With this goal, the Brazilian government is creating a centralized digital library with the business economic activity descriptions of all companies in the country. This library will serve the three government levels: Federal; the 27 States; and more than 5.000 Brazilian counties. We estimate that the data related to nearly 1.5 million companies will have to be processed every year into more than 1.000 possible different activities. It is important to highlight that the large number of possible categories makes this problem particularly complex when compared with others presented in the literature [1]. Moreover, the possibility of each activities' description to be assigned more than one category, *i.e.*, a multi-label assignment, turns this task even harder.

The economic activities categorization is just one from many other multi-label problem cases. To treat similarly problems, it has been proposed in the literature a variety of metrics and classifiers that are specialized to solve problems such as these. Some of these classifiers are: the ML-kNN, that is based on the kNN [2], Rank-SVM [3], a modified version of SVM, ADTBoost.MH [4] and BoosTexter [5], that are both techniques based on decision trees.

In this paper we presented a slightly modified version of the standard structure of the probabilistic neural network (PNN) [6], so that we could deal with the multi-label problem faced in this work. We have chosen the PNN classifier because of its implementation simplicity and high computational speed in the training stage, when compared to other algorithms, such as SVM and Backpropagation Neural Networks. The complexity of SVM, for example, grows quadratically with the size of the dataset, being thus a bottleneck for large dataset problems [7]. We compared the PNN performance against the ML-kNN, Rank-SVM, ADTBoost.MH and Boostexter applying them to some literature benchmark databases,

and to our economic activities database. In general, our classifier showed to be superior to the other ones in the experiments we have done.

This work is organized as follows. In Section II, we detail more the characteristics of the problem and its importance for the government institutions in Brazil. We describe our probabilistic neural network algorithm in Section III. In Section IV, the experimental results are discussed. A revision about related works is done in Section V. Finally, we present our conclusions and indicate some future paths for this research in Section VI.

## II. The Problem of Economic Activities Classification

In many countries, companies must have a contract (*Articles of Incorporation* or *Corporate Charter*, in USA) with the society where they can legally operate. In Brazil, this contract is called a *social contract* and must contain the *statement of purpose* of the company – this statement of purpose describes the *business activities* of the company and must be categorized into a legal business activity by Brazilian government officials. For that, all legal business activities are cataloged using a table called National Classification of Economic Activities – *Classificação Nacional de Atividade Econômicas*, (CNAE) [8].

To perform the categorization, the government officials (at the Federal, State and County levels) must find *the semantic correspondence* between the company economic activities description and one or more entries of the CNAE table. There is a numerical code for each entry of the CNAE table and, in the categorization task, the government official attributes one or more of such codes to the company at hand. This can happen on the foundation of the company or in a change of its social contract, if that modifies its economic activities.

The work of finding the semantic correspondence between the company economic activities description and a set of entries into the CNAE table are both very difficult and labor-intensive task. This is because of the subjectivity of each local government officials who can focus on their own particular interests so that some codes may be assigned to a company, whereas in other regions, similar companies, may have a totally different set of codes. Sometimes, even inside of the same state, different level of government officials may count on a different number of codes for the same company for performing their work of assessing that company. Having inhomogeneous ways of classifying any company everywhere in all the three levels of the governmental administrations can cause a serious distortion on the key information for the long time planning and taxation. Additionally, the continental size of Brazil makes this problem of classification even worse.

To add, the number of codes assigned by the human specialist to a company can vary greatly, in our dataset we have seen cases where the number of codes varied from 1 up to 109. However, in the set of assigned codes, the first code is the main code of that company. The remaining codes have no order of importance.

For all these reasons, the computational problem addressed by us is mainly that of automatically *suggesting* the human classifier the semantic correspondence between a textual description of the economic activities of a company and one or more items of the CNAE table. Or, depending on the level of certainty the algorithms have on the automatic classification, we may consider bypassing thus the human classifier.

### A. Evaluating the Results

Typically, text categorization is mainly evaluated by the *Recall* and *Precision* metrics [9] in the single-labled cases. Nonetheless, other authors have already proposed different metrics for multi-label categorization problems [5, 2].

Formalizing the problem, we have at hand, text categorization may be defined as the task of assigning documents to a predefined set of categories, or classes [1]. In multi-label text categorization a document may be assigned to one or more categories. Let $\mathcal{D}$ be the domain of documents, $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ a set of pre-defined categories, and $\Omega = \{d_1, d_2, \ldots, d_{|\Omega|}\}$ an initial corpus of documents previously categorized by some human specialists into subsets of categories of $\mathcal{C}$.

In multi-label learning, the training (-and-validation) set $TV = \{d_1, d_2, \ldots, d_{|TV|}\}$ is composed of a number documents, each associated with a subset of categories in $\mathcal{C}$. $TV$ is used to train and validate (actually, to tune eventual parameters of) a categorization system that associates the appropriate combination of categories to the characteristics of each document in the $TV$. The test set $Te = \{d_{|TV|+1}, d_{|TV|+2}, \ldots, d_{|\Omega|}\}$, on the other hand, consists of documents for which the categories are unknown to the automatic categorization systems. After being trained, as well as tuned, by the $TV$, the categorization systems are used to predict the set of categories of each document in $Te$.

A multi-label categorization system typically implements a real-valued function of the form $f : \mathcal{D} \times \mathcal{C} \rightarrow \mathbb{R}$ that returns a value for each pair $\langle d_j, c_j \rangle \in \mathcal{D} \times \mathcal{C}$ that, roughly speaking, represents the evidence for the fact that the test document $d_j$ should be categorized under the category $c_j \in C_j$, where $C_j \subset \mathcal{C}$. The real-valued function $f(.,.)$ can be transformed into a ranking function $r(.,.)$, which is an one-to-one mapping onto $\{1, 2, \ldots, |\mathcal{C}|\}$ such that, if $f(d_j, c_1) > f(d_j, c_2)$, then $r(d_j, c_1) < r(d_j, c_2)$. If $C_j$ is the set of proper categories for the test document $d_j$, then a successful categorization system tends to rank categories in $C_j$ higher than those not in $C_j$. Additionally, we also use a threshold parameter so that those categories that are ranked above the threshold $\tau$ (*i.e.*, $c_k | f(d_j, c_k) \geq \tau$) are the only ones to be assigned to the test document.

We have used five multi-label metrics discussed in [5, 2] to evaluate the performance of the classifiers: *hamming loss*, *one-error*, *coverage*, *ranking loss*, and *average precision*. We now present each of these metrics:

*Hamming Loss (hloss)* evaluates how many times the test document $d_j$ is misclassified, *i.e.*, a category not belonging to the document is predicted or a category belonging to the document is not predicted.

$$\text{hloss}_j = \frac{1}{|C|} |P_j \Delta C_j|, \tag{1}$$

where $|C|$ is the number of categories and $\Delta$ is the symmetric difference between the set of predicted categories $P_j$ and

the set of appropriate categories $C_j$ of the test document $d_j$. The predicted categories are those which rank higher than the threshold $\tau$.

