5-2020

Lexicon Based Approaches to Sentiment Analysis of Spanish Tweets: A Comparative Study

Jean Roca

Lexicon Based Approaches to Sentiment Analysis of Spanish Tweets: A Comparative Study

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
04, 2020

by

Jean Pierre Roca

Abstract

Sentiment analysis is a natural language processing technique that aims to classify text based on the emotions expressed in them. It is a research area that has been around for almost 20 years and has seen a lot of development. The works presented in this paper attempts to target a less-developed area in sentiment analysis known as multilingual sentiment analysis. More specifically, multilingual sentiment analysis of micro-texts. Using the existing WordNet lexicon and a domain-specific lexicon for a corpus of Spanish tweets, we analyze the effectiveness of these techniques.

1. Introduction

1.1 Background

Sentiment analysis is a field of natural language processing (NLP) that is very active. It provides researchers with a way to extract emotions and opinions from text in a variety of ways with great accuracy due to its recent development. One of the main issues with sentiment analysis is that it is very language dependent and much of the efforts in this field have utilized English text. Emotions and attitudes are not expressed the same across languages. The lack of resources for sentiment analysis in other languages has led to a growing interest in a field known as multilingual sentiment analysis.

The work presented in this paper targets Spanish micro-texts and compares the effectiveness of existing classifiers using resources originally developed for English texts with a domain-specific classifier. The data we will be using is a corpus of tagged Twitter publications originally written in Spanish. Twitter is a popular social media platform in which users can express their thoughts and opinions to their followers via short publications. These publications are known as "*tweets"* and they take the form of micro-texts with a maximum character length of 280 as of late 2017. It is important to note, however, that the data used in this research was made available by TASS 2017: Workshop on Semantic Analysis at SEPLN which at the time had gathered tweets at their previous maximum character length of 140.

The reduced length of these texts poses a challenge for most classifiers that are typically developed using larger amounts of text. Additionally, Spanish tweets tend to use more SMS

abbreviations and slang due to the nature of the platform. The research presented in this paper will analyze the impact of these challenges on the accuracy of lexicon-based classification.

1.2 Goals

The work presented in this paper aims to accomplish three goals. The first goal will be to determine the accuracy of TextBlob's sentiment classifier using a Spanish corpus of tagged tweets.

The second goal will be to generate a domain-specific lexicon for the Spanish tweets and use this lexicon to classify and determine the accuracy against the same corpus of Spanish tweets.

The third goal will be to compare the effectiveness of these classifiers with different lexicons and determine possible areas of improvement with hopes of developing a more accurate sentiment classifier for Spanish micro-texts.

2. Related Work

2.1 Sentiment Analysis of Tweets

There are three main approaches that people can take when it comes to sentiment classification. These approaches are known as the lexicon approach, machine learning approach, and the hybrid approach. The machine learning approach is considered a supervised learning approach while the lexicon approach is considered an unsupervised learning approach and the hybrid approach a mix of the two as the name implies.

The effectiveness of the machine learning approach on tweets was most recently investigated in [3] where an accuracy of 84.15% was achieved. This is a slight drop from [2] published 2 years earlier where an accuracy of 87.42% was achieved with a more advanced machine learning technique known as gradient boosting. In this paper we briefly explore the accuracy of a simpler machine learning approach utilizing a Naïve Bayes classifier as implemented in the Natural Language Tool Kit (NLTK) for Python.

The most recent works in lexicon approaches for tweets in languages other than English can be seen in [4] and [5]. In [4] the authors were able to achieve a maximum accuracy of 64.5% on the same data set we used for this paper using only a lexicon for detection and correction of frequent errors expressed in tweets. In [5] published more recently the authors took on a similar approach to ours where they used a sentiment dictionary in English and then manually translated them to Turkish words to use on their Turkish tweet data set and obtain a maximum accuracy of 57%. In this paper we explore improvements on these techniques that are capable of yielding higher accuracies for classifying tweets.

2.2 Domain-Specific Sentiment Lexicon

Words and their meaning vary greatly based on the domain or context they are found in. It is common to find a variation of “really cold A/C” in a listing about a car. In this context the word *cold* is seen as positive and therefore one can associate a positive sentiment value to the word. However, when we describe a person as being *cold* this is typically seen as a negative trait that would cause the word *cold* to have a negative sentiment value when used in this context.

