Techniques for Sentiment Analysis and Topic Detection of Spanish Tweets: Preliminary Report*

Técnicas de análisis de sentimientos y deteccion de asunto de tweets en español: informe preliminar

Antonio Fernández Anta
Institute IMDEA Networks
Madrid, Spain

Philippe Morere†
ENSEIRB-MATMECA
Bordeaux, France

Luis Núñez Chiroque
Institute IMDEA Networks
Madrid, Spain

Agustín Santos
Institute IMDEA Networks
Madrid, Spain

Resumen: Análisis de sentimientos y detección de asunto son nuevos problemas que están en la intersección del procesamiento de lenguaje natural y la minería de datos. El primero intenta determinar si un texto es positivo, negativo o neutro, mientras que el segundo intenta identificar la temática del texto. Un esfuerzo significante está siendo invertido en la construcción de soluciones efectivas para estos dos problemas, principalmente para textos en inglés. Usando un corpus de tweets en español, presentamos aquí un análisis comparativo de diversas aproximaciones y técnicas de clasificación para estos problemas. Los datos de entrada son preprocesados usando técnicas y herramientas propuestas en la literatura, junto con otras específicamente propuestas aquí y que tienen en cuenta las peculiaridades de Twitter. Entonces, se han utilizado clasificadores populares (de hecho se han usado todos los clasificadores de WEKA). No todos los resultados obtenidos son presentados, debido a su alto número.
Palabras clave: Análisis de sentimientos, detección de asunto.

Abstract: Sentiment analysis and topic detection are new problems that are at the intersection of natural language processing (NLP) and data mining. Sentiment analysis attempts to determine if a text is positive, negative, or neither, while topic detection attempts to identify the subject of the text. A significant amount of effort has been invested in constructing effective solutions for these problems, mostly for English texts. Using a corpus of Spanish tweets, we present a comparative analysis of different approaches and classification techniques for these problems. The data is preprocessed using techniques and tools proposed in the literature, together with others specifically proposed here that take into account the characteristics of Twitter. Then, popular classifiers have been used. (In particular, all classifiers of WEKA have been evaluated.) Due to its high number not all the results obtained will be presented here.
Keywords: Sentiment analysis, topic detection.

1 Introduction

With the proliferation of online reviews, ratings, recommendations, and other forms of online opinion expression, there is a growing interest in techniques for automatically extract the information they embody. Two of the problems that have been posed to achieve this are sentiment analysis and topic detection, which are at the intersection of natural language processing (NLP) and data mining. *Sentiment analysis* attempts to determine if a text is positive, negative, or neither, possibly providing degrees within each type. On its hand, *topic detection* attempts to identify the main subject of a given text. Research in both problems is very active, and a num-

* Partially funded by the Spanish Ministerio de Economía y Competitividad.
† Work partially done while visiting Institute IMDEA Networks.

ber of methods and techniques have been proposed in the literature to solve them. Most of these techniques focus on English texts and study large documents. In our work, we are interested in languages different from English and micro-texts. In particular, we are interested in sentiment and topic classification applied to Spanish Twitter microblogs. Spanish is increasingly present over the Internet, and Twitter has become a popular method to publish thoughts and information with its own characteristics. For instance, publications in Twitter take the form of *tweets* (i.e., Twitter messages), which are micro-texts with a maximum of 140 characters. In Spanish tweets it is common to find specific Spanish elements (SMS abbreviations, hashtags, slang). The combination of these two aspects makes this a distinctive research topic, with potentially deep industrial applications.

The motivation of our research is twofold. On the one hand, we would like to know whether usual approaches that have been proved to be effective with English text are also so with Spanish tweets. On the other, we would like to identify the best (or at least good) technique for Spanish tweets. For this second question, we would like to evaluate those techniques proposed in the literature, and possibly propose new ad hoc techniques for our specific context. In our study, we try to sketch out a comparative study of several schemes on term weighting, linguistic preprocessing (stemming and lemmatization), term definition (e.g., based on uni-grams or n-grams), the combination of several dictionaries (sentiment, SMS abbreviations, emoticons, spell, etc.) and the use of several classification methods. When possible, we have used tools freely available, like the *Waikato Environment for Knowledge Analysis* (WEKA, an open source software which consists of a collection of machine learning algorithms for data mining) (at University of Waikato, 2012).

1.1 Related Work

As mentioned above, sentiment analysis, also known as opinion mining, is a challenging Natural Language Processing (NLP) problem. Due to its tremendous value for practical applications, it has experienced a lot of attention, and it is perhaps one of the most widely studied topic in the NLP field. Pang and Lee (Pang and Lee, 2008) have a comprehensive survey of sentiment analysis and opinion mining research. Liu (Liu, 2010), on his hand, reviews and discusses a wide collection of related works. Although, most of the research conducted focuses on English texts, the number of papers on the treatment of other languages is increasing every day. Examples of research papers on Spanish texts are (Brooke, Tofiloski, and Taboada, 2009; Martínez-Cámara, Martín-Valdivia, and Ureña-López, 2011; Martínez Cámara et al., 2011).