*One-error (one-error$_j$)* evaluates if the top ranked category is present in the set of appropriate categories $C_j$ of the test document $d_j$.

$$\text{one-error}_j = \begin{cases} 0 & \text{if } [\arg\max_{c\in\mathcal{C}} f(d_j, c)] \in C_j \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

where $[\arg\max_{c\in\mathcal{C}} f(d_j, c)]$ returns the top ranked category for the test document $d_j$.

*Coverage (coverage$_j$)* measures how far we need to go down the rank of categories in order to cover all the possible categories assigned to a test document.

$$\text{coverage}_j = \max_{c\in C_j} r(d_j, c) - 1, \quad (3)$$

where $\max_{c\in C_j} r(d_j, c)$ returns the maximum rank for the set of appropriate categories of the test document $d_j$.

*Ranking Loss (rloss$_j$)* evaluates the fraction of category pairs $\langle c_k, c_l\rangle$, for which $c_k \in C_j$ and $c_l \in \bar{C}_j$, that are reversely ordered for the test document $d_j$:

$$\text{rloss}_j = \frac{|\{(c_k, c_l)|f(d_j, c_k) \leq f(d_j, c_l)\}|}{|C_j||\bar{C}_j|}, \quad (4)$$

where $(c_k, c_l) \in C_j \times \bar{C}_j$, and $\bar{C}_j$ is the complementary set of $C_j$ in $\mathcal{C}$.

*Average Precision (avgprec$_j$)* evaluates the average of precisions computed after truncating the ranking of categories after each category $c_i \in C_j$ in turn:

$$\text{avgprec}_j = \frac{1}{|C_j|}\sum_{k=1}^{|C_j|} precision_j(R_{jk}), \quad (5)$$

where $R_{jk}$ is the set of ranked categories that goes from the top ranked category until a ranking position $k$ where there is a category $c_i \in C_j$ for $d_j$, and $precision_j(R_{jk})$ is the number of pertinent categories in $R_{jk}$ divided by $|R_{jk}|$.

For $p$ test documents, the overall performance is obtained by averaging each metric, that is, $\text{hloss} = \frac{1}{p}\sum_{j=1}^{p}\text{hloss}_j$, $\text{one-error} = \frac{1}{p}\sum_{j=1}^{p}\text{one-error}_j$, $\text{coverage} = \frac{1}{p}\sum_{j=1}^{p}\text{coverage}_j$, $\text{rloss} = \frac{1}{p}\sum_{j=1}^{p}\text{rloss}_j$, and $\text{avgprec} = \frac{1}{p}\sum_{j=1}^{p}\text{avgprec}_j$. The smaller the value of *hamming loss*, *one-error*, *coverage* and *ranking loss*, and the larger the value of *average precision*, the better the performance of the categorization system. The performance is optimal when hloss = one-error = rloss = 0 and avgprec = 1.

## III. Probabilistic Neural Network (PNN)

The Probabilistic Neural Network (PNN) was first proposed by Donald Specht in 1990 [6]. This is an artificial neural network for nonlinear computing which approaches the Bayes optimal decision boundaries. This is done by estimating the *probability density function* of the training dataset using the Parzen nonparametric estimator [10].

According to the work presented in [11, 12, 13, 14, 15, 16], this type of neural network can yield similar results, sometimes superior, in pattern recognition problems when compared to the other techniques, such as: Backpropagation Neural Network, SVM and kNN.

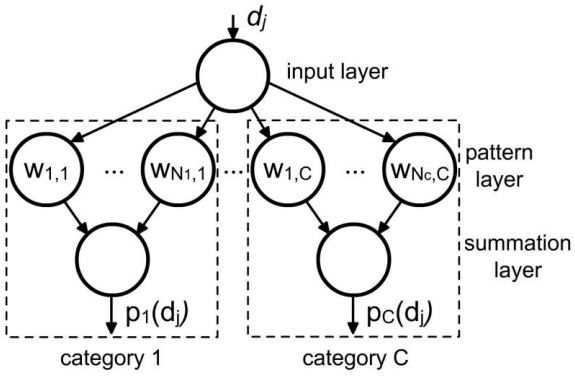


**Figure. 1**: The modified PNN architecture.

The original PNN algorithm was designed for single-label problems. Thus, we slightly modified its standard architecture, so that it is now capable of solving multi-label problem addressed in this work.

In our modified version, instead of four, the PNN is composed of only three layers: the *input* layer, the *pattern* layer and the *summation* layer, as depicted in Figure 1. Thus, like in the original structure, this version of PNN needs only one training step, thus its train is very fast comparing to the others feed-forward neural networks [17].

The train consists in assigning each training sample $w$ of category $i$ to a neuron of pattern layer of category $i$. Thus, the weight vector of this neuron is the characteristics vector of the sample. In addition, the number of pattern layer's neurons of each category is equal to its number of samples.

For each $d_j$ test instance passed by the input layer to a neuron in the pattern layer, it computes the output for the $d_j$. The computation is as showed in Equation 6.

$$F_{k,i}(d_j) = \frac{1}{2\pi\sigma^2} exp\left(\frac{d_j^t w_{k,i} - 1}{\sigma^2}\right), \quad (6)$$

where the $d_j$ is the pattern characteristics input vector, and the $w_{k,i}$ is the $k^{th}$ sample for a neuron of category $i$, $k \in N_i$, whereas $N_i$ is the number of neurons of category $i$. In addition, $d_j$ and $w_{k,i}$ were normalized so that $d_j^t d_j = 1$ and $w_{k,i}^t w_{k,i} = 1$. $\sigma$ is the Gaussian standard deviation, which determines the receptive field of the Gaussian curve.

The next step is the summation layer. In this layer, all weight vectors are summed, Equation 7, in each cluster $i$ producing $p_i(d_j)$ values, where $C$ is the total number of categories.

$$p_i(d_j) = \sum_{k=1}^{N_i} F_{k,i}(d_j), \quad i = 1, 2, \ldots, C \quad (7)$$

Finally, for the selection of the categories which will be assigned by neural network to each sample, we consider the most likely categories pointed out by the summation layer based on a chosen threshold $\tau$. For example, whether $p_i(d_j) > \tau$, so the category $i$ is assigned to the sample $d_j$.

Differently from other types of networks, such as those feed forward based [18], the PNN proposed needs few parameters to be configured: the $\sigma$, (see in Equation 6) and the determination of threshold value. The $\sigma$ is used to narrow the receptive field of the Gaussian curve in order to strictly select only

the more likelihood inputs for a given category. Other advantages of the probabilistic neural networks is that it is easy to add new categories, or new training inputs, into the already running structure, which is good for the on-line applications [17]. Moreover, it is reported in the literature that it is also easy to implement this type of neural network in parallel. On the other hand, one of its drawbacks is the great number of neurons in the pattern layer, which can be, nevertheless, mitigated by an optimization on the number of the neuron [19].

