Inf Retrieval (2009) 12:400–415
DOI 10.1007/s10791-008-9083-7

Using the Web as corpus for self-training text categorization

Rafael Guzmán-Cabrera · Manuel Montes-y-Gómez · Paolo Rosso · Luis Villaseñor-Pineda

Received: 11 May 2008 / Accepted: 28 November 2008 / Published online: 23 December 2008

Abstract Most current methods for automatic text categorization are based on supervised learning techniques and, therefore, they face the problem of requiring a great number of training instances to construct an accurate classifier. In order to tackle this problem, this paper proposes a new semi-supervised method for text categorization, which considers the automatic extraction of unlabeled examples from the Web and the application of an enriched self-training approach for the construction of the classifier. This method, even though language independent, is more pertinent for scenarios where large sets of labeled resources do not exist. That, for instance, could be the case of several application domains in different non-English languages such as Spanish. The experimental evaluation of the method was carried out in three different tasks and in two different languages. The achieved results demonstrate the applicability and usefulness of the proposed method.

Keywords Text categorization · Semi-supervised learning · Self-training · Web as corpus · Authorship attribution

R. Guzmán-Cabrera (✉)
Facultad de Ingeniería Mecánica, Electrica y Electrónica, Universidad de Guanajuato, Guanajuato, Mexico
e-mail: guzmanc@salamanca.ugto.mx

R. Guzmán-Cabrera · P. Rosso
Natural Language Engineering Lab., Polytechnic University of Valencia, Valencia, Spain
e-mail: prosso@dsic.upv.es

M. Montes-y-Gómez · L. Villaseñor-Pineda
Laboratorio de Tecnologías del Lenguaje, Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Mexico
e-mail: mmontesg@inaoep.mx

L. Villaseñor-Pineda
e-mail: villasen@inaoep.mx

1 Introduction

Nowadays, there is a lot of information available in digital format. This situation has produced a growing need for tools that help people to find, organize and analyze all these resources. In particular, *text categorization* (Sebastiani 2002), the automatic assignment of free text documents to one or more predefined categories, has emerged as a very important component in many information management tasks. Most of these tasks are of thematic nature, such as newswire and spam filtering (Aas and Eikvil 1999), whereas others are non-thematically restricted,[1] for instance, authorship attribution (Chaski 2005; Holmes 1994) and sentiment classification (Yu 2006).

There are two main approaches for text categorization: the knowledge engineering approach and the *supervised learning approach*. In the first one, the classification rules are manually constructed by domain experts, whereas in the second, the classifiers are automatically built from a set of labeled (already categorized) documents by applying machine learning techniques. Evidently, due to the high cost associated with the manual construction of classifiers, most current methods for text categorization are based on *supervised* learning techniques.

In order to construct an accurate classifier under the supervised learning approach several issues must be addressed. For instance, it is very important to define an adequate representation of documents as well as to determine an appropriate discriminative function that maps documents to classes. In particular, some recent works have proposed different document representations based on *n*-grams (Bekkerman and Allan 2004) and on syntactic and semantic information (Moschitti and Basili 2004), and they also have applied a number of statistical and machine learning techniques (Sebastiani 2002).

In addition to these issues, a major difficulty of supervised techniques is that they commonly require high-quality training data in order to construct an accurate classifier. Unfortunately, in many real-world applications training sets are *extremely small* and very imbalanced (i.e., the number of examples in some classes is significantly greater than the number of examples in the others).

In order to overcome these problems, many researches have recently been working on *semi-supervised learning* algorithms as well as on different solutions to the class-imbalance problem (for an overview refer to Chawla et al. 2004; Seeger 2000). In line with these works, in this paper we propose a new semi-supervised method for text categorization. This method has two distinctive characteristics: (i) it does not require a predefined set of unlabeled training examples, instead, it considers their automatic extraction from the Web; and (ii) it applies a self-training approach that selects instances considering not only their labeling confidence by a base classifier, but also their correspondence with a web-based labeling.[2] These two characteristics allow the proposed method to work with very few training examples and, therefore, to be less sensitive to the class imbalance problem.

Furthermore, given that the proposed method exclusively relies on lexical information, it can be considered as a *domain independent*, and even as a *language independent* approach. Nevertheless, it is important to emphasize that its usage will be more pertinent

[1] The Nora project (www.noraproject.org) and the Monk project (www.monkproject.org) are two research efforts related to these kind of tasks.

[2] Given that each unlabeled example is downloaded from the Web using a set of automatically defined class queries, each of them has a default category or web-based label.

for those cases (namely, domains and languages) where large sets of labeled resources do not exist, which, for instance, could be the case of several application domains in different non-English languages (Kilgarriff and Grefenstette 2003). In addition, our method is expected to be more appropriate for those languages having a reasonable presence on the Web. Therefore, based on these two conditions, we consider that the proposed method is especially relevant for Spanish, where there are not enough tagged resources, but, on the contrary, there exist abundant information on the Web.

