Multi-view Semi-supervised Learning: An Approach to Obtain Different Views from Text Datasets

Edson Takashi Matsubara Maria Carolina Monard Gustavo E. A. P. A. Batista
University of São Paulo – USP
Institute of Mathematics and Computer Science – ICMC
Laboratory of Computational Intelligence – LABIC
P. O. Box 668, 13560-970, São Carlos, SP, Brazil
{edsontm, mcmonard, gbatista}@icmc.usp.br

Abstract. The supervised machine learning approach usually requires a large number of labelled examples to learn accurately. However, labelling can be a costly and time consuming process, especially when manually performed. In contrast, unlabelled examples are usually inexpensive and easy to obtain. This is the case for text classification tasks involving on-line data sources, such as web pages, email and scientific papers. Semi-supervised learning, a relatively new area in machine learning, represents a blend of supervised and unsupervised learning, and has the potential of reducing the need of expensive labelled data whenever only a small set of labelled examples is available. Multi-view semi-supervised learning requires a partitioned description of each example into at least two distinct views. In this work, we propose a simple approach for textual documents pre-processing in order to easily construct the two different views required by any multi-view learning algorithm. Experimental results related to text classification are described, suggesting that our proposal to construct the views performs well in practice.

1. Introduction

Due to the rapidly increasing amount of textual data available and the range of interesting and important problems arising in text analysis, there has been a growing interest in applying machine learning methods to text. By combining unsupervised and supervised learning, the need for labelled training data can often be greatly reduced, allowing for the development of more powerful models and methods. Methods that have been proposed under this paradigm are known as semi-supervised learning, and can be considered as the middle road between supervised and unsupervised learning. Semi-supervised algorithms learn a concept definition by combining a small set of labelled and a large set of unlabelled examples.

The multi-view semi-supervised CO-TRAINING method [1] dealt with in this work applies to datasets that have a natural separation of their features into two disjoint sets. In other words, each example is described by two disjoint views, however each view is sufficient for inducing a classifier. Afterwards, a supervised learning system is trained separately using each view, producing two different classifiers. These classifiers are used to label the unlabelled examples, assigning a confidence level to each classification. Unlabelled examples classified with high confidence are used to enlarge the pool of labelled examples; this process is repeated to increment the labelled set until a stop criteria is reached.

In this work we propose and evaluate a simple approach to obtain the two disjoint views, needed by CO-TRAINING, from any textual data base. In order to evaluate the proposed approach, we perform an experimental evaluation with a set of documents extracted from scientific articles published in the *Lecture Notes on Artificial Intelligence* series. The experimental results were obtained using PRETEXT [2], a computational environment for text pre-processing that implements our approach to construct the views from text data, and an implementation of the CO-TRAINING algorithm using *Naive Bayes* as the underlying learner.

The rest of this paper is organized as follows: Section 2 reports some related work on semi-supervised learning. Section 3 describes the CO-TRAINING algorithm and some extended features present in our implementation of this algorithm. Section 4 presents our proposed approach to construct the different views. Section 5 reports the results obtained in the experimental evaluation and Section 6 concludes this paper.

2. Related Work

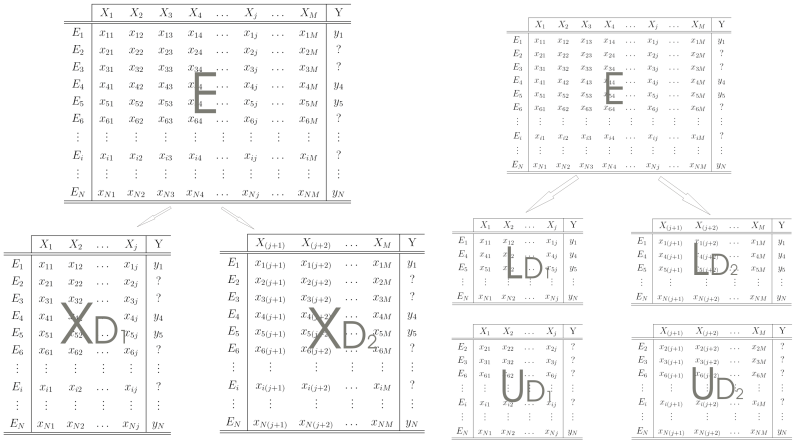
Semi-supervised learning algorithms can be divided into single-view and multiple-view [3]. In a single-view scenario, the algorithms have access to the entire set of domain features. In a multi-view setting, the domain features are presented in subsets (views) that are sufficient for learning the target concept. Single-view algorithms can be split up into transductives [4], Expectation Maximization (EM) variations [5], background knowledge based algorithms [6] and seeded clustering algorithms [7]. Multi-view algorithms are based on the assumption that the views are both *compatible* and *uncorrelated*. If all examples are labelled identically by the target concepts in each view, the dataset is *compatible*. Two views are *uncorrelated* if given the label of any example, its descriptions in each view are independent.

