USING ONTOLOGIES TO IMPROVE DOCUMENT CLASSIFICATION WITH TRANSDUCTIVE SUPPORT VECTOR MACHINES

Roxana Aparicio[1] and Edgar Acuna[2]

[1]Institute of Statistics and Computer Information Systems, University of Puerto Rico, Rio Piedras Campus, Puerto Rico
roxana.aparicio@upr.edu
[2]Department of Mathematical Sciences, University of Puerto Rico, Mayaguez Campus, Puerto Rico
edgar.acuna@upr.edu

ABSTRACT

Many applications of automatic document classification require learning accurately with little training data. The semi-supervised classification technique uses labeled and unlabeled data for training. This technique has shown to be effective in some cases; however, the use of unlabeled data is not always beneficial.

On the other hand, the emergence of web technologies has originated the collaborative development of ontologies. In this paper, we propose the use of ontologies in order to improve the accuracy and efficiency of the semi-supervised document classification.

We used support vector machines, which is one of the most effective algorithms that have been studied for text. Our algorithm enhances the performance of transductive support vector machines through the use of ontologies. We report experimental results applying our algorithm to three different datasets. Our experiments show an increment of accuracy of 4% on average and up to 20%, in comparison with the traditional semi-supervised model.

KEYWORDS

Semi-supervised Document Classification, Text Mining, Support Vector Machines, Ontologies

1. INTRODUCTION

Automatic document classification has become an important subject due the proliferation of electronic text documents in the last years. This problem consists in learn to classify unseen documents into previously defined categories. The importance of make an automatic document classification is evident in many practical applications: Email filtering [1], online news filtering [2], web log classification [3], social media analytics [4], etc.

Supervised learning methods construct a classifier with a training set of documents. This classifier could be seen as a function that is used for classifying future documents into previously defined categories. Supervised text classification algorithms have been successfully used in a wide variety of practical domains. In experiments conducted by Namburú et al. [5], using high accuracy classifiers with the most widely used document datasets, they report up to 96% of accuracy with a binary classification in the Reuters dataset. However, they needed 2000 manually labeled documents to achieve this good result [5].

DOI : 10.5121/ijdkp.2013.3301

The problem with supervised learning methods is that they require a large number of labeled training examples to learn accurately. Manual labeling is a costly and time-consuming process, since it requires human effort. On the other hand, there exists many unlabeled documents readily available, and it has been proved that in the document classification context, unlabeled documents are valuable and very helpful in the classification task [6].

The use of unlabeled documents in order to assist the text classification task has been successfully used in numerous researches [7], [8], [9], [10]. This process has received the name of semi-supervised learning. In experiments conducted by Nigam, on the 20 Newsgroups dataset, the semi-supervised algorithm performed well even with a very small number of labeled documents [9]. With only 20 labeled documents and 10,000 unlabeled documents, the accuracy of the semi-supervised algorithm was 5% superior than the supervised algorithm using the same amount of labeled documents.

Unfortunately, semi-supervised classification does not work well in all cases. In the experiments found in literature some methods perform better than others and for distinct datasets the performance differs [5]. There are some datasets that do not benefit from unlabeled data or even worst, sometimes, unlabeled data decrease performance. Nigam [9] suggests two improvements to the probabilistic model in which he tries to contemplate the hierarchical characteristics of some datasets.

Simultaneously, with the advances of web technologies, ontologies have increased on the World-Wide Web. Ontologies represent shared knowledge as a set of concepts within a domain, and the relationships between those concepts. The ontologies on the Web range from large taxonomies categorizing Web sites to categorizations of products for sale and their features. They can be used to reason about the entities within that domain, and may be used to describe the domain. In this work we propose the use of ontologies in order to assist the semi-supervised classification.

2. MOTIVATION

In certain applications, the learner can generalize well using little training data. Even when it is proved that, for the case of document classification, unlabeled data could improve efficiency. However, the use of unlabeled data is not always beneficial, and in some cases it decreases performance.

Ontologies provide another source of information, which, with little cost, helps to attain good results when using unlabeled data. The kind of ontologies that we focus in this work give us the words we expect to find in documents of a particular class.

Using this information we could guide the direction of the use of unlabeled data, respecting the particular method rules. We just use the information provided by the ontologies when the learner needs to make a decision, and we give the most probable label when otherwise arbitrary decision is to be made.

The advantages of using ontologies are twofold:

- They are easy to get since they are either readily available or they could be built with little cost.
- Improve the time performance of the algorithm by speeding up convergence.

