Procesamiento del Lenguaje Natural, Revista nº 50 marzo de 2013, pp 45-52 recibido 28-11-12 revisado 13-02-13 aceptado 19-02-13

Sentiment Analysis and Topic Detection of Spanish Tweets: A Comparative Study of NLP Techniques [*]

Análisis de sentimientos y detección de asunto de tweets en español: un estudio comparativo de técnicas de PLN

Antonio Fernández Anta[1] **Luis Núñez Chiroque**[1]
Philippe Morere[2†] **Agustín Santos**[1]
[1] Institute IMDEA Networks, Madrid, Spain
[2] ENSEIRB-MATMECA, Bordeaux, France
{antonio.fernandez, luisfelipe.nunez, philippe.morere, agustin.santos}@imdea.es

Resumen: Se está invirtiendo mucho esfuerzo en la construcción de soluciones efectivas para el análisis de sentimientos y detección de asunto, pero 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.

Palabras clave: Análisis de sentimientos, detección de asunto.

Abstract: A significant amount of effort is been invested in constructing effective solutions for sentiment analysis and topic detection, but mostly for English texts. Using a corpus of Spanish tweets, we present a comparative analysis of different approaches and classification techniques for these problems.

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 extracting 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. Research in both problems is very active, and a number 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 micro-blogs. 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 hand, we would like to identify the best (or at least a good) technique for processing 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.

1.1 Related Work

Sentiment analysis is a challenging NLP problem. Due to its tremendous value for

[*] Partially funded by Factory Holding Company 25, S.L. The authors want to thank Fernando Pérez for useful discussions.

[†] Partially done while at Institute IMDEA Networks.

ISSN 1135-5948

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 (2008) have a comprehensive survey of sentiment analysis and opinion mining research. 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; Sidorov et al., 2012).

Most of the algorithms for sentiment analysis and topic detection use a collection of data to train a classifier, which is later used to process the real data. Data is preprocessed before being used in the classifier in order to correct errors and extract the main features. Many different techniques have been proposed for these two 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, Bhayani, and Huang (2009) report similar accuracies with classifiers based on Naive Bayes, Maximum Entropy, and SVM.

Regarding preprocessing the data, 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) as basic terms. Again, there is no clear conclusion on which is the best option, since Pak and Paroubek (2010) report the best performance with bi-grams, while 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. (2011) propose the use of several dictionaries: an emoticon dictionary and an acronym dictionary. 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 other types of data can be directly applied to tweets. Twitter data poses new and different challenges, as discussed by 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 (2004) has also shown that the bag-of-features approach is topic-dependent and 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. (2010) report very good results using WEKA, 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).

Most of the references above have to do with sentiment analysis, due to its popularity. 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 real time nature of Twitter data, most works (Mathioudakis and Koudas, 2010; Vakali, Giatsoglou, and Antaris, 2012) are interested in breaking news detection and tracking. They propose methods for the classification of tweets in an open (dynamic) set of topics. Instead, we are interested in a closed (fixed) set of topics. However, we explore all the indexing and clustering techniques proposed, since most of them could also be applied to sentiment analysis.

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.

Stemming and lemmatization. One of the main difference between Spanish and English is that the latter is a weakly inflected language in contrast to Spanish, a highly inflected one. One interesting questions is to compare how well the usual stemming and lemmatization processes perform with Spanish words.

Word processing and correction. We have used several dictionaries in order to correct the words and replace emoticons, SMS abbreviations, and slang terms by their meaning in correct Spanish. 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 tweets.

Valence shifters. An alternative to the usual direct term-counting method is the processing of valence shifters and negative words (*not, neither, very, little*, etc). We think that those 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. Additionaly, the hashtags are a great indicator of the topic of a tweet, whereas retrieving keywords from the webpage 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.

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.

2.1 Attributes Definition and Preprocessing

n-grams As we mentioned, classifiers will consider sets of n words (n-grams), with unigrams as a special case. The value of n could be defined in our algorithm. When using n-grams, n is a parameter that highly influences performance. We found that, in practice, having n larger than 3 did not improve the results, so we limit n by that value.

Of course, it is possible to combine n-grams with several values of n. The drawback of this 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. This threshold has been set to 85%, which means that more than 85% of the occurrences of an attribute 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. 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. In summary, the list of attributes is built from the input data (if so decided) preprocessed as determined and, potentially, by additional data (like the affective dictionary). Once this process is completed, the list of attributes and the list of vectors obtained from the tweets are passed to the classifier.