The CO-TRAINING algorithm introduces the theoretical foundations of multi-view learning, and other multi-view learning algorithms have been proposed, such as: CO-EM [8] which combines EM and CO-TRAINING; CO-TESTING [3] which combines active and semi-supervised learning, and CO-EMT [3] an extension of CO-TESTING with CO-EM. The use of Support Vector Machines (SVM) instead of *Naive Bayes* (NB) as the underlying learner is proposed in [9, 10]. An improved version of CO-EM using SVM is proposed in [11] showing experimental results that outperform other algorithms.

Applications of CO-TRAINING include email classification [9], named entity recognition [12], wrapper induction [13], and classification of web pages [1]. However, multi-view learning algorithms are highly dependent on the application. For example, the views in [1] consist of words in the hyperlinks pointing to the pages and words in the Web pages, while the first and second views in [9] consist of the body and the head of emails, respectively. Thus, the views from a dataset can be obtained in different ways. In this work we propose a simple and general way to obtain two views from textual documents.

3. The CO-TRAINING Algorithm

Given a set of N examples $E = \{E_1, ..., E_N\}$ defined by a set of M features $\mathbf{X} = \{X_1, X_2, ..., X_M\}$, CO-TRAINING needs two disjoint views, namely view D_1 and D_2, of the set of examples E. We shall refer to these two views as $\mathbf{X}_{D_1}$ and $\mathbf{X}_{D_2}$ such that $\mathbf{X} = \mathbf{X}_{D_1} \cup \mathbf{X}_{D_2}$ and $\mathbf{X}_{D_1} \cap \mathbf{X}_{D_2} = \emptyset$, and where each view is sufficient to induce a classifier. For simplicity, let us consider $\mathbf{X}_{D_1} = \{X_1, X_2, \ldots X_j\}$ and $\mathbf{X}_{D_2} = \{X_{j+1}, X_{j+2}, \ldots X_M\}$ — Figure 1(a). For unlabelled data we consider the y



(a) Disjoint views $\mathbf{X}_{D_1}$ and $\mathbf{X}_{D_2}$ of E

(b) Subsets L_{D_1}, U_{D_1}, L_{D_2} and U_{D_2} used by CO-TRAINING

Figure 1. The two views used as input for CO-TRAINING

value as "?". Furthermore, there are few examples in set E for which the value of the label y is known. The E set can be divided into two subsets L (Labelled) and U (Unlabelled) examples. The subset $L \subset E$ composed by the labelled example is further divided into two disjoint views L_{D_1} and L_{D_2} where $L = L_{D_1} \cup L_{D_2}$ and $L_{D_1} \cap L_{D_2} = \emptyset$. Similarly, the subset of unlabelled examples $U \subset E$ is split up into two disjoint views U_{D_1} and U_{D_2} where $U = U_{D_1} \cup U_{D_2}$ and $U_{D_1} \cap U_{D_2} = \emptyset$. These four subsets L_{D_1}, L_{D_2}, U_{D_1} and U_{D_2}, illustrated in Figure 1(b), constitute the input to CO-TRAINING described by Algorithm 1.