## IV. Experimental Evaluation

We evaluate the categorization performance of PNN on two real-world multi-label learning problems: (i) categorization of Web pages and (ii) categorization of free-text descriptions of economic activities. We also compare PNN categorization performance with that of: the multi-label lazy learning approach ML-kNN [2], the multi-label kernel method Rank-SVM [3], the multi-label decision tree ADTBoost.MH [4], and the boosting-style algorithm BoosTexter [5]. We believe that these categorizers are representative of some of the most effective multi-label text categorization methods currently available.
In the following subsections, we briefly describe the comparing algorithms and analyze our experimental results.

### A. Comparing Algorithms

The ML-kNN [2] categorizer is a version of kNN [20] especially designed for multi-label categorization. In this categorizer, the $k$ nearest neighbors of a test document $d_j$ are identified in $TV$. The Euclidean distance is used to measure distances between documents. Then, the maximum a posteriori (MAP) principle is used to determine the category set for $d_j$, using statistical information obtained from the category sets of the neighbors of $d_j$, i.e., the number of neighboring documents belonging to each possible category.
The BoosTexter [5] categorizer uses the boosting machine-learning technique to the problem of multi-label text categorization. The purpose of boosting is to find a highly accurate categorization rule by combining many simple and moderately inaccurate categorization rules (called weak hypotheses). The boosting algorithm finds a set of weak hypotheses by calling a weak learner repeatedly in a series of rounds. In the BoosTexter system, two different boosting algorithms are tested, using a one-level decision tree weak learner: AdaBoost.MH [21], specifically designed to minimize Hamming loss, and AdaBoost.MR, aimed at minimizing ranking loss.
The ADTBoost.MH [4] categorizer is a multi-label alternating decision tree (ADT) learning algorithm based on both AdaBoost.MH [21], which combines a one-level decision tree with boosting, and ADTBoost [22], which uses boosting as a method for learning alternating decision trees (ADTrees). ADTBoost.MH is an extension of AdaBoost.MH, that permits a better readability of the categorization rule ultimately produced using tree representations of large set of rules, as well as an extension to ADTBoost, in order to handle multi-label categorization problems.
The Rank-SVM [3] categorizer is a Support Vector Machine (SVM) [23] like learning system to handle multi-label problems. The multi-label model is built from two different subsystems. The first one ranks the labels by defining a linear model that maximizes the margin and at the same time minimizes the ranking loss. The second one predicts a threshold and all categories ranking above the threshold are considered to belong to the answer category set.

### B. Yahoo's databases

In the first set of experiments, we have used 11 multi-labeled databases of text from *yahoo.com* domain [1]. Initially, each database passed by a process of simple feature selection based on the number of documents that contains a specific term to reduce the dimensionality of each one. Actually, only 2% terms with highest document frequency were selected and the others were removed. Then, each document was represented by a vector, where each dimension represents the number of times a word appeared in the document [9]. Such pre-processing was performed in [2]. In addition, each database has 2000 samples to training and 3000 to test, and the average number of classes is 30. More information about these databases are available in Table 1.
To evaluate the performance of the algorithms the metrics presented in Section II were used, and the results were obtained directly from [2], with exception of the PNN's results. In [2] it was not used any process of exhausting search for optimization of the classifiers. The parameters of ML-kNN were obtained from a experiment on a image database, the ones of BoosTexter and ADTBoost.MH were selected values that, according with the authors in [2], could not alter significantly their performance on Yahoo's databases. For Rank-SVM was used the parameters that achieved the best result in [24]. To turn in a fair comparison with the other techniques, we tested for PNN only the order of magnitude of the variance's value. For this, we used the part of training of Arts & Humanities' database from Yahoo and we used the variance's values 10, 1 and 0.1 on a cross-validation experiment. The value chosen was 0.1, because it returned the best result. The threshold employed for Hamming Loss metric was 0.5, same value used for ML-kNN, therefore, this parameter was not optimized for PNN.
The results obtained for Yahoo's databases are presented in Tables from 2 to 6. Each one of the tables represents a metric, where each row is a data set and each column is a classifier. The term "Average" in the last row means the average value of the metric achieved by each classifier to every databases. It is important to observe that the results of the ADTBoost.MH classifier to Ranking Loss were not reported because, according with [2], the algorithm of this classifier did not supply such information.
To accomplish an clearer evaluation of the classifiers, we have adopted two criterions derived from [2]. The first criterion creates one partial order "$\succ$" that evaluates the performance between two classifiers for each metric. In that way, if the classifier $A1$ has a better performance than $A2$ to a given metric, so we have $A1 \succ A2$. In order to perform this task, we used two-tailed paired t-test at 5% significance level.
However, the presented criterion is insufficient to obtain the performance of classifiers as a whole, therefore, we used a

[1] Databases and codes of the proposed PNN are available at http://www.inf.ufes.br/∼elias/ijcisim2009.rar

| *Data Set* | *Categories* | *Terms* | *Set* | *DC* | *CD* | *MNC* | *RC* |
|---|---|---|---|---|---|---|---|
| *Arts&Humanities* | *26* | *462* | *Train* | *44,50%* | *1,63* | *11* | *19,23%* |
| | | | *Test* | *43,63%* | *1,64* | *14* | *19,23%* |
| *Business&Economy* | *30* | *438* | *Train* | *42,20%* | *1,59* | *10* | *50,00%* |
| | | | *Test* | *41,93%* | *1,59* | *12* | *43,33%* |
| *Computers&Internet* | *33* | *681* | *Train* | *29,60%* | *1,49* | *17* | *39,39%* |
| | | | *Test* | *31,27%* | *1,52* | *17* | *36,36%* |
| *Education* | *33* | *550* | *Train* | *33,50%* | *1,47* | *7* | *57,58%* |
| | | | *Test* | *33,73%* | *1,46* | *6* | *57,58%* |
| *Entertainment* | *21* | *640* | *Train* | *29,30%* | *1,43* | *9* | *28,57%* |
| | | | *Test* | *28,20%* | *1,42* | *17* | *33,33%* |
| *Health* | *32* | *612* | *Train* | *48,05%* | *1,67* | *7* | *53,13%* |
| | | | *Test* | *47,20%* | *1,66* | *13* | *53,13%* |
| *Recreation&Sports* | *22* | *606* | *Train* | *30,20%* | *1,41* | *13* | *18,18%* |
| | | | *Test* | *31,20%* | *1,43* | *17* | *18,18%* |
| *Reference* | *33* | *793* | *Train* | *13,75%* | *1,16* | *5* | *51,52%* |
| | | | *Test* | *14,60%* | *1,18* | *12* | *54,55%* |
| *Science* | *40* | *743* | *Train* | *34,85%* | *1,49* | *7* | *35,00%* |
| | | | *Test* | *30,57%* | *1,43* | *9* | *40,00%* |
| *Social&Science* | *39* | *1047* | *Train* | *20,95%* | *1,27* | *9* | *56,41%* |
| | | | *Test* | *22,83%* | *1,29* | *10* | *58,97%* |
| *Society&Culture* | *27* | *636* | *Train* | *41,90%* | *1,71* | *13* | *25,93%* |
| | | | *Test* | *39,97%* | *1,68* | *16* | *22,22%* |
| *(Average)* | *30,55* | *655,27* | *Treino* | *33,53%* | *1,48* | *9,82* | *39,54%* |
| | | | *Test* | *33,19%* | *1,48* | *13* | *39,72%* |

| *LEGEND:* | *DC: percentage of documents belonging to more than one category* |
|---|---|
| | *CD: average number of categories of each document* |
| | *MNC: maximum number of categories assigned to an instance* |
| | *RC: percentage of rare categories (categories with less than 1% instances in the data set belong to it)* |

