TASS: A Naive-Bayes strategy for sentiment analysis on Spanish tweets*

TASS: Una estrategia Naive-Bayes para el análisis del sentimiento en tweets en español

Pablo Gamallo
Centro de Investigação em Tecnologias da Língua (CITIUS)
Univ. de Santiago de Compostela
pablo.gamallo@usc.es

Marcos Garcia
Cilenis, S.L.
marcos.garcia@cilenis.com

Santiago Fernández-Lanza
CITIUS, Univ. de Santiago de Comp.
santiago.fernandez.lanza@usc.es

Resumen: En este artículo, se describe la estrategia que subyace al sistema presentado por nuestro grupo para la tarea de análisis de sentimiento en el TASS 2013. El sistema se basa principalmente en un clasificador Naive-Bayes orientado a la detección de la polaridad en tweets escritos en español. Los experimentos realizados han mostrado que los mejores resultados se han alcanzado utilizando clasificadores binarios que distinguen apenas entre dos categorías de polaridad: positivo y negativo. Para poder identificar más niveles de subjetividad, hemos incorporado al sistema umbrales de separación con los que distinguir valores de polaridad fuertes, medios y débiles o neutros. Además, para poder detectar si un tweet tiene o no tiene polaridad, el sistema incorpora también una regla básica basada en la búsqueda de palabras con polaridad dentro del texto analizado. Los resultados de la evaluación muestran valores razonablemente altos (cerca del 67% de precisión) cuando el sistema se aplica para detectar cuatro categorías de sentimiento.
Palabras clave: Análisis del sentimiento, Minería de opiniones, Clasificación Naive Bayes, Twitter

Abstract: This article describes the strategy underlying the system presented by our team for the sentiment analysis task at TASS 2013. The system is mainly based on a naive-bayes classifier for detecting the polarity of Spanish tweets. The experiments have shown that the best performance is achieved by using a binary classifier distinguishing between just two sharp polarity categories: positive and negative. To identify more polarity levels, the system is provided with experimentally set thresholds for detecting strong, average, and weak (or neutral) values. In addition, in order to detect tweets with and without polarity, the system makes use of a very basic rule that searchs for polarity words within the analysed text. Evaluation results show a good performance of the system (about 67% accuracy) when it is used to detect four sentiment categories.
Keywords: Sentiment Analysis, Opinion Mining, Naive Bayes Classification, Twitter

1 Introduction

Sentiment Analysis consists in finding the opinion (e.g. positive, negative, or neutral) from text documents such as movie reviews or product reviews. Opinions about movies, products, etc. can be found in web blogs, social networks, discussion forums, and so on. Companies can improve their products and services on the basis of the reviews and comments of their costumers. Recently, many works have stressed the microblogging service Twitter. As Twitter can be seen as a large source of short texts (tweets) containing user opinions, most of these works make sentiment analysis by identifying user attitudes and opinions toward a particular topic or

* This work has been supported by Ministerio de Ciencia e Innovación, within the project OntoPedia, ref: FFI2010-14986.

product (Go et al., 2009). The task of making sentiment analysis from tweets is a hard challenge. On the one hand, as in any sentiment analysis framework, we have to deal with human subjectivity. Even humans often disagree on the categorization on the positive or negative sentiment that is supposed to be expressed on a given text (Villena et al., 2013). On the other hand, tweets are too short text to be linguistically analyzed, and it makes the task of finding relevant information (e.g. opinions) much harder.

The workshop TASS (Workshop on Sentiment Analysis at SEPLN) is an experimental evaluation workshop that includes, among other experiments, a specific task directly related to sentiment analyisis. In particular, Task 1, called, "Sentiment Analysis at global level", consists in performing an automatic sentiment analysis to determine the global polarity (using 6 levels) of each message in the test set. The 6 levels are the following: positive (P), negative (N), very positive (P+), very negative (N+), neutral (NEU), and no polarity at all (NONE). The results of our system in this task were above-average. On the other hand, task 3, "Sentiment Analysis at entity level", consists in performing an automatic sentiment analysis, similar to Task 1, but determining the polarity at entity level of each message. In this task, our system achieved the highest score among 4 participants.

In this article, we describe the learning strategies we developed so as to perform these two tasks.

2 Naive Bayes Classifier

Most of the algorithms for sentiment analysis are based on a classifier trained using a collection of annotated text data. Before training, data is preprocessed so as to extract the main features. Some classification methods have been proposed: Naive Bayes, Support Vector Machines (SVM), KNN, etc. However, and according to (Go et al., 2009), it is not clear which of these classification strategies is the more appropriate to perform sentiment analysis.