Most of the algorithms for sentiment analysis and topic detection use a collection of data to train a classifier that is later used to process the real data. The (training and real) data is processed before being used for (building or applying) the classifier in order to correct errors and extract the main features (to reduce the required processing time or memory). Many different techniques have been proposed for these phases. For instance, different classification methods have been proposed, like Naive Bayes, Maximum Entropy, Support Vector Machines (SVM), BBR, KNN, or C4.5. In fact, there is no final agreement on which of these classifiers is the best. For instance, Go *et al.* (Go, Bhayani, and Huang, 2009) report similar accuracy with classifiers based on Naive Bayes, Maximum Entropy, and SVM.

Regarding preprocessing the data (texts in our case), one of the first decisions to be made is which elements will be used as basic terms. Laboreiro *et al.* (Laboreiro et al., 2010) explore tweets tokenization (or symbol segmentation) as the first key task for text processing. Once single words or terms are available, typical choices are using uni-grams, bi-grams, n-gram, or parts-of-speech (POS). Again, there is no clear conclusion on which is the best option, since Pak and Paroubek (Pak and Paroubek, 2010) report the best performance with bi-grams, while Go (Go, Bhayani, and Huang, 2009) present better results with unigrams. The preprocessing phase may also involve word processing the input texts: stemming, spelling and/or semantic analysis. Tweets are usually very short, having emoticons like :) or :-), or abbreviated (SMS) words like "Bss" for "Besos" ("kisses"). Agarwal *et al.* (Agarwal et al., 2011) propose the use of several dictionaries: an emoticon dictionary and an acronym dic-

tionary. Other preprocessing tasks that have been proposed are contextual spell-checking and name normalization (Kukich, 1992).

One important question is whether the algorithms and techniques proposed for a type of data can be directly applied to tweets. This could be very convenient, since a corpus of Spanish reviews of movies (from Muchocine[1]) has already been collected and studied (Cruz et al., 2008; Martínez Cámara et al., 2011). Unfortunately, Twitter data poses new and different challenges, as discussed by Agarwal *et al.* (Agarwal et al., 2011) when reviewing some early and recent results on sentiment analysis of Twitter data (e.g., (Go, Bhayani, and Huang, 2009; Bermingham and Smeaton, 2010; Pak and Paroubek, 2010)). Engström (Engström, 2004) has also shown that the bag-of-features approach is topic-dependent and Read (Read, 2005) demonstrated how models are also domain-dependent.

These papers, as expected, use a broad spectrum of tools for the extraction and classification processes. For feature extraction, *FreeLing* (Padró et al., 2010) has been proposed, which is a powerful open-source language processing software. We use it as analyzer and for lemmatization. For classification, Justin *et al.* (Justin et al., 2010) report very good results using WEKA (at University of Waikato, 2012; Hall et al.,), which is one of the most widely used tools for the classification phase. Other authors proposed the use of additional libraries like LibSVM (Chang and Lin, 2011). In contrast, some authors (e.g., (Phuvipadawat and Murata, 2010)) propose the utilization of Lucene (Lucene, 2005) as index and text search engine.

Most of the references above have to do with sentiment analysis, since this is a very popular problem. However, the problem of topic detection is becoming also popular (Sriram et al., 2010), among other reasons, to identify trending topics (Allan, 2002; Bermingham and Smeaton, 2010; Lee et al., 2011). Due to the the realtime nature of Twitter data, most works (Mathioudakis and Koudas, 2010; Sankaranarayanan et al., 2009; Vakali, Giatsoglou, and Antaris, 2012; Phuvipadawat and Murata, 2010) are interested in breaking news detection and tracking. They propose methods for the classification of tweets in an open (dynamic) set of topics. Instead, in work we are interested in a closed (fixed) set of topics. However, we explore all the index and clustering techniques proposed, since most of them could be applied to sentiment analysis process.

[1] http://www.muchocine.net

1.2 Contributions

In this paper we have explored the performance of several preprocessing, feature extraction, and classification methods in a corpus of Spanish tweets, both for sentiment analysis and for topic detection. The different methods considered can be classified into almost orthogonal families, so that a different method can be selected from each family to form a different configuration. In particular, we have explored the following families of methods.

Term definition and counting In this family it is decided what constitutes a basic term to be considered by the classification algorithm. The different alternatives are using single words (uni-grams), or groups of words (bi-grams, tri-grams, n-grams) as basic terms. Of course, the aggregation of all these alternatives is possible, but it is typically never used because it results in a huge number of different terms, which makes the processing hard or even impossible. Each of the different terms that appears in the input data is called by classification algorithms an *attribute*. Once the term formation is defined, the list of attributes in the input data is found, and the occurrences of each attributed are counted.

Stemming and lemmatization One of the main difference between Spanish and English is that English is a weakly inflected language in contrast to Spanish, a highly inflected one. A part of our work is the stemming and lemmatization process. In order to reduce the feature dimension (number of attributes), each word could be reduced to either its *lemma* (canonical form) (e.g., "cantábamos" is reduced to its infinitive "cantar") or its *stem* (e.g., "cantábamos" is reduced to "cant"). One interesting questions is to compare how well the usual stemming and lemmatization processes perform with Spanish words.

Word processing and correction Several dictionaries are available to correct the

words and thus reduce the noise caused by mistakes. A spell checker can be used to correct typos. Other grammar dictionaries can replace emoticons, SMS abbreviations, and slang terms by their meaning in correct Spanish. In addition, any special-term dictionary can be applied to get a context in a tweet (i.e., an affective word list can give us the tone of a text, which is relevant for sentiment analysis). Finally, it is possible to use a morphological analyzer to determine the type of each word. Thus, a word-type filter can be applied to the tweets.

