A new input representation for multi-label text classification

Rodrigo Alfaro
Departamento de Informática,
Universidad Técnica Federico Santa María
and Escuela de Ingeniería Informática,
Pontificia Universidad Católica de Valparaíso.
Valparaíso,Chile.
Email: rodrigo.alfaro@ucv.cl

Héctor Allende
Departamento de Informática,
Universidad Técnica Federico Santa María
and Facultad de Ingeniería,
Universidad Adolfo Ibáñez.
Valparaíso,Chile.
Email: hallende@inf.utfsm.cl

***Abstract*—Automatic text classification is the task of assigning unseen documents to a predefined set of classes or categories. Text Representation for classification have been traditionally approached with $tf.idf$ due to its simplicity and good performance. Multi-label automatic text classification has been traditionally tackled in the literature either by transforming the problem to apply binary techniques or by adapting binary algorithms to work with multiple labels. We present $tf.rfl$, a novel text representation for the multi-label classification approach. Our proposal focuses on modifying the data set input to the algorithm, differentiating the input by the label to evaluate. Performance of $tf.rfl$ was tested with a known benchmark and compared to alternative techniques. The results show improvement compared to alternative approaches in terms of Hamming Loss.**

***Keywords*-Multi-label, Text classification, Text representation, Machine learning.**

I. Introduction

Large amounts of digital text available on the web contain useful information for different purposes. The amount of digital text it is expected to increase significantly in the near future, making the development of data analysis applications an urgent need. Text classification (or categorization) is defined as the assignment of a Boolean value to each pair$\langle d_j, c_i\rangle \in \mathcal{D} \times \mathcal{C}$, where $\mathcal{D}$ is the domain of documents and $\mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$ is the set of predefined labels [1].

Binary classification is the most simple and widely studied case, in which a document is classified into one of two mutually exclusive categories or classes. The binary classification can be extended for solving multi-class problems. Moreover, if a document can be categorized with one label or multiple labels at once will be called Single-Label or Mult-Labels Problem [1].

Tsoumakas [2] presents a formal description of multi-label methods, in this paper, $L = \{\lambda_j : j = 1...q\}$ where λ_j correspond to the j-th label, is used to denote the finite set of labels in a multi-label learning task and $D = \{f(\mathbf{x_i}; Y_i); i = 1...m\}$ to denote a set of multi-label training data, where x_i is the feature vector and $Y_i \subseteq L$ the set of labels of the $i-$th example. The methods to solve this problem are grouped in two types: problem transformation and algorithm adaptation. The first type of methods are algorithm independent and transform the multi-label learning task into one or more single-label classification tasks. Thus, this type of methods can be implemented using efficient binary algorithms. The most common problem transformation method (PT4) learns $|L|$ binary classifiers $H_l : X \to \{l, \neg l\}$, one for each different label l in L. PT4 transforms the original data set into $|L|$ data sets $D_{l:l=1...|L|}$. Each D_l labels every example in D has l, if l is contained in the example, or $\neg l$ otherwise. PT4 gives the same solution as the single-label multi-class problem using a binary classifier. For the classification of a new instance x this method generates a set of labels as the union of the labels generated by the $|L|$ classifiers: $H_{PT4}(x) = \bigcup_{l \in L}\{l\} : H_l(x) = l$. The second type of methods extends specific learning algorithms for handling multi-label data directly. Extension is achieved by adjustments such as modifications to classical formulations of statistics or information theory. The pre-processing of documents for better representation can be considered also in this type.

Multi-label Classification is an important problem on real applications and it can be observed in many domains, such as functional genomics, text categorization, music mining, image classification and others.

The purpose of this paper is to present a new representation for documents based on label dependent term-weighting. This representation is a generalization of the $tf.rf$ representation applied to two class single-label classification problems as shown by Lan [3].

This paper is organized as follows. In section 2, we briefly introduce to Multi-label Text Classification. In section 3, we make an analysis of text representation. Our proposal of new representation is illustrated in section 4. In section 5 we compare the performance of $tf.rfl$ with other algorithms. The last section is devoted to concluding remarks.