Stemming and Lemmatization When creating the list of attributes, 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 (lemmatization or stemming, respectively). We have used the FreeLing software to perform the lemmatization process. The Snowball software stemmer has been used in our experiments. We have decided to always use one of the two processes.

Word Processing and Correction As mentioned above, one of the possible preprocessing steps before extracting attributes and vectors is to correct spelling errors. If correction is done, the algorithm uses the Hunspell dictionary to perform it.

Another optional preprocessing step 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 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*).

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. The filtering is done using the FreeLing software, which is used to extract the type of each word.

When performing sentiment analysis, we have found useful to have an *affective dictionary*. 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 approach, 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.

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.

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. If the valence shifter is positive, the weight is multiplied by 3, while if it is negative by 0.5.

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 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). 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. 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. 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 in the pages induce an unexpected noise which degrades the results. The approach we have chosen is to keep only the keywords of the pages. We chose to retrieve only 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 papers (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 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.

2.2 Classification Methods

For classification, we use WEKA[1], which is a collection of machine learning algorithms that can be used for classification and clustering. It includes algorithms for classification, regression, clustering attribute selection and association rule mining. Almost all popular classification algorithms are included (Bayesian methods, decision tree learners, random trees and forests, etc.).

For each experiment we set up a configuration that tells our algorithm which attributes to choose and how to create vectors of attributes. The output of this algorithm is a WEKA file for a specific configuration and the input data. Once this file is available, we are able to run all the available classification algorithms that WEKA provides. However, due to space limit we will below concentrate on only a few.

3 Experimental Results

3.1 Data Sets and Experiments Configurations

In our experiments we have used a corpus of tweets provided for the TASS workshop at the SEPLN 2012 conference as input data set. This set contains about 70,000 tweets. Additionaly, over 7,000 of the tweets were given as a small training set with both topic and sentiment 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.

[1] http://www.cs.waikato.ac.nz/ml/weka, accessed August 2012.

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
Lemma/Stem (**L**/**S**)	L	L	S	L	L	L	L	L	L	L	L	L	L	L
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				
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.

Precision	Recall	F-Measure	Class
0.468	0.619	0.533	música
0.316	0.318	0.317	economía
0.565	0.503	0.532	entretenimiento
0.721	0.814	0.765	política
0.386	0.354	0.37	cine
0.175	0.241	0.203	literatura
0.551	0.442	0.491	otros
0.194	0.162	0.176	tecnología
0.419	0.5	0.456	deportes
0.5	0.409	0.45	fútbol
0.579	0.584	0.578	Weighted Avg.

Table 1: Detail of Configuration 2 of topic detection with Complement Naive Bayes.

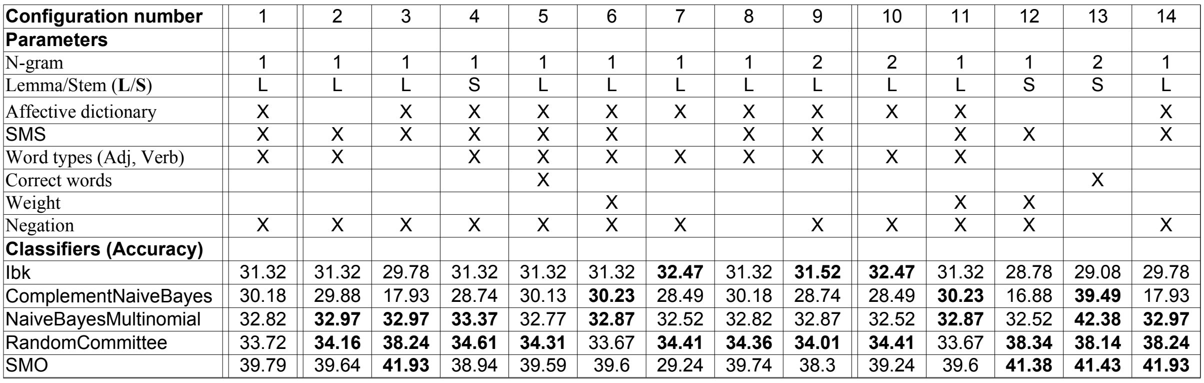
For the TASS workshop we tested multiple configurations with all the WEKA classifiers to choose the one with the highest accuracy. Different configurations gave the best results for sentiment analysis and topic detection. As described, our initial approach was to compare every possible configuration and all classification methods of WEKA. Unfortunately, it was unfeasible to execute all possible configurations with all possible classification methods. Hence, we made some decisions to limit the number of experiments.

In this paper, we have chosen to present 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. Then, we have chosen for each of the two problems (topic and sentiment) a basic configuration similar to the one submitted to the TASS workshop. 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).