Valence shifters By default, once the decision of what constitutes a basic term is made, each term has the same weight is a tweet. A clear improvement to this term-counting method is the process of valence shifters and negative words. Example of negative words are "no", "ni", or "sin" ("not", "neither", "without"), while examples of valence shifters are "muy" or "poco" ("very", "little"). These words are useful for sentiment classification since they change and/or revert the strength of a neighboring term.

Tweet semantics The above approaches can be improved by processing specific tweet artifacts such as author tags, or hashtags and URLs (links) provided in the text. The author tags act like a history of the tweets of a specific person. Because this person will most likely post tweets about the same topic, this might be relevant for topic detection. Additionaly, the hashtags are a great indicator of the topic of a tweet, whereas retrieving keywords from the web-page linked within a tweet allows to overpass the limit of the 140 characters and thus improves the efficiency of the estimation. Another way to overpass this limit is to investigate the keywords of a tweet in a search-engine to retrieve other words of the same context.

Classification methods In addition to these variants, we have explored the full spectrum of classification methods provided by WEKA.

We can construct a large set of (more than 100 thousand) different methods by combining features from all the described families. As this number of combinations is too high, we had to reduce it by manually, choosing a subset of all the methods that is manageable and we think is the most relevant. We hope the reader finds the subset we present satisfactory.

The rest of the paper is structured as follows. In Section 2 we describe in detail the different techniques that we have implemented or used. In Section 3 we describe our evaluation scenario and the results we have obtained. Finally, in Section 4 we present some conclusions and open problems.

2 Methodology

In this section we give the details of how the different methods considered have been implemented in our system. A summary of these parameters is presented in Table 1.

2.1 Term Definition and Processing

n-grams As we mentioned, a *term* is the basic element that will be considered by the classifiers. These terms will be sets of n words (n-grams), with the case when terms are single words (unigrams) as a special case. The value of n is defined in our algorithm with the parameter *n-gram* (see Table 1). The reason for considering the use of n-grams with $n > 1$ (instead of restricting always the terms to individual words) is because they are particularly efficient to recognize common expressions of a language. Also, by keeping a word into its context, it is possible to differentiate its different meanings. For example, in the sentences "estoy cerca" ("I am close") and "cierro la cerca" ("I close the fence"), using 2-grams will allow to detect the two different meanings of the word "cerca". As the words stay in their context, an n-gram carries more information than the sum of the information of its n words: it also carries the context information. (Using uni-grams every single word is a term, and any context information is lost.)

When using n-grams, n is a parameter that highly influences performance. Having a high value of n allows catching more context information, since the combinations of words are less probable. On another side, rare combinations means less occurrences in the data set, which means that a bigger data set is needed to have good results. Also, the larger n is, the longer the attribute list is. In addition, since tweets are short, choosing a large n would result in n-grams of almost the size of a tweet, which would make little sense. We found that, in practice, having n larger than 3 did not improve the results, so

Parameter/flag	Description	Process
n-gram	Number of words that form a term	Both
Only n-gram	Whether words are also terms	Both
Use input data	Whether the input data is used to define attributes	Both
Lemma/Stem	Which technique is used to extract the root of words	Both
Correct words	Whether a dictionary is used to correct misspellings	Both
SMS	Whether an emoticons and SMS dictionary is used	Both
Word types	Types of words to be processed	Both
Affective dictionary	Whether an affective dictionary is used to define attributes	Sentiment
Negation	Whether negations are considered	Sentiment
Weight	Whether valence shifters are considered	Sentiment
Hashtags	Whether hashtags are considered as attributes	Topic
Author tags	Whether author tags are considered as attributes	Topic
Links	Whether data from linked web pages is used	Topic
Search engine	Whether a search engine is used	Topic

Table 1: Parameters and flags that define a configuration of our algorithm.

we limit n to be no larger than 3.

Of course, it is possible to combine the n-grams with several values of n. We only consider the possibility of combining two such values, and one has to be $n = 1$. This is controlled with the flag *Only n-gram* (see Table 1), which says whether only n-grams (with $n > 1$) are considered as terms or also individual words (unigrams) are considered. In the latter case, the lists of attributes of both cases are merged. The drawback of merging is the high number of entries in the final attribute list. Hence, when doing this, a threshold is used to remove all the attributes that appear too few times in the data set, as they are considered as noise. We force that the attribute appears at least 5 times in the data set to be considered. Also, a second threshold is used to remove ambiguous attributes. For example, the entry "ha sido" ("has been") can be found in tweets independently of its topic or sentiment and can be safely removed. This threshold has been set to 85%, which means that more than 85% of the occurrences of this entry have to be for a specific topic or sentiment.

Processing Terms The processing of terms involves first building the list of attributes, which is the list of different terms that appear in the data set of interest. In principle, the data set used to identify attributes is formed at least by all the tweets that are provided as input to the algorithm, but there are cases in which we do not use them. For instance, when using an affective dictionary (see below) we may not use the input data. This is controlled with a parameter that we denote *Use input data* (see Table 1). Moreover, even if the input data is processes, we may filter it and only keep some of it; for instance, we may decide to use only nouns. This can be controlled with the parameter *Word types* (see Table 1), which is described below. In summary, the list of attributes is built from the input data (if so decided) preprocessed as determined by the rest of parameters (e.g., filtered *Word types*) and from potentially the additional data (like the affective dictionary).

