Improving Text Classification through Web Corpora

Rafael Guzmán-Cabrera\textsuperscript{1,2}, Manuel Montes-y-Gómez\textsuperscript{3}, Paolo Rosso\textsuperscript{2}, Luis Villaseñor-Pineda\textsuperscript{3}

\textsuperscript{1}Facultad de Ingeniería Mecánica, Electrónica y Electrónica
Universidad de Guanajuato, Mexico
guzmanc@salamanca.ugto.mx

\textsuperscript{2}Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain
pross@dsic.upv.es

\textsuperscript{3}Laboratorio de Tecnologías del Lenguaje
Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico
{mmontesg,villasen}@inaoep.mx

Abstract. A major difficulty with supervised approaches for text classification is that they require a great number of training instances (manually labeled examples) in order to construct an accurate classifier. In this paper we propose a semi-supervised method for text classification that is specially suited to work with very few training examples. This method consists of two main processes. The first one considers the automatic extraction of unlabeled examples from the Web. The second one focuses on the learning procedure, in particular, on the integration of the unlabeled examples into the original training set. Preliminary results indicate that our proposal can significantly improve the classification accuracy in scenarios where there are less than ten training examples available per class.

1 Introduction

Nowadays there is a lot of digital information available from the Web as well as from several private repositories. This situation has produced a growing need for tools that help people to find, filter and analyze all these resources. In particular, text classification [1], the assignment of free text documents to one or more predefined categories based on their content, has emerged as a very important component in many information management tasks.

The state-of-the-art approach for automatic text classification considers the application of a number of statistical and machine learning techniques, including regression models, Bayesian classifiers, support vector machines, nearest neighbor classifiers and neuronal networks [1, 5]. A major difficulty with this kind of supervised techniques is that they commonly require a great number of labeled examples (training instances) to construct an accurate classifier. Unfortunately, because a human expert must manually label these examples, the training sets are extremely small for many
application domains. In order to overcome this problem, recently many researchers have been working on semi-supervised learning algorithms (for an overview see [6]). It has been showed that augmenting the training set with additional information it is possible to improve the classification accuracy using different learning algorithms such as naïve Bayes [4], support vector machines [2], and nearest-neighbor algorithms [8].

In this paper we propose a new method for semi-supervised text classification. This method is different from previous approaches in three main concerns. First, it is specially suited to work with very few training examples. Whereas previous methods consider groups of ten and even hundreds of training examples, our method allows working with less than ten labeled examples per class. Second, it does not require a predefined set of unlabeled examples. It considers the automatic extraction of related untagged data from the Web. Finally, given that it deals with very few training examples, it does not aim including a lot of additional information in the training phase; on the contrary, it only incorporates a small group of examples that considerably augment the dissimilarities among classes.

It is important to point out that the Web has been lately used as a corpus in many natural language tasks [3]. In particular, Zelikvitz and Kogan [9] 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 that it also mines the Web for additional information (extra-unlabeled examples). Nevertheless, our method applies finer procedures to construct the set of queries related to each class and to combine the downloaded information.

The evaluation of the method considers a set of Spanish news about natural disasters. The experimental results demonstrate that the proposed approximation can be effectively used in classification scenarios having very few training examples.

The rest of the paper is organized as follows. Section 2 presents a general overview of the proposed method. It describes the process of corpora acquisition from the Web as well as the learning algorithm that allows incorporating new unlabeled data to the original training corpus. Section 3 shows some experimental results on news classification. Finally, Section 4 depicts our conclusions and discusses some future work.

2 Proposed Method

Figure 1 shows the general scheme of the proposed method. It consists of two main processes. The first one aims to collect from the Web an extended set of text examples (that can be entire web pages or just snippets) related to the given application domain. On the other hand, the purpose of the second process is to increase the classification accuracy by gradually augmenting the originally small training set with a selection of examples downloaded from the Web.

The following sections describe in detail these two processes.

2.1 Corpora Acquisition

As we mentioned, 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 com-
Query Construction. In order to form queries that can be the input to a search engine, it is necessary to previously determine the set of relevant words for each class in the training corpus. The criterion used for this purpose is based on a combination of the frequency of occurrence and the information gain\(^1\) of words. We consider that a word \(w_i\) is relevant for 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:

\[
\frac{f_w^C}{|C'|} > \frac{1}{|C'| \sum_{w \in C'} f_w^C}, \text{ where } C' = \{w \in C | f_w^C > 1\}
\]

2. The information gain of \(w_i\) with respect to \(C\) is positive. That is, if \(IG_{w_i}^C > 0\).

Once obtained the set of relevant words per class, it is possible to construct the corresponding set of queries. Founded on the method by Zelikovitz and Kogan [9], we decide to construct queries of three words. This way, we create as many queries per class as all three-word combinations of its relevant words.