In order to evaluate the proposed method we carried out three different experiments; the first two correspond to thematic classification applications, whereas the later focuses on a non-thematic classification task. In particular, the first experiment considered the classification of natural disasters news reports in Spanish. Its goal was to evaluate the method in a typical non-English task consisting of very few training examples (to be precise, less than ten training instances per class). The second experiment used a subset of the English Reuters-21578 corpus. Its aim was to show the language independence of the method as well as to evaluate its performance in a larger test collection. Finally, the last experiment focused on the problem of authorship attribution. The main purpose of this last experiment was to investigate the applicability of the proposed method in a non-thematic classification task. In order to make this evaluation more interesting, it was focused on Spanish poem classification where documents are usually very short and their vocabulary and structure are very different from everyday—Web—language.

The rest of the paper is organized as follows. Section 2 describes some related work in semi-supervised text categorization. Section 3 presents our web-based self-training approach for text categorization. Section 4 discusses the evaluation results from the three suggested experimental scenarios. Finally, Sect. 5 draws our conclusions and presents some ideas for future work.

2 Semi-supervised learning

As we previously mentioned, the traditional approach for text categorization (and in some extent, to all kinds of classification tasks) considers only labeled instances in the training phase. An inconvenience of this supervised approach is that labeled instances are often difficult and expensive (in terms of human effort) to obtain. On the other hand, unlabeled data may be easy to collect, but there are only few effective ways to use them (Zhu 2005). In this scenario, the *semi-supervised learning* approach emerged as a solution to address these problems. Mainly, it considers the usage of large amount of unlabeled data, together with labeled data, in order to build better classifiers.

Due to the fact that semi-supervised learning requires less human effort and gives higher accuracy, it has been of great interest both in theory and in practice. In particular, the idea of learning classifiers from a combination of labeled and unlabeled data is certainly not new in the statistics community. In 1968, it was suggested that labeled and unlabeled data could be combined to build classifiers with the likelihood of maximization by testing all possible class assignments (Hartley and Rao 1968). More recently, in the field of machine learning, the combined use of labeled and unlabeled examples has been found effective for different tasks (Seeger 2000). Specifically, there are several semi-supervised methods for text categorization, which in turn are based on different learning algorithms, such as Naïve Bayes (Nigam et al. 2000; Peng et al. 2004), Support Vector Machines (Joachims 1999), and nearest-neighbor algorithms (Zelikovitz and Hirsh 2002). Our method differs from all these previous approaches in two main concerns:

– It does not require a predefined set of unlabeled data; instead, it considers their automatic extraction from the Web. This characteristic is very important since there are several application domains where it is extremely difficult to collect example documents. That, for instance, could be the case of the task of authorship attribution, where there are only a few documents per given author. In this case, our method does not exactly find these particular documents, but allows locating some text fragments (snippets) that show similar word distributions and, therefore, that could be considered as additional training instances.
– It applies a self-training approach that selects instances considering not only their labeling confidence by a base classifier, but also their correspondence with a web-based labeling (i.e., a kind of a priori category given by the queries used to download the unlabeled examples). Due to the fact that our method uses these two complementary sources of information for predicting the class from unlabeled examples, it is less dependent on the number of labeled instances and, consequently, it is more adequate for working with very few training examples.

3 Proposed method

Figure 1 shows the general architecture of the proposed method. It consists of two main processes: the first deals with the corpora acquisition from the Web, and the second focuses on the semi-supervised learning problem. The following sections describe in detail these two processes.

3.1 Corpora acquisition

This process considers the automatic extraction of unlabeled examples from the Web. In order to do this, it first constructs a number of queries by combining the most significant words from each class; and then, using these queries, it looks at the Web for some additional training examples related to the given classes.

At this point, it is important to comment that even though the idea of using the Web as corpus may not initially sound intuitive, there are already a number of successful efforts