Once the list of attributes is constructed, a vector is created for each tweet in the input data. This vector has one position for each attribute, so that the value at that position is the number of occurrences of the attribute in the tweet. This value can be modified in some tweets if the occurrence of an attribute is near a valence shifter (see below). Once this process is completed, the list of attributes and the list of vectors obtained from the tweets are the data passed to the classifier.

2.2 Stemming and Lemmatization

When creating the list of attributes from a collection of terms, different forms of the same word will be found (e.g., singular/plural, masculine/feminine). Including each form as a different attribute would make the list unnecessarily long. Hence, typically only the root of the words is used in the attribute list. The root can take the form of the lemma or the stem of the word. The pro-

cess of extracting it is called lemmatization or stemming, respectively. Lemmatization preserves the meaning and type of a word (e.g., words "buenas" and "buenos" become "bueno"). We have used the FreeLing software to perform this processing, since it can provide the lemma of those words that are in its dictionary. After lemmatization, there are no plurals or other inflected forms, but still two words with the same root but different type may appear. Stemming on its hand reduces even more the list of attributes. A stem is a word whose affixes has been removed. Stemming might lose the meaning and any morphological information that the original word had (e.g., words "aparca", verb, and "aparcamiento", noun, become "aparc"). The Snowball (Sno, 2012) software stemmer has been used in our experiments.

We have decided to always use one of the two processes. Which one is used in a particular configuration is controlled with the parameter *Lemma/Stem* (see Table 1).

2.3 Word Processing and Correction

As mentioned above, one of the possible preprocessing steps of the data before extracting attributes and vectors is to correct spelling errors. Whether or not this step is taken is controlled with the flag *Correct words* (see Table 1). If correction is done, the algorithm uses the Hunspell dictionary (Hun, 2012) (an open source spell-checker) to perform it.

Another optional preprocessing step (controlled with the flag *SMS*) expands the emoticons, shorthand notations, and slang commonly used in SMS messages which is not understandable by the Hunspell dictionary. The use of these abbreviations is common in tweets, given the limitation to 140 characters. An SMS dictionary (dic, 2012) is used to do the preprocessing. It transforms the SMS notations into words understandable by the main dictionary. Also, the emoticons are replaced by words that describe their meaning. For example `:-)` is replaced by *feliz* ("happy") and `:-(` by *triste* ("sad"). The emoticons tend to have a strong emotional semantic. Hence, this process helps estimating the sentiment of the tweets with emoticons.

We have observed that the information of a sentence is mainly located in a few keywords. These keywords have a different type according to the information we are interested in. For topic estimation, the keywords are mainly nouns and verbs whereas for sentiment analysis, they are adjectives and verbs. For example, in the sentence *La pelicula es buena* ("The movie is good"), the only word that is carrying the topic information is the noun *pelicula*, which is very specific to the cinema topic. Besides, the word that best reflects the sentiment of the sentence is the adjective *buena*, which is positive. Also, in the sentence *El equipo ganó el partido* ("The team won the match"), the verb *ganó* is carrying information for both topic and sentiment analysis: the verb *ganar* is used very often in the soccer and sport topics and has a positive sentiment. We allow to filter the words of the input data using their type with the parameter *Word types* (see Table 1). The filtering is done using the FreeLing software, which is used to retrieve the type of each word.

When performing sentiment analysis, we have found useful to have an *affective dictionary,* whose use is controlled with the flag *Affective dictionary* (see Table 1). We have used an affective dictionary developed by Martín García (García, 2009). This dictionary consist of a list of words that have a positive or negative meaning, expanded by their polarity "P" or "N" and their strength "+" or "-". For example, the words *bueno* ("good") and *malo* ("bad") are respectively positive and negative with no strength whereas the words *mejor* ("best") and *peor* ("worse") are respectively positive and negative with a positive strength. As a first approach, we have not intensively used the polarity and the strength of the affective words in the dictionary. Its use only forces the words that contain it to be added as attributes. This has the advantage of drastically reducing the size of the attribute list, specially if the input data is filtered. Observe that the use of this dictionary for sentiment analysis is very pertinent, since the affective words carry the tweet polarity information. In a more advanced future aproach, the characteristics of the words could be used to compute weights. Since not all the words in our affective dictionary may appear in the corpus we have used, we have built *artificial* vectors for the learning machine. There is one artificial vector per sentiment analysis category (positive+, positive, negative, negative+, none), which has been built counting one occurrence of those words

whose polarity and strength match with the appropriate category.

2.4 Valence Shifters

There are two different aspects of valence shifting that are used in our methods. First, we may take into account negations that can invert the sentiment of positive and negative terms in a tweet. Second, we may take weighted words, which are intensifiers or weakeners, into account. Whether these cases are processed is controlled by the flags *Negation* and *Weight* (see Table 1).

Negations are words that reverse the sentiment of other words. For example, in the sentence *La pelicula* ***no*** *es buena* ("The movie is **not** good"), the word *buena* is positive whereas it should be negative because of the negation *no*. The way we process negations is as follows. Whenever a negative word is found, the sign of the 3 terms that follow it is reversed. This allows us to differentiate a positive *buena* from a negative *buena*. The area of effect of the negation is restricted to avoid false negative words in more sophisticated sentences.