3. THEORETICAL BACKGROUND

3.1. Support Vector Machines

The learning method of Support Vector Machines (SVM) was introduced by Vladimir Vapnik et al [11]. Supervised support vector machine technique has been successfully used in text domains [12].

Support Vector Machines is a system for efficiently training linear learning machines in kernel-induced feature spaces. Linear learning machines are learning machines that form linear combinations of the input variables [13]. The formulation of SVM is as follows:

Given a training set $S = \{(\boldsymbol{x_i}, \boldsymbol{y_i}); i = 1,2, \dots, m\}$, that is linearly separable in the feature space implicitly defined by the kernel $\mathrm{K}(\mathbf{x}, \mathbf{z})$ and suppose the parameters $\boldsymbol{\alpha}^*$ and b^* solve the following quadratic optimization problem:

$$\begin{aligned} maximize_{w,b} \;\; & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x_i}.\boldsymbol{x_j}) \\ \text{s.t.} \;\; & \sum_{i=1}^{m} \alpha_i y_i = 0 \\ & \alpha_i \geq 0, i = 1, \dots, m \end{aligned} \qquad \textbf{(0-1)}$$

Then the decision rule given by $sgn(f(x)$,where $f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i{}^* y_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b^*$ is equivalent to the maximal margin hyperplane in the feature space implicitly defined by the kernel $\mathrm{K}(\mathbf{x}, \mathbf{z})$ and that hyperplane has geometric margin

$$\gamma = \left(\sum_{j \in SV} \alpha_j{}^* \right)^{-\frac{1}{2}}.$$

3.2. Transductive Support Vector Machines

Transductive learning refers to the estimation of the class of the unlabeled working set. In contrast with the inductive approach where the learner induces a function with low error rate; transductive learning aims to classify a given set of unlabeled examples with as few errors as possible. The most representative technique of transductive learning is Transductive Support Vector Machines (TSVM). It was introduced by Joachims [8] with particular application in Text Classification.

TSVM maximizes margin not only on the training, but also on the test set. For transductive learning, the learner L receives as input a training set $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots (\boldsymbol{x}_l, y_l)$ and a test set T=$\{\boldsymbol{x_1^*}, \boldsymbol{x_2^*}, \dots, \boldsymbol{x_k^*}\}$ (from the same distribution) [8].

Including T=$\{\boldsymbol{x_1^*}, \boldsymbol{x_2^*}, \dots, \boldsymbol{x_k^*}\}$, the corresponding labeling $y_1^*, y_2^*, \dots, y_k^*$ to be found and the slack variables $\xi_j^*,\ j = 1, \dots \mathrm{k}$ for the unlabeled data in the derivation, we arrive to the following optimization problem for the non-separable case:

$$min: W(y_1^*, y_2^*, \dots, y_k^*, \boldsymbol{w}, b, \xi_1, \dots, \xi_k^*) = \frac{1}{2}\|w\| + C\sum_{i=0}^{n} \xi_i + C^* \sum_{j=0}^{k} \xi_j^*$$

s.t.

$$(w.x_i + b) \geq 1 - \xi_i \qquad \xi_i \geq 0, i = 1, \dots n \qquad \text{0-2}$$

$$y_j^*\left(w.x_j^* + b\right) \geq 1 - \xi_j^* \qquad \xi_j^* \geq 0, j = 1, \ldots \mathrm{k}$$
$$y_j^* \in \{-1,1\} \qquad j = 1, \ldots \mathrm{k}$$

C and C^* are parameters set by the user. They allow trading off margin size against misclassifying training examples or excluding test examples.

Solving this problem means finding a labeling $y_1^*, y_2^*, \ldots, y_k^*$ of the test data and a hyperplane $\langle \boldsymbol{w}, b \rangle$, so that this hyperplane separates both training and test data with maximum margin.

For a very small number of test examples, this problem can be solved simply by trying all possible assignments of $y_1^*, y_2^*, \ldots, y_k^*$ to the two classes. However, this approach becomes intractable for large test sets. Joachims [14], proposed an algorithm that repeatedly optimize approximations to the TSVM training problem using local search. Local search algorithms start with some initial instantiation of the variables. In each iteration, the current variable instantiation is modified so that it moves closer to a solution. This process is iterated until no more improvement is possible.

3.3. Ontologies

The evolution of the web has originated new forms of information sharing. The continuous growing of information in the WWW makes the existence of explicit semantics that supports machine processing of information necessary. The concept of Ontologies was originated in the Artificial Intelligence community as an effort to formalize descriptions of particular domains.