We decided to use a classification strategy based on Naive Bayes (NB) because it is a simple and intuitive method whose performance is similar to other approaches. NB combines efficiency (optimal time performance) with reasonable accuracy. The main theoretical drawback of NB methods is that it assumes conditional independence among the linguistic features. If the main features are the tokens extracted from texts, it is evident that they cannot be considered as independent, since words co-occuring in a text are somehow linked by different types of syntactic and semantic dependencies. However, even if NB produces an oversimplified model, its classification decisions are surprinsingly accurate (Manning, Raghadvan, y Schütze, 2008).

3 Preprocessing

As we will describe in the next section, the main features of the model are lemmas extracted using lemmatization. Given that the language of microblogging requires a special treatment, we propose a pre-processing task to correct and normalize the tweets before lemmatizing them.

The main preprocessing tasks we considered are the following:

- removing urls, references to usernames, and hashtags
- reduction of replicated characters (e.g. *amoooor* $\rightarrow$ *amor*)
- normalizing the text by using a small list of abbreviations (e.g. *x* $\rightarrow$ *por*)
- identifying emoticons and interjections and replacing them with polarity or sentiment expressions (e.g. *:-)* $\rightarrow$ *good*)

4 Features

4.1 Lemmas (UL)

To characterise the features underlying the classifier, we make use of unigrams of lemmas instead of tokens to minimize the problems derived from the sparse distribution of words. Moreover, only lemmas belonging to lexical categories are selected as features, namely nouns, verbs, adjectives, and adverbs. So, grammatical words, such as determiners, conjunctions, and prepositions are removed from the model.

To configure the feature representation, the frequency of each selected lemma in a tweet is stored.

4.2 Multiwords (MW)

There is no agreement on which the best option (unigrams?, bigrams?, ...) is for sentiment analysis. In (Pak y Paroubek, 2010),

the best performance is achieved with bigrams, while (Go et al., 2009) show that the better results are reached with unigrams. An alternative option is to make use of a selected set of n-grams (or multiwords) identified by means of regular patterns of PoS tags. Multiword expressions identified by means of PoS tags patterns can be conceived as linguistically motivated terms, since most of them are pairs of words linked by syntactic dependencies.

So, in addition to unigrams of lemmas, we also consider multiwords extracted by an algorithm based on patterns of PoS tags. In particular, we used the following set of patterns:

- NOUN-ADJ
- NOUN-NOUN
- ADJ-NOUN
- NOUN-PRP-NOUN
- VERB-NOUN
- VERB-PRP-NOUN

The instances of bigrams and trigrams extracted with these patterns ared added to the unigrams to build the language model. Multiword extraction was performed using our tool *GaleXtra*[1], released under GPL license and described in (Barcala et al., 2007).

4.3 Polarity Lexicon

We build a polarity lexicon with both *positive* and *negative* entries from different sources:

- Spanish Emotion Lexicon (SEL) (Sidorov, 2012) contains 2,036 words that are associated with a probability with respect to at least one basic emotion: joy, anger, fear, sadness, surprise, and disgust. In order to transform emotions into polarity values (positive and negative), words denoting joy were tagged as "positive", while words referring to anger, fear, sadness, and disgust were tagged as "negative". Most of words denoting surprise were removed from the lexicon since they are either ambiguous or not relevant concerning polarity. As a result, we generated a polarity lexicon derived from SEL with 1,890 lemmas, 722 of them are positive and 1,168 are negative.
- A list of synonyms (ExpandSEL) was automaticaly extracted by expanding SEL using a dictionary of synonyms, called *Fuzzy Dictionary of Synonyms and Antonyms*, and an expansion algorithm described in (Lanza, Graña, y Sobrino, 2003).
- A list of polarity words was semi-automatically extracted from the training corpus (CorpusLex). We built two ranked lists with most frequent words occurring in, respectively, positive and negative tweets. Then, the two lists were manually revised by selecting only polarity words.
- Finally, we also used a polarity lexicon automatically generated by using the multilingual sense-level aligned WordNet structure (WNLex). This lexicon was generated within the work described in (Perez-Rosas, Banea, y Mihalcea, 2012). As this resource contains 10% errors, it has not been expanded with synonyms so as to prevent error expansion.

In Table 1, the size of the polarity lexicons are shown. Let us note that the final dictionary is the union (by removing word duplications) of its parts. Our polarity lexicons only contains lemmas with the two basic polarity: positive or negative. There is not neutral words nor strenght values (e.g. "+" or "−").

The final polarity lexicon L is used in two different ways.