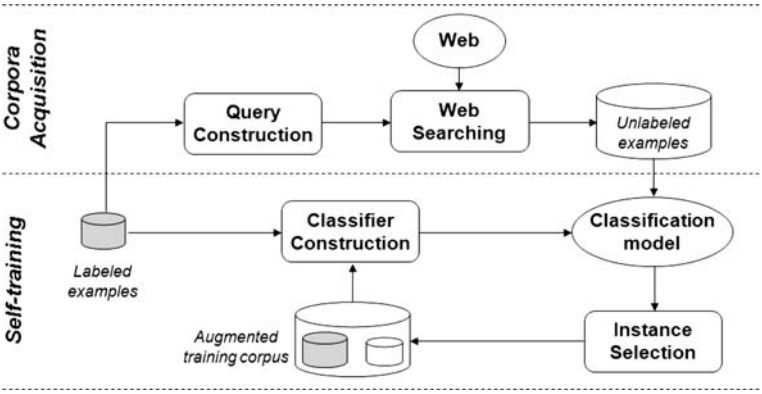

Fig. 1 General architecture of our web-based self-training text categorization method

concerning different natural language tasks (Kilgarriff and Grefenstette 2003). In particular, in (Zelikovitz and Kogan 2006) the authors proposed a method for mining the Web to improve text classification by creating a background text set. Our method is similar to this approach in the sense that it also mines the Web for additional information (extra-unlabeled examples). Nevertheless, as we will describe below, our method applies finer procedures to construct the set of queries related to each class and to combine the downloaded information.

3.1.1 Query construction

To construct the set of queries for searching the Web, it is necessary to previously determine the set of relevant words from each class in the training corpus. The criterion used for this purpose is based on a combination of two characteristics of the given words: on the one hand, their frequency of occurrence, and on the other hand, their information gain. Explicitly, we consider that a word w_i is relevant for a class C if:

1. The frequency of occurrence of w_i in C is greater than the average occurrence of all words (happening more than once) in that class. That is:

$$f_{w_i}^C > \frac{1}{|C'|}\sum_{\forall w \in C'} f_w^C, \ \textit{where}\ C' = \{w \in C | f_w^C > 1\} \tag{1}$$

2. The information gain of w_i in the given training set is positive ($IG_{w_i} > 0$). The idea of this condition is to select those words that help reducing the uncertainty of the value of the class from the given set of examples.

Having obtained the set of relevant words per each class it is possible to construct their corresponding set of queries. Founded on the method proposed in (Zelikovitz and Kogan 2006), we decided to construct queries of three words. This way, we created as many queries per class as all three-word combinations of its relevant words. We measured the significance of a query $q = \{w1, w2, w3\}$ to the class C as indicated below:

$$\Gamma_C(q) = \sum_{i=1}^{3} f_{w_i}^C \times IG_{w_i} \tag{2}$$

Because the selection of relevant words relies on a criterion based on their frequency of occurrence and their information gain, the number of queries per class is not the same even though they include the same number of training examples. In addition, an increment in the number of examples does not necessarily represent a growth in the number of built queries.

3.1.2 Web searching

The next action is using the defined queries to extract a set of additional unlabeled text examples from the Web. Based on the observation that most significant queries tend to retrieve the most relevant web pages, our method for searching the Web determines the number of downloaded examples per query in a direct proportion to its Γ-value. Therefore, given a set of M queries $\{q_1,\ldots,q_M\}$ for a class C, and considering that we want to download a total of N additional examples per class, the number of examples to be extracted by a query q_i is determined as follows:

$$\Psi_C(q_i) = \frac{N}{\sum_{k=1}^{M} \Gamma_C(q_k)} \times \Gamma_C(q_i) \quad (3)$$

It is important to notice that, because each downloaded example corresponds exactly to one particular query, it is possible to consider that these examples belong to a particular class (the same class of the query that was used to retrieved them). This information, which we previously mentioned as web-based labeling, represents a kind of prior category for the unlabeled examples, and thus, it can be of great help in improving the performance of the semi-supervised learning approach.

3.2 Semi-supervised learning

The objective of this second process is to increase the classification accuracy by gradually enlarging the originally small training set with the examples downloaded from the Web.

In particular, we designed this process following the well-known *self-training approach* (Solorio 2002). In this approach, a classifier is initially trained using the small amount of labeled data; then, this classifier is used to classify the unlabeled data, and the most confident examples—in conjunction with their predicted label—are added to the training set; finally, the classifier is re-trained and the procedure is repeated.

In our case, as we previously explained, the selection of the most confident examples not only considers their labeling confidence by a base classifier, but also their correspondence with the web-based labeling. Following, we detail our new self-training algorithm:

1. Build a weak classifier (C_l) using a specified learning method (l) and the available training set (T).
2. Classify the unlabeled web examples (E) using the constructed classifier (C_l). In other words, estimate the class for all downloaded examples.
3. Select the best m examples per class ($E_m \subseteq E$; in this case E_m represents the union of the best m examples from all classes) based on the following two conditions:
   (a) The estimated class of the example corresponds to the class of the query used to download it. In some way, this *filter* works as an ensemble of two classifiers: C_l and the Web (expressed by the set of queries).
   (b) The example has one of the m-highest confidence predictions for the given class.
4. Combine the selected examples with the original training set ($T \leftarrow T \cup E_m$) in order to form a new training collection. At the same time, eliminate these examples from the set of downloaded instances ($E \leftarrow E - E_m$).
5. Iterate σ times over steps 1 to 4 or repeat until $E_m = \emptyset$. In this case σ is a user specified threshold.
6. Construct the final classifier using the enriched training set.

