International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
ISSN: 2150-7988 Vol.1 (2009), pp.155-169
http://www.mirlabs.org/ijcisim

Multi-Label Text Categorization with a Data Correlated VG-RAM Weightless Neural Network

Alberto F. De Souza, Bruno Zanetti Melotti, Claudine Badue
Universidade Federal do Espírito Santo, 29075-910 - Vitória-ES - Brazil

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

Abstract: **In multi-label text categorization, one or more labels (or categories) can be assigned to a single document. In many such categorization tasks, 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) 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 and ML-KNN categorizers using ten multi-label text categorization performance metrics. Our experimental results show that VG-RAM WNN-COR has an overall better performance than VG-RAM WNN and ML-KNN for the set of metrics considered.**

Keywords: VG-RAM Weightless Neural Networks, machine learning, multi-label text categorization, label correlation, categorization of economic activities, multi-label text categorization performance metrics

I. 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 [16]. However, in real-world problems, multi-label categorization is frequently necessary [15, 5, 4, 17, 3, 6, 13, 20, 21]. 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) [16]. 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, which is frequently not the case. Fortunately, several techniques for multi-label categorization have been proposed, such as multi-label decision trees [4], kernel methods [5, 3] or neural networks [13, 20], and many of them specifically for multi-label text categorization [15, 17, 6, 13, 20]. 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 fast training and test, and easy implementation [2, 9]. 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). 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.

We evaluate the performance of VG-RAM WNN-COR on the categorization of free-text descriptions of economic activities. The automation of the categorization of economic activities of companies from business descriptions in free text format is a huge challenge for the Brazilian governmental administration in the present day. So far, this task has been carried out by humans, not all of them properly trained for the job. When this problem is tackled by humans, the subjectivity on their categorization brings a problem: different human categorizers can give different results when working on the same business description. This can cause distortions in the information used for planning, taxation and other governmental obligations of the three Brazilian administrative levels: County, State and Federal. Furthermore, the number of possible categories considered is very large, more than 1000 in the Brazilian scenario, which makes the categorization problem even harder to be solved.

We analyze the performance of VG-RAM WNN-COR using ten multi-label text categorization performance metrics: *one-error* [14], *coverage* [15], *ranking loss* [14], *average precision* [10], *R-precision* [10], *Hamming loss* [14], *exact match* [8], *precision* [10, 16], *recall* [10, 16], and F_1 [10, 16]. We also compare the VG-RAM WNN-COR performance with that of standard VG-RAM WNN and Multi-Label k-Nearest Neighbors (ML-KNN) [21] categorizers. Our experimental evaluation shows that VG-RAM WNN-COR has an overall better performance than VG-RAM WNN and ML-KNN on the categorization of economic activities for the set of metrics considered.

This paper is organized as follows. After this introduction, Section II defines the multi-label text categorization problem. Section III describes our VG-RAM WNN and VG-RAM WNN-COR categorizers, and Section IV the ML-KNN categorizer. Section V presents our experimental methodology, and Section VI our experimental results. Our conclusions follow in Section VII.

II. 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 [16]. 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 of 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 trained and tune on 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 of the form $f : \mathcal{D} \times C \rightarrow \mathbb{R}$ that returns a degree of belief for each pair $\langle d_j, c_i \rangle \in \mathcal{D} \times C$, that is, a number between 0 and 1 that, roughly speaking, represents the confidence with which 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 $r(.,.)$, such that, if $f(d_j, c_i) > f(d_j, c_k)$, then $r(d_j, c_i) < r(d_j, c_k)$, and if $f(d_j, c_i) < f(d_j, c_k)$, then $r(d_j, c_i) > r(d_j, c_k)$. If $f(d_j, c_i) = f(d_j, c_k)$ we have a tie.

When there are no ties, i.e., $f(d_j, c_i) \neq f(d_j, c_k)$ for all $i \neq k$, $f(.,.)$ can be transformed into a ranking function $r(.,.)$ that is an one-to-one mapping onto $\{1, 2, \ldots, |C|\}$. However, if there are ties ($f(d_j, c_i) = f(d_j, c_k)$ for some $i \neq k$), the categories can be ranked in many different ways. In this paper, we adopted the ranking method called ordinal ranking [18], that assigns distinct ordinal ranking positions to all categories, including those tied. In this method, the assignment of distinct ordinal ranking positions to tied categories is done at random.

Let C_j be the set of pertinent categories of the test document d_j and $\hat{C}_j$ the set of categories predicted for d_j. A successful categorization system will tend to rank categories in C_j higher than those not in C_j. Those categories c_i ranked above a threshold τ_i are then predicted to the test document d_j, i.e., $\hat{C}_j^{\tau_i} = \{c_i | f(d_j, c_i) \geq \tau_i\}$.

III. 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 [9].

A. 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	**X_1**	**X_2**	**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 $\#2$.

B. 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.

C. *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 V-B). 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.

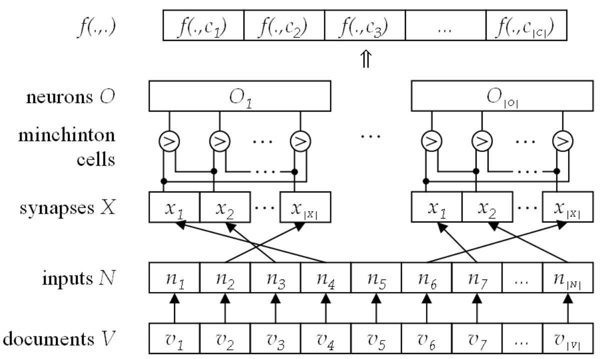


Figure. 3: VG-RAM WNN and VG-RAM WNN-COR text categorization setup.

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). In this way, the output of the network implements the function $f(.,.)$, defined in Section II.

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.

IV. ML-kNN

The Multi-Label k-Nearest Neighbors (ML-KNN) [21] categorizer is a version of the k-Nearest Neighbors (KNN) [16] especially designed for multi-label categorization. In this categorizer, the k nearest neighbors of d_j are identified in TV. The Euclidean distance is used to find the nearest neighbors of d_j. Then, for the given k, the maximum a posteriori (MAP) principle is employed for determining the belief for each pair $\langle d_j, c_i \rangle \in \mathcal{D} \times C$ 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.

Zhang and Zhou [21] evaluated the performance of ML-KNN on several multi-label learning problems. In their experiments, ML-KNN achieved higher performance than well-established algorithms, such as Boostexter [15], the multi-label kernel method Rank-SVM [5], and the multi-label decision tree ADTBoost.MH [4]. This has motivated us to use ML-KNN as a baseline in the VG-RAM WNN-COR evaluation.