Other valence shifters are words that change the degree of the expressed sentiment. Examples of these are, for instance *muy* ("very"), which increases the degree, or *poco* ("little"), which decreases it. These words were included in the dictionary developed by Martín García (García, 2009) as words with positive or negative strength but no polarity. If the flag *Weight* is set, our algorithm finds these words in the tweets, and changes the weight of the 3 terms following them. If the valence shifter has positive strength the weight is multiplied by 3, while if it is negative by 0.5.

2.5 Twitter Artifacts

It has been noticed that with the previous methods, not all the potential data contained in the tweets is used. There are several frequent element in tweets that carry a significant amount of information. Among others we have the following.

- *Hashtags* (any word which starts with "#"). They are used for identify messages about the same topic. Hashtags are very helpful for topic estimation since some of them may carry more topic information than the rest of the tweet. For example, if a tweet contains `#BAR`, which is the hashtag of the Barcelona soccer team, it can almost doubtlessly be classified in a soccer tweet.
- *References* (a "@" followed by the username of the referenced user). It is used to reference other Twitter users. Any user can be referenced. For example, `@username` means the tweet is answering a tweet of *username*, or referring to his/her. References are interesting because some users appear more frequently in certain topics and will more likely tweet about them. A similar behaviour can be found for sentiment.
- *Links* (a URL). Because of the character limitation of the tweets, users often include URLs of webpages where more details about the message can be found. This may help obtaining more context, specially for topic detection.

In our algorithms, we have the possibility of including hashtags and references as attributes. This is controlled by the flags *Hashtags* and *Author tags* (see Table 1), respectively. We believe that these options are just a complement to previous methods and cannot be used alone, because we have found that the number of hashtags and references in the tweets is too small.

We also provide the possibility of adding to the terms of a tweet the terms obtained from the web pages linked from the tweet. This is controlled by the flag *Links*. A first approach could have been retrieving the whole source code of the linked page, get all the terms it contains, and keep the ones that match the attribute list. Unfortunately, there are too many terms, and the menus of the pages induce an unexpected noise which degrades the results. The approach we have chosen is to only keep the keywords of the pages. We chose to only retrieve the text within the HTML tags `h1`, `h2`, `h3` and `title`. The results with this second method are much better since the keywords are directly related to the topic.

Because of the short length of the tweets, our estimations often suffer from a lack of words. We found a solution to this problem in several paper (Banerjee, Ramanathan, and Gupta, 2007; Gabrilovich and Markovitch, 2005; Rahimtoroghi and Shakery, 2011) that use web sources (like Wikipedia or the Open Directory) to complete tweets. The web is a

mine of information and search-engines can be used to retrieve it. We have used this technique to obtain many keywords and a context from just a few words taken from the tweets. For implementation reasons, Bing (Bin, 2012) was chosen for the process. The title and description of the 10 first results of the search are kept and processed in the same way as the words of the tweet. We found out that we have better results by searching in Bing with only the nouns contained in the tweet; therefore, this is the option we chose. The activation of this option is controlled with the flag *Search engine*.

2.6 Classification Methods

The Waikato Environment for Knowledge Analysis (WEKA) (at University of Waikato, 2012) is a collection of machine learning algorithms that can be used for classification and clustering. The workbench includes algorithms for classification, regression, clustering attribute selection and association rule mining. Almost all popular classification algorithms are included. WEKA includes several Bayesian methods, decision tree learners, random trees and forests, etc. It also provides several separating hyperplane approaches and lazy learning methods.

Since we use WEKA as learning machine, it is worth knowing that each element in the learning machine data set will be called an *attribute*, and each element of the data itself will be called a *vector*. (These correspond to the attributes and vectors we have been handling above.) WEKA uses a specific file format ARFF (Attribute-Relation File Format) to reference the attributes and the vectors it uses to learn. This file is first composed of a list of all the attributes whose order is directly related to the order of the vectors' values. The second part of the file is composed by a list of vector, each one representing a tweet. Thus, each tweet adds a vector (line) to the file whereas an attribute adds a line in the first part of the file and a value in each vector.

The different parameters described in Table 1 form a configuration that tells our algorithm which attributes to choose and how to create the vectors. The output of this algorithm is an ARFF file for the configuration and the input data. In general, some of the parameters intend to reduce the size of this file, mainly for two reasons. First, it has been noticed that WEKA is more efficient when there is a smaller number of attributes. Second, a smaller file avoids having lack of memory issues: a great amount of memory, which is proportional to the file size, is needed while WEKA builds a model.

Once the ARFF file is available, we are able to run all the available classification algorithms that WEKA provides. However, due to time limit we will below concentrate on only a few.

3 Experimental Results

3.1 Data Sets

We have used a corpus of tweets provided for the TASS workshop at the SEPLN 2012 conference (TAS, 2012) as input data set. This set contains about 70,000 tweets provided as tuples *ID*, *date*, *userID*. Additionaly, over 7,000 of the tweets were given as a small training set with both topic (chosen *politics, economy, technology, literature, music, cinema, entertainment, sports, soccer* or *others*) and sentiment (or polarity, chosen *strong positive, positive, neutral, negative, strong negative* or *none*) classification. The data set was shuffled for the topics and sentiments to be randomly distributed. Due to the large time taken by the experiments with the large data set, most of the experiments presented have used the small data set, using 5,000 tweets for training and 2,000 for evaluation.

