Improving VG-RAM WNN Multi-label Text Categorization via Label Correlation

Alberto F. De Souza, Claudine Badue,
Bruno Zanetti Melotti, Felipe T. Pedroni, and Fernando Líbio L. Almeida

Universidade Federal do Espírito Santo
29075-910 - Vitória - ES - Brazil

alberto@lcad.inf.ufes.br, claudine@lcad.inf.ufes.br,
bruno@lcad.inf.ufes.br, fpedroni@lcad.inf.ufes.br, flibio@lcad.inf.ufes.br

Abstract

In multi-label text databases one or more labels, or categories, can be assigned to a single document. In many such databases there can be correlation on the assignment of subsets of the set of categories. This can be exploited to improve machine learning techniques devoted to multi-label text categorization. In this paper, we examine a Virtual Generalizing Random Access Memory Weightless Neural Network (VG-RAM WNN for short) architecture that takes advantage of the correlation between categories to improve text-categorization performance. We compare the performance of this architecture, that we named Data Correlated VG-RAM WNN (VG-RAM WNN-COR), with that of standard VG-RAM WNN using four multi-label categorization performance metrics: one-error, ranking loss, average precision and hamming loss. Our experimental results show that VG-RAM WNN-COR has an overall better performance than VG-RAM WNN for the set of metrics considered.

1. Introduction

Most works on text categorization in the literature are focused on single-label text categorization problems, where each document may only have a single label [15]. However, in real-world problems, multi-label categorization is frequently necessary [14, 8, 5, 16, 3, 9, 12, 17, 18]. From a theoretical point of view, single-label categorization is more general than multi-label, since an algorithm for single-label categorization can also be used for multi-label categorization: one needs only to transform the multi-label categorization problem into n independent single-label problems, where n is the number of possible labels, or categories [15]. However, this equivalence only holds if the n categories are stochastically independent, that is, the association of a category c_i to a document is independent of the association of another category, c_j, to the same document; however, this is frequently not the case. Multi-label categorization systems can take advantage of the correlation between categories in order to improve their performance.

Virtual Generalizing Random Access Memory Weightless Neural Networks (VG-RAM WNN for short) is an effective machine learning technique which offers simple implementation and fast training and test [2, 10]. In this paper we present a new VG-RAM WNN architecture that exploits the correlation between categories. We named this architecture Data Correlated VG-RAM WNN (VG-RAM WNN-COR, for short). Different from standard VG-RAM WNN's neurons, which can only assign a single category to a document, in VG-RAM WNN-COR each neuron can assign one or more categories to a document simultaneously.

Several techniques for multi-label categorization have been proposed, such as multi-label decision trees [5], kernel methods [8, 3] or neural networks [12, 17], and many of them specifically for multi-label text categorization [14, 16, 9, 12, 17]. In a previous work [6], we compared the VG-RAM WNN performance with that of the multi-label lazy learning technique (ML-KNN) proposed by Zhang and Zhou [18]. Their technique achieved higher performance than many well-established algorithms in several multi-label problems [18]; however, our experiments showed that VG-RAM WNN outperforms ML-KNN in a number of multi-label text categorization metrics.

We evaluate the performance of VG-RAM WNN-COR on the categorization of companies according to their economic activities. The automation of the categorization of companies according to their economic activities described in free text format is a huge challenge for the Brazilian gov-

ernmental administration in the present day. So far, this task has been carried out by humans, not all of them properly trained for the job. In our evaluation of VG-RAM WNN-COR, we use four multi-label categorization performance metrics: one-error, ranking loss, average precision and hamming loss. Our experimental evaluation shows that VG-RAM WNN-COR outperforms VG-RAM WNN, showing better performance in three out of four evaluation metrics (two-tailed paired t-test at 5% significance level).

This paper is organized as follows. Section 2 presents the multi-label text categorization problem and Section 3 our VG-RAM WNN and VG-RAM WNN-COR categorizers. Section 4 presents our experimental methodology and analyzes our experimental results. Our conclusions follow in Section 5.