II. Multi-label Text Classification

Automatic classification of multi-label text has not been thoroughly addressed in the literature. Although many multi-label data sets are available in the literature, most of the

techniques for automatic text classification consider them only as single-label data set. One of the first approaches developed was Boostexter, an algorithm based on Boosting for multi-label text [4]. From the categories of solution methods presented in [2], problem transformation is the most widely used. However, automatic classification of multi-label text has been solved also by algorithms that capture directly the characteristics of the multi-label problem. Zhang, for example, solved the multi-label problem using Artificial Neural Networks with multiple outputs [5].

Regardless of the solution approaches to the problem and the algorithms to solve it, according to Joachims [6] the text classification task has complexities of high-dimensional feature space, heterogeneous use of terms, and high level of redundancy. Multi-label problems have the additional complexities of large number of tags per document, existence of labels in 2 or 3 tier hierarchies, and that the same text can have more than 10 tags simultaneously. All this multi-label problem characteristics require different methods of evaluation and statistical tests than those used in traditional single-label problem.

III. Problem Representation

Performance of reasoning systems crucially depends on problem representation. The same task may be easy or difficult, depending on the way we describe it [7]. Explicit representation of important information enhances machine performance. Also, a more complex representation can work better with simpler algorithms.

Document representation has a high impact on the task of classification [8]. Some elements used for representing documents are: N-grams, single-word, phrases, or logical terms and statements. The vector space model is one of the most widely used models for ad-hoc information retrieval, mainly because of its conceptual simplicity and the appeal of the underlying metaphor of using spatial proximity for semantic proximity [9]. To solve the problem of how to weight terms in the vector space model, the frequency of occurrence of a word in a document, could be used as its term weight. However, there are more effective methods for term weighting. The basic information used in term weighting is term frequency, document frequency, or sometimes collection frequency. There are different mappings of text to input space in text classification. Leopold, for example, combines mappings with different kernel functions in Support Vector Machines [10].

In the vector space model (VSM), the contents of a document are represented by a vector in the term space: $d = \{w_1; ...; w_k\}$, where k is the size of the term (feature) set. Terms can be of several levels, such as syllables, words, phrases, or any other semantic and/or syntactic unit used to identify the content of a text. Different terms have different importance within a text, thus the relevance indicator w_i (usually between 0 and 1) represents how much the term t_i contributes to the semantics of the document d. According to Lan [11], two important decisions for choosing a representation based on VSM are: 1) What should be a term? For example sub-word, word, multi-word or meaning, and 2) How to weight a term? Term weighting can be binary, $tf.idf$ of Salton [12], using feature selection metrics such as χ^2, information gain (IG), or gain ratio (GR). Term weighting methods improve the effectiveness of the text classification by assigning appropriate weights to terms. Although text classification has been studied for several decades, the term weighting methods for text classification are usually borrowed from the traditional information retrieval (IR) field, for example, the boolean model, the $tf.idf$, and its various variants.

Table 1 shows the variables that we will consider in a term-weighting method for multi-label problems.

Table I
VARIABLES UTILIZED IN A TERM-WEIGHTING IN MULTI-LABEL PROBLEM FOR A TERM t WITH FOUR LABELS

	t	$\bar{t}$
$label_1$	a_{t1}	d_{t1}
$label_2$	a_{t2}	d_{t2}
$label_3$	a_{t3}	d_{t3}
$label_4$	a_{t4}	d_{t4}

Where:

- a_{ti} is the number of documents in the category i containing the term t,
- d_{ti} is the number of documents in the category i that do not contain the term t,

Bag-of-Words Representation. The most widely used document representation for text classification is $tf.idf$ [1], where for two class problem ($label_1$ is $class^+$ and $label_2$ is $class^-$) each dimension of the vector is computed as:

$$tf.idf_{td} = f_{td} * log\Big(\frac{N}{N_t}\Big) \tag{1}$$

In equation 1 f_{td} is the frequency of term t in the document d, $N = (a_{t1} + d_{t1} + a_{t2} + d_{t2})$ the number of documents, and $N_t = (a_{t1} + a_{t2})$ the number of documents containing the term t.

Relevance Frequency Representation. Recently, in [11] Lan proposed $tf.rf$, an improved VSM representation based on two classes single-label ($label_1$ is $class^+$ and $label_2$ is $class^-$) problem:

$$tf.rf_{td} = f_{td} * log_2\Big(2 + \frac{a_{t1}}{max\big(1, a_{t2}\big)}\Big) \tag{2}$$

Where f_{td} is the frequency of term t in the document d, a_{t1} is the number of documents in the positive category containing the term t, and a_{t2} is the number of documents in the negative category containing the term t.