3.2 Configurations for the Submitted Results

We tested multiple configurations with all the WEKA classifiers to choose the one with the highest accuracy to be submitted to the TASS challenge. Different configurations gave the best results for sentiment analysis and topic detection. For instance, for topic detection the submitted results were obtained with a Complement Naive Bayes classifier on attributes and vectors obtained from the input data by not applying lemmatization nor stemming, filtering the words and keeping only nouns, and using hastags and author tags. The reported accuracy by the challenge organizers in the large data set is 45.24%.

Regarding sentiment (polarity), the submitted results were obtained by first classifying the tweets in 5 subsets by using the topic detection algorithm, and then running the sentiment analysis algorithm within each

subset. The latter used Naive Bayes Multinomial on data preprocessed by using the affective dictionary, filtering words and keeping only adjectives and verbs (adjectives were stemmed, and verbs were lemmatized), using the SMS dictionary, and processing negations at the sentence level. The accuracy reported in the large data set was of 36.04%.

Since the mentioned results were submitted, we have worked on making the algorithm more flexible, so it is simpler to activate and deactivate certain processes. This has led to a slightly different behaviour from the submitted version, but we believe it has resulted in an improvement in accuracy.

3.3 Process to Obtain the New Experimental Results

As mentioned, the algorithm used for obtaining the new experimental results, is more flexible and can be configured with the parameters defined in Table 1. In addition, all classification methods of WEKA can be used. Unfortunately, it is unfeasible to execute all possible configurations with all possible classification methods. Hence, we have made some decisions to limit the number of experiments.

First, we have chosen only five classification algorithms from those provided by WEKA. In particular, we have chosen the methods Ibk, Complement Naive Bayes, Naive Bayes Multinomial, Random Committee, and SMO. This set tries to cover the most popular classification techniques. Several configurations of the parameters from Table 1 will be evaluated with these 5 methods.

Second, we have chosen for each of the two problems (topic and sentiment) a basic configuration. In each case, the basic configuration is as close as possible to the configuration used to obtain the submitted results. (Since the algorithm has been modified to add flexibility, the exact submitted configuration could not be used.) The reason for choosing these as basic configurations is that they were found to be the most accurate among those explored before submission. Then, starting from this basic configuration a sequence of derived configurations are tested. In each derived configuration, one of the parameters of the basic configuration was changed, in order to explore the effect of that parameter in the performance. Finally, for each classification method a new configuration is created and tested with the parameter settings that maximized the accuracy.

The accuracy values computed in each of the configurations with the five methods with the small data set are presented in Figures 1 and 2. In both figures, Configuration 1 is the basic configuration. The derived configurations are numbered 2 to 9. (Observe that each accuracy value that improves over the accuracy with the basic configuration is shown on boldface.) Finally, the last 5 configurations of each figure correspond to the parameters settings that gave highest accuracy in the prior configurations for a method (in the order Ibk, Complement Naive Bayes, Naive Bayes Multinomial, Random Committee, and SMO).

3.4 Topic Estimation Results

As mentioned, Figure 1 presents the accuracy results for topic detection on the small data set, under the basic configuration (Configuration 1), configurations derived from this one by toggling one by one every parameter (Configurations 2 to 9), and the seemingly best parameter settings for each classification method (Configurations 10 to 14). Observe that there are no derived configuration with the search engine flag set. This is because the ARFF file generated in that configuration after searching the web as described above (even for the small data set) was extremely large and the experiment could not be completed

The first fact to be observed in Figure 1 is that Configuration 1, which is supposed to be similar to the one used for the submitted results, seems to have a better accuracy with some methods (more than 56% versus 45.24%). However, it must be noted that this accuracy has been computed with the small data set (while the value of 45.24% was obtained with the large one). A second observation is that in the derived configurations there is no parameter that by changing its setting drastically improves the accuracy. This also applies to the rightmost configurations, that combine the best collection of parameter settings.

Finally, it can be observed that the largest accuracy is obtained by Configuration 2 with Complement Naive Bayes. This configuration is obtained from the basic one by simply removing the word filter that allow only

Configuration number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Parameters														
n-gram	1	1	1	1	1	1	1	2	1	1	1	1	2	1
Only n-gram														
Lemma/Stem (L/S)	L	L	S	L	L	L	L	L	L	L	L	L	L	L
Use input data	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SMS					X						X		X	
Word types (Nouns C&P)	X		X	X	X	X	X	X	X	X			X	X
Correct words				X										
Hashtags	X	X	X	X	X		X	X	X	X	X	X		X
Author Tags	X	X	X	X	X	X		X	X		X	X	X	X
Links									X	X				
Search engine														
Classifiers (Accuracy)														
Ibk	36,62	30,54	36,37	36,62	36,77	31,17	**37,97**	32,64	**38,57**	32,47	30,49	30,54	33,83	36,62
ComplementNaiveBayes	56,75	**58,45**	56,25	56,75	**57**	55,75	53,66	53,56	53,56	51,67	**58,25**	**58,45**	52,02	56,75
NaiveBayesMultinomial	56,35	**57,1**	55,61	56,35	56,25	55,46	53,71	55,61	54,11	53,26	**56,95**	**57,1**	56	56,35
RandomCommittee	53,56	52,47	52,62	53,56	**53,91**	**53,66**	52,52	**55,06**	52,72	52,27	51,92	52,47	38,15	53,56
SMO	56,3	55,06	55,95	56,3	56,55	55,51	55,26	55,9	55,16	54,21	42,38	55,06	54,81	56,3

Figure 1: Accuracy (%) of different configurations for topic detection in the small data set.

nouns. Looking most closely at this combination of parameter configuration and method, we can obtain other performance parameters, presented in Table 2. The meaning of these can be found in WEKA. This combination has a 58.45% of correctly classified instances, and a relative absolute error of 54.07%.