The term 'ontology' in the context of information management is defined as a formal, explicit specification of a shared conceptualization [15]. A conceptualization refers to an abstract model of some phenomenon in the world which identifies the relevant concepts, relations and constraints. These concepts, relations and constraints must be explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Finally, an ontology represents shared knowledge, that is a common understanding of the domain between several parties. Ontologies enable semantic mapping between information sources [16].

In other words, ontology specifies a domain theory. It is a formal description of concepts and their relations, together with constraints on those concepts and relations [17].

There are many different types of ontologies. Typically they are classified according to their expressiveness. Alexiev et. al. [17] classifiy ontologies into two groups:

1. Light-weight ontologies. The ontologies in this group are those with the lower level of expressiveness, they are:
   - controled vocabulary: a list of terms.
   - thesaurus: relation between terms are provided.
   - informal taxonomy: there is an explicit hierarchy, but there is not strict inheritance.
   - formal taxonomy: there is strict inheritance.
   - frames or classes: a frame (or class) contains a number of properties and these properties are inherited by subclasses and instances.
2. Heavy-weight ontologies.
   - value restrictions: values of properties are restricted.
   - general logic constraints: values may be constraint by logical or mathematical formulas.
   - first-order logic constraints: very expressive ontology languages that allow first-order logic constraints between terms and more detailed relationships such as disjoint classes, disjoint coverings, inverse relationships, part-whole relationships, etc. feasible for any optimal w and b. Likewise if $d_j = 1$ then $z_j = 0$.

4. USING ONTOLOGIES WITH TSVM

4.1 Extracting information from ontologies

Given a collection of ontologies related to the classification problem, we build a vector of the form:

$$v_{ont} = (\boldsymbol{x_i}, y_i), i = 1, \dots, c$$

where c is the number of classes in the classification problem. Each $\boldsymbol{x_i}$ is a vector of words that are known to be good discriminator for class y_i.

Additionally, to each word could be associated a weight w_j that corresponds to the importance of word x_j in discriminating its corresponding class.

In this work, we focus in binary classification, hence the set of classes is $\{1, -1\}$. Our proposal is to incorporate the ontologies in the algorithm, in order to use this information to help make the decision of which unlabeled examples are worth switch to labeled samples. We use the information of a probabilistic label given to each unlabeled document by the ontologies. The intention is not to push too hard to conform strictly to the ontologies, but use them as a piece of information at that point of the algorithm.

In order to use the information provided by the ontology, we first assign to each unlabeled document d^* a probabilistic label z induced by the ontologies.

Let $d = (w_1, w_2, \dots, w_k)$ be a document, and let $\{(a_1, a_2, \dots, a_p, +1), (b_1, b_2, \dots, b_n, -1)\}$ be the ontology for a binary problem. Then we assign to the unlabeled document d a label y_{ont}using the following rule:

$$y_{ont} = \underset{i}{\text{argmax}}\left(\sum_{w_j \in v_{ont_i}} w_j\right)$$

The algorithm for assigning the labels induced by ontologies is shown in Figure 1.

ALGORITHM ONTOLOGY_LABEL

Input: training examples $\boldsymbol{x_1^*}, \dots, \boldsymbol{x_k^*}$
ontology $\{(\boldsymbol{a_1}, \boldsymbol{a_2}, \dots, \boldsymbol{a_p}, +1), (\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_n}, -1)\}$

Output: probabilistic labels of the test examples $z_1^*, \dots, z_k^*$

//Words are ordered by id in both document vectors and ontology vectors
// Repeat for all unlabeled examples

```
for (i = 0, i < k, i++){
        posweight := 0
        negweight := 0
        for (j = 0, j < p, j++){
                if (a_j ∈ x_i^*) { posweight := posweight + weight(x ∈ x_i^* | x = a_j)}
        }
        for (j = 0, j < n, j++){
                if (b_j ∈ x_i^*) { negweight := negweight + weight(x ∈ x_i^* | x = b_j)}
        }
        y_ont_i := max (posweight, negweight)
        If (posweight < negweight) y_ont_i := −y_ont_i;
}
```

Figure 1 Algorithm for calculating the label induced by the Ontology

4.2 Incorporating ontologies to TSVM

For weighted ontologies, we just multiply corresponding weights of document word and ontology word.