V. Experimental Methodology

We employed a series of experiments to compare VG-RAM WWN-COR with VG-RAM WNN and ML-KNN. For that, we (i) used two data sets composed of textual descriptions of economic activities of companies categorized manually according lawful Brazilian economic activities. We (ii) preprocessed these data sets using standard IR techniques, and used the resulting data to (iii) tune VG-RAM WNN-COR, VG-RAM WNN, and ML-KNN categorizers and (iv) perform experiments for comparing VG-RAM WWN-COR with VG-RAM WNN and ML-KNN using multi-label text categorization performance metrics. The following subsections present the details of the parts (i), (ii), and (iii) of our experimental evaluation of VG-RAM WNN-COR. The experimental results, or part (iv), are presented in the next section.

A. Data Sets

The categorization 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) [7]. 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 and ML-KNN on the categorization of economic activities, we employed two data sets, each of which composed of textual descriptions of economic activities of companies categorized into a subset of CNAE categories by Brazilian government officials trained in this task.

The first data set, called EX100, consists of 6911 documents (textual descriptions) categorized into 105 different economic activities (categories). Each one of these categories occurs in exactly 100 different documents of this data set, i.e., there are 100 instances of documents of each category; the average number of categories per document is roughly 1.52 (standard deviation 0.79). The characteristics of EX100 allows examining the performance of categorizers in the case where the categories (or labels) are evenly distributed across the documents. This data set also contains the official brief description of each one of the 105 CNAE categories and their corresponding code.

The second data set, called AT100, consists of 10495 documents categorized into 762 categories. Each category appears in *up to* 100 different documents, i.e., there are between 1 and 100 instances of documents of each category; the average number of categories per document is roughly 1.49 (standard deviation 0.86). The characteristics of AT100 allows examining the performance of categorizers in the case where there are rare categories. This data set also contains the official brief description of each one of the 762 CNAE categories and their corresponding code.

We partitioned EX100 into 10 subsets of 691 documents (the last one had 692) and AT100 into 10 subsets of 1049 documents (the last one had 1054) in order to perform 10-fold cross-validation experiments.

B. Data Preprocessing

We transformed all words in our data sets into their uninflected form (term), i.e., the dictionary form of the word (known as lemma [10]), and then removed all prepositions using the Diadorim electronic dictionary of the Brazilian Portuguese language [12]. After that, we identified all distinct terms in each training set, TV, i.e., the vocabulary of interest. Note that, as we are using 10-fold cross-validation, we have 10 training sets for EX100 and 10 for AT100 and, therefore, 20 vocabularies of interest. Using the vocabulary of interest associated with each training set, we transformed all documents of the 20 training set/test set pairs into their corresponding 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 was computed according to the standard normalized *tfidf* weighting function [16].

The average size of the vocabulary of interest is roughly 3609.8 terms (standard deviation 21.17) for EX100, and roughly 5377.6 terms (standard deviation 19.45) for AT100. Table V-B shows the sizes of the vocabularies of interest of EX100 and AT100 for the 20 training set/test set pairs.

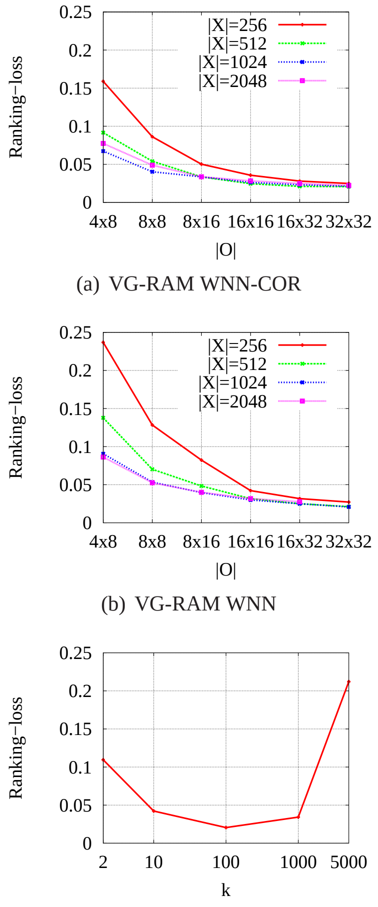
Fold	$\|V\|$	
	EX100	**AT100**
1	3605	5392
2	3614	5404
3	3634	5406
4	3594	5386
5	3600	5363
6	3654	5360
7	3578	5363
8	3612	5386
9	3601	5363
10	3606	5353

Table 1: The size of the vocabulary of interest of each one of the 20 training set/test set pairs.

C. Categorizers Validation

The VG-RAM WNN-COR, VG-RAM WNN and ML-KNN categorizers possess parameters that can be optimized for achieving best performance in a given data set. To tune (or to validate) these categorizers, we used a single training(-and-validation) set, TV, for each data set detailed above. We divided each of these two TV sets into 10 subsets, and used the first nine to train and the last one to tune the parameters of the categorizers for each data set according to the *ranking loss* [14] metric (see Section VI-A). This metric evaluates the fraction of category pairs $\langle c_i, c_k \rangle$, $c_i \in C_j$ and $c_k \in \bar{C}_j$, that are or may be reversely ordered ($f(d_j, c_i) \leq f(d_j, c_k)$) in the ranking of categories for the test document d_j of a given data set. We chose the metric *ranking loss* for validation because it is not affected by ties, can be used for evaluating the whole ranking produced by the categorizers, and is commonly used for evaluating rank-based text categorization systems [14, 15, 5, 21].

Figure 4 and Figure 5 present the results of the validation experiments employed for tuning the number of neurons and synapses per neuron of the VG-RAM WNN-COR and VG-RAM WNN, and the parameter k of ML-KNN, for the EX100 and AT100 data sets, respectively. As Figure 4(a) shows, for the EX100 data set, the performance of VG-RAM WNN-COR increases (*ranking loss* 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 network have about 32×32 (1024) neurons and 512 synapses per neuron; while, for the AT100 data set (Figure 5(a)), the performance levels off when the network have about 32×32 (1024) neurons and 1024 synapses per neuron. Therefore, in the experimental evaluation of VG-RAM WNN-COR with EX100 we used 32×32 (1024) neurons and 512 synapses per neuron, while with AT100 we used 32×32 (1024) neurons and 1024 synapses per neuron. Applying the same reasoning and using the results shown in Figure 4(b) and Figure 5(b), for VG-RAM WNN we chose $|O| = 32 \times 32$ (1024) and $|X| = 1024$ for EX100, and $|O| = 32 \times 32$ (1024) and $|X| = 512$ for AT100. Finally, we found that, in the case of ML-KNN, k equal to 100 nearest neighbors produces the best performance results for both the EX100 and AT100 data sets (see Figure 4(c) and Figure 5(c)).