Initially, a small pool $U' \subset U$ of unlabelled examples is created. U' examples consist of two views, U'_{D_1} and U'_{D_2}, which are withdrawn from U_{D_1} and U_{D_2}, respectively. It is important to note that $U' = U'_{D_1} \cup U'_{D_2}$ and $U'_{D_1} \cap U'_{D_2} = \emptyset$. After the creation of U' the main loop of Algorithm 1 starts. The training examples L_{D_1} and L_{D_2} are used to induce two classifiers h_{D_1} and h_{D_2}, respectively. Using these two classifiers, examples from U'_{D_1} and U'_{D_2} are labelled and inserted in R'_{D_1} and R'_{D_2}, respectively. After that, the labelled examples in R'_{D_1} and R'_{D_2} are given to the function $bestExamples$ which is responsible for selecting the "best" examples to be inserted in L_{D_1} and L_{D_2}. $bestExamples$ only considers examples from R'_{D_1} and R'_{D_2} that have the same class label. After the examples are inserted in L_{D_1} and L_{D_2} the process is repeated until a stop criteria is reached. Currently, two stop criterias are implemented and can be reached: either the user defined maximum number of iterations or the U' sets become empty.

We have implemented several extended features in our implementation of CO-TRAINING. For instance, the $bestExamples$ function has some parameters that enable the user to set how the examples from R'_{D_1} and R'_{D_2} are selected. Two of these parameters are: (i) the minimum probability to label an example, and; (ii) the maximum number of examples of each class that may be inserted in L. These two parameters are very important, the first one defines a minimum confidence level to label an example; the second one influences the examples class distribution in L_{D_1} and L_{D_2}.

Next, we describe the proposed procedure to obtain the two disjoint views $\mathbf{X}_{D_1}$ and $\mathbf{X}_{D_2}$ from texts.

Algorithm 1: CO-TRAINING

Input: $L_{D_1}, L_{D_2}, U_{D_1}, U_{D_2}, k$
Output: L_{D_1}, L_{D_2}
Build U'_{D_1} and U'_{D_2} as described;
$U_{D_1} = U_{D_1} - U'_{D_1}$;
$U_{D_2} = U_{D_2} - U'_{D_2}$;
for $i = 0$ *to* k **do**
  Induce h_{D_1} from L_{D_1};
  Induce h_{D_2} from L_{D_2};
  $R'_{D_1} = h_{D_1}(U'_{D_1})$ set of classified examples from U'_{D_1};
  $R'_{D_2} = h_{D_2}(U'_{D_2})$ set of classified examples from U'_{D_2};
  $(R_{D_1}, R_{D_2}) = bestExamples(R'_{D_1}, R'_{D_2})$;
  $L_{D_1} = L_{D_1} \cup R_{D_1}$;
  $L_{D_2} = L_{D_2} \cup R_{D_2}$;
  if $U_{D_1} = \emptyset$ **then return**(L_{D_1}, L_{D_2}) **else**
    Randomly select examples from U_{D_1} and U_{D_2} to replenish U'_{D_1} and U'_{D_2} respectively;
  end
end
return(L_{D_1}, L_{D_2});

4. Constructing two disjoint views

The attribute-value representation of documents used in Text Mining provides a natural framework to create the two disjoint views needed by CO-TRAINING. However, the attribute-value representation is characterized by very high dimensional data since every word in the document may be treated as an attribute. In this work, we use a text preprocessing computational tool we have implemented, called PRETEXT [2], to efficiently decompose text into words (stems) using the bag-of-words approach, as well as reducing the dimensionality of its representation, making text accessible to most learning algorithms that require each example be described by a vector of fixed dimensionality. The documents are written either in Portuguese, Spanish or English. Our tool is based on the Porter's stemming algorithm for the English language, which was adapted for Portuguese and Spanish. In addition, the tool includes facilities to reduce the dimensionality of datasets using the well known Zipf's law and Luhn cut-offs.

In the identification of terms as bag-of-words, a term can be represented by simple words (*1-gram*), which are represented by the stem of simple words in our tool, or composed words (*2* and *3-gram*) that occur in the document. Each term is used as an attribute of the dataset represented in the attribute-value format. It can be observed that the two views needed by CO-TRAINING can easily be constructed using this approach. In this work, we have used *1-gram* representation for one view and *2-gram* representation for the other view. Furthermore, PRETEXT has several known measures implemented to represent the value of terms in the documents. In this work we have used the term frequency measure, which counts the number of occurrences of a term in a document.