2. Multi-Label Text Categorization

Text categorization may be defined as the task of assigning categories (or labels), from a predefined set of categories, to documents [15]. In multi-label text categorization, one or more categories may be assigned to a document.

Let $\mathcal{D}$ be the domain of documents, $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ a set of pre-defined categories, and $\Omega = \{d_1, \ldots, d_{|\Omega|}\}$ an initial corpus of documents previously categorized manually by a domain expert into subsets of categories of $\mathcal{C}$. In multi-label learning, the training(-and-validation) set $TV = \{d_1, \ldots, d_{|TV|}\}$ is composed of a number documents, each associated with a subset of categories of $\mathcal{C}$. 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 $Te = \{d_{|TV|+1}, \ldots, d_{|\Omega|}\}$, on the other hand, consists of documents for which the categories are unknown to the categorization system. After being (tunned and) trained with TV, the categorization system is used to predict the set of categories of each document in Te.

A multi-label categorization system typically implements a real-valued function $f : \mathcal{D} \times \mathcal{C} \rightarrow \mathbb{R}$ that returns a value for each pair $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$ that, roughly speaking, represents the evidence for the fact that the test document d_j should be categorized under the category c_i. The real-valued function $f(.,.)$ can be transformed into a ranking function $r(.,.)$, which is a one-to-one mapping onto $\{1, 2, \ldots, |\mathcal{C}|\}$, such that if $f(d_j, c_k) > f(d_j, c_l)$, then $r(d_j, c_k) < r(d_j, c_l)$.

If C_j is the set of pertinent categories for the test document d_j, then a successful categorization system will tend to rank categories in C_j higher than those not in C_j. Those categories c_i that rank above a threshold τ_i (i.e., $c_i | f(d_j, c_i) \geq \tau_i$) are then assigned to the test document d_j.

3. VG-RAM WNN and VG-RAM WNN-COR

RAM-based neural networks [1], also known as weightless neural networks (WNN), do not store knowledge in their connections but in Random Access Memories (RAM) inside the network's nodes, or neurons. In spite of their remarkable simplicity, WNN are very effective as pattern recognition tools, offering fast training and test, and easy implementation [2]. However, if the network input is too large, the memory size of the neurons of WNN becomes prohibitive, since it must be equal to 2^n, where n is the input size. Virtual Generalizing RAM (VG-RAM) networks are RAM-based neural networks that only require memory capacity to store the data related to the training set [10].

3.1. VG-RAM WNN Neurons

VG-RAM WNN neurons store the input-output pairs seen during training, instead of only the output. In the test phase, the memory of VG-RAM neurons is searched associatively by comparing the input presented to the network with all inputs in the input-output pairs learned. The output of each VG-RAM neuron is taken from the pair whose input is nearest to the input presented—the distance function employed by VG-RAM neurons is the hamming distance. If there is more than one pair at the same minimum distance from the input presented, the neuron's output is chosen randomly among these pairs.

lookup table	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	**Y**
entry #1	1	1	0	category 1
entry #2	0	0	1	category 2
entry #3	0	1	0	category 3
	↑	↑	↑	↓
input	1	0	1	**category 2**

Figure 1. VG-RAM WNN lookup table.

Figure 1 shows the lookup table of a VG-RAM neuron with three synapses (X_1, X_2 and X_3). This lookup table contains three entries (input-output pairs), which were stored during the training phase (entry $\#1$, entry $\#2$ and entry $\#3$). During the test phase, when an input vector (input) is presented to the network, the VG-RAM test algorithm computes the distance between this input vector and each input of the input-output pairs stored in the lookup table. In the example of Figure 1, the hamming distance from the input to entry $\#1$ is two, because both X_2 and X_3 bits do not match the input vector. The distance to entry $\#2$ is one, because X_1 is the only non-matching bit. The distance to entry $\#3$ is three, as the reader may easily verify. Hence, for this input vector, the algorithm evaluates the neuron's output, Y, as category 2, since it is the output value stored in entry $\#2$.