(a) VG-RAM WNN-COR

(b) VG-RAM WNN

(c) ML-KNN

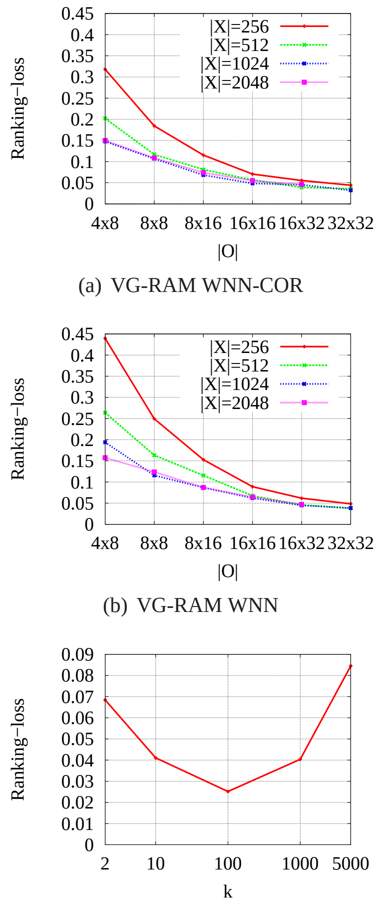
Figure. 4: Results of validation experiments aimed at tuning the VG-RAM WNN-COR, VG-RAM WNN, and ML-KNN categorizers for EX100 data set.

VI. Experimental Results

The metrics used in the literature to evaluate text categorization performance can roughly be divided into two groups:

(i) **Evaluation metrics for ranked sets**, which evaluate the whole ranking of categories derived from the real-valued function $f(.,.)$; these include *one-error* [14], *coverage* [15], *ranking loss* [14], *average precision* [10], and *R-precision* [10];

(ii) **Evaluation metrics for unranked sets**, which evaluate the set of categories predicted for the test document d_j, $\hat{C}_j$ (see Section II), among which the most frequent are *Hamming loss* [14], *exact match* [8], *precision* [10, 16], *recall* [10, 16], and F_β [10, 16].



(a) VG-RAM WNN-COR

(b) VG-RAM WNN

(c) ML-KNN

Figure. 5: Results of validation experiments aimed at tuning the VG-RAM WNN-COR, VG-RAM WNN, and ML-KNN categorizers for AT100 data set.

In the following two subsections, we present the experiments we have used to compare the VG-RAM WNN-COR performance against that of VG-RAM WNN and ML-KNN.

A. Results with Metrics for Ranked Sets

***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:

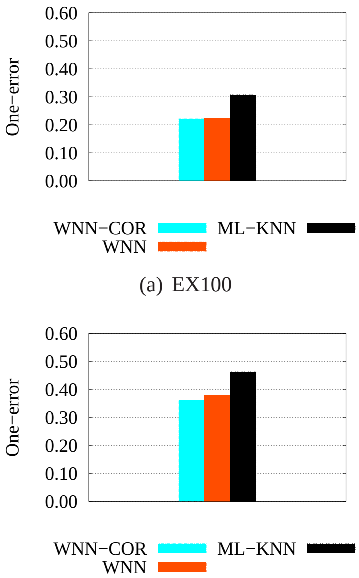
$$\textit{one-error}_j = \begin{cases} 0 & \text{if } [\arg\max_{c_i \in C} f(d_j, c_i)] \in C_j \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

where $[\arg\max_{c_i \in C} f(d_j, c_i)]$ returns the top ranked category for the test document d_j.
The overall performance is obtained by:

$$\textit{one-error} = \frac{1}{|Te|} \sum_{j=1}^{|Te|} \textit{one-error}_j \quad (2)$$

The smaller the value of one-error, the better the performance of the categorization system. The performance is perfect when *one-error* $= 0$.
Figure 6 shows the VG-RAM WNN-COR, VG-RAM WNN and ML-KNN performance in terms of *one-error* for EX100 and AT100 (the smaller the better). As the figure shows,

VG-RAM WNN-COR has about the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). This is to be expected since, when we have enough examples of each category (EX100), the benefits of data correlation may diminish; while, when certain categories are not well represented in the data set (AT100), data correlation between those and others in the data set, when captured, may allow better categorization performance. Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).



(a) EX100

(b) AT100

Figure. 6: *One-error* (the smaller the better)

***Coverage* (*coverage*$_j$)** measures how far we need to go down the ranking of categories for the test document d_j in order to cover all its pertinent categories:

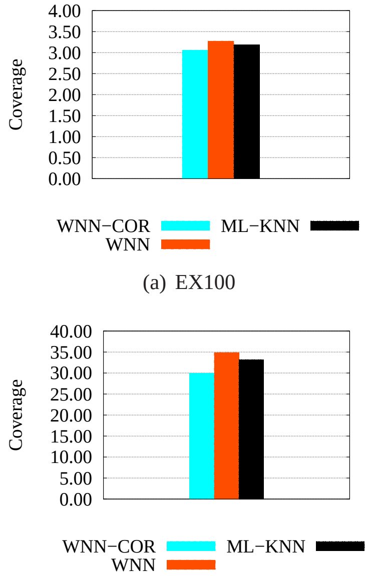
$$coverage_j = \max_{c_i \in C_j} r(d_j, c_i) - 1, \quad (3)$$

where $\max_{c_i \in C_j} r(d_j, c_i)$ returns the maximum rank for the set of pertinent categories of d_j, C_j. The overall performance is given by:

$$coverage = \frac{1}{|Te|} \sum_{j=1}^{|Te|} coverage_j. \quad (4)$$

The smaller the value of *coverage*, the better the performance of the categorization system. The performance is perfect when *coverage* $= \frac{1}{|Te|} \sum_{j=1}^{|Te|} (|C_j| - 1)$.

Figure 7 shows the VG-RAM WNN-COR, VG-RAM WNN and ML-KNN performance in terms of *coverage* for EX100 and AT100 (the smaller the better). As the figure shows, VG-RAM WNN-COR outperforms VG-RAM WNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level). This happens because data correlation allows VG-RAM WNN-COR to move pertinent categories up in the ranking, reducing the coverage. Although Figure 7 may suggests that VG-RAM WNN-COR outperforms ML-KNN for EX100 and AT100, the performance advantage is only significant for AT100 (two-tailed paired t-test at 5% significance level). However, it is important to note that, exploring data correlation, VG-RAM WNN may outperform ML-KNN.