Once we have obtained the probabilistic label for all unlabeled documents, we can make the following modification in the transductive approach presented by Joachims [8]:

ALGORITHM TSVM

Input: labeled examples $(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)$

unlabeled examples $\boldsymbol{x_1^*}, \dots, \boldsymbol{x_k^*}$

labels induced by ontologies $\boldsymbol{z_1^*}, \dots, \boldsymbol{z_k^*}$ for unlabeled documents

Output: Predicted labels of the unlabeled examples $y_1^*, \dots, y_k^*$

1. Train an inductive SVM M_1using the labeled data $(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)$.
2. Classify unlabeled documents $\boldsymbol{x_1^*}, \dots, \boldsymbol{x_k^*}$ using M_1
3. Loop1: While there exist unlabeled documents
    1. Increase the influence of unlabeled data by incrementing the cost factors (parameters in the algorithm)
    2. Loop 2: While there exist unlabeled examples that do not meet the restriction of the optimization problem
       Select unlabeled examples to switch given that are misclassified according to the ontology induced label $\boldsymbol{z_1^*}, \dots, \boldsymbol{z_k^*}$
        1. Retrain

Return labels $y_1^*, \dots, y_k^*$ for unlabeled documents

Figure 2 Algorithm for training transductive support vector machines using ontologies.

4.3 Time Complexity of the Algorithm

Using the sparse vector representation the time complexity of the dot products depend only on the number of non-zero entries.

Let *m* the maximum number of non-zero entries in any of the training examples, let *q* be the rows of the Hessian. For each iteration, most time is spent on the Kernel evaluations needed to compute the Hessian. Since we used a linear Kernel, this step has time complexity $O(q^2m)$.

5. EXPERIMENTAL EVALUATION

5.1 Datasets

We used three well known data sets among researchers in text mining and information retrieval. These datasets are the following:

1. Reuters-1 (RCV1)

This data set is described in detail by Lewis et. al. [18]. We randomly selected a portion of documents from the most populated categories. The quantity of selected documents is proportional to the total amount of documents in each category. In Table 1, we show the quantity of selected documents, for each category. For the negative class of each category, we randomly selected the same amount of documents from the other categories.

Table 1 Number of labeled and unlabeled documents used in experiments for 10 categories of Reuters dataset.

CATEGORY	LABELED	UNLABELED	TOTAL
Accounts/earnings	1325	25069	26394
Equity markets	1048	20296	21344
Mergers/acquisitions	960	18430	19390
Sports	813	15260	16073
Domestic politics	582	11291	11873
War, civil war	1001	17652	18653
Crime, law enforcement	466	7205	7671
Labour issues	230	6396	6626
Metals trading	505	9025	9530
Monetary/economic	533	5663	6196

2. 20 Newsgroups

The 20 Newsgroups data set was collected by Ken Lang, consists of 20017 articles divided almost evenly among 20 different UseNet discussion groups.

This data set is available from many online data archives such as CMU Machine Learning Repository [19] .

For our experiments we used 10000 documents corresponding to 10 categories. For each class we used 100 labeled documents and 900 unlabeled documents.

3. WebKB

The WebKB data set described at [20], it contains 8145 web pages gathered from universities computer science departments. The collection includes the entirety of four departments, and additionally, an assortment of pages from other universities. The pages are divided into seven categories: student, faculty, staff, course, project, department and other.

In this work, we used the four most populous categories (excluding the category other): student, faculty, course and project. A total of 4199 pages, distributed as shown in Table 2:

Table 2 Number of labeled and unlabeled documents used in experiments for WebKB dataset.

CATEGORY	LABELED	UNLABELED	TOTAL
Course	93	837	930
Department	18	164	182
Faculty	112	1012	1124
Student	164	1477	1641

5.2 Performance measures

In order to evaluate and compare the classifiers, we used the most common performance measures, which we describe below. The estimators for these measures can be defined based on the following contingency table:

Table 3 Contingency table for binary classification.

	LABEL $y = +1$	**LABEL** $y = -1$
Prediction $\boldsymbol{f(x) = +1}$	PP	PN
Prediction $\boldsymbol{f(x) = -1}$	NP	NN

Each cell of the table represents one of the four possible outcomes of a prediction $f(\boldsymbol{x})$ for an example $(\boldsymbol{x}, y)$.