According to Lan, using this representation in different single-label data sets improves the performance of two-class classifiers [11]. For multi-class problems, Lan used one-versus-all method.

Note that this representation is for single-label and does not consider frequency information of the term being evaluated in other classes or categories, it only considers the relationship of the appearance of the term in the class under evaluation (positive) versus all the other classes (negative).

IV. Our Proposal of a New Representation for Multi-Label

As it has been presented, on one hand, $tf.idf$ representation of documents, considers only the frequency of terms in the document (tf) and the frequency of terms in all documents (idf), disregarding the class or label to which the documents belong. On the other hand, $tf.rf$ also considers the frequency of terms in the document (tf) and the frequency of terms in all documents of the class evaluation (rf). That is, in $tf.rf$, each document will be represented by a different vector when assessing if it belongs to a class. From a theoretical point of view, this extension of the rf representation of text would: differentiate the representation of a document according to the label wanted to evaluate, achieving larger differences between documents belonging to different labels, reducing the dimension of the feature space according to the relevance for each label and harnessing the good performance of binary classifiers.

Then, $tf.rfl$ is composed of term frequency and relevance frequency for a label, is a new representation, based on $tf.rf$, for a multi-label problem.

$$tf.rfl_{tdl} = f_{td} * log_2\Big(2 + \frac{a_{tl}}{max\big(1, a_{tl}^{/i}\big)}\Big) \tag{3}$$

In equation 3, the term $a_{tl}^{/i}$ is the average number of documents containing the term t for each document labeled other that l:

$$a_{tl}^{/i} = \frac{1}{|L| - 1} \sum_{L_i \neq l} a_{tL_i} \tag{4}$$

where $|L|$ is the total number of labels.

Thus, the proposed term-weighting method includes information about the frequency of occurrence of a term t in each set of documents labeled other than the label being evaluated. It is expected that $a_{tl}^{/i}$ will be higher if the term t appears more frequently in documents with label l than in documents with others labels $l^{/i}$, and it will be lower if the term t is more frequent in documents with labels other than l.

Our proposal is based on the $tf.rfl$ representation and the SVM binary ensemble, and it comprehends: transforming the problem to PT4 form [2], then for each document d, building the $tf.rfl$ representation for each label l, and classifying using l binary classifiers.

V. Experiments

Testing of the proposed $tf.rfl$ was done using the Reuters-21578 Distribution 1.09. The Reuters-21578 data set consists of 21,578 Reuters newswire documents that appeared in 1987, where less than half of the documents have human-assigned topic labels. The data set used and the validation mechanism are the same as used in [5], i.e. subsets of the k categories with the largest number of articles for $k = 3, ..., 9$ are selected resulting in seven different data sets denoted as First3, First4, ..., First9. Also, 10-fold cross validation is performed on each data set. Our classification method reports the average values of three runs. Table 2 presents the data set characteristics.

Table II
Characteristics of the pre-processed data set summarizes.

Data Set	Number of Categories	Number of Documents	Vocabulary Size
First3	3	7,258	529
First4	4	8,078	598
First5	5	8,655	651
First6	6	8,817	663
First7	7	9,021	677
First8	8	9,158	683
First9	9	9,190	686

First, the original problem is transformed to the PT4 form, dividing into 9 input data sets for nine binary classifiers, where each machine work one-against-others labels. Three representations were constructed for the data set, the classical $tf.idf$, $tf.rf$ and our proposal $tf.rfl$. An ensemble of binary SVM classifiers was used. Each machine considered a linear kernel and its other parameters were optimized in terms of maximizing the classification margin between each pair of classes. The ensemble was implemented with LibSVM [13], where each machine worked with random sampling , 2/3 examples for training and 1/3 for testing. Note that all $tf.ifd$ representations are the same, regardless of the label wanted to evaluate, while $tf.rfl$ representations are different for each label.

Multi-label classification methods requires different performance metrics than those used in traditional single-label classification methods, measures that have been proposed in the past can be grouped in to bipartitions and rankings [14]. As in [4] and [5], evaluation of the results in this research was performed using Hamming Loss, considering bipartition, which evaluates how many times an instance-label pair is misclassified.

$$hloss(h) = \frac{1}{p} \sum_{i=1}^{p} \frac{1}{Q} |h(\mathbf{x_i}) \Delta Y_i| \tag{5}$$

Performance is better when $hloss(h)$ is near 0.