3.2. VG-RAM WNN-COR Neurons

While in VG-RAM WNN each neuron is trained to output a single category for each input vector, in VG-RAM WNN-COR each neuron may be trained to output a set of categories for each input vector.

Figure 2 illustrates the lookup table of a VG-RAM WNN-COR neuron with three synapses (X_1, X_2 and X_3) and three entries (input-output pairs) stored during the training phase (entry $\#1$, entry $\#2$ and entry $\#3$). Similar to VG-RAM WNN, when an input vector is presented to the network in the test phase, the VG-RAM WNN COR test algorithm computes the distance between this input vector and each input of the input-output pairs in the lookup table. In the example of Figure 2, the hamming distance from the input to entries $\#1$, $\#2$, and $\#3$ is two, one, and three, respectively. As the input of entry $\#2$ is the nearest to the network input, the output of the VG-RAM WNN COR neuron is given by categories 1 and 3, i.e. the value of Y represents both categories, 1 and 3.

lookup table	X_1	X_2	X_3	Y
entry #1	1	1	0	category 2
entry #2	0	0	1	category 1, 3
entry #3	0	1	0	category 1, 2, 3
	↑	↑	↑	↓
input	1	0	1	**category 1, 3**

Figure 2. VG-RAM WNN-COR lookup table.

3.3. Text Categorization with VG-RAM WNN and VG-RAM WNN-COR

To categorize text documents using VG-RAM WNN, we represent a document as a multidimensional vector $V = \{v_1, \ldots, v_{|V|}\}$, where each element v_i corresponds to a weight associated to a specific term in the vocabulary of interest (see Section 4.2). We use single layer VG-RAM WNN (Figure 3) whose neurons' synapses $X = \{x_1, \ldots, x_{|X|}\}$ are randomly connected to the network's input $N = \{n_1, \ldots, n_{|N|}\}$, which has the same size of the vectors representing the documents, i.e., $|N| = |V|$. Note that $|X| < |V|$ (our experiments have shown that $|X| < |V|$ provides better performance). Each neuron's synapse x_i forms a minchinton cell with the next, x_{i+1} ($x_{|X|}$ forms a minchinton cell with x_1) [11]. The type of the minchinton cell we have used returns 1 if the synapse x_i of the cell is connected to an input element n_j whose value is larger than that of the element n_k to which the synapse x_{i+1} is connected (i.e. $n_j > n_k$); otherwise, it returns zero.

During training, for each document in the training set, the corresponding vector V is connected to the VG-RAM WNN's input N and the neurons' outputs $O = \{o_1, \ldots, o_{|O|}\}$ to one of the categories of the document. All neurons of the VG-RAM WNN are then trained to output this category with this input vector. The training for this input vector is repeated for each category associated with the corresponding document. During test, for each test document, the inputs are connected to the corresponding vector and the number of neurons outputting each category is counted. The network's output is computed by dividing the count of each category by the number of neurons of the network. This output is organized as a vector whose size is equal to the number of categories. The value of each vector element varies from 0 to 1 and represents the percentage of neurons which presented the corresponding category as output (the sum of the values of all elements of this vector is always equal to 1). This way, the output of the network implements the function $f(.,.)$, defined in Section 2.

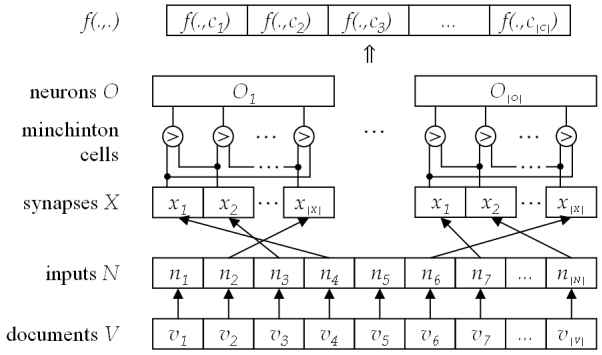


Figure 3. VG-RAM WNN and VG-RAM WNN-COR text categorization setup.