5.2.1 Error rate and Accuracy

Error rate is probability that the classification function f predicts the wrong class.

$$Err(f) = \Pr\left(f(\boldsymbol{x}) \neq y | f\right)$$

It can be estimated as:

$$Err(f) = \frac{PN + NP}{PP + PN + NP + NN}$$

Accuracy measures the ratio of correct predictions to the total number of cases evaluated.

$$A(f) = \frac{PP + NN}{PP + PN + NP + NN}$$

5.2.2 Precision / Recall breakeven point and Fβ-Measure

Recall is defined as the probability that a document with label $y = 1$ is classified correctly. It could be estimated as follows:

$$Rec_{test}(f) = \frac{PP}{PP + NP}$$

Precision is defined as the probability that a document classified as $f(x) = 1$ is classified correctly. It could be estimated as follows

$$Prec_{test}(f) = \frac{PP}{PP + PN}$$

Precision and recall are combined to give a single measure, to make it easier to compare learning algorithms.

Fβ-Measure is the weighted harmonic mean of precision and recall. It can be estimated from the contingency table as:

$$F_{\beta,test}(f) = \frac{(1+\beta^2)PP}{(1+\beta^2)PP + PN + \beta^2 NP}$$

Precision / Recall breakeven point (PRBEP) is the value at which precision and recall are equal. β is a parameter. The most commonly used value is $\beta = 1$, giving equal weight to precision and recall.

5.3 Experimental results

The experiments evaluate the quality and efficiency of the algorithm.

For Twenty newsgroups dataset, the experiments are shown in Table 4, for selected 10 categories. Each category consists of 2000 examples from which 10 percent are labeled documents. In this table we can see an improvement with respect to the TSVM in the accuracy for three categories. The highest improvement is reached for category soc.religion.christian.

Table 4 Accuracy of TSVM y TSVM + ontologies for ten categories of Twenty Newsgroups.

Category	TSVM	TSVM+ont	GAIN
alt.atheism	81.25	88.12	6.87
comp.graphics	93.67	94.3	0.63
misc.forsale	89.38	94.38	5
rec.autos	77.36	76.1	-1.26
rec.motorcycles	74.68	74.68	0
sci.electronics	66.88	66.88	0
sci.med	75.32	74.68	-0.64
soc.religion.christian	73.58	94.34	20.76
talk.politics.guns	97.45	97.45	0
rec.sport.baseball	86.16	86.16	0

Table 5 Precision and Recall of TSVM y TSVM + ontologies for ten categories of Twenty Newsgroups.

Category	TSVM	TSVM+ont
alt.atheism	71.15%/100.00%	80.90%/97.30%
comp.graphics	88.51%/100.00%	89.53%/100.00%
misc.forsale	82.61%/98.70%	91.46%/97.40%
rec.autos	96.30%/60.47%	96.15%/58.14%
rec.motorcycles	96.08%/56.32%	96.08%/56.32%
sci.electronics	90.91%/44.94%	90.91%/44.94%
sci.med	91.07%/60.00%	90.91%/58.82%
soc.religion.christian	62.73%/98.57%	89.61%/98.57%
talk.politics.guns	96.25%/98.72%	96.25%/98.72%
rec.sport.baseball	100.00%/73.81%	100.00%/73.81%

Table 5 shows the values of precision and recall for the same dataset. In this table we note that precision improves in all cases in which accuracy has been improved by the use of ontologies.

We also note that in two cases there has been a little lost in accuracy by the use of ontologies. We conjecture that the reason is that the selected ontologies might not agree with the manual labeling.

For Web-Kb dataset, the experiments are shown in Table 6, for the four categories that are commonly used by researchers [6], [14]. We use all the available documents for each category. Ten percent of the documents were labeled and the rest was selected as unlabeled documents. In Table 6 we can see an improvement in the accuracy for three categories. Table 7 shows the precision and recall measures for Web-Kb dataset. This table shows an increment in precision even in the category in which ontologies do not report an improvement in comparison with TSVM.

Table 6 Accuracy of TSVM y TSVM + ontologies for 4 categories of Web-Kb dataset.

Category	TSVM	TSV+ont	Gain
Course	96.5	96.84	0.34
Department	93.48	94.6	1.12
Faculty	85.29	84.8	-0.49
Student	83.94	84.34	0.4

Table 7 Precision and Recall of TSVM y TSVM + ontologies for 4 categories of Web-Kb dataset.

Category	TSVM	TSV+ont
Course	97.05%/98.77%	97.70%/98.50%
Department	74.85%/88.65%	81.63%/85.11
Faculty	90.22%/73.13%	90.96%/71.09%
Student	86.20%/86.79%	87.66%/85.65%

Table 8 Accuracy of TSVM y TSVM + ontologies for 10 categories of Reuters dataset.