(a) EX100

(b) AT100

Figure. 7: *Coverage* (the smaller the better)

***Ranking loss* (*ranking-loss*$_j$)** evaluates the fraction of category pairs $\langle c_i, c_k \rangle$, $c_i \in C_j$ and $c_k \in \bar{C}_j$, that are reversely ordered ($f(d_j, c_i) \leq f(d_j, c_k)$) in the ranking of categories for the test document d_j:

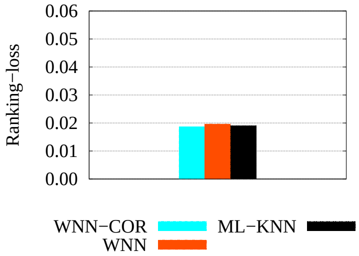
$$\begin{aligned} ranking\text{-}loss_j \quad = \quad & \frac{1}{|C_j||\bar{C}_j|} |\{(c_i, c_k) | f(d_j, c_i) \leq f(d_j, c_k), \\ & (c_i, c_k) \in C_j \times \bar{C}_j\}|, \end{aligned} \quad (5)$$

where $\bar{C}_j$ is the complementary set of C_j in C. The overall performance is computed as:

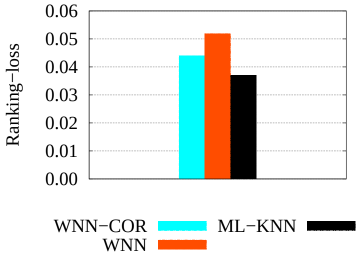
$$ranking\text{-}loss = \frac{1}{|Te|} \sum_{j=1}^{|Te|} ranking\text{-}loss_j. \quad (6)$$

The smaller the value of *ranking loss*, the better the performance of the categorizer. The performance is perfect when *ranking-loss* $= 0$.

Figure 8 shows the VG-RAM WNN-COR, VG-RAM WNN and ML-KNN performance in terms of *ranking loss* for EX100 and AT100 (the smaller the better). As the figure shows, VG-RAM WNN-COR outperforms VG-RAM WNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level). VG-RAM WNN-COR exhibits about the same performance of ML-KNN for EX100, but an inferior performance than ML-KNN for AT100 (two-tailed paired t-test at 5% significance level). This happens because, for documents associated with rare categories, the neural network may not output any pertinent category, which results in a larger *ranking loss*—by definition, for any given document ML-KNN always output a different than zero belief for all categories. Note that this has an smaller impact in VG-RAM WNN-COR than in VG-RAM WNN thanks to data correlation.

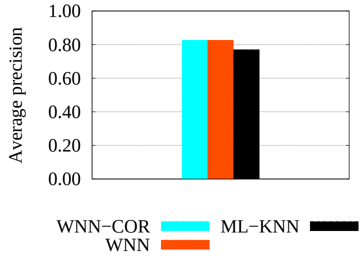


(a) EX100

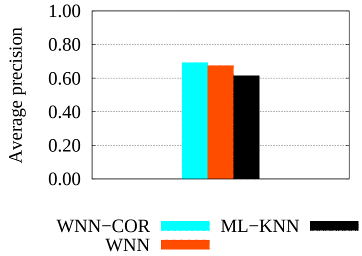


(b) AT100

Figure. 8: *Ranking loss* (the smaller the better)



(a) EX100



(b) AT100

Figure. 9: *Average precision* (the larger the better)

Average precision **(avg-precision$_j$)** evaluates the average of precisions computed truncating the ranking of categories for the test document d_j after each category $c_i \in C_j$ in turn:

$$\textit{avg-precision}_j = \frac{1}{|C_j|} \sum_{k=1}^{|C_j|} \frac{|\hat{C}_j^k \cap C_j|}{|\hat{C}_j^k|}, \quad (7)$$

where $|C_j|$ is the number of pertinent categories of the test document d_j, and $\hat{C}_j^k$ is the set of predicted categories that goes from the top of the ranking until the ranking position k. If there is a category $c_i \in C_j$ at position k and $f(d_j, c_i) = 0$, then the precision value obtained for $\hat{C}_j^k$ in Equation (7) is taken to be 0.
The overall performance is calculated as:

$$\textit{avg-precision} = \frac{1}{|Te|} \sum_{j=1}^{|Te|} \textit{avg-precision}_j. \quad (8)$$

The larger the value of *average precision*, the better the performance of the categorization system. The performance is perfect when *avg-precision* $= 1$.
Figure 9 shows the categorizers' performance in terms of *average precision* for EX100 and AT100 (the larger the better). As the figure shows, VG-RAM WNN-COR has about the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level). These results are in line with those of *one-error* and have the same explanations (see above).

R-precision **(*R-precision*$_j$)** evaluates the precision computed with the $|C_j|$ top ranked categories for d_j:

$$\textit{R-precision}_j = \frac{|\hat{C}_j^{|C_j|} \cap C_j|}{|\hat{C}_j^{|C_j|}|}, \quad (9)$$

where $\hat{C}_j^{|C_j|}$ is the set of $|C_j|$ top ranked categories. Note that categories c_i in the set of $|C_j|$ top ranked categories for which $f(d_j, c_i) = 0$ should not be inserted into $\hat{C}_j^{|C_j|}$. In this case, $|\hat{C}_j^{|C_j|}|$ may be smaller than $|C_j|$.
The overall performance is obtained by:

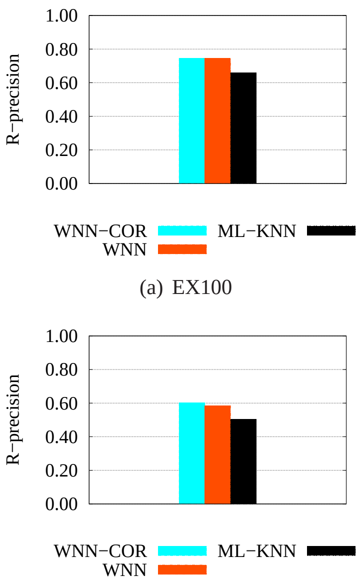
$$\textit{R-precision} = \frac{1}{|Te|} \sum_{j=1}^{|Te|} \textit{R-precision}_j. \quad (10)$$

The larger the value of *R-precision*, the better the performance of the categorizer. The performance is perfect when *R-precision* $= 1$.
Figure 10 shows the categorizers' performance in terms of *R-precision* for EX100 and AT100 (the larger the better). Similarly to the case of *average precision*, VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).

B. Results with Metrics for Unranked Sets

The metrics examined in this section evaluate the set of categories predicted for a given d_j, $\hat{C}_j$, instead of a ranking, as the metrics described in the previous section. Because of that, we need a means of thresholding the ranking of categories derived from $f(.,.)$. There are various techniques for determining the threshold τ_i for each category c_i [19, 16]. We evaluate the performance of all categorizers examined under a perfect thresholding policy; i.e., we choose the cardinality of the predicted set of categories for d_j, $|\hat{C}_j|$, to be equal to $|C_j|$ (or approximately equal to $|C_j|$). Thus, as we have done for the metric *R-precision* (see above), we derive $\hat{C}_j$ from the $|C_j|$ top ranked categories for d_j and call it $\hat{C}_j^{|C_j|}$.