To categorize text documents using VG-RAM WNN-COR we use the same setup of the VG-RAM WNN illustrated in Figure 3. In the training phase, for each document in the training set, the corresponding vector V is connected to the input of the VG-RAM WNN COR, N, and the output of its neurons, O, to the set of categories assigned to the document. Each neuron of the VG-RAM WNN-COR is trained to output this set with this input vector. During the test phase, for each test document, the corresponding vector V is connected to the input of the network, N. The function $f(.,.)$ is computed by dividing the number of votes for each category by the total number of categories outputted by the network. The number of votes for each category is obtained by counting their occurrences in all sets outputted by the network.

4. Experimental Evaluation

The following subsections present the details of our experimental evaluation of VG-RAM WNN-COR.

4.1. Data Set

The classification of companies according to their economic activities is an important step of the process of obtaining information for statistical analysis of the economy within a city, state or country. In Brazil, all economic activities recognized by law are cataloged in a table called *"Classificação Nacional de Atividades Econômicas (CNAE)"* (National Classification of Economic Activities) [4]. Government officials must find the semantic correspondence between textual descriptions of economic activities of companies and one or more entries of the CNAE table for each new company or any that changes its set of economic activities.

To compare the performance of VG-RAM WNN-COR with that of VG-RAM WNN on the categorization of economic activities, we used a data set composed of 3281 textual descriptions of economic activities of companies categorized into a subset of 764 CNAE categories. The categorization of each company in this data set were performed by Brazilian government officials trained in this task. This data set also contains the official brief description of each one of the 1183 CNAE categories existing today. We partitioned the whole set of economic activities descriptions into ten subsets of 328 documents (the last one had 329) in order to perform ten-fold cross validation experiments.

To tune VG-RAM WNN and VG-RAM WNN-COR parameters (number of neurons and number of synapses per neuron) we used nine of the ten sets of documents mentioned above. We divided it again into 10 subsets and used the first nine for training and the last one for tuning the networks.

4.2. Data Preprocessing

We removed a set of stop words and stemmed the resulting words of the data set following the procedure for Brazilian Portuguese developed by Dias [7]. This removes stop words such as articles, preposition, pronouns, etc., and stems the remaining words removing Portuguese gender, plurals, augmentative, diminutive, etc., producing the vocabulary of interest.

After that, each document in the data set was transformed into the multidimensional vector of weights, $V = \{v_1, \ldots, v_{|V|}\}$, where $|V|$ is the number of terms that occurs at least once in the current training set. Each element v_i corresponds to the weight associated to each word i of the vocabulary of interest present in the document. This weight is computed according to the standard normalized *tfidf* weighting function [15].

4.3. Evaluation Metrics

We have used four multi-label evaluation metrics proposed in [13, 14] for examining the classification performance of VG-RAM WNN-COR, namely *one-error*, *ranking loss*, *average precision*, and *hamming loss*. The metrics one-error, ranking loss, and average precision evaluate the whole ranking derived from the real-valued function $f(.,.)$, while hamming loss evaluates the exact set of categories predicted for the test document d_j. We present each of these metrics below.

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

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

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

$$\text{rloss}_j = \frac{|\{(c_k, c_l) | f(d_j, c_k) \leq f(d_j, c_l)\}|}{|C_j||\bar{C}_j|} \quad (2)$$

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 (average-precision$_j$) evaluates the average of precisions computed after truncating the ranking of categories after each category $c_i \in C_j$ in turn:

$$\text{avgprec}_j = \frac{1}{|C_j|} \sum_{k=1}^{|C_j|} precision_j(R_{jk}) \quad (3)$$

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 $|R_{jk}|$. If there is a category $c_i \in C_j$ at the position k and $f(d_j, c_i) = 0$ then $precision_j(R_{jk}) = 0$.

Hamming Loss (hloss$_j$) 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 pertinent to the document is not predicted:

$$\text{hloss}_j = \frac{1}{|\mathcal{C}|} |P_j \Delta C_j| \quad (4)$$

where P_j is the set of categories predicted for the test document d_j, $|\mathcal{C}|$ is the number of categories, and Δ is

the symmetric difference between the set of predicted categories P_j and the set of pertinent categories C_j for the test document d_j.