Category	TSVM	TSV+ont	Gain
Accounts/earnings	96.30	96.45	0.15
Equity markets	92.5	93.7	1.20
Mergers/acquisitions	96.2	96.4	0.20
Sports	96.46	96.46	0.00
Domestic politics	83.4	83.9	0.50
War, civil war	94.06	95.98	1.92
Crime, law enforcement	92.7	95.14	2.44
Labour issues	85.5	87.15	1.65
Metals trading	96.20	97.48	1.28
Monetary/economic	85.2	89.7	4.50

The third set of experiments corresponds to Reuters dataset, and are shown in Table 8. We selected a sample for the ten most populated categories. In this table we can see an improvement in the accuracy in nine of the ten selected categories. There is no lost reported in any of the categories.

Table 9 shows the corresponding precision and recall measures for this experiment. We note again an increment in precision for all categories. With this dataset it was easier to find related ontologies since categories are well defined. This might be the reason why ontologies were beneficial in nine categories and had no effect in just one category.

Table 9 Precision and Recall of TSVM y TSVM + ontologies for 10 categories of Reuters dataset.

Category	TSVM	TSV+ont
Accounts/earnings	96.30%/96.30%	97.16%/95.70%
Equity markets	92.50%/92.50%	93.71%/92.20%
Mergers/acquisitions	96.20%/96.20%	97.74%/95.00%
Sports	100.00%/94.06%	100.00%/94.06%
Domestic politics	83.40%/83.40%	85.61%/81.50%
War, civil war	89.11%/99.96%	92.37%/99.95%
Crime, law enforcement	89.20%/99.99%	92.54%/100.00%
Labour issues	85.50%/85.50%	86.88%/88.10%
Metals trading	96.20%/96.20%	99.96%/92.09%
Monetary/economic	85.20%/85.20%	95.21%/81.80%

5.3.1 Influence of the ontologies

Figure 3 shows the effect of using ontologies for class soc.religion.christian of Twenty Newsgroups dataset. For a total of 2000 documents, we vary the size of the labeled documents.

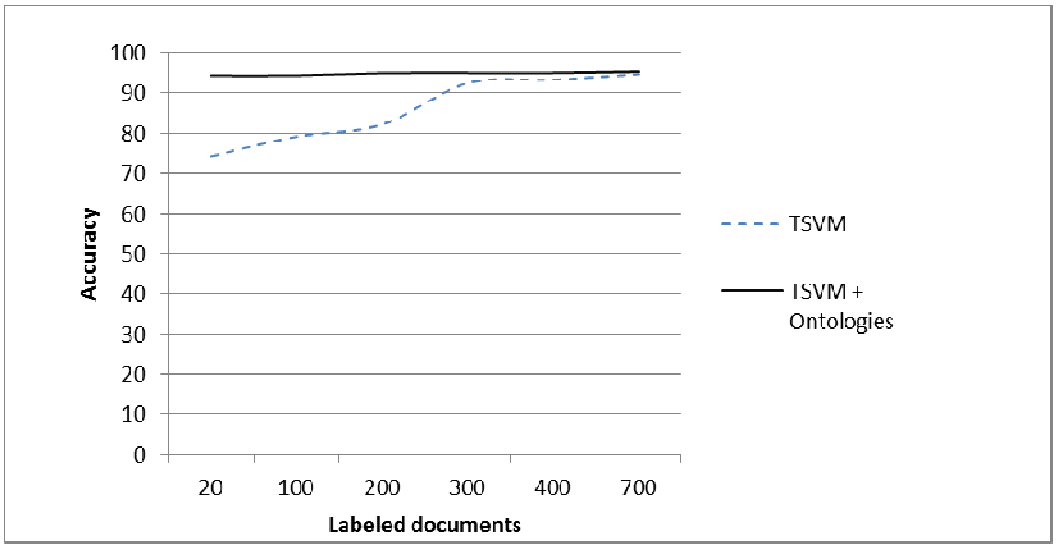

Figure 3. Accuracy of TSVM and TSVM using ontologies for one class of 20 Newsgroups for 2000 documments varying the amount of labeled documents.

In this particular case, the use of ontologies was equivalent to using about twenty percent more of labeled data (400 labeled documents).

5.4 Time efficiency

In Table 10, we present the training times in cpu-seconds for both TSVM and TSVM + ontologies for different datasets sizes. We conduct our experiments in a Dell Precision Workstation 650 with Intel Xeon dual processor, 2.80GHz. It has a 533MHz front side bus, a 512K cache and 4GB SDRAM memory at 266MHz.

We note that there is no significant overhead of the use of the ontologies.