*Table 1*: Information about data sets from *yahoo.com* used in the experiments.

| Data Set | ML-kNN | BoosTexter | ADTBoost.MH | Rank-SVM | PNN |
|---|---|---|---|---|---|
| Arts&Humanities | 0.0612 | 0.0652 | **0.0585** | 0.0615 | 0.0630 |
| Business&Economy | **0.0269** | 0.0293 | 0.0279 | 0.0275 | 0.0307 |
| Computers&Internet | 0.0412 | 0.0408 | 0.0396 | **0.0392** | 0.0447 |
| Education | **0.0387** | 0.0457 | 0.0423 | 0.0398 | 0.0437 |
| Entertainment | 0.0604 | 0.0626 | **0.0578** | 0.0630 | 0.0640 |
| Health | 0.0458 | **0.0397** | **0.0397** | 0.0423 | 0.0514 |
| Recreation&Sports | 0.0620 | 0.0657 | **0.0584** | 0.0605 | 0.0634 |
| Reference | 0.0314 | 0.0304 | **0.0293** | 0.0300 | 0.0307 |
| Science | **0.0325** | 0.0379 | 0.0344 | 0.0340 | 0.0353 |
| Social&Science | **0.0218** | 0.0243 | 0.0234 | 0.0242 | 0.0281 |
| Society&Culture | **0.0537** | 0.0628 | 0.0575 | 0.0555 | 0.0596 |
| Average | 0.0432 | 0.0459 | **0.0426** | 0.0434 | 0.0468 |

*Table 2*: Hamming Loss obtained by classifiers on the Yahoo's databases.

| Data Set | ML-kNN | BoosTexter | ADTBoost.MH | Rank-SVM | PNN |
|---|---|---|---|---|---|
| Arts&Humanities | 0.6330 | **0.5550** | 0.5617 | 0.6653 | 0.5597 |
| Business&Economy | **0.1213** | 0.1307 | 0.1337 | 0.1237 | 0.1317 |
| Computers&Internet | 0.4357 | 0.4287 | 0.4613 | **0.4037** | 0.4457 |
| Education | 0.5207 | 0.5587 | 0.5753 | **0.4937** | 0.5463 |
| Entertainment | 0.5300 | **0.4750** | 0.4940 | 0.4933 | 0.5530 |
| Health | 0.4190 | **0.3210** | 0.3470 | 0.3323 | 0.4080 |
| Recreation&Sports | 0.7057 | 0.5557 | **0.5547** | 0.5627 | 0.6037 |
| Reference | 0.4730 | 0.4427 | 0.4840 | **0.4323** | 0.4780 |
| Science | 0.5810 | 0.6100 | 0.6170 | **0.5523** | 0.6123 |
| Social&Science | **0.3270** | 0.3437 | 0.3600 | 0.3550 | 0.3753 |
| Society&Culture | 0.4357 | 0.4877 | 0.4845 | **0.4270** | 0.4647 |
| Average | 0.4711 | 0.4463 | 0.4612 | **0.4401** | 0.4708 |

*Table 3*: One-Error obtained by classifiers on the Yahoo's databases.

| Data Set | ML-kNN | BoosTexter | ADTBoost.MH | Rank-SVM | PNN |
|---|---|---|---|---|---|
| Arts&Humanities | 5.4313 | 5.2973 | 5.1900 | 9.2723 | **4.8503** |
| Business&Economy | 2.1840 | 2.4123 | 2.4730 | 3.3637 | **2.1087** |
| Computers&Internet | 4.4117 | 4.4887 | 4.4747 | 8.7910 | **4.0380** |
| Education | **3.4973** | 4.0673 | 3.9663 | 8.9560 | 3.4980 |
| Entertainment | 3.1467 | 3.0883 | 3.0877 | 6.5210 | **3.0663** |
| Health | 3.3043 | 3.0780 | 3.0843 | 5.5400 | **3.0093** |
| Recreation&Sports | 5.1010 | 4.4737 | 4.3380 | 5.6680 | **4.2773** |
| Reference | 3.5420 | 3.2100 | 3.2643 | 6.9683 | **2.9097** |
| Science | 6.0470 | 6.6907 | 6.6027 | 12.401 | **5.9930** |
| Social&Science | **3.0340** | 3.6870 | 3.4820 | 8.2177 | 3.1357 |
| Society&Culture | 5.3653 | 5.8463 | **4.9545** | 6.8837 | 5.3350 |
| Average | 4.0968 | 4.2127 | 4.0834 | 7.5075 | **3.8383** |

*Table 4*: Coverage obtained by classifiers on the Yahoo's databases.

| Data Set | ML-kNN | BoosTexter | ADTBoost.MH | Rank-SVM | PNN |
|---|---|---|---|---|---|
| Arts&Humanities | 0.1514 | 0.1458 | N/A | 0.2826 | **0.1306** |
| Business&Economy | 0.0373 | 0.0416 | N/A | 0.0662 | **0.0367** |
| Computers&Internet | 0.0921 | 0.0950 | N/A | 0.2091 | **0.0826** |
| Education | **0.0800** | 0.0938 | N/A | 0.2080 | 0.0803 |
| Entertainment | 0.1151 | 0.1132 | N/A | 0.2617 | **0.1103** |
| Health | 0.0605 | **0.0521** | N/A | 0.1096 | 0.0526 |
| Recreation&Sports | 0.1913 | 0.1599 | N/A | 0.2094 | **0.1556** |
| Reference | 0.0919 | 0.0811 | N/A | 0.1818 | **0.0732** |
| Science | 0.1167 | 0.1312 | N/A | 0.2570 | **0.1166** |
| Social&Science | **0.0561** | 0.0684 | N/A | 0.1661 | 0.0601 |
| Society&Culture | 0.1338 | 0.1483 | N/A | 0.1716 | **0.1315** |
| Average | 0.1024 | 0.1028 | N/A | 0.1930 | **0.0936** |

*Table 5*: Ranking Loss obtained by classifiers on the Yahoo's databases.