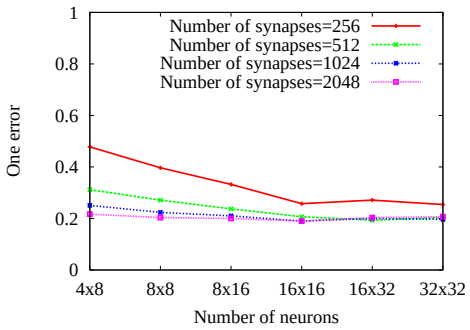
In this paper, instead of deriving the set of predicted categories for each test document d_j via a threshold τ_i for each category c_i, such that c_i is predicted to d_j only if $f(d_j, c_i) \geq \tau_i$, we derived the predicted set for d_j by truncating the ranking of categories in the position $k = |C_j|$. In this way, we evaluate the performance of categorizers under a perfect thresholding policy, by which the cardinality of the predicted set of categories is equal to the cardinality of the pertinent set.

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

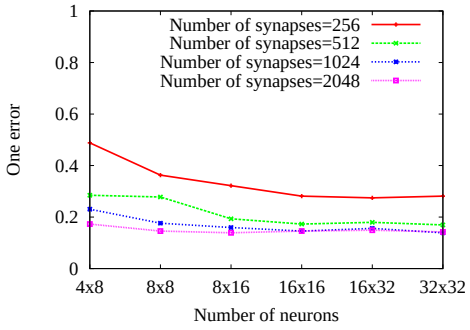
4.4. Experimental Results

To tune the parameters of the neural networks under study we used the metric one-error due to its simplicity and consequent easy understanding. Figure 4 presents the results of the validation experiments employed for tuning the number of neurons and synapses per neuron of the VG-RAM WNN and VG-RAM WNN-COR. As Figure 4 shows, the performance of both networks increase (one error decreases) with the number of neurons in the x-axis and with the number of synapses per neuron represented by each curve, but levels off when the networks have about 256 (16×16) neurons and 1024 synapses per neuron. Therefore, we used 256 neurons and 1024 synapses per neuron for both VG-RAM WNN and VG-RAM WNN-COR in the final evaluation experiments.

Table 1 shows the results of our performance comparison between VG-RAM WNN and VG-RAM WNN-COR. To produce the results shown in this table, for each of the ten folds mentioned in Section 4.1, we trained the networks with 4133 documents—1181 descriptions of CNAE categories and 2952 (nine folds) economic activities descriptions—and tested with 328 (one fold) descriptions of economic activities. Table 1 presents the average of the ten results obtained for each metric where the best result on each metric is shown in bold face. As this table shows, VG-RAM WNN-COR outperforms VG-RAM WNN in all metrics, achieving an one-error 3.6% smaller, ranking loss 37.4% smaller, average precision 1.2% higher and hamming loss 3.2% smaller.



(a) VG-RAM WNN



(b) VG-RAM WNN-COR

Figure 4. Results of validation experiments aimed at tuning the number of neurons and synapses per neuron of the networks.

To present a clearer view of the relative performance of the algorithms, a partial order $\succ$ is defined for each evaluation metric, where A1 $\succ$ A2 means that the performance of algorithm A1 is significantly better than that of algorithm A2 on the specific metric (two-tailed paired t-test at 5% significance level). If the performance is not significantly better, we say A1 $\equiv$ A2. The partial order on the two comparing algorithms in terms of each evaluation metric is shown in Table 2.

It is important to note that it is possible that A1 performs better than A2 in terms of some metrics but equivalent or worse in others. In this case, it is hard to judge which algorithm is superior. So, in order to give an overall perfor-

Table 1. Categorizers' performance.

VG-RAM	One-Error	Ranking Loss	Average Precision	Hamming Loss
WNN	0.17805	0.06646	0.72927	0.00279
WNN-COR	**0.17165**	**0.04161**	**0.73817**	**0.00270**

Table 2. Partial order.