Table 10 Training time in seconds for different dataset sizes.

LABELED	UNLABELED	TOTAL	TSV(s)	TSV+ONT (s)
10	100	110	0.05	0.04
50	500	550	0.09	0.07
100	1000	1100	0.14	0.15
200	2000	2200	7.37	7.19
500	5000	5500	315.48	471.85
1000	10000	11000	1162.63	1121.65

Figure 4 shows the variation of the training time in cpu-seconds, in logarithmic scale, with respect to the number of documents for the two algorithms. As we can note, there is no substantial difference between them. In some cases, TSVM + ontologies performs better. This could be due the reduction in the number of iterations when we use ontologies as shown in Table 11.

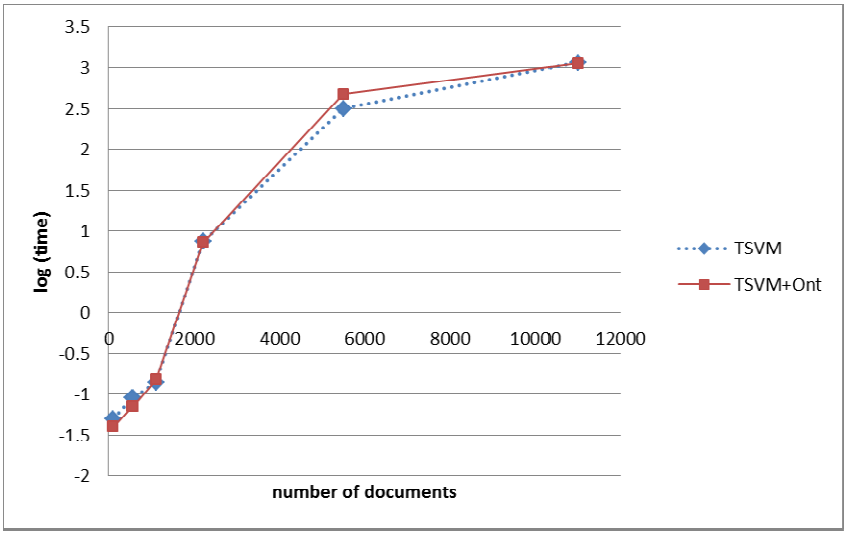

Figure 4 Training time of TSVM and TSVM using ontologies for different documents sizes.

Table 11 Number of iterations for different dataset sizes.

LABELED	UNLABELED	TOTAL	TSV(s)	TSV+ONT (s)
10	100	110	0.05	0.04
50	500	550	0.09	0.07
100	1000	1100	0.14	0.15
200	2000	2200	7.37	7.19
500	5000	5500	315.48	471.85
1000	10000	11000	1162.63	1121.65

6. RELATED WORK

Traditionally, ontologies were used to help pre-processing text documents, such as the use of WordNet to find synonyms to be considered as one word or token.

A distinct approach is presented in [21]. He extracts facts and relationships from the web and builds ontologies. He uses these ontologies as constraints to learn semi-supervised functions at one in a coupled manner.

Recently, Chenthamarakshan [22] presented an approach in which they first map concepts in an ontology to the target classes of interest. They label unlabeled examples using this mapping, in order to use them as training set for any classifier. They called this process concept labeling.

7. CONCLUSIONS

The title is to be written in 20 pt. Garamond font, centred and using the bold and “Small Caps” formats.

In this work, we studied and implemented the use of ontologies to help the semi-supervised document classification task. We compared the performance of these algorithms in three benchmark data sets: 20 Newsgroups, Reuters and WebKb.

Our experiments improve the accuracy of TSVM in many cases. For twenty newsgroups datasets, we obtain the best results having an improvement up to 20 percent.

We note that precision improves in all cases in which accuracy has been improved by the use of ontologies. Furthermore, we improve precision in almost all cases even in the categories in which ontologies do not report an improvement in comparison with TSVM.

We have shown that the influence of ontologies in some cases reached up to 20 percent of data which in our particular experiment it was equivalent to using about 400 labeled documents.

We also evaluate the time performance. Experimental evaluations show that the running time of the learning TSVM algorithm is not significantly affected by the use of the ontologies in most cases.

We show that we can benefit from domain knowledge, where experts create ontologies in order to guide the direction of the semi-supervised learning algorithm. We also have suggested a way to determine if the available ontologies will benefit the semi supervised process. In that way, if it is not, one can always select other ontologies.

Ontologies represent a new source of reliable and structured information that can be used at different levels in the process of classifying documents, and this concept can be extended to the use of ontologies in other areas.