When the context of a word is not considered in sentiment analysis, the results are not as accurate as they could be. The use of a domain-specific lexicon has proven to improve the accuracy of classifiers in the past [1]. Building a domain-specific lexicon helps determine the most accurate sentiment value for a word in a given context.

The research presented in this paper utilizes techniques for generating domain-specific lexicons developed as discussed in [1] with some minor modifications to work for the data set provided. The effectiveness of this domain-specific lexicon will be compared to the effectiveness of an existing English lexicon that does not consider context in an effort to obtain a more accurate sentiment classification.

3. Implementation and Methods

3.1 Pre-Processing

Figure 1 illustrates what a tweet looks like before pre-processing. In this experiment we were required to restructure the data provided in order to perform our analysis on sentiment classification. As we can observe, the tweets found in the TASS data set were in XML format and included various fields such as user, language, and date that did not provide meaningful insight to us. Therefore, we extracted the tweet ID, sentiment polarity, and tweet from this tagged data set and stored it in a separate file. For this experiment we chose to ignore the tweets tagged with neutral sentiment and focused solely on the positive and negative tweets.

Figure 1 (Original Tweet)

Once we had our data separated, we cleaned up the tweets some more in order to remove any whitespaces and made sure all the tweets were in UTF-8. This was an important step as the Spanish tweets contained some illegal characters that were initially not understood by Python. It is also important to note that we did not remove any stop words, punctuations, or links as these

would be later accounted for in our algorithms for classification. Once this pre-processing was done, we stored the clean tweets in a 3-tuple list as demonstrated in Figure 2.

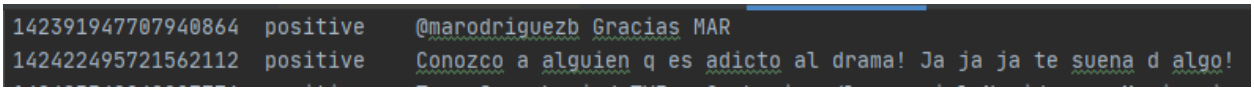


Figure 2 (Processed Tweet)

3.2 Classifiers

The first classifier that was developed tested the accuracy of the popular natural language processing library known as TextBlob. This Python library contains an extensive polarity lexicon in English and also the ability to translate text into English using the Google Translate API. Table 1 shows the entries for the word “cool” that are found within TextBlob’s lexicon.

Word	Polarity	Subjectivity	Sense
cool	0.6	0.9	"fashionable and attractive at the time"
cool	0.1	0.4	"neither warm nor very cold"

Table 1 (Lexicon Entry for “cool”)

TextBlob uses its built-in sentiment lexicon in order to extract a polarity value for a given word in a sentence and then determines the polarity of a sentence using these values for all the words in the sentence. In this example the word “cool” has two different entries in the sentiment lexicon. In order to calculate either positive or negative sentiment we pay attention to the polarity values. At the time of classifying, TextBlob will average the polarity values for each word in a sentence and then multiply them up. If the result is positive, then the sentence has a positive value. Example 1 shows how a tweet gets classified using this approach from start to finish.

1. {“Muy buenos dias!”, “pos”} //gathered from data set
2. {“Very good days!”, “”} //translate with Google API and remove classification
3. Polarity = polarity(very) * polarity(good) = 1.3 * 0.7 = 0.91 //calculate polarity
4. If polarity value is > 0 then tweet is positive //classify tweet
5. If classifications match, then tweet has been correctly classified //determine accuracy

Example 1 (TextBlob Translation Classifier)

In our classifier, we first mapped each tweet from our data set as a key with its tagged value of either positive or negative. After this, we translated the tweet using TextBlob’s translation method and extracted the sentiment polarity value returned for this now English tweet. This value is a value within the range of -1.0 and 1.0 where negative values represent negative sentiment polarity and vice versa. In some cases, polarity values can surpass 1.0, in which case they max out at 1.0 or set at a minimum -1.0 if they were to go below. Additionally, words that do not appear in the sentiment lexicon get ignored as “days” did in Example 1. For the last step, we mapped the translated tweet with its sentiment value and compared it to the original Spanish tweets sentiment value to determine accuracy. Due to the limitations of Google’s API needed for translation, we chose to only use half of the data set or roughly about 2500 tweets for this part of the experiment.