4 Experimental evaluation

This section presents the experimental evaluation of the proposed method. This evaluation was carried out in three different experiments, which consider thematic and non-thematic tasks in Spanish as well as in English document collections.

The following subsection describes the common experimental setup for all experiments; thereafter, in the subsequent subsections, we describe the objectives and discuss the results from each one of the experiments.

4.1 Experimental Setup

Searching the Web. In order to accomplish the search and download the information from the Web we used the well-known Google search engine that we accessed through its API.[3] In general, we downloaded 1,000 additional examples—snippets—per class, but eliminated duplicated examples. Duplicated examples may exist because two different queries from the same class (for instance, $\langle Baja + California + hurricane \rangle$ and $\langle hurricane + kilometers + storm \rangle$) can retrieve very similar sets of documents.

Document preprocessing. We removed all punctuation marks and numerical symbols, that is, we only considered alphabetic tokens. We also removed stop words and converted all words to lowercase. It is important to clarify that these preprocessing actions were applied on both labeled and unlabeled documents.

Learning algorithms. We selected two state-of-the-art machine-learning algorithms for text categorization, namely Naïve Bayes (NB) and Support Vector Machines (SVM) (Aas and Eikvil 1999; Sebastiani 2002). In all experiments, we used the implementations facilitated by the WEKA machine-learning environment (Witten and Frank 1999), and used as features all words occurring more than once in the given training set.

Evaluation measure. The effectiveness of the method was measured by the classification accuracy, which indicates the percentage of documents that have been correctly classified from the entire document test set. In all cases, we evaluated the statistical significance of the results using the t-test and a $p = 0.005$ (Smucker et al. 2007).

4.2 Experiment 1: classifying Spanish natural disasters news reports

This first experiment focused on the classification of Spanish news reports about natural disasters. Its objective was to evaluate the proposed method in a typical non-English scenario, consisting of very few training examples from a specific domain application. In addition to this general objective, this experiment also considered the evaluation of some components of our method: in particular, the evaluation of the proposed web-based filtering used for selecting the best unlabeled examples.

In the following, we describe the evaluation corpus and the results from this experiment.

4.2.1 Evaluation corpus

For this experiment, our evaluation corpus was a set of Spanish newspaper articles about natural disasters.[4] This corpus was collected from the "Reforma" Mexican newspaper.[5] It consists of 240 documents grouped in four different classes: forest fires ($C1$), hurricanes ($C2$), floods ($C3$), and earthquakes ($C4$).

For the experimental evaluation, we organized this corpus as follows: four different training sets formed by 1, 2, 5 and 10 examples per class respectively, and a fixed test set of 200 examples (50 documents per class).

[3] http://www.google.com/apis.

[4] http://ccc.inaoep.mx/~mmontesg/resources/Desastres.sgm.

[5] http://www.reforma.com.

4.2.2 Experimental results

4.2.2.1 Baseline results Baseline results correspond to the direct application of the selected classifiers (namely, Naïve Bayes and SVM) on the test data. The second columns of Tables 1 and 2 show these results for the four different training conditions. They mainly indicate that traditional classification approaches achieve poor performance levels when dealing with very few training examples.

4.2.2.2 Results of our method In order to evaluate the usefulness of the proposed method we performed the following two different experiments by varying the parameter m of the algorithm of Sect. 3.2:

1. At each iteration we added only one additional example per class to the training set; in other words, we set $m = 1$.
2. At each iteration we added to the training set as many unlabeled examples as the number of instances in the original training collection. That is, we set $m = |T|$.

The results from these two experiments, using Naïve Bayes and SVM as base classifiers, are shown in Tables 1 and 2 respectively. In this tables, the asterisks (*) next to the accuracy percentages indicate that the achieved improvement over the baseline result was statistically significant.