| Data Set | ML-kNN | BoosTexter | ADTBoost.MH | Rank-SVM | PNN |
|---|---|---|---|---|---|
| Arts&Humanities | 0.5097 | 0.5448 | 0.5526 | 0.4170 | **0.5645** |
| Business&Economy | **0.8798** | 0.8697 | 0.8702 | 0.8694 | 0.8763 |
| Computers&Internet | 0.6338 | **0.6449** | 0.6235 | 0.6123 | 0.6398 |
| Education | **0.5993** | 0.5654 | 0.5619 | 0.5702 | 0.5889 |
| Entertainment | 0.6013 | **0.6368** | 0.6221 | 0.5637 | 0.5991 |
| Health | 0.6817 | **0.7408** | 0.7257 | 0.6839 | 0.7047 |
| Recreation&Sports | 0.4552 | 0.5572 | **0.5639** | 0.5315 | 0.5396 |
| Reference | 0.6194 | **0.6578** | 0.6264 | 0.6176 | 0.6441 |
| Science | **0.5324** | 0.5006 | 0.4940 | 0.5007 | 0.5073 |
| Social&Science | **0.7481** | 0.7262 | 0.7217 | 0.6788 | 0.7113 |
| Society&Culture | **0.6128** | 0.5717 | 0.5881 | 0.5717 | 0.5993 |
| Average | 0.6249 | **0.6378** | 0.6318 | 0.6015 | 0.6341 |

*Table 6*: Average Precision obtained by classifiers on the Yahoo's databases.

second criterion. In this one is applied a system based on rewards and punishes. For example, for the case of $A1 \succ A2$, $A1$ is rewarded with $+1$ and $A2$ is punished with $-1$. Then, we compare the classifiers two a two through of the sum of their rewarded and punished between them. This step is different from the one that had been done in [2], and the motive for this modification is the inconsistency in such method [2]. In this case, if $A1$ have positive value in relation to $A2$, so $A1$ is superior to $A2$, *i.e.*, $A1 > A2$. Thus, the results obtained by these two criterions are shown in Table 7. The accumulated score of each algorithm is also shown in the parentheses.

Analyzing the result of the second criterion shown in Table 7, we observe that ML-kNN, ADTBoost.MH and BoosTexter are highly competitive among themselves. On the other hand, Rank-SVM was the most inferior in the experimental results. Finally, the PNN presented a slightly superior performance in relation to ML-kNN, BoosTexter and Rank-SVM, and drawn with ADTBoost.MH.

### C. Economic activities database

Although the PNN and ADTBoost.MH have presented similar performance on Yahoo's databases, the ADTBoost.MH has a very slow training phase, and it will be inadequate to apply it on economic activities database, since we inted to use an algorithm for optmization of parameters, as Genetic Algorithm. Then, we chose to use PNN and ML-kNN, because ML-kNN has the training phase faster, after of PNN. Thus, we employed a serie of experiments to compare PNN with ML-kNN.

We used a dataset containing 3264 documents of free text business descriptions of Brazilian companies categorized into a subset of 764 CNAE categories. This dataset was obtained from real companies placed in Vitoria County in Brazil. The CNAE codes of each company in this dataset were assigned by Brazilian government officials trained in this task. Then we evenly partitioned the whole dataset into four subsets of equal size of 816 documents. We joined to this categorizing dataset the brief description of each one of the 764 CNAE categories, totalizing $4028$ documents. Hence, in all training (-and-validation) set, we adopted the 764 descriptions of CNAE categories and a subset of 816 business description documents, and, as the test set, the other three subsets of business descriptions totalizing $2448$ documents. As a result, we carried out a sequence of four experiments with each of these algorithms. Results are reported as average categorization accuracy across the experiments.

We preprocessed the dataset via term selection –a total of 1001 terms were found in the database after removing stop words and trivial cases of gender and plural; only words appearing in the CNAE table were considered. After that, each document in the dataset was described as a multidimensional vector using the *Bag-of-Words* representation, *i.e.*, each dimension of the vector corresponds to the number of times a term of the vocabulary appears in the corresponding document. Table 8 summarizes the characteristics of this dataset[3]. In Table 8, #C denotes the number of categories, #t denotes the number of terms in the vocabulary, NTD denotes the average number of terms per document, DC denotes the percentage of documents belonging to more than one category, CD denotes the average number of categories of each document, and RC denotes the percentage of rare categories, *i.e.*, those categories associated with less than $1\%$ of the documents of the dataset.

In both PNN and ML-kNN algorithms, their parameters were optimized for each category of the dataset. In the probabilistic neural network case, one value of $\sigma$ for each category and one value of threshold were selected by a Genetic Algorithm. For the ML-kNN, we also optimized the number of nearest neighbors, value of threshold and one $\delta$ for each category,

[2] further details is mentioned in Appendix A, where is illustrated a little example for explanation.

[3] dataset available at http://www.inf.ufes.br/∼elias/vitoria.tar.gz.

| HL-Hamming Loss; OE-One-error; C-Coverage; RL-Ranking Loss; AP-Average Precision<br>A1-ML-kNN; A2-BoosTexter; A3-ADTBoost.MH; A4-Rank-SVM; A5-PNN | |
|---|---|
| Criterion 1 | |
| HL | $A1 \succ A5$, $A3 \succ A2$, $A4 \succ A2$, $A3 \succ A5$, $A4 \succ A5$ |
| OE | $A2 \succ A3$, $A2 \succ A5$ |
| C | $A1 \succ A4$, $A5 \succ A1$, $A2 \succ A4$, $A5 \succ A2$, $A3 \succ A4$, $A5 \succ A3$, $A5 \succ A4$ |
| RL | $A1 \succ A4$, $A5 \succ A1$, $A2 \succ A4$, $A5 \succ A2$, $A5 \succ A4$ |
| AP | $A2 \succ A4$, $A3 \succ A4$, $A5 \succ A4$ |
| Criterion 2 | |
| PNN(1) > **ML-kNN** > Rank-SVM(-2)<br>PNN(1) > **BoosTexter** > Rank-SVM(-2)<br>**ADTBoost.MH** > Rank-SVM(-2)<br>{ML-kNN(2), BoosTexter(2), ADTBoost.MH(2), PNN(2)} > **Rank-SVM**<br>**PNN** > {ML-kNN(-1), BoosTexter(-1), Rank-SVM(-2)} | |

*Table 7*: Relative performance of the classifiers by two criterions on the Yahoo's databases.

| | #C | #t | Training set | | | | Test/validation set | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | NTD | DC | CD | RC | NTD | DC | CD | RC |
| CNAE | 764 | 1001 | 4.65 | 0.00% | 1.00 | 100.00% | 10.92 | 74.48% | 4.27 | 85.21% |

*Table 8*: Characteristics of the CNAE dataset.

which is a parameter that alters slightly the categories' priori probabilities obtained from the database.
To tune these parameters we divided the training set (-and-validation) set into a training set, which was used to inductively build the categorizer, and a validation set, which was used to evaluate the performance of the categorizer in the series of experiments aimed at parameter optimization. The training set is composed of 764 descriptions of CNAE categories and the validation set of 816 business description documents described previously.