The second classifier developed used the Naïve Bayes classifier [7] as implemented in the NLTK for Python. This machine learning technique relies on a theorem of probability known as Bayes' Theorem that relies on conditional probability outlined as so:

$$P(positive|me\ gusta\ la\ comida) = \frac{P(me\ gusta\ la\ comida|positive)P(positive)}{P(me\ gusta\ la\ comida)}$$

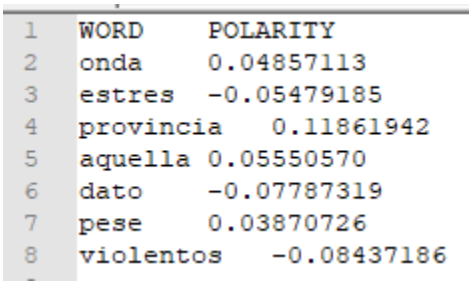
These probabilities are further broken down when we consider the "naïve" part of the algorithm. This part assumes that each word found in a sentence is independent of the rest. The above formula broken down in the Naïve Bayes' classifier would become:

$$P(positive|me\ gusta\ la\ comida)$$
$$= \frac{P(me|positive)P(gusta|positive)P(la|positive)P(comida|positive)P(positive)}{P(me\ gusta\ la\ comida)}$$

These individual words will be repeated among the data set several times and therefore we are able to calculate probabilities for each of them and train the classifier with this data. The classifier was trained using 80% of our data (4000 tweets) and then tested on the remaining 20% of the data (1000 tweets).

Finally, the last classifier that was implemented focused on building a domain-specific lexicon for this set of data and using this lexicon to classify the tweets. The techniques used to generate this lexicon were a combination of text mining and information theoretic techniques as detailed

in [1]. There was some slight modifications that were made in order to generate the lexicon for this data set, mainly encoding of tweets, but once we obtained the lexicon with sentiment values for the most frequent words found in the data set we tested the lexicon. Figure 3 shows a small portion of the lexicon that was generated. This lexicon consisted of 3456 unique words that were found most often in the data set.



	WORD	POLARITY
1		
2	onda	0.04857113
3	estres	-0.05479185
4	provincia	0.11861942
5	aquella	0.05550570
6	dato	-0.07787319
7	pese	0.03870726
8	violentos	-0.08437186

Figure 3 (Domain-Specific Lexicon)

The algorithm developed for this part, as seen on Example 2, was simple and consisted in adding the polarity values for each word found in the tweet. If the sum of all these words turned out to be positive, then we would assign a positive value to the tweet and vice versa. The tweets were mapped to their values after performing this addition and compared against the original value for that tweet to determine accuracy. Single letters, hyperlinks, and emojis were ignored for this experiment.

1. {“Muy buenos dias!”, “pos”} //gathered from data set
2. Polarity = polarity(muy) + polarity(buenos) + polarity(dias) = 0.097 + 0.24 + 0.18 = 0.517 //calculate polarity
3. If polarity value is > 0 then tweet is positive //classify tweet
4. If classifications match, then tweet has been correctly classified //determine accuracy

Example 2 (Domain-Specific Classifier)

Generating the lexicon is the most challenging part of this classifier, but once that is done classifying the tweets becomes a matter of simple addition. It is important to note that generating the lexicon is also affected by a user-established threshold. For this experiment a value of 3 was used for the threshold. This meant that in order for a word to appear in our lexicon, it had to have shown up at least 3 times in the tweets data set. Outside of disregarding words that were not frequent in our data set, this threshold also allowed us to account for potential typos that some tweets might have. The classifier saw its best accuracy when using this threshold.

4. Evaluations

4.1 Data Set

To begin our experiment the first step was to obtain a large enough data set of Spanish tweets. Initially, our approach was to utilize the Twitter developer API in order to actively collect tweets for a given language. This approach was successful in gathering tweets, however, due to limitations on the usage of the API we would not be able to gather a large enough sample set of data. Additionally, these tweets would not be tagged with a sentiment value and would require manual tagging for each one. This is what led us to use the publicly available TASS 2017 data set of tagged Spanish tweets. This is the same data set used in [4] and is composed of 2870 positive tweets and 2182 negative for a total of 5052 tweets used in our research. These 5052 tweets had a total of 24,780 unique tokens of size n >= 2.