Table 1 Accuracy percentages using Naïve Bayes as base classifier ($m = 1$ and $m = |T|$)

Training examples	Baseline result	m-Value	Our method		
			1st iteration	2nd iteration	3rd iteration
1	51.7	$m = 1$	**78.3***	77.3*	76.0*
2	56.7		70.0*	**86.0***	**86.1***
5	80.4		82.2	85.1	**92.1***
10	77.1		83.1	87.2*	**91.3***
1	51.72	$m = \|T\|$	**78.3***	77.3*	76.0*
2	56.71		86.5*	**87.6***	86.5*
5	80.41		**97.0***	96.5*	95.6*
10	77.14		97.2*	**97.5***	96.5*

Table 2 Accuracy percentages using SVM as base classifier ($m = 1$ and $m = |T|$)

Training examples	Baseline result	m-Value	Our method		
			1st iteration	2nd iteration	3rd iteration
1	50.0	$m = 1$	49.1	51.0	**55.3**
2	58.3		62.3	**68.1***	67.0
5	77.1		76.4	80.1	**87.0***
10	80.4		82.1	85.2	**90.1***
1	50.0	$m = \|T\|$	49.1	51.0	**55.3**
2	58.3		68.2*	74.0*	**74.5***
5	77.1		93.5*	92.5*	**96.0***
10	80.4		**96.5***	96.1*	95.1*

Table 3 Accuracy percentages of the method without considering the web-based labeling for the selection of the best m unlabeled instances

Training examples	Baseline result		m-Value	Our method					
				NB (iterations)			SVM (iterations)		
	NB	SVM		1st	2nd	3rd	1st	2nd	3rd
1	51.7	50.0	$m = 1$	59.5	**61.5**	54.0	46.5	44.0	34.5
2	56.7	58.3		76.5	79.5	**83.5**	**66.5**	62.5	66.5
5	80.4	77.1		**88.0**	81.5	80.0	**84.5**	79.0	79.0
10	77.1	80.4		90.5	90.0	88.0	**91.5***	86.0	79.5
1	51.7	50.0	$m = \|T\|$	59.5	**61.5**	54.0	46.5	44.0	34.5
2	56.7	58.3		76.0*	74.0*	**77.5***	56.5	52.0	**61.5**
5	80.4	77.1		**83.1**	83.0	80.5	61.0	66.5	66.5
10	77.1	80.4		**86.0***	77.5	84.5	67.5	71.0	65.0

In general, we may conclude that the results are very satisfactory since they indicate that our method could clearly outperform the baseline results using any of the given classifiers. In particular, setting $m = |T|$ lead to accuracy improvements of almost 25%. Nevertheless, these results also show that this particular classification task was not as difficult as expected, since we could significantly improve the original classification results by adding only a few snippets (that indeed, are very small documents) to the original training set.

In addition to the previous one, we carried out another experiment in order to evaluate the usefulness of the proposed *web-based labeling* for the selection of the best m unlabeled examples (refer to the 3rd step of the algorithm of Sect. 3.2). Table 3 resumes the results of this experiment. These results are interesting since they indicate that:

- Our web-based labeling filter allowed a better selection of the unlabeled instances. In particular, the results of this last experiment were worse than those from the previous one; only a few of them showed a statistically significant improvement over the baseline results (indicated by an asterisk (*)).
- In this experiment, in the majority of the cases, the best results were obtained at the very first iteration. This behavior shows that our filter carries out an important role in the treatment of unlabeled examples with medium and low confidence predictions, which tend to be candidates for being inserted in higher iterations. For that reason in the previous experiment (particularly in the case of SMV) the best results were generally achieved after three iterations.

4.3 Experiment 2: classifying English news reports

The purpose of this experiment was twofold. On the one hand, to validate the language independence of the proposed method, and on the other hand, to evaluate its performance in a larger test collection. In order to accomplish this objective we considered the classification of news reports from a subset of the Reuters, which consists of more than 10,000 English news reports from ten different categories.

In the following we present some data related to the evaluation corpus and the results obtained in this experiment.

Table 4 English training and test data sets

Category	Training set	Test set
ACQ	1650	798
CORN	182	71
CRUDE	391	243
EARN	2877	1110
GRAIN	434	194
INTEREST	354	159
MONEY-FX	539	262
SHIP	198	107
TRADE	369	182
WHEAT	212	94
Total	7206	3220

4.3.1 Evaluation corpus

For this experiment, we selected the subset of the 10 largest categories of the Reuters-21578 collection.[6] In particular, we considered the ModApte split distribution that includes all labeled documents published before 04/07/87 as training data (i.e., 7206 documents) and all labeled documents published after 04/07/87 as test set (i.e., 3220 documents). Table 4 shows some numbers from this collection.

4.3.2 Experimental results

As can be seen in Table 4, the collection considered for this experiment presents an important class imbalance problem (Chawla et al. 2004). Due to this situation, and considering that our method is not immune to this problem, but it is specially suited to work with very few training examples, we decided to configure our experiment as follows. First, we applied a random under-sampling (Hoste 2005) with the aim of assembling a small but balanced training corpus, and thereafter, we used our self-training approach to compensate the missing information by adding new—highly discriminative—training instances (i.e., snippets downloaded from the Web).