We measure the significance of a query \(q = \{w_1, w_2, w_3\}\) to the class \(C\) as indicated below:

\[\text{For details on the calculus of the information gain refer to [1].}\]
\[ \Gamma_c(q) = \sum_{i=1}^{3} f_{w_i}^C \times IG_{w_i}^C \]

**Web Searching.** Once constructed the set of queries per class, the next action is using these queries to extract from the Web a set of additional unlabeled text examples. 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 \(\Gamma\)-value. Therefore, given a set of \(M\) queries \(\{q_1, \ldots, q_M\}\) for 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:

\[ \Psi_c(q_i) = \frac{N}{\sum_{k=1}^{M} \Gamma_c(q_k)} \times \Gamma_c(q_i) \]

### 2.2 Semi-supervised learning

As we previously mentioned, the purpose of this process is to increase the classification accuracy by gradually augmenting the originally small training set with the examples downloaded from the Web.

Our algorithm for semi-supervised learning is an adaptation of a method proposed elsewhere [7]. It mainly considers the following steps:

1. **Build a weak classifier** \((C_l)\) using a specified learning method \((l)\) and the training set available \((T)\).
2. **Classify the downloaded examples** \((E)\) using the constructed classifier \((C_l)\). In order words, estimate the class for all downloaded examples.
3. **Select the best** \(m\) **examples** \((E_m \subseteq E)\) based on the following two conditions:
   a. The estimate 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.
4. **Combine the selected examples with the original training set** \((T \leftarrow T \cup E_m)\) in order to form a new training set. At the same time, eliminate these examples from the set of downloaded instances \((E \leftarrow E - E_m)\).
5. **Iterate** \(\sigma\) **times over steps 1 to 4** or repeat until \(E_m = \emptyset\). In this case \(\sigma\) is a user specified threshold.
6. **Construct the final classifier** using the enriched training set.

It is important to point out that the proposed algorithm could be applied in combination with several different classification techniques (e.g., naïve Bayes, support vector machines, nearest-neighbor algorithms, etc.). Even more, it also can be used in conjunction with an ensemble of classifiers as proposed by Solorio [7]. In this case,
an adequate selection criterion (refer to step 3) would help diminishing the variance
of the predictions made by the set of classifiers.

3 Experimental Evaluation

3.1 Experimental Setup

Corpus. It is a set of Spanish newspaper articles about natural disasters. It consists of
210 documents grouped in four different categories: forest fires (C1), hurricanes
(C2), inundations (C3), and earthquakes (C4).

For experimental evaluation we organized the corpus as follows. On the one hand,
we considered four different training sets, formed by 1, 2, 5 and 10 examples per
class respectively. On the other hand, we used a fixed test set of 200 examples (50
examples per class).

Searching the Web. We used Google as search engine. In all cases we downloaded
1,000 additional examples per class. In particular, we collected snippets instead of
complete web pages.

Learning methods. We selected the support vector machines (SVM) and naïve
Bayes (NB) as the base classifiers for our experiments. This decision was supported
on the fact that they are considered the state-of-the-art methods for text classification
[1, 5].

Evaluation measure. The method effectiveness is described by the classification
accuracy. This measure indicates the percentage of documents that have been cor-
rectly classified from the entire document set.

Baseline. Baseline results correspond to the direct application of the selected classifi-
cers on the test data. Table 1 shows the results for the four different training condi-
tions. They mainly evidence that traditional classification approaches achieve poor
performance levels when dealing with very few training examples.

<table>
<thead>
<tr>
<th>Number of training examples</th>
<th>SVM</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50.01</td>
<td>51.72</td>
</tr>
<tr>
<td>2</td>
<td>58.33</td>
<td>56.71</td>
</tr>
<tr>
<td>5</td>
<td>77.14</td>
<td>80.41</td>
</tr>
<tr>
<td>10</td>
<td>80.42</td>
<td>77.14</td>
</tr>
</tbody>
</table>

3.2 Experimental Results

This section presents some results related to the main processes of the proposed
method, namely, the corpora acquisition from the Web and the semi-supervised learn-
ing approach.

The central task for corpora acquisition is the automatic construction of a set of
queries that express the relevant content of each class. Table 2 shows some numbers
on this task.
Table 2. Some numbers about query construction

<table>
<thead>
<tr>
<th>Number of training examples</th>
<th>Relevant words per class</th>
<th>Queries per class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

It is noticeable that, because the selection of relevant words relies on a criterion based on their frequency of occurrence and their information gain, there is not the same number of queries per class even thought there were used the same number of training examples. In addition, it is also visible that an increment on the number on examples not necessarily represents a growth on the number of built queries.

Nevertheless, it is important to clarify that using more examples allows to construct more general and consequently more relevant queries. For instance, using only two examples about hurricanes we constructed queries such as \(<\text{Baja} + \text{California} + \text{hurricane}>\) and \(<\text{hurricane} + \text{town} + \text{time}>\), whereas using ten examples we could obtain queries such as \(<\text{hurricane} + \text{kilometers} + \text{storm}>\) and \(<\text{hurricane} + \text{wind} + \text{alert}>\).

With these queries we collected from the Web a set of 1,000 snippets per class, obtaining a total of 4,000 additional unlabeled examples. Then we add some of these examples to the original training set. Mainly, we performed three different experiments by varying the parameter \(m\) of the algorithm of Section 2.2.