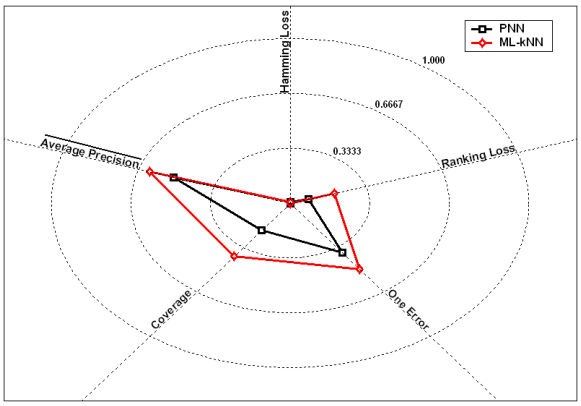


**Figure. 2**: Experimental results of each multi-label categorizer on the economic activities dataset.

After tuning, the multi-label categorizers were trained with the 764 descriptions of CNAE categories and tested with the 2448 documents of the test set. The process of validation and test was performed three times. Figure 2 presents the average experimental results of the PNN and ML-kNN algorithms on the economic activities dataset in terms of *hamming loss*, *ranking loss*, *one-error*, *coverage* and *average precision*.
In Figure 2, each metric is represented by a ray emanating from the center of the circle. Its values varies from 0.0, in the center, to 1.0, on the border of the circle. The result yielded by an algorithm, with respect to a given metric, is then plotted over the appropriated rays. The smaller the value of the *hamming loss*, *ranking loss one-error*, and *coverage* metrics, the better. On the other hand, the larger the value of the *average precision*, the better. A normalization on the *coverage* results was devised so that its value could fit between 0 and 1. Therefore, we draw the actual value divided by $|\mathcal{C}| - 1$, where $|\mathcal{C}|$ is the number of classes, that is, $|\mathcal{C}| = 764$ in our case. In order to draw the results of the *average precision* in the same way we have done for the other metrics, we are plotting, in Figure 2, the average precision $= 1 - (average\ precision)$.
As shown by the innermost lines in Figure 2, PNN outperforms ML-kNN in terms of *ranking loss* (0.1168 smaller), *one-error* (0.1216 smaller), *coverage* (0.1933 smaller), *average precision* (0.1067 higher), and presents a drawn result in terms of *hamming loss* (0.0055 for both classifiers).
Table 9 shows the results of our performance comparison between PNN and ML-kNN.

| | PNN | ML-kNN |
|---|---|---|
| hamming loss | 0.0055 | 0.0055 |
| ranking loss | **0.0798** | 0.1966 |
| one-error | **0.3736** | 0.4952 |
| coverage | **0.2050** | 0.3983 |
| average precision | **0.5120** | 0.6187 |

*Table 9*: Performance comparison between PNN and ML-kNN.

For statistical analysis of the results, we carried out the two-tailed paired t-test at 5% significance level and the PNN surpass ML-kNN in all the metrics, with exception one: the hamming loss, in which have no statistical difference.

## V. Related Work

To the best knowledge of the authors, the closest work we have found so far on text classification using the Probabilistic Neural Networks is that by [25]. In that work, the neural network was applied to identify and to classify web sites of the e-commerce. If the web site were recognized as commerce, it would be classified into one out of the eleven defined categories considered within their experiment. Otherwise, the web site is classified as not being of the e-commerce class. Therefore, there were indeed twelve different categories in that classification task: 11 for valid e-commerce classes and one for an invalid e-commerce. In their experiments the network was designed with 5958 neurons in the pattern layer and 12 neurons in the summation layer, where each train-

ing sample was a vector with 432 dimensions. In spite of the great amount of neurons used in the network, the authors mentioned that they did not have problems with memory's limitations and application speed. The results of the experiments obtained $80\%$ of accuracy in the classification into twelve existent categories, and $92\%$ in the task of the identification of a valid web site of e-commerce. The authors considered satisfactory these results and good enough for their purpose in the proposed problem.

The work in [26] is the first one on dealing with the problem of automatically classify the economic activities based on free text. In their work, they compared the results achieved between a Nearly Neighbors algorithm approach and a Weightless Neural Network, called VG-RAM WNN, using the equivalent to $1-$ *one-error* metric to evaluate the performance of their experiments, defined in Section II. In the first algorithm the performance was of $63.36\%$, while VG-RAM WNN showed to be slightly better, with a performance of $67.56\%$. However, the use of a single metric seemed not to be the most appropriated one for the evaluation of multi-labled problems. Then, a different approach was performed by [27]. In this new work a neural netowrk was used with 83 arrays of small standard PNN for the classification of a set of 3696 business activities descriptions in free text format of Brazilian companies into a subset of 415 economic activities classes, recognized by Brazilian law. Unlike in the previous work, the authors used the Recall and Precision as metrics for the evaluation of their experiments. Although it was achieved a reasonable value for the Recall, the value for the Precision was very low, since almost every neural networks returned at least one class to each instance of the test.

A PNN with a slightly modified architecture to treat problems of multi-label classification was proposed in [28], which is the same PNN presented in this work. Such neural network presents some advantage over the array of small standard PNN approach, used in [27], because of only one PNN is used to solve the whole problem of multi-label classification, whereas in the previous approach we needed to build many neural networks (83 in that case), which complicates the process of optimization.

A recent work on the subject discussed in this paper was published in [29]. A comparison between VG-RAM WNN and ML-kNN for the a set of multi-label web pages database, including the economic activities database, was carried out by the authors in [29]. The results were, when compared to the work in [26], greatly improved and more appropriated metrics were introduced in that work. The results reported in that work were such that in the problem of categorization of free-text descriptions of economic activities, VG-RAM WNN outperformed ML-KNN in terms of the four multi-label evaluation metrics adopted in that paper, while, in the categorization of web pages, on average, VG-RAM WNN outperformed ML-KNN in terms of three metrics and showed similar categorization performance in terms of the one metric.

In the more general studies of multi-label problems, the work in [2] compared three algorithms: BoosTexter, ADTBoost.MH and Rank-SVM, against a proposed ML-kNN algorithm. All algorithms were designed to treat the problem of multi-labled classification. Three different datasets were used to carry out the experiments: one in bioinformatic data [3, 30], one in natural scene classification data [31], and one in automatic web page categorization data [32]. In [2], ML-kNN algorithm presented better performance than the others algorithms for both bioinformatic and natural scene datasets and achieved the same performance as the BoosTexter algorithm for the web page datasets.

Another very close general multi-label problem to one we are presenting in this paper, concerning with the economic activities classification, is the patent categorization [33]. Both are based on free text descriptions of a variety topics. Also similar is that manually classifying a large volume of patents documents, managed by patent offices, is a labor-intensive and time-consuming task. A patent document may cite another patent document, or articles, for comparing or contrasting reasons. Therefore, besides using the content categorization approach, the authors in [33] proposed to extract and use the direct hyperlink citation relationships among patent documents in order to improve the quality of the whole process of classification. Hyperlink citation is a similar strategy some researchers have been widely applied to web page classification studies. The experiments were conducted on a nanotechnology-related patent dataset from the USPTO. The training dataset contained 13,913 instances, and the testing data set 4,358 data instances. The average of category for document was 36, and the total of categories were up to 426. The results by the $K_{Gra}$ kernel proposed approach yielded $86.67\%$ accuracy overcoming the $81\%$ of manually processing and the results of previous work [34].