LEX1 Only those lemmas that are found in both the training corpus and L are selected as unigram features. So, lexicon L allows us to reduce the feature space by reducing the computational cost of the system. Besides, in (Saralegi y Vicente, 2012) it was claimed that a feature space with only polarity words contains less noisy features and, then, allows the system to achieve better precision. The Results obtained in the experiments described in section 6 will show that this is not always true.

LEX2 Since not all the words in the lexicon L are found in the training cor-

[1] http://gramatica.usc.es/~gamallo/gale-extra/index.htm

polarity	SEL	ExpandSEL	CorpusLex	WNLex	*L* (Final Lexicon)
positive	722	610	331	476	1852
negative	1302	838	247	871	2712
Union	1890	1448	578	1347	4564

Table 1: Polarity lexicons

pus, we have built artificial tweets as follows: each new artificial tweet consists of just one lemma of L, its polarity (positive or negative) in the lexicon, and an estimated frequency, namely the average frequency of lemmas in the training corpus. Given that these tweets are only provided with binary values (positive and negative), the features selected from them should be used for training binary classifiers with just only positive and negative categories.

4.4 Valence Shifters (VS)

We take into account negative words that can shift the polarity of specific lemmas in a tweet. In the presented work, we will make use of only those valence shifters that reverse the sentiment of words, namely *negations*. The strategy to identify the scope of negations relies on the PoS tags of the negative word as well as of those words appearing to its right in the sequence. The algorithm is as follows:

Whenever a negative word is found, its PoS tag is considered and, according to its syntactic properties, we search for a polarity word (noun, verb, or adjective) within a window of 2 words after the negation. If a polarity word is found and is syntactically linked to the negative word, then its polarity is reversed. For instance, if the negation word is the adverb "no", the system only reverses the polarity of verbs or adjectives appearing to its right. Nouns are not syntactically linked to this adverb. By contrast, if the negation is the determiner "ninguno", only the polarity of nouns can be reversed. Our strategy to deal with negation scope is not so basic as those described in (Yang, 2008) and (Anta et al., 2013), which are just based on a rigid window after the negation word: 1 and 3 words, respectively.

5 Strategies

Three different naive-bayes classifiers have been built, according to three different strategies:

Baseline This is a naive-bayes classifier that learns from the original training corpus how to classify the six categories found in the corpus: positive (P), strong positive (P+), negative (N), strong negative (N+), neutral (NEU), and no polarity at all (NONE). So, no modification has been introduced in the training corpus.

Binary The second classifier was trained on a simplified training corpus. Two reductions were made on the corpus: P+ and N+ tweets were converted to just P and N, respectively, and both NEU and NONE tweets were not taken into account. As a result, a basic binary classifier was trained that only identifies both Positive and Negative tweets. In order to account for degrees of polarity, the probability values given by the classifiers were normalized, and some probability thresholds were empirically set by considering the category distribution in the training corpus. More precisely, to take into account the six categories found in the annotated corpus, we set three thresholds: low positive and negative values are classified as NEU; high positive values are taken as P+, while high negative values are considered as N+. In addition, in order to detect tweets with and without polarity, the following basic rule is used: if the tweet contains at least one word that is aso found in the polarity lexicon, then the tweet has some degree of polarity. Othewise, the tweet has no polarity at all and is classified as NONE. The binary classifier is actually suited to specify the basic polarity between positive and negative, reaching a precision closed to 90% in a corpus with just these two categories.

2Binaries The third type of classifier is similar to the previous one (Binary), except for the NONE category, which is detected by a different binary classifier

that decides whether a tweet has polarity (YES) or not (NONE). So, tweet classification consists of two steps: first, a classifier identifies NONE tweets and sends those that are provided with polarity to a basic positive-negative classifier, including the same type of thresholds as in the previous strategy: the thresholds allows us to detect P+, N+, and NEU.

6 Experiments

6.1 Training corpus

In our experiments we have used as input data set the training corpus of tweets provided for the TASS workshop at SEPLN 2013 conference. This set contains 7216 tweets, which were tagged with polarity values among 6 categories: P+ (strong positive), P (positive), NEU (neutral), N (negative), N+ (strong negative), and no sentiment (NONE). The distribution of the polarity categories are shown in table 2.

Polarity	numb. of tweets	% of tweets
P+	1764	24%
P	1019	14%
NEU	610	8%
N	1221	17%
N+	903	13%
NONE	1702	24%
Total	**7219**	**100%**

Table 2: Distribution of polarity categories in the training corpus

6.2 Evaluated classifiers

We have implemented and evaluated five classifiers by making use of the three strategies described in section 5, as well as the features defined in 4. The five classifiers are the following:

BASE-LEX1 This system was implemented on the basis of the "Baseline" strategy and the following two features: unigrams of lemmas (UL), polarity lexicon used as in LEX1 (i.e., with feature reduction), and valence shifter (VS).