4.2 Metrics

For the purpose of this experiment our evaluation metric will be determined by the accuracy of each of the classifiers we developed and tested. Accuracy will be a ratio of the number of correct classifications over the number of total tweets for that classification. The classifications we will be measuring in this experiment will just be positive and negative sentiment. Additionally, we will be measuring an overall accuracy for the classifier determined by the total number of correct classifications for both positive and negative sentiment over the total number of tweets used in the particular classifier.

4.3 Methods

All the classifiers used in this experiment handled the pre-processing of tweets the same way. They all parsed the original XML the same way to extract the tweets and worked with the same data set. Due to limitations of requests made to the Google Translate API, the TextBlob Translation Classifier had to use a little less than half of the data set. The Naïve Bayes classifier was trained with 80% of our data set and then tested on the remaining 20% of the data set. Finally, the domain-specific lexicon was generated using the entire data set and then tested against the same set of tweets and their sentiment values.

4.4 Results

The results of our TextBlob Classifier that consisted in translating the Spanish tweets and using the English sentiment lexicon can be seen in Table 2. The translation classifier had an accuracy of 66.74% for positive tweets and an accuracy of 68.29% for negative tweets for an overall accuracy of 67.14%.

	Correctly Classified	Total Tweets	Accuracy
Positive Tweets	1168	1750	66.74%
Negative Tweets	407	596	68.29%
Overall	1575	2346	67.14%

Table 2 (TextBlob Translation Classifier Results)

The results of our second classifier that utilized NLTK's implementation of Naïve Bayes can be found in Table 3. Table 4 includes the top 3 most common words found in tweets and their ratios of occurrence among negative and positive tweets. This classifier yielded a positive accuracy of 76.45%, a negative accuracy of 77.07%, and an overall accuracy of 76.70%.

	Correctly Classified	Total Tweets	Accuracy
Positive Tweets	461	603	76.45%
Negative Tweets	306	397	77.07%
Overall	767	1000	76.70%

Table 3 (NLTK Naïve Bayes Classifier Results)

	negative/positive tweet ratio
contains(gracias)	1/24.5
contains(grinan)	22.9/1
contains(deficit)	17.2/1

Table 4 (NLTK Naïve Bayes Most Common Words)

The results of our final classifier using the domain-specific lexicon are shown in Table 5. This algorithm had an accuracy of 68.40% for positive tweets and 91.70% for negative tweets for an overall accuracy of 78.46%.

	Correctly Classified	Total Tweets	Accuracy
Positive Tweets	1963	2870	68.40%
Negative Tweets	2001	2182	91.70%
Overall	3964	5052	78.46%

Table 5 (Domain-Specific Lexicon Classifier Results)

Chart 1 shows the accuracies grouped by classifiers.

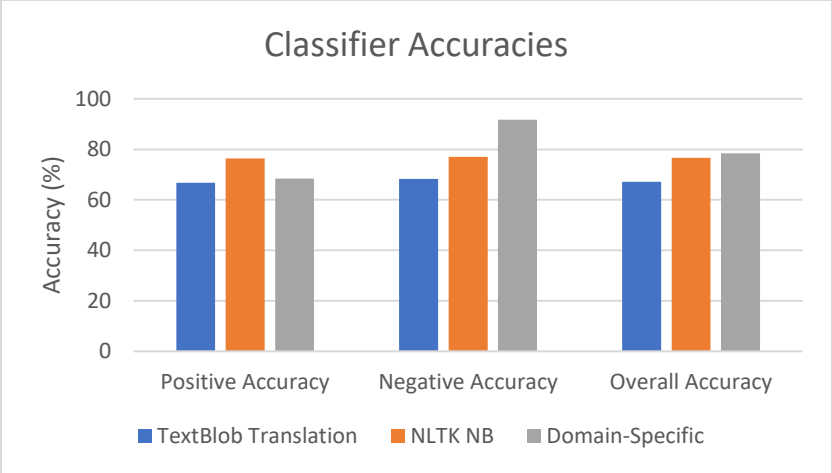


Chart 1 (Classifier Accuracies)