REFERENCES

[1] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, A Bayesian Approach to Filtering Junk E-Mail, 1998.

[2] C. Chan, A. Sun, and E. Lim, "Automated Online News Classification with Personalization," in *4th International Conference of Asian Digital Library*, 2001.

[3] J. Yu, Y. Ou, C. Zhang, and S. Zhang, "Identifying Interesting Customers through Web Log Classification," *IEEE Intelligent Systems*, vol. 20, no. 3, pp. 55-59, 2005.

[4] P. Melville, V. Sindhwani, and R. Lawrence, "Social media analytics: Channeling the power of the blogosphere for marketing insight.," in *Workshop onInformation in Networks*, 2009.

[5] S. Namburú, T. Haiying, Jianhui L., and K. Pattipati, "Experiments on Supervised Learning Algorithms for Text Categorization," in *Aerospace Conference, 2005 IEEE*, 2005.

[6] Kamal Nigam, Andrew McCallum, S. Thrun, and Tom Mitchell, "Learning to classify text from labeled and unlabeled documents," in *Tenth Conference on Artificial intelligence*, Madison, Wisconsin, United States, 1998.

[7] K. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," *Advances in Neural Information Processing Systems 12*, pp. 368-374, 1998.

[8] T. Joachims, "Transductive inference for text classification using support vector machines," in *Sixteenth International Conference of Machine Learning*, 1999.

[9] Kamal Nigam, "Using Unlabeled Data to Improve Text Classification," School of Computer Science, Carnegie Mellon University, Doctoral Dissertation 2001.

[10] A. Krithara, M. Amini, J. Renders, and C. Goutte, "Semi-supervised Document Classification with a Mislabeling Error Model," in *Advances in Information Retrieval, 30th European Conference on IR Research (ECIR'08)*, Glasgow, UK, 2008, pp. 370-381.

[11] E. Boser, M. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Fifth Annual Workshop on Computational Learning theory, COLT '92*, Pittsburgh, Pennsylvania, United States, 1992, pp. 27-29.

[12] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Tenth European Conference on Machine Learning*, 1998.

[13] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines and other Kernel-based Learning Methods*.: Cambridge University Press, 2002.

[14] T. Joachims, *Learning to classify text using support vector machines*.: Kluwer Academic Publishers, 2001.

[15] T. Gruber, "A translation approach to portable ontology specifications," *KNOWLEDGE ACQUISITION*, vol. 5, pp. 199-220, 1993.

[16] L. Lacy, *Owl: Representing Information Using the Web Ontology Language*., 2005.

[17] V. Alexiev et al., *Information Integration with Ontologies: Experiences from an Industrial Showcase*.: Wiley, 2005.

[18] D. Lewis, Y. Yang, T. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research.," *Journal of Machine Learning Research*, vol. 5, pp. 361-397, 2004.

[19] UCI. (2011) Center for Machine Learning and Intelligent Systems. [Online]. HYPERLINK "http://archive.ics.uci.edu/ml/index.html" http://archive.ics.uci.edu/ml/index.html

[20] M. Craven et al., "Learning to extract symbolic knowledge from the World Wide Web.," in *Fifteenth National Conference on Artificial Intellligence.*, 1998.

[21] A. Carlson, J. Betteridge, C. Richard, E. Hruschka, and T. Mitchell, "*Coupled semi-supervised learning for information extraction*", in ACM international conference on Web search and data mining, 2010.

[22] Vijil Chenthamarakshan, Prem Melville, Vikas Sindhwani, and Richard. Lawrence, "*Concept Labeling: Building Text Classifiers with Minimal Supervision*," in International Joint Conference on Artificial Intelligence (IJCAI), 2011.

Authors

Dr. Roxana Aparicio holds a Ph.D. degree in Computer and Information Sciences and Engineering from the University of Puerto Rico - Mayaguez Campus. She received her MS degree in Scientific Computing from the University of Puerto Rico and her BS in Computer Enginnering from the University San Antonio Abad, Cusco, Peru. Currently she is professor in the Institute of Statistics and Information Systems of the University of Puerto Rico - Río Piedras Campus.

Dr. Edgar Acuna holds a Ph.D. degree in Statistics from the University of Rochester, New York. He received his BS in Statistics from the University La Molina, Peru and his MS in Applied Mathematics from the Pontificia Universidad Catolica, Lima, Peru. Currently he is professor of Statistics and CISE in the department of Mathematical Sciences of the University of Puerto Rico-Mayaguez Campus.