BIN-LEX1 It relies on the Binary strategy with the following features: unigrams of lemmas (UL), polarity lexicon used as in LEX1, and valence shifter (VS).

BIN-LEX2 It relies on the Binary strategy with the following features: unigrams of lemmas (UL), polarity lexicon used as in LEX2, and valence shifter (VS).

BIN-LEX2-MW It relies on the Binary strategy with the following features: unigrams of lemmas (UL), multiwords (MW), polarity lexicon used as in LEX2, and valence shifter (VS).

2BIN-LEX2 It relies on the 2Binary strategy with the following features: unigrams of lemmas (UL), polarity lexicon used as in LEX2, and valence shifter (VS).

All the classifers have been implemented with Perl language. They rely on the naive-bayes algorithm and incorporate the preprocessing tasks defined in section 3. In previous experiments, we observed that the strategy based on LEX2 works slightly better than than based on LEX1. On the other hand, it is not trivial to adapt LEX2 to the Baseline strategy, because our polarity lexicons only contain either positive or negative tokens. So, we decided to implement the Baseline classifier using just LEX1.

6.3 Evaluation

To evaluate the classification performance of the five classifiers, a 10-fold cross-validation procedure has been conducted on the training dataset. The results are shown in table 3, where the names of the evaluated systems are in the first column, the global accuracy obtained counting just 4 categories (P, NEU, N, and NONE) is shown in the second column, the global accuracy computing from 6 categories (P+, P, NEU, N, N+, and NONE) is in the third column, and the specific fscores of the 6 categories are in the remaining columns.

For 4 polarity levels, the best classifiers are both BIN-LEX2 and BIN-LEX2-MW. The difference betweent these two systems is not statistically significant (paired t-test with $p < 0,05$ inferred as significant), but BIN-LEX2 uses less features than BIN-LEX2-MW and, thereby, is more efficient in terms of computational cost. For 6 polarity levels, the highest accuracy is reached by BASE-LEX1 while the second one is by BIN-LEX2.

Let us note that the classifiers relying on the Binary strategy achieve the best performance when they are used on only 4 cate-

System	Acc (4 cat.)	Acc. (6 cat.)	P+	P	NEU	N	N+	NONE
BASE-LEX1	.592	**.439**	.57	.22	.06	.369	.38	.545
BIN-LEX1	.6	.421	.548	.037	.035	.345	.346	.566
BIN-LEX2	**.617**	.426	.566	.047	.05	.316	.391	.567
BIN-LEX2-MW	**.617**	.425	.563	.037	.05	.307	.398	.567
2BIN-LEX2	.581	.363	.543	.092	.066	.35	.393	.278

Table 3: Accuracy obtained from the training corpus for both 4 and 6 polarity levels, displayed in the 2nd and 3rd columns, respectively. Columns 4th to 9th show F-scores of each polarity level

.

System	Acc (4 cat.)	Acc. (6 cat.)	P+	P	NEU	N	N+	NONE
BASE-LEX1	.621	.54	.69	.22	.109	.464	.484	.522
BIN-LEX1	.635	.546	.679	.026	.037	.364	.349	.602
BIN-LEX2	**.668**	**.558**	.72	.033	.028	.323	.392	.613
BIN-LEX2-MW	.654	.544	.703	.034	.035	.375	.39	.602
2BIN-LEX2	.566	.442	.701	.031	.029	.406	.375	.185

Table 4: Accuracy obtained from the test data for both 4 and 6 polarity levels, displayed in the 2nd and 3rd columns, respectively. Columns 4th to 9th show F-scores of each level

.

gories, i.e. when they are used to indicate whether a text expresses a positive, negative, or neutral sentiment, or no sentiment at all. By contrast, their performance deacreases significatively when they make use of the 6 polarity levels. As it was expected, these classifiers are able to distinguish with high accuracy between positive and negative texts, but they are not suited to detect polarity at finer granularity levels, such as weak and strong values within positive and negative statements.

The worst system is that based on the 2Binaries strategy: 2BIN-LEX2. As the last column shows, the fscore of NONE value is very low, probably because the first classifier (YES or NONE polarity) does not work properly. This is due to the fact that the task of distinguishing between polarity and none polarity is more difficult than that of distinguishing between positive and negative. NONE tweets are indeed very heterogeneous and sparse. Besides, the errors of the first classifier introduces too much noise in the following steps of the classification process.