3.2 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 no configuration uses a search engine. This is because we found that the ARFF file generated iafter searching the web as described above (even for the small data set) was extremely large and the experiment



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
Lemma/Stem (**L**/**S**)	L	L	L	S	L	L	L	L	L	L	L	S	S	L
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.

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 submitted to TASS, 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 allows only nouns. Looking more closely at this combination of parameter configuration and method, we can obtain other performance parameters, presented in Table 1. The meaning of these can be found in the WEKA documentation. This combination has a 58.45% of correctly classified instances, and a relative absolute error of 54.07%.

3.3 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.

Precision	Recall	F-Measure	Class
0.368	0.285	0.321	negative+
0.354	0.43	0.389	negative
0.145	0.064	0.089	neutral
0.317	0.14	0.194	positive
0.461	0.715	0.561	positive+
0.525	0.469	0.495	none
0.404	0.424	0.4	Weighted Avg.

Table 2: Detail of Configuration 13 of sentiment analysis with Naive Bayes Multinomial.

As before, we take a closer look at the best combination in Table 2. 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 Conclusions

We have presented a comprehensive set of experiments classifying Spanish tweets according to sentiment and topic. 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 represent a thorough study.

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 though, since they show that it is possible to use classical methods for analyzing Spanish texts. The largest accuracy obtained (58% for topics and 42% for sentiment) are not too far from other values reported in the TASS workshop. However, these values reflect that there is still a lot of room for improvement, justifying further efforts.

References

Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In LSM '11, pp. 30–38.

Allan, James. 2002. Topic detection and tracking. Kluwer Academic Publishers.

Banerjee, Somnath, Krishnan Ramanathan, and Ajay Gupta. 2007. Clustering short texts using wikipedia. In SIGIR'07, pp. 787–788.

Bermingham, Adam and Alan F. Smeaton. 2010. Classifying sentiment in microblogs: is brevity an advantage? In *CIKM 2010*, pp. 1833–1836.

Brooke, Julian, Milan Tofiloski, and Maite Taboada. 2009. Cross-Linguistic Sentiment Analysis: From English to Spanish. In *RANLP 2009*.

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

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. *Proc. del Lenguaje Natural*, 41:73–80.

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

Gabrilovich, Evgeniy and Shaul Markovitch. 2005. Feature generation for text categorization using world knowledge. In IJCAI'05, pp. 1048–1053.

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

Justin, T., R. Gajsek, V. Struc, and S. Dobrisek. 2010. Comparison of different classification methods for emotion recognition. In *MIPRO 2010*, pp. 700 –703.

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

Laboreiro, Gustavo, Luís Sarmento, Jorge Teixeira, and Eugénio Oliveira. 2010. Tokenizing micro-blogging messages using a text classification approach. In AND '10, pp. 81–88.

Lee, K., D. Palsetia, R. Narayanan, M.M.A. Patwary, A. Agrawal, and A. Choudhary. 2011. Twitter trending topic classification. In *ICDMW 2011*, pp. 251–258.

Liu, Bing. 2010. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, 2nd Edition*, Taylor & Francis Group.

Martínez-Cámara, Eugenio, M. Martín-Valdivia, and L. Ureña-López. 2011. Opinion classification techniques applied to a spanish corpus. In *NLDB 2011*, pp. 169–176.

Mathioudakis, Michael and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *SIGMOD'10*, pp. 1155–1158.

Padró, Lluís, Samuel Reese, Eneko Agirre, and Aitor Soroa. 2010. Semantic services in freeling 2.1: Wordnet and ukb. In *Global Wordnet Conference 2010*, pp. 99–105.

Pak, Alexander and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC'10*.

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

Rahimtoroghi, Elahe and Azadeh Shakery. 2011. Wikipedia-based smoothing for enhancing text clustering. In *AIRS'11*, pp. 327–339.

Read, Jonathon. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *ACLstudent'05*, pp. 43–48.

Grigori Sidorov, Sabino Miranda-Jiménez, Francisco Viveros-Jiménez, Alexander Gelbukh, Noé Castro-Sánchez, Francisco Velásquez, Ismael Díaz-Rangel, Sergio Suárez-Guerra, Alejandro Treviño, and Juan Gordon. 2012. Empirical Study of Machine Learning Based Approach for Opinion Mining in Tweets In *MICAI'12*.

Sriram, Bharath, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *SIGIR '10*, pp. 841–842.

Vakali, Athena, Maria Giatsoglou, and Stefanos Antaris. 2012. Social networking trends and dynamics detection via a cloud-based framework design. In *WWW '12 Companion*, pp. 1213–1220.