Table 5 shows the accuracy results corresponding to different levels of data reduction. It is important to notice that by using only 100 training examples per class it was almost possible to reach the baseline result (which corresponds to the use of the whole training set, i.e., 7206 instances).

In order to evaluate the impact of our web-based self-training categorization method we performed two different experiments. In the first one, we used only 10 training examples per class, whereas, in the second, we employed 100 instances per class. In both experiments, we performed ten iterations, adding at each step the best 10 examples per class. Table 6 shows the accuracy results of these experiments. In this table we also indicate with an asterisk (*) the cases where our method achieved a statistically significant improvement over the initial accuracy value.

From Table 6 it is possible to observe the impact of our self-training method. For instance, when using only 10 training examples per class the method achieved a notable 14% increase in the accuracy (from 58.6 to 70.6). Nevertheless, it is clear that given the

[6] http://www.daviddlewis.com/resources/testcollections/reuters21578.

Table 5 Accuracy percentages for different sizes of the training set

Training examples per class	Accuracy percentage
10	58.6
20	73.7
30	77.3
40	79.3
50	81.8
80	82.8
100	84.1
Baseline	84.7

Table 6 Accuracy percentage after the training corpus enrichment

	Accuracy percentage	
	Using 10 labeled instances per class	Using 100 labeled instances per class
Initial value	58.6	84.1
Iteration 1	66.9*	84.6
Iteration 2	68.7*	84.7
Iteration 3	69.6*	84.8
Iteration 4	70.3*	86.6*
Iteration 5	**70.6***	86.8*
Iteration 6	68.6*	**86.9***
Iteration 7	69.0*	86.7*
Iteration 8	69.0*	86.7*
Iteration 9	68.5*	86.7*
Iteration 10	68.7*	86.7*

complexity of this test collection (that contains some semantically related classes such as grain, corn and wheat) it is necessary to start with more training examples.

In the case of the second experiment (which made use of 100 training examples per class), the increment in the accuracy was not as high as in the first experiment. It only increased the accuracy from 84 to 86.9%. However, it is important to point out that this difference was statistically significant, and that it was enough to outperform the baseline result (84.7%). This indicates that our method obtained a higher accuracy using only 1000 labeled examples instead of considering the whole set of 7206 instances.

4.4 Experiment 3: authorship attribution of Spanish poems

This last experiment aimed to validate the proposed method in a non-thematic classification task. In order to do that, it considered the task of authorship attribution, which consists of automatically determining the corresponding author of an anonymous text.

It is important to comment that, even though there are several different approaches for authorship attribution, which vary from those using stylometric measures (Holmes 1994; Malyutov 2006) and syntactic cues (Chaski 2005; Stamatatos et al. 2001; Diederich et al. 2003), to those based on word-based features (Argamon and Levitan 2005; Kaster et al. 2005; Coyotl-Morales et al. 2006; Zhao and Zobel 2005), most of them rely on the same

Table 7 Statistics from the authorship attribution corpus

Poets	Number of documents	Word tokens	Number of phrases	Average word tokens per document	Average phrases per document
Efran Huerta	48	11352	510	236.5	22.3
Jaime Sabines	80	12464	717	155.8	17.4
Octavio Paz	75	12195	448	162.6	27.2
Rosario Castellanos	80	11944	727	149.3	16.4
Rubén Bonifaz	70	12481	720	178.3	17.3

simple idea that, in order to identify the author of a text, its writing style is more important than the topic. In line with this idea, the aim of this final experiment was to find out whether or not it was possible to extract from the Web *style information*, and whether or not it could be incorporated to the training set in order to improve the accuracy of the classification method.

The following sections describe the corpus used in this experiment, the baseline results, as well as the improvement in the accuracy obtained applying the proposed method.

4.4.1 Evaluation corpus

Given that there is not a standard data set for evaluating authorship attribution methods, we had to assemble our own corpus. This corpus was gathered from the Web and consists of 353 poems written by five different authors.[7] Table 7 summarizes some numbers about this corpus. It is important to notice that the collected poems are very short texts (172 words in average), and that all of them correspond to contemporary Mexican poets. In particular, we were very careful to select modern writers in order to avoid the identification of authors by the use of anachronisms.

4.4.2 Searching for the baseline configuration

Due to the difficulty in comparing our method with other previous works—mainly caused by the absence of a standard evaluation corpus—we carried out several experiments in order to establish a proper baseline. These experiments considered the use of four different kinds of word-based features: (i) functional words, (ii) content words, (iii) the combination of functional and content words, and (iv) word n-grams. Table 8 shows the results obtained with each one of these configurations.