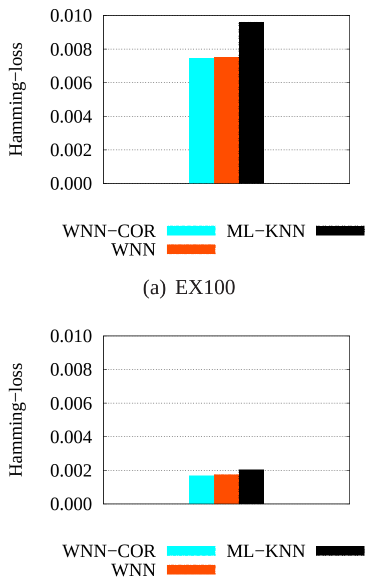
Hamming loss **(*Hamming-loss*$_j$)** evaluates how many times the test document d_j is misclassified (i.e., a category not belonging to the document is predicted or a category belonging to the document is not predicted), normalized by the total



(a) EX100

(b) AT100

Figure. 10: *R-precision* (the larger the better)



(a) EX100

(b) AT100

Figure. 11: *Hamming loss* (the smaller the better)

number of categories:

$$\textit{Hamming-loss}_j = \frac{|\hat{C}_j^{|C_j|} \ominus C_j|}{|C|}, \tag{11}$$

where $\ominus$ indicates the symmetric difference between the set of predicted categories, $\hat{C}_j^{|C_j|}$, and the set of pertinent categories of d_j, C_j.
The overall performance is calculated as:

$$\textit{Hamming-loss} = \frac{1}{|Te|} \sum_{j=1}^{|Te|} \textit{Hamming-loss}_j. \tag{12}$$

The smaller the value of *Hamming loss*, the better the performance of the categorizer. The performance is perfect when *Hamming-loss* $= 0$.
Figure 11 shows the categorizers' performance in terms of *Hamming loss* for EX100 and AT100 (the smaller the better). As in the case of *average precision* (see previous subsection), VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).

***Exact match* (*exact-match*$_j$)** evaluates how frequently all and only all pertinent categories are present in the set of predicted categories for d_j:

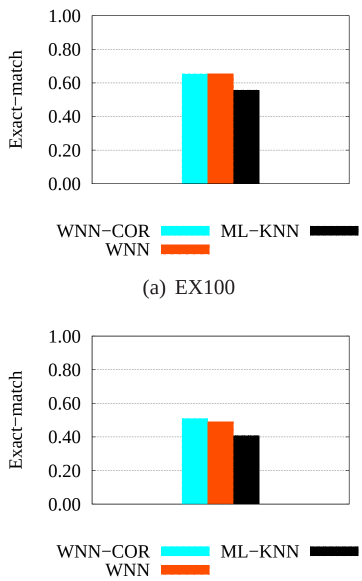
$$\textit{exact-match}_j = \begin{cases} 1 & \text{if } \hat{C}_j^{|C_j|} = C_j; \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The overall performance is obtained by:

$$\textit{exact-match} = \frac{1}{|Te|} \sum_{j=1}^{|Te|} \textit{exact-match}_j. \tag{14}$$

The larger the value of *exact match*, the better the performance of the categorizer. The performance is perfect when *exact-match* $= 1$.

Figure 12 shows the categorizers' performance in terms of *exact match* for EX100 and AT100 (the larger the better). As before, VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).



(a) EX100

(b) AT100

Figure. 12: *Exact match* (the larger the better)

***Precision on a per-category basis* (*precision*$_i^c$)** evaluates the fraction of test documents categorized under the category c_i that are truly associated with c_i, and can be estimated using the contingency table for the category c_i, shown in Table VI-B, as:

$$\textit{precision}_i^c = \frac{TP_i}{TP_i + FP_i}, \tag{15}$$

where FP_i (false positives for c_i) is the number of test documents that have been incorrectly categorized under c_i, TN_i (true negatives) is the number of test documents that have been correctly not categorized under c_i, TP_i (true positives) is the number of test documents that have been correctly categorized under c_i, and FN_i (false negatives) is the number of test documents that have been incorrectly not categorized under c_i.

Category c_i		Expert judgments YES	Expert judgments NO
Categorizer judgments	YES	TP_i	FP_i
	NO	FN_i	TN_i

Table 2: The contingency table for the category c_i.

The average of *precision*$_i^c$ can be computed in two different ways:

(i) Macroaveraging evaluates the average over the results for different categories:

$$macro\text{-}precision^c = \frac{\sum_{i=1}^{|C|} precision_i^c}{|C|}. \qquad (16)$$

(ii) Microaveraging evaluates the sum over all individual decisions in terms of the contingency table for the category c_i:

$$micro\text{-}precision^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)}. \qquad (17)$$

The larger the value of *macro-precision*^{c} and *micro-precision*^{c}, the better the performance of the categorizer. The performance is perfect when *macro-precision*$^c = 1$ and *micro-precision*$^c = 1$.
Figure 13 and Figure 14 show the categorizers' performance in terms of *macro-precision*^{c} and *micro-precision*^{c}, respectively, for EX100 and AT100 (the larger the better). Again, VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).

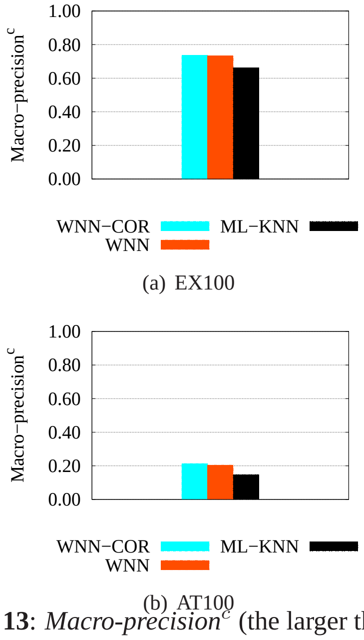


Figure. 13: *Macro-precision*^{c} (the larger the better)

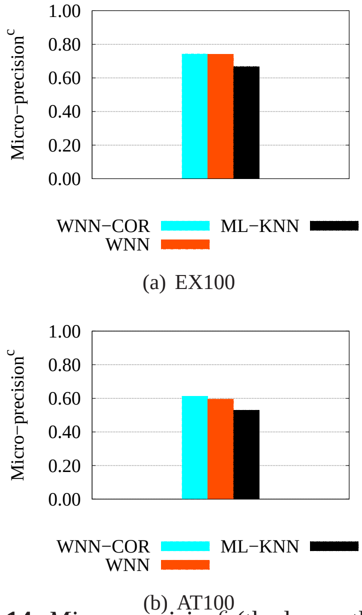


Figure. 14: *Micro-precision*^{c} (the larger the better)

***Recall on a per-category basis* (*recall*$_i^c$)** evaluates the fraction of test documents truly associated with the category c_i that are categorized under c_i, and can also be estimated using the contingency table for the category c_i shown in Table VI-B, as:

$$recall_i^c = \frac{TP_i}{TP_i + FN_i}. \qquad (18)$$

Estimates of *macro-recall*^{c} and *micro-recall*^{c} are calculated as:

$$macro\text{-}recall^c = \frac{\sum_{i=1}^{|C|} recall_i^c}{|C|}; \qquad (19)$$

$$micro\text{-}recall^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)}. \qquad (20)$$

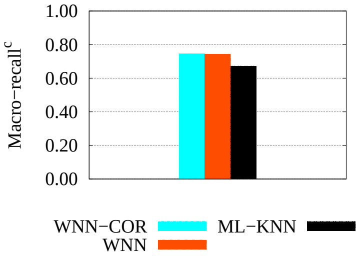
The larger the value of *macro-recall*^{c} and *micro-recall*^{c}, the better the performance of the categorizer. The performance is perfect when *macro-recall*$^c = 1$ and *micro-recall*$^c = 1$.