4.5 Analysis

The results that were obtained from our experiment are an improvement to those seen in the past for similar subjects. Our experiment has shown a significant improvement when using a domain-specific lexicon when compared to a general lexicon. This improvement is also attributed to the fact that in the domain-specific lexicon we did not require any translation from English to Spanish as we did when we got the results from Table 2. Accuracy for the TextBlob Translation classifier was impacted by the translation step. It is also interesting to note that in our domain-specific classifier we obtained a very high accuracy for negative tweet classification when compared to positive tweet classification. This led the domain specific approach to have the highest accuracy, but the Naïve Bayes classifier was the most precise. This is most likely due to the data set centered on politics in Spain at around the time they were gathered. This would lead proper nouns like “Grinan”, a Spanish politician” to carry very negative weights and as a result making negative tweets easier to identify from positive tweets.

The results from the Naïve Bayes classification were very descriptive. Not only did we get positive and negative classification accuracies, but we also got a list of the top words and their ratios of appearance for each classification. This is where we can identify very positive words like “gracias” or “thank you” in English to have high occurrences in positive tweets. As mentioned previously, this data set consisted vastly of tweets from 2011-2012 and this is why the Spanish politician Griñan appeared in many negative tweets due to political issues from that time. This also attributed to the increase in accuracy for negative tweets in this classifier.

5. Conclusion

5.1 Conclusions

The work in this paper has served as a contemporary study on different approaches for multilingual sentiment analysis. The work presented is different from other studies by utilizing a domain-specific lexicon to classify sentiment and comparing the effectiveness of this technique to previous techniques. The overall accuracy of 78.46% is greater than the accuracies found in previous work done on sentiment analysis of Spanish tweets and there is definitely some improvement that can be made to our work. The TextBlob Python library also included very helpful functions that facilitated this experiment. The overall accuracy of the translating approach was still decent at 67.14% and proved to be a viable option for classifying tweets. The goals for this experiment were met with an added bonus of exploring a machine learning approach with a Naïve Bayes classifier. The Naïve Bayes classifier proved to be more effective than the regular lexicon approach and did not require a very large data set in order to do so. Improvements that could be done in this experiment would be to consider optimizing our algorithms to consider emoticons and punctuation. Currently, we consider slang in the domain-specific approach, but we fail to acknowledge a very important part of internet speech which is expressed with emoticons and punctuation. There is not much work being done in Spanish sentiment analysis when compared to the work done in English and the work in this paper also aims to encourage people to continue researching this field with hopes of improvements.

5.2 Future Work

In the future we plan to investigate an even more specific area of sentiment analysis known as topic detection [6] as well as improving the current work we have discussed in this paper. The data set that was used in this experiment is old and outdated due to new character limitations on Twitter. The way people interact on the internet has also changed greatly since 2012 and this is something to consider with future work in this field. It would be interesting to compare the effectiveness of the techniques discussed in this paper with a more updated data set.

6. References

[1] Labille, Kevin & Gauch, Susan & Alfarhood, Sultan. (2017). Creating Domain-Specific Sentiment Lexicons via Text Mining.

[2] Kose, Suha. (2018). Twitter Sentiment Analysis.

[3] Rathaur, Rohit. (2019). twitter sentiments analysis. 10.13140/RG.2.2.32223.41126.

[4] Navas-Loro, M., & Rodríguez-Doncel, V. (2017). OEG at TASS 2017 : Spanish Sentiment Analysis of tweets at document level.

[5] Shehu, Harisu Abdullahi & Tokat, Sezai & Sharif, Md. Haidar & Uyaver, Sahin. (2019). Sentiment analysis of Turkish Twitter data. AIP Conference Proceedings. 2183. 080004. 10.1063/1.5136197.

[6] Fernández Anta, Antonio & Morere, Philippe & Chiroque, Luis & Santos, A.. (2013). Sentiment analysis and topic detection of Spanish Tweets: A comparative study of NLP techniques. Procesamiento de Lenguaje Natural. 50. 45-52.

[7] Daniel Jurafsky and James H. Martin. (2009). Speech and Language Processing (2nd Edition). Prentice-Hall, Inc., USA.