In this metric, for fewer categories Boos-Texter is better than $tf.rfl$. For more categories (First5, First6, First7, First8 and First9 data sets) $tf.rfl$ is better than all others algorithms. Table 3 shows these results.

Table III
EXPERIMENTAL RESULTS OF SVM ENSEMBLES WITH $tf.idf$, $tf.rf$ AND $tf.rfl$ COMPARED WITH OTHERS LEARNING ALGORITHMS IN TERMS OF HAMMING LOSS. BP-MLL* AND BOOSTEXTER* AS REPORTED BY [5].

Data set	SVM Ens tf.idf	SVM Ens tf.rf	**SVM Ens tf.rfl**	Bp-MLL*	Boos-Texter*
First3	0.02797	0.02814	0.02716	0.0368	**0.0236**
First4	0.02641	0.02687	0.02590	0.0256	**0.0250**
First5	0.02590	0.02611	**0.02526**	0.0257	0.0260
First6	0.02477	0.02522	**0.02412**	0.0271	0.0262
First7	0.02246	0.02287	**0.02186**	0.0252	0.0249
First8	0.02083	0.02118	**0.02026**	0.0230	0.0229
First9	0.01981	0.02012	**0.01930**	0.0231	0.0226
Average	0.02402	0.02436	**0.02341**	0.02664	0.02446

To evaluate the results developed a test based on two-tailed paired t-test at 5 percent significance level. According to these results SVM Ens $tf.rfl$ is better than SVM Ens $tf.idf$ (4.2595×10^{-6}), SVM Ens $tf.rf$ (2.0376×10^{-7}) and Bp-MLL (3.74×10^{-2}). Where the p-value show in the parentheses further gives a quantification of the significance level. The results show improvement statistically significant compared to alternative approaches in terms Hamming Loss.

VI. REMARKS AND CONCLUSIONS

Multi-label Classification is an important and increasingly developing field of Information Retrieval and Machine Learning. Text Representation and classification have been traditionally approached with $tf.idf$ due to its simplicity and good performance. Changes in input representation can use knowledge about the problem, a label, or class to which the document belongs. Other representations could be developed for overcoming the problem directly and without problem transformations. New benchmarks should be used for validating the results, however, the preprocessing of multi labeled texts must be standardized.

In this paper we have presented a novel text representation for the multi-label classification approach. This representation considers the label to which the document belongs. This is a combination between problem transformation and algorithm adaptation. The performance of this representation was tested in combination with an ensemble of SVM over a known benchmark. The results show improvement statistically significant compared to alternative approaches in terms Hamming Loss. We believe that the contribution of the proposed multi-label representation lies in terms of better natural understanding of the problem. For future work, we plan to compare ours to others $tf.idf$ variation representations and to investigate other label dependent representations and procedures for reducing the dimensional feature space, according to the relevance for each label.

REFERENCES

[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[2] G. Tsoumakas and I. Katakis, "Multi label classification: An overview," *International Journal of Data Warehouse and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[3] M. Lan, C.-L. Tan, and H.-B. Low, "Proposing a new term weighting scheme for text categorization," in *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*. AAAI Press, 2006, pp. 763–768.

[4] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," in *Machine Learning*, 2000, pp. 135–168.

[5] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge Data Engineering.*, vol. 18, no. 10, pp. 1338–1351, 2006.

[6] T. Joachims, *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer-Springer, 2002.

[7] E. Fink, "Automatic evaluation and selection of problem-solving methods: Theory and experiments," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 16(2), pp. 73–105, 2004.

[8] M. Keikha, N. Razavian, F. Oroumchian, and H. S. Razi, "Document representation and quality of text: An analysis." in *In Survey of Text Mining II: Clustering, Classifcation, and Retrieval*. Springer-Verlag, London, 2008, pp. 135–168.

[9] C. Manning and H. Schtze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[10] E. Leopold and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?" *Machine Learning*, vol. 46, no. 1-3, pp. 423–444, January 2002.

[11] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 721–735, 2009.

[12] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management: an International Journal*, vol. 24, no. 5, pp. 513–523, 1988.

[13] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[14] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook, 2nd edition*. O. Maimon, L. Rokach (Ed.), Springer, 2010.