Figure 15 and Figure 16 show the categorizers' performance in terms of *macro-recall*^{c} and *micro-recall*^{c}, respectively, for EX100 and AT100 (the larger the better). VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). Both VG-RAM WNN-COR and VG-RAM WNN outperform ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).

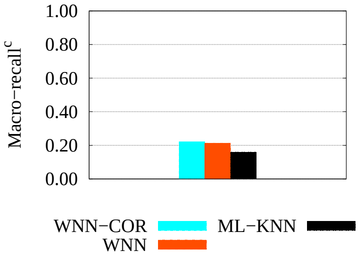
F_β *on a per-category basis* ($F_{\beta}{}_i^c$) evaluates the weighted harmonic mean of *precision*$_i^c$ and *recall*$_i^c$:

$$F_{\beta}{}_i^c = \frac{(\beta^2 + 1) precision_i^c \times recall_i^c}{\beta^2 precision_i^c + recall_i^c}. \qquad (21)$$

In this formula, β may be seen as the relative degree of importance attributed to *precision*$_i^c$ and *recall*$_i^c$ [16]. If $\beta = 0$ then $F_{\beta}{}_i^c$ coincides with *precision*$_i^c$, whereas if $\beta = +\infty$ then

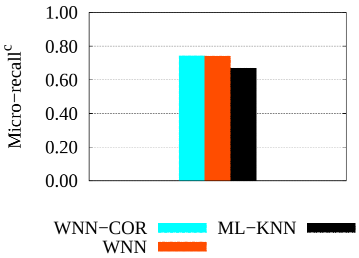


(a) EX100

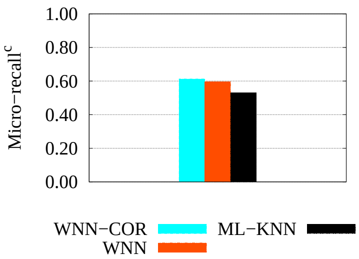


(b) AT100

Figure. 15: *Macro-recall*^{c} (the larger the better)



(a) EX100



(b) AT100

Figure. 16: *Micro-recall*^{c} (the larger the better)

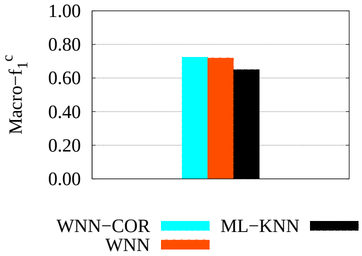
$F_{\beta}{}_i^c$ coincides with $recall_i^c$. Usually, a value $\beta = 1$ is used, which attributes equal importance to $precision_i^c$ and $recall_i^c$. Estimates of $macro\text{-}F_\beta{}^c$ and $micro\text{-}F_\beta{}^c$ are given by:

$$macro\text{-}F_\beta{}^c = \frac{1}{|C|} \sum_{i=1}^{|C|} F_{\beta}{}_i^c; \tag{22}$$

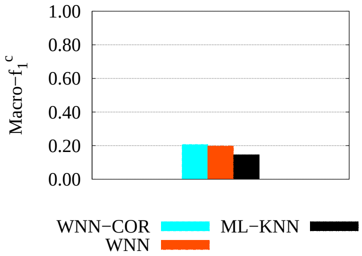
$$micro\text{-}F_\beta{}^c = \frac{(\beta^2+1)micro\text{-}precision^c \times micro\text{-}recall^c}{\beta^2 micro\text{-}precision^c + micro\text{-}recall^c}. \tag{23}$$

The larger the value of $macro\text{-}F_\beta{}^c$ and $micro\text{-}F_\beta{}^c$, the better the performance of the categorizer. The performance is perfect when $macro\text{-}F_\beta{}^c = 1$ and $micro\text{-}F_\beta{}^c = 1$.

Figure 17 and Figure 18 show the categorizers' performance in terms of $macro\text{-}F_1^c$ and $micro\text{-}F_1^c$, respectively, for EX100 and AT100 (the larger the better). VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100 (two-tailed paired t-test at 5% significance level). VG-RAM WNN-COR outperforms ML-KNN for EX100 and AT100 (two-tailed paired t-test at 5% significance level).

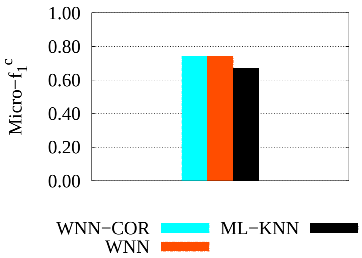


(a) EX100

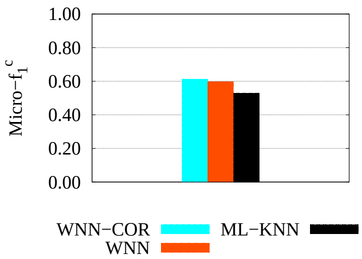


(b) AT100

Figure. 17: *Macro-*$F_1{}^c$ (the larger the better)



(a) EX100



(b) AT100

Figure. 18: *Micro-*$F_1{}^c$ (the larger the better)

Precision on a per-document basis **(*precision*$_j^d$)** evaluates the fraction of predicted categories that are pertinent for the test document d_j, and can be estimated in terms of the contingency table for d_j shown in Table VI-B as:

$$precision_j^d = \frac{TP_j}{TP_j + FP_j}, \tag{24}$$

where FP_j (false positives for d_j) is the number of categories that have been incorrectly predicted for d_j; and TN_j (true negatives), TP_j (true positives), and FN_j (false negatives) are defined accordingly.

Document d_j		Expert judgments	
		YES	NO
Categorizer judgments	YES	TP_j	FP_j
	NO	FN_j	TN_j

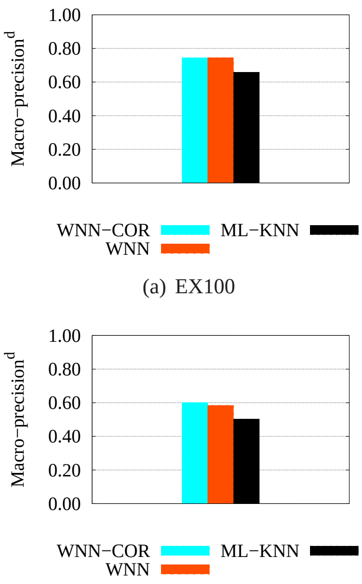
Table 3: The contingency table for the test document d_j.