3.5 Sentiment Estimation Results

Figure 2, on its turn, shows the accuracy computed for the basic configuration (Configuration 1), the derived configurations (2 to 9), and the best settings per classification method (10 to 14) for sentiment analysis with the small data set. As before, it can be observed that the accuracy of Configuration 1 with SMO is better than the reported accuracy of the results submitted (39.79% versus 36.04%). It also holds that no parameter seems to make a huge difference. However in this case the combination of parameters seem to have some impact, since the best combination, formed by Configuration 13 and method Naive Bayes Multinomial, has significant better accuracy than any other configuration with the same method. However, other methods (e.g., SMO) has a more homogenous set of values.

As before, we take a closer look at the best combination in Table 3. This combination is able to classify correctly 851 instances (and incorrectly 1157), with an accuracy of 42.38%, and relative absolute error of 77.29%.

4 Conclusion

We have presented a comprehensive set of experiments classifying Spanish tweets according to sentiment and topics. In these experiments we have evaluated the use of stemmers and lemmatizers, n-grams, word types, negations, valence shifters, link processing, search engines, special Twitter semantics (hashtags), and different classification methods. This collection of techniques and approaches represent a thorough study comparable with others present in the literature.

The first conclusion of our study is that none of the techniques explored is the silver bullet for Spanish tweet classification. None made a clear difference when introduced in the algorithm. The second conclusion is that tweets are very hard to deal with, mostly due to their brevity and lack of context. The results of our experiments are encouraging, since they show that it is possible the utilization of classical methods for analyzing Spanish texts. The largest accuracy obtained (58% for topics and 42% for sentiment) are not too far from other reported values (TAS, 2012). However, these values reflect that there is still a lot of room for improvement, justifying further efforts.

References

2012. Bing. http://www.bing.com/, accessed August 2012.

2012. diccionariosms.com.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.619	0.039	0.468	0.619	0.533	0.79	música
0.318	0.049	0.316	0.318	0.317	0.635	economía
0.503	0.085	0.565	0.503	0.532	0.709	entretenimiento
0.814	0.192	0.721	0.814	0.765	0.811	política
0.354	0.014	0.386	0.354	0.37	0.67	cine
0.241	0.017	0.175	0.241	0.203	0.612	literatura
0.442	0.102	0.551	0.442	0.491	0.67	otros
0.162	0.013	0.194	0.162	0.176	0.575	tecnología
0.5	0.009	0.419	0.5	0.456	0.745	deportes
0.409	0.014	0.5	0.409	0.45	0.698	fútbol
0.584	0.117	0.579	0.584	0.578	0.734	Weighted Avg.

Table 2: Detail of Configuration 2 of topic detection with Complement Naive Bayes.

Configuration number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Parameters														
N-gram	1	1	1	1	1	1	1	1	2	2	1	1	2	1
Only n-gram														
Lemma/Stem (L/S)	L	L	L	S	L	L	L	L	L	L	L	S	S	L
Use input data	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Affective dictionary	X		X	X	X	X	X	X	X	X	X			X
SMS	X	X	X	X	X	X		X	X		X	X		X
Word types (Adj, Verb)	X	X		X	X	X	X	X	X	X	X			
Correct words					X								X	
Weight						X					X	X		
Negation	X	X	X	X	X	X	X		X	X	X	X		X
Classifiers (Accuracy)														
Ibk	31,32	31,32	29,78	31,32	31,32	31,32	**32,47**	31,32	**31,52**	**32,47**	31,32	28,78	29,08	29,78
ComplementNaiveBayes	30,18	29,88	17,93	28,74	30,13	**30,23**	28,49	30,18	28,74	28,49	**30,23**	16,88	**39,49**	17,93
NaiveBayesMultinomial	32,82	**32,97**	**32,97**	**33,37**	32,77	**32,87**	32,52	32,82	32,87	32,52	**32,87**	32,52	**42,38**	**32,97**
RandomCommittee	33,72	**34,16**	**38,24**	**34,61**	**34,31**	33,67	**34,41**	**34,36**	**34,01**	**34,41**	33,67	**38,34**	**38,14**	**38,24**
SMO	39,79	39,64	**41,93**	38,94	39,59	39,6	29,24	39,74	38,3	39,24	39,6	**41,38**	**41,43**	**41,93**

Figure 2: Accuracy (%) of different configurations for sentiment analysis in the small data set.

http://www.diccionariosms.com, accessed August 2012.

2012. Hunspell: open source spell checking, stemming, morphological analysis and generation under gpl, lgpl or mpl licenses. http://hunspell.sourceforge.net/, accessed August 2012.