## VI. Conclusions and Future Work

The problem of classifying huge number of economic activities description in free text format every day is a huge challenge for the Brazilian governmental administration. This problem is crucial for the long term planning in all three levels of the administration in Brazil.

In this study, we presented an experimental evaluation of the Probabilistic Neural Network performance on multi-label text classification. We performed a comparative study of PNN and other classifiers on a Yahoo and economic activities databases. To evaluate these algorithms a set of metrics usally applied for this type of problem were used. The achieved results showed that the proposed approach can be as well as or even better than other widespread techniques in literature. A direction for a future work includes a study to improve the PNN's performance, such as, to examinate the correlation on assigned codes, to use techniques to feature selection and selection of the best training samples. Furthermore, researches to turn the PNN working in online environment, keeping the reduced dimension, ever are being done.

## VII. Acknowledgments

We would like to thank Andréa Pimenta Mesquita, CNAE classifications coordinators at Vitoria City Hall, for providing us with the dataset we used in this work. We would also like to thank Min-Ling Zhang for all the help with the ML-kNN categorization tool and Web Page data sets. This work is partially supported by the Internal Revenue Brazilian Ser-

vice (*Receita Federal do Brasil*), the CNPq, the Brazilian government research agency, under the grants (308207/2004-1, 471898/2004-0, 620165/2006-5), *Financiadora de Estudos e Projetos*—FINEP-Brasil (grants CT-INFRA-PRO-UFES/2005, CT-INFRA-PRO-UFES/2006) and *Fundação Espírito Santense de Tecnologia*—FAPES-Brasil (grant 41936450/2008).

## References

[1] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[2] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A Lazy Learning Approach to Multi-Label Learning," *Pattern Recogn.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[3] A. Elisseeff and J. Weston, "A Kernel Method for Multi-Labelled Classification," in *NIPS*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA, USA: MIT Press, 2001, pp. 681–687.

[4] F. D. Comité, R. Gilleron, and M. Tommasi, "Learning multi-label alternating decision trees from texts and data," *Lecture Notes in Computer Science*, pp. 35 – 49, 2003.

[5] R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[6] D. F. Specht, "Probabilistic Neural Networks," *jnn*, vol. 3, no. 1, pp. 109–118, 1990.

[7] J. xiong Dong, A. Krzyzak, and C. Y. Suen, "Fast SVM Training Algorithm with Decomposition on Very Large Data Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 603–618, April 2005.

[8] CNAE, *Classificação Nacional de Atividades Econômicas Fiscal*, 1st ed. Rio de Janeiro, RJ: IBGE – Intituto Brasileiro de Geografia e Estatística, http://www.ibge.gov.br/concla.

[9] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, 1st ed. New York: Addison-Wesley, 1998.

[10] E. Parzen, "On the estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. vol. 3, pp. 1065 – 1076, 1962.

[11] C. C. Fung, V. Iyer, W. Brown, and K. W. Wong, "Comparing the Performance of Different Neural Networks Architectures for the Prediction of Mineral Prospectivity," pp. 394–398, August 2005.

[12] C.-J. Huang and W.-C. Liao, "A Comparative Study of Feature Selection Methods for Probabilistic Neural Networks in Cancer Classification," *IEEE International Conference on Tools with Artificial Intelligence*, pp. 451 – 458, November 2003.

[13] W. Jatmiko, T. Fukuda, K. Sekiyama, and B. Kusumoputro, "Optimized Probabilistic Neural Networks in Recognizing Fragrance Mixtures using Higher Number of Sensors," *IEEE Sensors*, p. 4, November 2005.

[14] I. Kalatzis, N. Piliouras, E. Ventouras, C. C. Papageorgiou, A. D. Rabavilas, and D. Cavouras, "Comparative Evaluation of Probabilistic Neural Network Versus Support Vector Machines Classifiers in Discriminating ERP Signals of Depressive Patients from Healthy Controls," *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, vol. vol. 2, pp. 981 – 985, September 2003.

[15] P. K. Patra, M. Nayak, S. K. Nayak, and N. K. Gobbak, "Probabilistic Neural Network for Pattern Classification," *IEEE Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. vol. 2, pp. 1200 – 1205, 2002.

[16] D. F. Specht, "Probabilistic Neural Networks for Classification, Mapping, or Associative Memory," *IEEE International Conference on Neural Networks*, vol. vol. 1, no. 24, pp. 525 – 532, July 1988.

[17] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.

[18] S. Haykin, *Neural Networks – A Comprehensive Foundation*, 2nd ed. New Jersey: Prentice Hall, 1998.

[19] K. Z. Mao, K. C. Tan, and W. Ser, "Probabilistic Neural-Network Structure Determination for Pattern Classification," *IEEE Transactions on Neural Networks*, vol. vol. 11, pp. 1009 – 1016, July 2000.

[20] C. D. Manning, P. Raghavan, and H. Schútze, *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2008.

[21] E. R. Schapire, Y. Singer, and A. Singhal, "Boosting and rocchio applied to text filtering," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, Melbourne, Australia.

[22] Y. Freund and L. Mason, "The Alternating Decision Tree Learning Algorithm," in *Proceedings of the 16th International Conference in Machine Learning (ICML'99)*, 1999, pp. 124–133.

[23] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, 1998, pp. 137–142.

[24] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," *Advances in Neural Information Processing Systems*, pp. 681 – 687, 2002.

[25] I. Anagnostopoulos, C. Anagnostopoulos, V. Loumos, and E. Kayafas, "Classifying Web Pages Employing a Probabilistic Neural Network," *IEEE Proceedings Software*, vol. vol. 151, no. 3, pp. 139 – 150, June 2004.

[26] A. F. D. Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, and L. Veronese, "Automated Free Text Classification of Economic Activities using VG-RAM Weightless Neural Networks," *7 International Conference on Intelligent Systems Design and Applications (ISDA)*, p. 6, 2007.

[27] E. de Oliveira, P. M. Ciarelli, and F. O. Lima, "The Automation of the Classification of Economic Activities from Free Text Descriptions Using an Array Architecture of Probabilistic Neural Network," *VIII Simpósio Brasileiro de Automação Industrial (SBAI)*, p. 5, October 2007.

[28] E. Oliveira, P. M. Ciarelli, C. Badue, and A. F. D. Souza, "A Comparison between a kNN Based Approach and a PNN Algorithm for a Multi-label Classification Problem," in *ISDA '08: Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 628–633.

[29] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese, and C. Badue, "Automated Multi-label Text Categorization with VG-RAM Weightless Neural Networks," *Neurocomputing*, vol. 72, no. 10–12, pp. 2209–2217, 2009.

[30] P. Pavlidis, J. Weston, J. Cai, and W. Grundy, "Combining Microarray Expression Data and Phylogenetic Profiles to Learn Functional Categories using Support Vector Machines," in *RECOMB*, Montréal, Canada, CA, 2001, pp. 242–248.

[31] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning Multi-Label Scene Classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[32] N. Ueda and K. Saito, "Parametric Mixture Models for Multi-Labeled Text," in *NIPS*, S. Thurn and K. Obermayer, Eds. Cambrige, MA, USA: MIT Press, 2003.