1. At each iteration we add to the training set one additional example per class (i.e., we set \(m = 1\)).
2. At each iteration we add to training set as many unlabeled examples as the number of instances in the original set (i.e., we set \(m = |T|\)).
3. In one single step we add to the training set all unlabeled examples satisfying the condition (a) of the algorithm. That is, we considered every examples which estimate class corresponds to the class of the query used to download it.

Tables 3 and 4 show the results of the first two experiments. They indicate that our method outperformed all base configurations, especially when the naïve Bayes was used as the base classifier. In particular, setting \(m = |T|\) lead to accuracy improvements on the range of 30%. Somehow, these results demonstrate the pertinence of the procedures for query construction and weighting.

Table 3. Experiment using \(m = 1\)

<table>
<thead>
<tr>
<th>Number of training examples</th>
<th>Baseline</th>
<th>1st iteration</th>
<th>Our method</th>
<th>2nd iteration</th>
<th>3rd iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>NB</td>
<td>SVM</td>
<td>NB</td>
<td>SVM</td>
</tr>
<tr>
<td>1</td>
<td>50.01</td>
<td>51.72</td>
<td>49.17</td>
<td>78.31</td>
<td>51.09</td>
</tr>
<tr>
<td>2</td>
<td>58.33</td>
<td>56.71</td>
<td>62.33</td>
<td>70.04</td>
<td>68.16</td>
</tr>
<tr>
<td>5</td>
<td>77.14</td>
<td>80.41</td>
<td>76.48</td>
<td>82.25</td>
<td>80.11</td>
</tr>
<tr>
<td>10</td>
<td>80.42</td>
<td>77.14</td>
<td>82.13</td>
<td>83.12</td>
<td>85.23</td>
</tr>
</tbody>
</table>
Table 4. Experiment using $m = 7$

<table>
<thead>
<tr>
<th>Number of training examples</th>
<th>Baseline</th>
<th>1st iteration</th>
<th>Our method</th>
<th>2nd iteration</th>
<th>3rd iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>NB</td>
<td>SVM</td>
<td>NB</td>
<td>SVM</td>
</tr>
<tr>
<td>1</td>
<td>50.01</td>
<td>51.72</td>
<td>49.02</td>
<td><strong>78.25</strong></td>
<td>51.53</td>
</tr>
<tr>
<td>2</td>
<td>58.33</td>
<td>56.71</td>
<td>68.22</td>
<td>86.54</td>
<td>74.04</td>
</tr>
<tr>
<td>5</td>
<td>77.14</td>
<td>80.41</td>
<td>93.54</td>
<td><strong>97.07</strong></td>
<td>92.51</td>
</tr>
<tr>
<td>10</td>
<td>80.42</td>
<td>77.14</td>
<td>96.53</td>
<td><strong>97.27</strong></td>
<td>96.16</td>
</tr>
</tbody>
</table>

Table 5 presents the results achieved by the third experiment. In this case, the results do not favor the proposed method, showing a fall in accuracy around 5 to 25%. In some way these results confirms our intuition that in scenarios having very few training instances it is better to include a small group of unlabeled examples that considerably augments the dissimilarities among classes than including a lot of doubtful-quality information.

Table 5. Experiment using all examples satisfying condition (a)

<table>
<thead>
<tr>
<th>Number of training examples</th>
<th>Baseline</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>NB</td>
</tr>
<tr>
<td>1</td>
<td>50.01</td>
<td>51.77</td>
</tr>
<tr>
<td>2</td>
<td>58.33</td>
<td>56.71</td>
</tr>
<tr>
<td>5</td>
<td>77.14</td>
<td>80.41</td>
</tr>
<tr>
<td>10</td>
<td><strong>80.42</strong></td>
<td>77.14</td>
</tr>
</tbody>
</table>

4 Conclusions and Future Work

In this paper we proposed a method for semi-supervised text classification that is specially suited to work with very few training examples. This method differs from previous approaches in that: (i) it automatically collects from the Web the set of unlabeled examples and, (ii) it only incorporates into the training phase a small group of unlabeled examples.

The experimental results on a set of newspaper articles about natural disasters demonstrate the viability of the method. In some way, they confirm our hypothesis that when dealing with very few training instances it is better to add a selected set of unlabeled examples (those that considerably augments the dissimilarities among classes) than incorporate a lot of doubtful-quality information. In particular, our method obtained the best results when we added to the training set as many unlabeled examples as the number of original labeled instances. It was also noticeable that our method achieved the best results only after two or three iterations.

As future work we plan to apply the proposed method to some non-topical classification problems such as authorship attribution and genre detection. We also want to investigate the helpfulness of the method to deal with unbalanced classes. On the other hand, we also plan to probe the method in other textual classification problems such as named entity classification and word sense disambiguation.
Acknowledgements

This work was done under partial support of CONACYT-Mexico (project grant 43990), and MCyT-Spain (TIN2006-15265-C06-04 research project) and PROMEP (UGTO-121)

References