5. Experimental Evaluation

We carried out an experimental evaluation using the LNAI dataset [14], a collection of title, abstracts and references of 277 (70%) articles from Inductive Logic Programming (ILP) and 119 (30%) articles from Case Based Reasoning (CBR) from *Lecture Notes in Artificial Intelligence* (LNAI). Using PRETEXT we constructed the *1-gram* and *2-gram* views. For both views, only stems that appeared more than once in all documents were

considered. After this pre-processing phase, there were 2,914 stems left (attributes) for the *1-gram* view and 3,245 for the *2-gram* view. Table 1 summarizes the datasets employed in this study. It shows the number of documents (#Doc) in the LNAI dataset, the number of attributes (#Attributes) in each view, and class distribution.

It is important to note that the LNAI dataset is completely labelled. It allows us to analyze the behavior of CO-TRAINING, comparing the labels assigned by CO-TRAINING in each iteration with the true labels. In other words, we use the CO-TRAINING algorithm in a simulated mode, in which the true labels are hidden from the algorithm. In order to obtain a lower bound of the error that CO-TRAINING can reach on this dataset, we measured the error rate of a *Naive Bayes* (NB) classifier using all examples and 10-fold cross-validation. This result (mean error and respective standard deviation) is shown in the last column (NB Error) of Table 1, as well as the prediction power of each individual view.

#Doc	View	#Attributes	Class	%Class	NB Error
396	1-gram	2914	ILP	30%	1.7 (3.7)
			CBR	70%	1.4 (1.9)
			Overall		1.5 (1.8)
	2-gram	3245	ILP	30%	1.8 (1.7)
			CBR	70%	1.5 (1.9)
			Overall		1.8 (1.7)

Table 1. LNAI two view dataset descriptions and NB errors

In order to measure the behavior of CO-TRAINING using 10-fold cross-validation, we adapted the sampling method as shown in Figure 2. First, a 10-fold for each view was created. Afterwards, pairs from each view were considered *i.e.*, first fold of view 1 with first fold of view 2, second fold of view 1 with second fold of view 2, and so on.

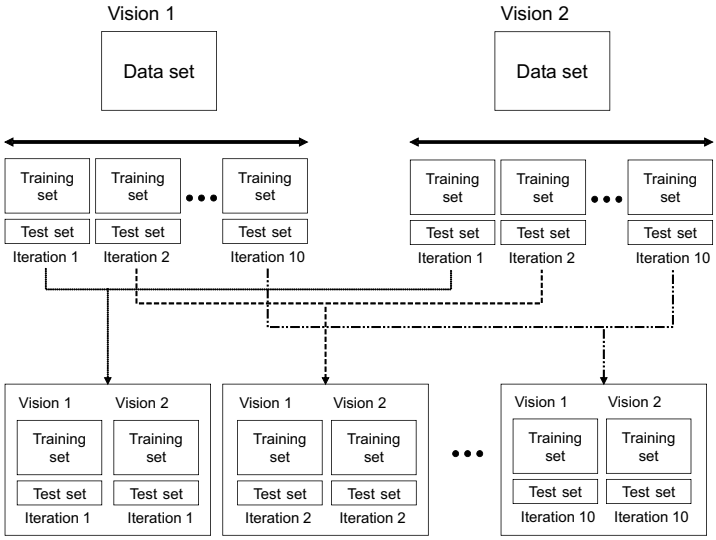


Figure 2. 10-*fold* construction for CO-TRAINING evaluation

As the main idea of semi-supervised learning is to use a large unlabelled sample to improve the performance of supervised learning algorithms when only a small set of labelled

examples is available, the first experiment aims to verify the behavior of CO-TRAINING using a different number of initial labelled examples. For this experiment, the number of examples of each class that may be inserted into L in each iteration is set to 2 for ILP and 2 for CBR classes. In addition, for this and subsequent experiments the minimum probability to label an example was set to 0.6.

Table 2 shows the results obtained, where: $|L_{ini}|$ and $|U_{ini}|$ refer respectively to the initial number of labelled and unlabelled examples. After execution $|L_{end}|$ shows the mean number of examples labelled by CO-TRAINING; $\#Errors$ and $\%Errors$ show the mean number and proportion of incorrectly labelled examples respectively where $\%Errors = \#Errors/(|L_{end}| - |L_{ini}|)$. Standard deviations are shown in brackets. In all cases the stop criteria was $U_{D_1} = \emptyset$ — Algorithm 1 — for k near 70.