Evaluation metric	Order
One-error	WNN-COR $\equiv$ WNN
Ranking loss	WNN-COR $\succ$ WNN
Average precision	WNN-COR $\succ$ WNN
Hamming loss	WNN-COR $\succ$ WNN
Total order	WNN-COR (3) $\succ$ WNN (-3)

mance assessment of an algorithm, we employed a score that takes into account its performance against that of the other algorithm on all metrics. Concretely, for each evaluation metric, if A1 $\succ$ A2 holds, then A1 is rewarded with a positive score $+1$ and A2 is penalized with a negative score -1. Based on the accumulated score of each algorithm on all evaluation metrics, a total order $\succ$ is defined on the two comparing algorithms, as shown in the last line of Table 2, where A1 $\succ$ A2 means that A1 performs better than A2. The accumulated score of each algorithm is also shown in the parentheses. As shown in Table 2, VG-RAM WNN-COR has an overall better performance than VG-RAM WNN for the set of metrics considered.

5. Conclusions

In this work, we presented an experimental evaluation of Data Correlated VG-RAM WNN (VG-RAM WNN-COR) on multi-label text classification and compared its performance with that of standard VG-RAM WNN. In order to do that, we used a database of textual descriptions of economic activities of companies categorized manually according to lawful Brazilian economic activities. Our experimental results showed that VG-RAM WNN-COR outperforms VG-RAM WNN, showing better performance in three out of four evaluation metrics (two-tailed paired t-test at 5% significance level).

6. Acknowledgments

We would like to thank *Receita Federal do Brasil*, *Conselho Nacional de Desenvolvimento Científico e Tecnológico* — CNPq-Brasil (grants 308207/2004-1, 471898/2004-0, 620165/2006-5, 309831/2007-5), *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 37711393/2007) for their support to this research work.

References

[1] I. Aleksander. Self-adaptive universal logic circuits. *IEEE Electronic Letters*, 2(8):231–232, 1966.

[2] I. Aleksander. *RAM-Based Neural Networks*, chapter From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines, pages 18–30. World Scientific, 1998.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[4] CNAE. Classificação Nacional de Atividades Econômicas - Fiscal (CNAE-Fiscal) 1.1. Technical report, Instituto Brasileiro de Geografia e Estatística (IBGE), Rio de Janeiro, RJ, 2003.

[5] F. D. Comité, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision tree from texts and data. In *Lecture Notes in Computer Science*, volume 2734, pages 35–49. Springer, 2003.

[6] 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*, 2008. To appear.

[7] M. A. L. Dias and M. G. Malheiros. Automatic extraction of keywords for the portuguese language. In *Proceedings of the 7th Workshop on Computational Processing of Written and Spoken Portuguese*, pages 204–207. Springer Berlin, Heidelberg, 2006.

[8] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, volume 14, pages 681–687. MIT Press, 2002.

[9] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua. A MFoM learning approach to robust multiclass multi-label text categorization. In *Proceedings of the 21st International Conference on Machine Learning*, pages 329–336, 2004.

[10] T. B. Ludermir, A. C. P. L. F. Carvalho, A. P. Braga, and M. D. Souto. Weightless neural models: a review of current and past works. *Neural Computing Surveys*, 2:41–61, 1999.

[11] R. J. Mitchell, J. M. Bishop, S. K. Box, and J. F. Hawker. *RAM-Based Neural Networks*, chapter Comparison of Some Methods for Processing Grey Level Data in Weightless Networks, pages 61–70. World Scientific, 1998.

[12] E. Romero, L. Màrquez, and X. Carreras. Margin maximization with feed-forward neural networks: a comparative study with svm and adaboost. *Neurocomputing*, 57:313–344, 2004.

[13] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 27(3):297–336, 1999.

[14] R. E. Schapire and Y. Singer. BoosTexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[15] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[16] N. Ueda and K. Saito. Parametric mixture models for multi-label text. In *Advances in Neural Information Processing Systems*, volume 15, pages 721–728. MIT Press, 2003.

[17] M.-L. Zhang and Z.-H. Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[18] M.-L. Zhang and Z.-H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.