2012. Snowball. http://snowball.tartarus.org/, accessed August 2012.

2012. Taller de análisis de sentimientos en la sepln / workshop on sentiment analysis at sepln (tass). http://www.daedalus.es/TASS, accessed August 2012.

Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Allan, James. 2002. Topic detection and tracking. Kluwer Academic Publishers, Norwell, MA, USA, chapter Introduction to topic detection and tracking, pages 1–16.

at University of Waikato, Machine Learning Group. 2012. Weka 3: Data mining software in Java. http://www.cs.waikato.ac.nz/ml/weka/, accessed August 2012.

Banerjee, Somnath, Krishnan Ramanathan, and Ajay Gupta. 2007. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development*

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.285	0.073	0.368	0.285	0.321	0.763	negative+
0.43	0.174	0.354	0.43	0.389	0.736	negative
0.064	0.028	0.145	0.064	0.089	0.577	neutral
0.14	0.047	0.317	0.14	0.194	0.616	positive
0.715	0.261	0.461	0.715	0.561	0.798	positive+
0.469	0.138	0.525	0.469	0.495	0.782	none
0.424	0.146	0.404	0.424	0.4	0.738	Weighted Avg.

Table 3: Detail of Configuration 13 of sentiment analysis with Naive Bayes Multinomial.

in information retrieval, SIGIR '07, pages 787–788, New York, NY, USA. ACM.

Bermingham, Adam and Alan F. Smeaton. 2010. Classifying sentiment in microblogs: is brevity an advantage? In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *CIKM*, pages 1833–1836. ACM.

Brooke, Julian, Milan Tofiloski, and Maite Taboada. 2009. Cross-Linguistic Sentiment Analysis: From English to Spanish. In *Proc. International Conference on Recent Advances in NLP*.

Chang, Chih-Chung and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May.

Cruz, Fermín L, Jose A Troyano, Fernando Enriquez, and Javier Ortega. 2008. Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento del Lenguaje Natural*, 41:73–80.

Engström, Charlotta. 2004. Topic dependence in sentiment classification. Master's thesis, University of Cambridge.

Gabrilovich, Evgeniy and Shaul Markovitch. 2005. Feature generation for text categorization using world knowledge. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 1048–1053, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

García, Miriam Martín. 2009. Sistema de clasificación automática de críticas de cine. Master's thesis, University Carlos III of Madrid. Proyecto Fin de Carrera, Ingeniería Superior de Telecomunicación.

Go, Alec, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update.

Justin, T., R. Gajsek, V. Struc, and S. Dobrisek. 2010. Comparison of different classification methods for emotion recognition. In *MIPRO, 2010 Proceedings of the 33rd International Convention*, pages 700 –703, may.

Kukich, Karen. 1992. Techniques for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439, December.

Laboreiro, Gustavo, Luís Sarmento, Jorge Teixeira, and Eugénio Oliveira. 2010. Tokenizing micro-blogging messages using a text classification approach. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10, pages 81–88, New York, NY, USA. ACM.

Lee, K., D. Palsetia, R. Narayanan, M.M.A. Patwary, A. Agrawal, and A. Choudhary. 2011. Twitter trending topic classification. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 251 –258, dec.

Liu, Bing. 2010. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca.*

Lucene. 2005. The Lucene search engine.

Martínez-Cámara, Eugenio, M. Martín-Valdivia, and L. Ureña-López. 2011. Opinion classification techniques applied to a spanish corpus. In Rafael Muñoz,

Andrés Montoyo, and Elisabeth Métais, editors, *Natural Language Processing and Information Systems*, volume 6716 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pages 169–176.

Martínez Cámara, Eugenio, M. Teresa Martín Valdivia, José M. Perea Ortega, and L. Alfonso Ureña López. 2011. Técnicas de clasificación de opiniones aplicadas a un corpus en español. *Procesamiento de Lenguaje Natural*, 47(0).

Mathioudakis, Michael and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 1155–1158, New York, NY, USA. ACM.

Padró, Lluís, Samuel Reese, Eneko Agirre, and Aitor Soroa. 2010. Semantic services in freeling 2.1: Wordnet and ukb. In Pushpak Bhattacharyya, Christiane Fellbaum, and Piek Vossen, editors, *Principles, Construction, and Application of Multilingual Wordnets*, pages 99–105, Mumbai, India, February. Global Wordnet Conference 2010, Narosa Publishing House.

Pak, Alexander and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Pang, Bo and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Phuvipadawat, S. and T. Murata. 2010. Breaking news detection and tracking in twitter. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 3, pages 120 –123, 31 2010-sept. 3.

Rahimtoroghi, Elahe and Azadeh Shakery. 2011. Wikipedia-based smoothing for enhancing text clustering. In *Proceedings of the 7th Asia conference on Information Retrieval Technology*, AIRS'11, pages 327–339, Berlin, Heidelberg. Springer-Verlag.

Read, Jonathon. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pages 43–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sankaranarayanan, Jagan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. 2009. Twitterstand: news in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 42–51, New York, NY, USA. ACM.

Sriram, Bharath, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 841–842, New York, NY, USA. ACM.

Vakali, Athena, Maria Giatsoglou, and Stefanos Antaris. 2012. Social networking trends and dynamics detection via a cloud-based framework design. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 1213–1220, New York, NY, USA. ACM.