% and $\|L_{ini}\|$	$\|U_{ini}\|$	$\|L_{end}\|$	$\#Errors$	$\%Errors$
2% 6	350	275.7 (2.9)	11.0 (2.5)	4.1% (0.9)
5% 17	339	276.8 (4.6)	9.5 (3.3)	3.7% (1.2)
7% 24	332	276.0 (3.4)	7.10 (2.9)	2.8% (1.1)
10% 34	322	279.8 (1.7)	7.4 (1.8)	3.0% (0.8)

Table 2. Mean number of CO-TRAINING incorrectly labelled examples varying $|L_{ini}|$

The performance of CO-TRAINING using the constructed views is very good for all $|L_{ini}|$ values, since few examples were labelled erroneously. Moreover, using NB as the underlying classifier, it is possible to construct a combined classifier h which computes the probability $P(y_v, E_i)$ of class y_v given the instance $E_i = (x_{D1_i}, x_{D2_i})$ by multiplying the class probabilities of h_{D_1} and h_{D_2}, *i.e.*, $P(y_v, E_i) = P(y_v|x_{D1_i})P(y_v|x_{D1_i})$. Table 3 shows the mean error and standard deviation of the classifiers h_{D_1}, h_{D_2} and h on the first and last iteration of CO-TRAINING, and Figure 3 shows the mean error in each iteration.

Iteration	% de $\|L_{ini}\|$	h_{D_1} 1-gram	h_{D_2} 2-gram	h
first	2%	13.4 (7.7)	20.0 (8.0)	11.1 (6.4)
last		5.3 (4.4)	4.0 (3.0)	3.0 (3.3)
first	5%	8.3 (4.9)	10.3 (4.3)	7.6 (4.6)
last		4.3 (3.4)	4.3 (2.9)	3.0 (2.3)
first	7%	6.6 (4.6)	8.3 (4.0)	5.8 (4.1)
last		4.0 (3.8)	3.5 (2.5)	3.0 (2.6)
first	10%	5.0 (3.9)	7.6 (2.9)	4.5 (3.7)
last		3.3 (4.5)	4.0 (3.3)	3.0 (3.3)

Table 3. Mean error of NB and combined classifiers on the first and last iterations

The maximum number of examples of each class inserted into L is an important parameter for CO-TRAINING [1]. We executed CO-TRAINING with three different settings: (i) $bestExamples$ function selects examples from U' in the same proportion of the class distribution; (ii) $bestExamples$ selects the same number of examples from each class, and; (iii) $bestExamples$ selects examples in the inverse proportion of the class distribution. It is important to note that the class distribution is known because the LNAI dataset is completely labelled. However, when only a small set of labelled examples is available, the class distribution might not be accurately estimated from the data. In these cases, class dis-

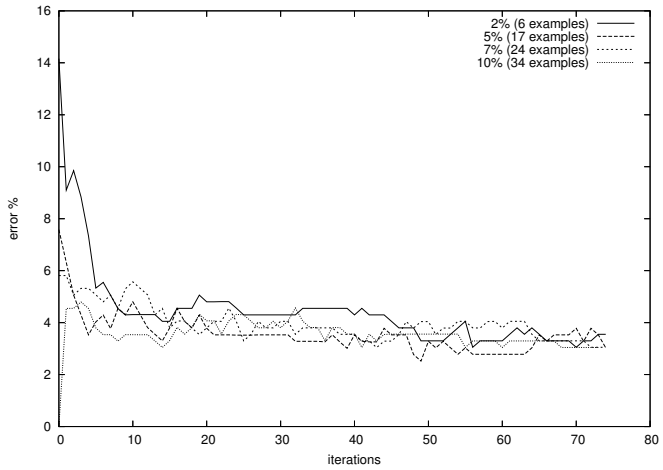


Figure 3. Mean error of combined classifiers for different values of $|L_{ini}|$

tribution might be estimated using domain knowledge, whenever this knowledge is available. The following experiment evaluates the impact of selecting examples at different distributions.