[33] X. Li, H. Chen, Z. Zhang, and J. Li, "Automatic Patent Classification using Citation Network Information: an Experimental Study in Nanotechnology," in *JCDL '07: Proceedings of the 2007 conference on Digital libraries*. New York, NY, USA: ACM, 2007, pp. 419–427.

[34] C. H. A. Koster, M. Seutter, and J. Beney, "Multi-classification of Patent Applications with Winnow," in *Ershov Memorial Conference*, 2003, pp. 546–555.

## Appendix A

In this Appendix it will be explained our reasons for not having used the same set of criterions presented in [2].
The original criterions are: the first criterion creates one partial order "$\succ$" that evaluates the performance between two classifiers for each metric. In that way, if the classifier $A1$ has a better performance than $A2$ to a given metric, so we have $A1 \succ A2$. In order to perform this task, we used two-tailed paired t-test at $p\%$ significance level. The second criterion is applied a system based on rewards and punishes. For example, for the case of $A1 \succ A2$ the classifier $A1$ is rewarded with $+1$ and the classifier $A2$ is punished with $-1$. Then we sum the rewards and punishes of each classifier to obtain total order "$>$". In this case, if $A1 > A2$, so $A1$ is superior to $A2$.
To show the inconsistency of the second criterion, initially we regard the following fictitious results, Table 10, reached by classifiers *A* and *B* for some databases and some metric, which the smaller value, the better.

| Data Set | A | B |
|---|---|---|
| Data 1 | 1 | 6 |
| Data 2 | 2 | 5 |
| Data 3 | 3 | 4 |
| Data 4 | 4 | 3 |
| Data 5 | 5 | 2 |
| Data 6 | 6 | 1 |

*Table 10*: Fictitious performance of *A* and *B*.

If we apply a two-tailed paired t-test at 5% significance level between the classifiers *A* and *B*, we will see that they are not different statistically.
We will imagine now that a result of a third classifier, in this case, classifier *C* is added as it is shown in Table 11.

| Data Set | A | B | C |
|---|---|---|---|
| Data 1 | 1 | 6 | 5 |
| Data 2 | 2 | 5 | 4 |
| Data 3 | 3 | 4 | 3 |
| Data 4 | 4 | 3 | 2 |
| Data 5 | 5 | 2 | 1 |
| Data 6 | 6 | 1 | 0 |

*Table 11*: Fictitious performance of *A*, *B* and *C*.

Again employing a two-tailed paired t-test at 5% significance level between two classifiers a time, according the first criterion, we obtain that *C* is different statistically of *B* and the pairs *A*, *B* and *A*,*C* are not different statistically. Using the second criterion, we achieve the order shown in Table 12, where the accumulated score of each algorithm is also shown in the parentheses.

| C(1) > A(0) > B(-1) |
|---|

*Table 12*: Fictitious result of performance order of *A*, *B* and *C*.

This information indicates that classifier *C* was superior to *A* and *B*, and *A* was superior to *B*. Such result is a quite unfair becuause of the insertion of a new classifier should not alter the relationship between *A* and *B*.
In the criterion used in this article, we will have the results presented in Table 13.
In this case it is shown that *C* is superior to *B*, but *A* is neither superior to *B* nand to *C*. The relationship among the classifiers is keept.
Unfortunately, this problem is more complex than the one illustrated here. For example, if we either add or remove some metric in our experiments, then the results could modify radically, for simple reason that we are summing between two classifiers the statistical test results done to each metric. But, in our experiments, we consider the same set of metrics for all algorithms, which avoids to happen such problem.

| |
|---|
| **A** |
| C(1) > **B** |
| **C** > B(-1) |

*Table 13*: Fictitious result of performance order of *A*, *B* and *C*.

## About the Authors

**Patrick Marques Ciarelli** received his B.Eng. and M.Sc. degrees in Electrical Engineering in 2006 and 2008, respectively, from the Universidade Federal do Espírito Santo (UFES), Brazil. Currently, he is doing doctorate in Electrical Engineering at UFES. His current research interests include neural networks, information retrieval, pattern recognition, and image processing.

**Dr. Elias Oliveira** is currently a Lecturer at the Information Science Department at the Universidade Federal do Espírito Santo (UFES), Brazil. He received the B.Sc. degree in Mathematics from Universidade Federal do Rio de Janeiro (UFRJ) and was awarded his Ph.D. in Computer Science in 2001, from School of Computing of the University of Leeds, UK. Dr. Oliveira is a collaborator of the Laboratório de Computação de Alto Desempenho (LCAD – High Performance Computing Laboratory) at UFES. His research interests include machine learning, evolutionary algorithms, information retrieval, text categorization, practical applications of constraint programming, and neural networks applications.

**Dr. Claudine Badue** is an Associate Researcher and member of the Laboratório de Computação de Alto Desempenho (LCAD — High Performance Computing Laboratory) at the Universidade Federal do Espírito Santo (UFES). In 1998, she received the B.Sc. degree in Computer Science from the Universidade Federal de Goiás (UFG), Brazil. She received the M.Sc. degree in Computer Science in 2001 and the Ph.D. degree in Computer Science in 2007, both from the Universidade Federal de Minas Gerais (UFMG), Brazil.

Her research interests are in the areas of information retrieval, text categorization, and performance analysis and modeling. She has been involved in research projects financed through Brazilian research agencies such as the National Research Council (CNPq) and the Espírito Santo Science and Technology Foundation (FEST). She has also been in the program committee and organizing committee of national and international conferences in Computer Science.

**Dr. Alberto Ferreira De Souza** is a Professor of Computer Science and Coordinator of the Laboratório de Computação de Alto Desempenho (LCAD — High Performance Computing Laboratory) at the Universidade Federal do Espírito Santo (UFES), Brazil. He received B.Eng. (Cum Laude) in Electronics Engineering and M.Sc. in Systems Engineering and Computer Science from Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Brazil, in 1988 and 1993, respectively; and Doctor of Philosophy (Ph.D) in Computer Science from the University College London, United Kingdom in 1999. He has authored/co-authored one USA patent and over 60 publications. He has edited proceedings of four conferences (two IEEE sponsored conferences), is a Standing Member of the Steering Committee of the International Conference in Computer Architecture and High Performance Computing (SBAC-PAD), and Coordinator of the Technical Committee on Computer Architecture and High Performance Computing of the Brazilian Computer Society (SBC).

Dr. De Souza held the following positions in UFES and elsewhere: member of the Board of Directors of the Regional Council of Engineering of Espírito Santo — CREA-ES (1995-1996), Vice-Dean of the School of Engineering — UFES (2001-2004), Director Superintendent of the Institute of Technology — UFES (2003-2004), and Pro-Provost of Planning and Development — UFES (2004-2007). At the present time, Alberto is Vice-President of the Administrative Council of the Espírito Santo Science and Technology Foundation (FEST) — FEST, and president of Steering Committee of the Vitória High Speed Metropolitan Area Network (METROVIX) — METROVIX.

Alberto Ferreira De Souza is Comendador of the order of Rubem Braga.