The results from this experiment were very interesting since they showed that: (i) functional words by themselves do not help to capture the writing style of short texts; (ii) content words contain some relevant information to distinguish among authors, even when all documents correspond to the same genre and discuss similar topics; (iii) the lexical collocations, captured by word n-grams, are useful for the task of authorship attribution; and (iv) due to the feature explosion and the small size of the given corpus, the use of higher n-gram sequences (in particular, trigrams) does not necessarily improve the classification performance.

[7] http://ccc.inaoep.mx/~mmontesg/resources/Poetas.sgm.

Table 8 Baseline results for authorship attribution (using Naïve Bayes and a 10-cross-fold validation strategy)

Features	Accuracy	Average precision	Average recall
Functional words	41.0	0.42	0.39
Content words	73.0	0.78	0.73
All kind of words	73.0	0.78	0.74
n-Grams (unigrams plus bigrams)	**78.8**	**0.84**	**0.79**
n-Grams (from unigrams to trigrams)	76.8	**0.84**	0.77

4.4.3 Experimental Results

For this last experiment, we organized the corpus in a different way with respect to the baseline experiment described in the previous section. Specifically, the corpus was divided into two data sets: training (with 80% of the labeled examples) and test (with 20% of the examples). The idea was to carry out the experiment in an almost-real situation, where it is not possible to know all the vocabulary in advance. This is a very important aspect to take into account in poem classification since poets tend to employ a very rich vocabulary. Table 9 shows some numbers about this collection.

Taking into account the results described in the previous section, we decided to use *n*-grams as document features. We performed two different experiments. In the first one we used bigrams as document features, whereas, in the second, we employed trigrams. Table 10 shows the results corresponding to the first three iterations of the method. As it can be observed, the integration of new information improved the baseline results; nevertheless, for this case it was impossible to achieve a statistically significant difference over the baseline results.

As a final comment, we consider that, in spite of being preliminary results, it is surprising to verify that it was feasible to extract useful examples from the Web for the task of

Table 9 Training and test sets for authorship attribution

Poets	Training set	Test set	Word forms (in training set)
Efrain Huerta	38	10	2827
Jaime Sabines	64	16	2749
Octavio Paz	60	15	2431
Rosario Castellanos	64	16	3280
Ruben Bonifaz	56	14	3552
Total	282	71	8377

Table 10 Accuracy percentage after the training corpus enrichment

Features	Baseline accuracy	Iteration		
		1st	2nd	3rd
Exp. 1 (unigrams plus bigrams)	78.9	80.3	**82.9**	80.3
Exp. 2 (from unigrams to trigrams)	74.6	74.7	78.8	**80.3**

authorship attribution. Our intuition suggested the opposite: given that poems tend to use unusual word combinations, the Web seemed not to be an adequate source of relevant information for this task.

5 Conclusions

In this paper we proposed a new semi-supervised method for text categorization. This method mainly differs from previous works in two main concerns. On the one hand, it does not require a predefined set of unlabeled examples, instead, it considers their automatic extraction from the Web; on the other hand, it applies an enriched self-training approach that selects unlabeled instances based on their labeling confidences as well as on their correspondence with an a priori web-based category.

The evaluation of the method was carried out in three different tasks and in two different languages. Some conclusions from these experiments are the following:

- Our method for constructing class-based queries was quite appropriate. The results indicated that in all experiments (one with very few labeled examples, the other with narrow classes, and another of non-thematic nature) it was possible to download relevant snippets that contribute to enhance the classification accuracy.
- The use of extra information, namely the web-based category of the downloaded examples, allows the self-training approach to improve the selection of unlabeled instances. Somehow, this characteristic makes our method specially suited to work with very few training examples.
- The capacity to work with very few training examples allows our method to be applied in classification problems having imbalanced classes. In this situations our method may be used in conjunction with under-sampling techniques.
- The proposed method can be defined as domain and language independent. Experimental results in three different tasks and in two different languages confirm this initial supposition.
- The method is portable to non-thematic tasks. In particular, the achieved results in authorship attribution evidenced this fact. However, it is important to mention that we were surprised about these results, since documents in this task are usually very short and their vocabulary and structure are very different from everyday web language.

Finally, it is it is important to point out that there is not a clear criterion to determine the parameters m and σ of our self-training method. For the presented experiments, we determined the number of unlabeled examples that must be incorporated into the training set at each iteration based on the following condition: the added information—expressed in number of words—must be proportionally small with respect to the original training data. Nevertheless, we are convinced that it is necessary to achieve a detailed analysis of current results as well as to perform further experiments in order to define better empirical criteria for selecting the values of these parameters.

In addition to this point, we are also interested in applying the method in other kind of text categorization tasks, such as Named Entity Classification and Word Sense Disambuguation.

Acknowledgements This work was done under partial support of CONACYT-Mexico (C01-39957), MCyT-Spain (TIN2006-15265-C06-04) and PROMEP (UGTO-121).