Table 4 shows the results for $|L_{ini}| = 17$ since similar results were obtained for the other three cases. Best results were always obtained selecting examples at a proportion similar to the class distribution — results in bold in Tables 4 and 5. Observe that except for the inverse proportion case, the error of h_{D_1} and h_{D_2} on the last iteration are acceptable comparing with the ones obtained by NB on the whole dataset — Table 1. This might indicate that selecting the same proportion of examples can be an acceptable choice when no further information related to class distribution is available.

$\|L_{ini}\|$	$\|U_{ini}\|$	(maj,min)	$\|L_{end}\|$	$\#Error$	$\%Error$
17	339	**(4,2)**	312.4 (2.0)	3.0 (0.9)	**1.0% (0.3)**
		(2,2)	276.8 (4.6)	9.5 (3.3)	3.7% (1.2)
		(2,4)	225.6 (5.2)	12.4 (4.7)	5.9% (2.2)

Table 4. CO-TRAINING performance for different proportions of examples selected in each iteration

Iteration	(maj,min)	h_{D_1} 1-gram	h_{D_2} 2-gram	h
first	**(4,2)**	7.8 (2.2)	11.1 (3.2)	7.3 (2.8)
last		2.3 (3.0)	2.5 (2.1)	**1.8 (2.1)**
first	(2,2)	8.3 (4.9)	10.4 (4.3)	7.6 (4.6)
last		4.3 (3.4)	4.3 (2.9)	3.0 (2.3)
first	(2,4)	7.6 (3.0)	11.4 (7.9)	7.8 (3.1)
last		5.8 (4.8)	5.3 (6.5)	4.5 (4.5)

Table 5. NB classifiers mean error and standard deviation on CO-TRAINING first and last iteration

6. Conclusions

In this work we propose a simple pre-processing method to construct the views required by multi-view semi-supervised learning algorithms. The proposed approach can be applied to

any set of textual documents. Experiments with CO-TRAINING on a set of documents extracted from scientific articles showed the applicability of this proposal, as well as encouraging initial results. We also show the importance of "tuning" CO-TRAINING execution aiming to obtain better results. Further research should provide a broader experimental evaluation on other textual datasets.

Acknowledgements. This work was partially supported by the Brazilian Research Councils CAPES and FAPESP.

References

[1] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. 11th Annu. Conf. on Comput. Learning Theory*, pages 92–100. ACM Press, New York, NY, 1998.

[2] Edson Takashi Matsubara, Claudia Aparecida Martins, and Maria Carolina Monard. Pretext: A pre-processing text tool using the bag-of-words approach. Technical Report 209, ICMC-USP, 2003. (in portuguese) `ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/RT_209.zip`.

[3] Ion Muslea. Active Learning With Multiple Views, 2002. Phd Dissertation, University Southerm California.

[4] V. Vapnik. *Statistical learning theory*. John Wiley & Sons, 1998.

[5] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Conference on Information and Knowledge Management*, pages 86–93, 2000.

[6] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.

[7] Marcelo Kaminski Sanches. Semi-supervised learning: an approach to label examples from a small pool of labeled examples, 2003. Master Dissertation, ICMC-USP, (in portuguese) `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-12102003-140536`.

[8] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[9] Svetlana Kiritchenko and Stan Matwin. Email classification with co-training. Technical report, University of Otawa, 2002.

[10] Michael Kockelkorn, Andreas Lüneburg, and Tobias Scheffer. Using transduction and multi-view learning to answer emails. In *Proceedings of the European Conference on Principle and Practice of Knowledge Discovery in Databases*, pages 266–277. Springer-Verlag, 2003.

[11] Ulf Brefeld and Tobias Scheffer. Co-EM Support Vector Learning. In *Proceedings of the International Conference in Machine Learning*. Morgan Kaufmann, 2004.

[12] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

[13] Ion Muslea, Steve Minton, and Craig Knoblock. Active + semi-supervised learning = robust multi-view learning. In *International Conference on Machine Learning*, pages 435–432. Morgan Kaufmann, 2002.

[14] Vinícios Melo, Marcos Secato, and Alneu Andrade Lopes. Automatic extraction and identification of bibliographical information from scientific articles (in portuguese). In *IV Workshop on Advances and Trend in AI*, pages 1–10, Chile, 2003.