The average of *precision*$_j^d$ can be computed in two different ways:

$$macro\text{-}precision^d = \frac{\sum_{j=1}^{|Te|} precision_j^d}{|Te|}; \tag{25}$$

$$micro\text{-}precision^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}(TP_j + FP_j)}. \tag{26}$$

The larger the value of *macro-precision*^{d} and *micro-precision*^{d}, the better the performance of the categorizer. The performance is perfect when *macro-precision*$^d = 1$ and *micro-precision*$^d = 1$. Figure 19 and Figure 20 show the categorizers' performance in terms of *macro-precision*^{d} and *micro-precision*^{d}, respectively, for EX100 and AT100 (the larger the better). VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100. VG-RAM WNN-COR outperforms ML-KNN for EX100 and AT100.



(b) AT100

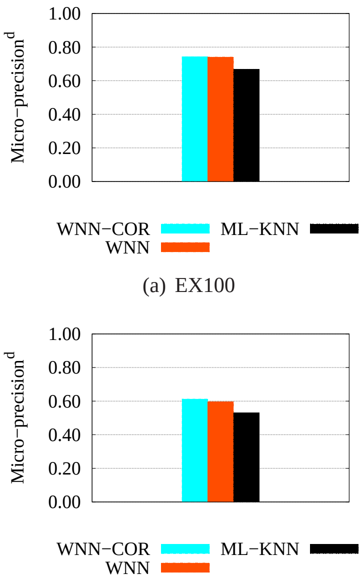
Figure. 19: *Macro-precision*^{d} (the larger the better)

Recall on a per-document basis (recall$_j^d$) evaluates the fraction of pertinent categories that are predicted for the test document d_j, and can also be estimated in terms of the contingency table for d_j shown in Table VI-B as:

$$recall_j^d = \frac{TP_j}{TP_j + FN_j}. \tag{27}$$

Estimates of *macro-recall*^{d} and *micro-recall*^{d} are calculated as:

$$macro\text{-}recall^d = \frac{\sum_{j=1}^{|Te|} recall_j^d}{|Te|}; \tag{28}$$

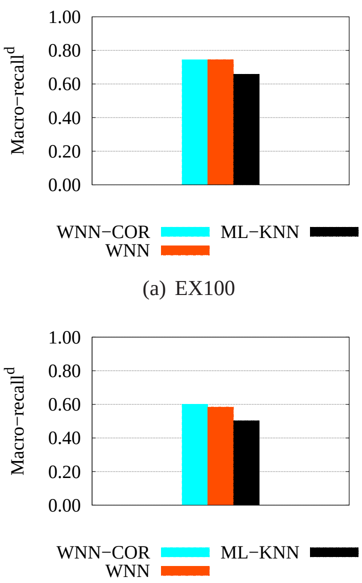


(b) AT100

Figure. 20: *Micro-precision*^{d} (the larger the better)

$$micro\text{-}recall^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}(TP_j + FN_j)}. \tag{29}$$

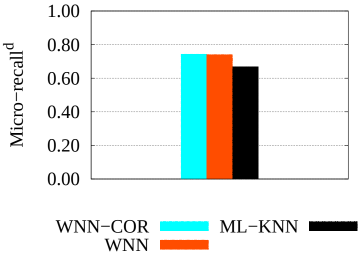
The larger the value of *macro-recall*^{d} and *micro-recall*^{d}, the better the performance of the categorizer. The performance is perfect when *macro-recall*$^d = 1$ and *micro-recall*$^d = 1$. Figure 21 and Figure 22 show the categorizers' performance in terms of *macro-recall*^{d} and *micro-recall*^{d}, respectively, for EX100 and AT100 (the larger the better). VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100. VG-RAM WNN-COR outperforms ML-KNN for EX100 and AT100.



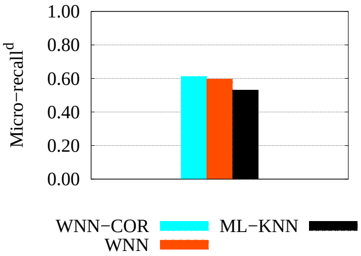
(b) AT100

Figure. 21: *Macro-recall*^{d} (the larger the better)

F_β on a per-document basis ($F_{\beta}{}_j^d$) evaluates the weighted

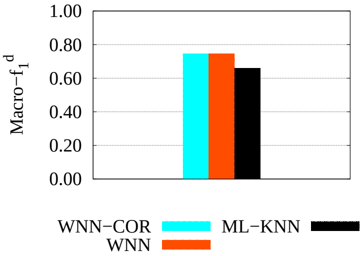


(a) EX100

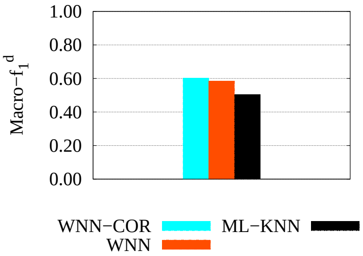


(b) AT100

Figure. 22: *Micro-recall*^{d} (the larger the better)



(a) EX100



(b) AT100

Figure. 23: *Macro-F*$_1{}^d$ (the larger the better)

harmonic mean of $precision_j^d$ and $recall_j^d$:

$$F_{\beta}{}_j^d = \frac{(\beta^2+1)precision_j^d \times recall_j^d}{\beta^2 precision_j^d + recall_j^d}. \tag{30}$$

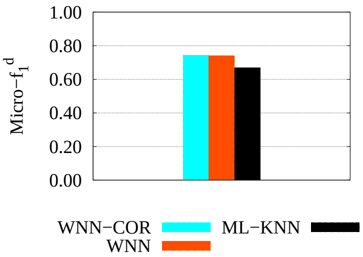
Estimates of *macro-F*$_\beta{}^d$ and *micro-F*$_\beta{}^d$ are given by:

$$macro\text{-}F_\beta{}^d = \frac{1}{|Te|}\sum_{j=1}^{|Te|} F_{\beta}{}_j^d; \tag{31}$$

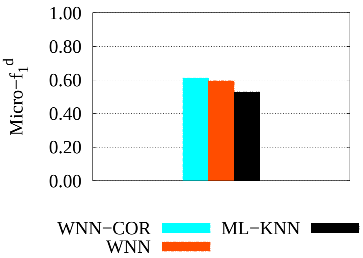
$$micro\text{-}F_\beta{}^d = \frac{(\beta^2+1)micro\text{-}precision^d \times micro\text{-}recall^d}{\beta^2 micro\text{-}precision^d + micro\text{-}recall^d}. \tag{32}$$