The results of both BASE-LEX1 and BIN-LEX2 were sent to participate in the task1 at the workshop TASS at SEPLN2013. Table 4 depicts the results on the evaluation of the test data. In this case, the best system in the two experiments (4 and 6 categories) is BIN-LEX2, which achieved the 3rd best performance in the task1-3l (just four categories), among 14 participants.

6.4 Polarity at the entity level

We also sent the results of BIN-LEX2 to participate in the sentiment analysis task at entity level (task3), which consists in performing an automatic sentiment analysis determining the polarity at entity level of each message in a different test corpus as task1. However, as Twitter messages are very short, we considered that the global polarity of the whole message should also be the polarity of the named entities occurring in the message. So, the entities found in a tweet were assigned the same polarity as the global tweet.

In task3, our system achieved the highest score among 4 participants: 41% accuracy.

7 Conclusions

We have presented a family of naive-bayes classifiers for detecting the polarity of Spanish tweets. The experiments have shown that the best performance is achieved by using a 2 level classifier trained to detect just two categories: positive and negative. To identify further polarity levels, the system is provided with experimentally set thresholds for detecting strong, average, and weak (or neutral) values. In addition, in order to detect tweets with and without polarity we can use a very basic strategy based on searching for polarity

words within the text/tweet. If the text contains at least one word that is also found in an external polarity dictionary, then the text has some degree of polarity. Otherwise it is tagged with the NONE value.

This strategy is well suited to deal with coarse granularity polarity detection. Its performance is significatively better when dealing with 4 (instead of 6) classification levels, i.e. when the objective is to detect positive, negative, neutral, and none polarity values. Our system improves 11 points in the test evaluation, from 55% with 6 levels to 66% accuracy with 4 levels, while the improvement average of the six best systems at the TASS competition is merely 8 points. In fact, the main drawback of our binary strategy is the use of thresholds for detecting degrees of polarity: strong positive and negative (high probability values), as well as neutral (low probability values). It follows that the best performance of our classifier with regard to the other competitors should be achieved with only 3 sharp categories: positive, negative, and none. In this experimental context, our system would not require the use of any unreliable threshold.

Bibliografía

[Anta et al.2013] Anta, Antonio Fernández, Luis Núnez Chiroque, Philippe Morere, y Agustń Santos. 2013. Sentiment analysis and topic detection of spanish tweets: A comparative study of nlp techniques. *Procesamiento del Lenguaje Natural*, 50:45–52.

[Barcala et al.2007] Barcala, M., E. Domínguez, P. Gamallo, M. López, E. Moscoso, G. Rojo, P. Santalla, y S. Sotelo. 2007. A corpus and lexical resources for multi-word terminology extraction in the field of economy. En *3rd Language & Technology Conference (LeTC'2007)*, páginas 355–359, Poznan, Poland.

[Go et al.2009] Go, Alec, Richa Bhayani, , y Lei Huang. 2009. Twitter sentiment classification using distant supervision. En *CS224N Technical report*. Standford.

[Lanza, Graña, y Sobrino2003] Lanza, S. Fernández, J. Graña, y A. Sobrino. 2003. Introducing FDSA (fuzzy dictionary of synonyms and antonyms): Applications on information retrieval and stand-alone use. *Mathware and Soft Computing*, 10(2-3):57–60.

[Manning, Raghadvan, y Schütze2008] Manning, Chris, Prabhakar Raghadvan, y Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, MA, USA.

[Pak y Paroubek2010] Pak, Alexander y Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. En *LREC-2010*.

[Perez-Rosas, Banea, y Mihalcea2012] Perez-Rosas, Veronica, Carmen Banea, y Rada Mihalcea. 2012. Learning sentiment lexicons in spanish. En *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.

[Saralegi y Vicente2012] Saralegi, X. y I. San Vicente. 2012. TASS: Detecting sentiments in spanish tweets. En *Workshop on Sentiment Analysis at SEPLN 2012*.

[Sidorov2012] Sidorov, Grigori. 2012. Empirical study of opinion mining in spanish tweets. *Lecture Notes in Artificial Intelligence*, 7629-7630.

[Villena et al.2013] Villena, J., S. Lana, E. Martínez, y J-C. González. 2013. TASS - workshop on sentiment analysis at SEPLN. *Procesamiento del Lenguaje Natural*, 50:37–44.

[Yang2008] Yang, K. 2008. WIDIT in TREC 2008 blog track: Leveraging multiple sources of opinion evidence. En *The Seventeenth Text Retrieval Conference (TREC-2008)*.