References

Aas, K., & Eikvil, L. (1999). *Text categorization: A survey*. Tech. Rep. 941. Norwegian Computing Center.

Argamon, S., & Levitan, S. (2005). Measuring the usefulness of function words for authorship attribution. In *Proceedings of ACH/ALLC Conference 2005*.

Bekkerman, R., & Allan, J. (2004). *Using bigrams in text categorization*. Tech. Rep. IR-408. Center of Intelligent Information Retrieval, UMass Amherst.

Chaski, C. (2005). Who's at the keyboard: Authorship attribution in digital evidence investigations. *International Journal of Digital Evidence, 4*(1), 1–13.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations, 6*(1), 1–6.

Coyotl-Morales, R. M., Villaseñor-Pineda, L., Montes-Y-Gómez, M., & Rosso, P. (2006). Authorship attribution using word sequences. In J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, & J. Kittler (Eds.), *CIARP* (Vol. 4225, pp. 844–853). Springer, Lecture Notes in Computer Science.

Diederich, J., Kindermann, J., Leopold, E., & Paass, G. (2003). Authorship attribution with support vector machines. *Applied Intelligence, 19*(1/2), 109–123.

Hartley, H. O., & Rao, J. N. K. (1968). Classification and estimation in analysis of variance problems. *Review of the International Statistical Institute, 36*(2), 141–147.

Holmes, D. I. (1994). Authorship attribution. *Computers and the Humanities, 28*, 87–106.

Hoste, V. (2005). *Optimization issues in machine learning of coreference resolution*. Ph.D. thesis, Faculteit Letteren en Wijsbegeerte, Universiteit Antwerpen, Belgium.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning* (pp. 200–209). San Francisco, CA: Morgan Kaufmann.

Kaster, A., Siersdorfer, S., & Weikum, G. (2005). Combining text and linguistic document representations for authorship attribution. In *SIGIR Workshop: Stylistic Analysis of Text for Information Access (STYLE)* (pp. 27–35).

Kilgarriff, A., & Grefenstette, G. (2003). Introduction to the special issue of the Web as corpus. *Computational Linguistics, 29*(2), 333–347.

Malyutov, M. B. (2006). Authorship attribution of texts: A review. In R. Ahlswede, L. Bäumer, N. Cai, H. K. Aydinian, V. Blinovsky, C. Deppe, & H. Mashurian (Eds.), *GTIT-C* (Vol. 4123, pp. 362–380). Springer, Lecture Notes in Computer Science.

Moschitti, A., & Basili, R. (2004). Complex linguistic features for text classification: A comprehensive study. In S. McDonald & J. Tait (Eds.), *Proceedings of the 26th European Conference on Information Retrieval (ECIR 2004)* (Vol. 2997, pp. 181–196). Sunderland, UK: Springer, Lecture Notes in Computer Science.

Nigam, K., Mccallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2/3), 103–134.

Peng, F., Schuurmans, D., Wang, S. (2004). Augmenting naive Bayes classifiers with statistical language models. *Information Retrieval, 7*(3–4), 317–345.

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

Seeger, M. (2000). *Learning with labeled and unlabeled data*. Tech. Rep. Edinburgh, UK: University of Edinburgh.

Smucker, M., Allan, J., & Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management* (pp. 623–632).

Solorio, T. (2002). *Using unlabeled data to improve classifier accuracy*. Master's thesis, Computer Science Department, INAOE, Mexico.

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2001). Computer-based authorship attribution without lexical measures. *Computers and the Humanities, 35*, 193–214.

Witten, I. H., & Frank, E. (1999). *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann.

Yu, B. (2006). *An evaluation of text classification methods for literary study*. Ph.D. thesis, Champaign, IL, USA.

Zelikovitz, S., & Hirsh, H. (2002). Integrating background knowledge into nearest-neighbor text classification. In S. Craw & A. D. Preece (Eds.), *ECCBR* (Vol. 2416, pp. 1–5). Springer, Lecture Notes in Computer Science.

Zelikovitz, S., & Kogan, M. (2006). Using web searches on important words to create background sets for LSI classification. In G. Sutcliffe & R. Goebel (Eds.), *FLAIRS Conference* (pp. 598–603). AAAI Press.

Zhao, Y., & Zobel, J. (2005). Effective and scalable authorship attribution using function words. In G. G. Lee, A. Yamada, H. Meng, & S. H. Myaeng (Eds.), *AIRS* (Vol. 3689, pp. 174–189). Springer, Lecture Notes in Computer Science.

Zhu, X. (2005). *Semi-supervised learning literature survey*. Tech. Rep. Computer Sciences, University of Wisconsin-Madison.