The larger the value of *macro-F*$_\beta{}^d$ and *micro-F*$_\beta{}^d$, the better the performance of the categorizer. The performance is perfect when *macro-F*$_\beta{}^d = 1$ and *micro-F*$_\beta{}^d = 1$.
Figure 23 and Figure 24 show the categorizers' performance in terms of *macro-F*$_1^d$ and *micro-F*$_1^d$, respectively, for EX100 and AT100 (the larger the better). VG-RAM WNN-COR presents the same performance of VG-RAM WNN for EX100, but outperforms it for AT100. VG-RAM WNN-COR outperforms ML-KNN for EX100 and AT100.
Note that the microaveraged metrics give an equal result, independently of being defined on a per-category basis or on a per-document basis. To understand why this is so, let $FP_{ij} = 1$ if the category c_i has been incorrectly predicted for the test document d_j, $FP_{ij} = 0$ otherwise; and $TP_{ij} = 1$ if c_i has been correctly predicted for d_j, $TP_{ij} = 0$ otherwise. Estimates of microaveraged precision on a per-category basis (*micro-precision*^{c}) and on a per-document basis (*micro-precision*^{d}) can be obtained, respectively, as:

$$\begin{aligned} micro\text{-}precision^c &= \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \\ &= \frac{\sum_{i=1}^{|C|}\sum_{j=1}^{|Te|} TP_{ij}}{\sum_{i=1}^{|C|}(\sum_{j=1}^{|Te|} TP_{ij} + \sum_{j=1}^{|Te|} FP_{ij})} \end{aligned} \tag{33}$$



(a) EX100



(b) AT100

Figure. 24: *Micro-F*$_1{}^d$ (the larger the better)

$$\begin{aligned} micro\text{-}precision^d &= \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}(TP_j + FP_j)} \\ &= \frac{\sum_{j=1}^{|Te|}\sum_{i=1}^{|C|} TP_{ij}}{\sum_{j=1}^{|Te|}(\sum_{i=1}^{|C|} TP_{ij} + \sum_{i=1}^{|C|} FP_{ij})} \end{aligned} \tag{34}$$

As one can observe in Equations (33) and (34), *micro-precision*^{c} is equal to *micro-precision*^{d}. Analogously, one can show that *micro-recall*^{c} and *micro-F*$_\beta{}^c$ are equal to *micro-recall*^{d} and *micro-F*$_\beta^c$, respectively.

C. Statistical T-Test

To present a clearer view of the relative performance of the algorithms, a partial order $\succ$ is defined on the set of all comparing algorithms 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 all comparing algorithms in terms of each evaluation metric for the EX100 and AT100 data sets is shown in Table 4 and Table 5, respectively.

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 performance assessment of an algorithm, we employed a score that takes into account its performance against that of the other algorithms 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 $>$ is defined on the set of all comparing algorithms, as shown in the last line of Table 4 and Table 5, where A1 $>$ A2 means that A1 performs better than A2 on the EX100 and AT100 data sets, respectively. The accumulated score of each algorithm is also shown in the parentheses. As shown in Table 4 and Table 5, VG-RAM WNN-COR has an overall better performance than VG-RAM WNN and ML-KNN on both the EX100 and AT100 databases for the set of metrics considered.

Evaluation metric	**WC $\times$ WN**	**WC $\times$ ML**	**WN $\times$ ML**
one-error	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
coverage	WC $\succ$ WN	WC $\equiv$ ML	WN $\equiv$ ML
ranking-loss	WC $\succ$ WN	WC $\equiv$ ML	WN $\equiv$ ML
avg-precision	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
R-precision	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
hamming-loss	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
exact-match	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-precision^c	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-precision^c	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-recall^c	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-recall^c	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-F_1^c	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-F_1^c	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-precision^d	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-precision^d	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-recall^d	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-recall^d	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-F_1^d	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-F_1^d	WC $\equiv$ WN	WC $\succ$ ML	WN $\succ$ ML
Total Order	**WC(**19**)** $>$ **WN(**15**)** $>$ **ML(**-34**)**		

Table 4: Results of t-test for EX100.

Evaluation metric	**WC $\times$ WN**	**WC $\times$ ML**	**WN $\times$ ML**
one-error	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
coverage	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
ranking-loss	WC $\succ$ WN	WC $\prec$ ML	WN $\prec$ ML
avg-precision	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
R-precision	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
hamming-loss	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
exact-match	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-precision^c	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-precision^c	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-recall^c	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-recall^c	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-F_1^c	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-F_1^c	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-precision^d	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-precision^d	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-recall^d	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-recall^d	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
macro-F_1^d	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
micro-F_1^d	WC $\succ$ WN	WC $\succ$ ML	WN $\succ$ ML
Total Order	**WC(**36**)** $>$ **WN(**-2**)** $>$ **ML(**-34**)**		

Table 5: Results of t-test for AT100.

VII. Conclusions

In this paper, we presented an experimental evaluation of Data Correlated VG-RAM WNN (VG-RAM WNN-COR) on multi-label text categorization and compared its performance with that of standard VG-RAM WNN and ML-KNN categorizers. In order to do that, we used two data sets composed 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 has an overall better performance than VG-RAM WNN and ML-KNN on the two databases for the set of metrics considered.

VIII. 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 de Apoio à Pesquisa do Estado do Espírito Santo*—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] 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, 2003.

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

[6] 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.

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

[8] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems*, volume 17, pages 649–656, 2005.

[9] 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.

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

[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] NILC. Diadorim: A lexical database for brazilian portuguese. `http://www.nilc.icmc.usp.br/nilc/tools/intermed.htm`, 2002.

[13] 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.

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

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

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

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

[18] Wikipedia, the free encyclopedia. Ranking. `http://en.wikipedia.org/wiki/Rank_order`, 2009.

[19] Y. Yang. A study of thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 137–145, 2001.

[20] 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.

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

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 Co-ordinator 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, and president of Steering Committee of the Vitória High Speed Metropolitan Area Network — METROVIX.

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

Bruno Zanetti Melotti received the B.Eng. degree in Computer Engineering from the Universidade Federal do Espírito Santo — UFES, Brazil, in 2006. Since 2007, he has been working with the Laboratório de Computação de Alto Desempenho — LCAD (High Performance Computing Laboratory) at UFES. Currently, he is pursuing his M.Sc. degree in Electrical Engineering at UFES. His research interests include information retrieval and data mining.

Dr. Claudine Badue is an Associate Researcher 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, data mining, and performance analysis and modeling. She has been involved in research projects financed through Brazilian research agencies, such as Conselho Nacional de Desenvolvimento Científico e Tecnológico — CNPq (Brazilian National Council for Scientific and Technological Development) and Fundação de Apoio à Pesquisa do Estado do Espírito Santo — FAPES (State of Espírito Santo Research Foundation). She has also been in the program committee and organizing committee of national and international conferences in Computer Science.