A Deep Learning Approach for Sentiment Analysis in Spanish Tweets

Gerson Vizcarra[1(✉)], Antoni Mauricio[1], and Leonidas Mauricio[2]

[1] Research and Innovation Center in Computer Science,
Universidad Católica San Pablo, Arequipa, Peru
{gerson.vizcarra,manasses.mauricio}@ucsp.edu.pe

[2] Department of Mechanical Engineering, Artificial Intelligence,
Image Processing and Robotic Lab, Universidad Nacional de Ingeniería,
Bldg. A - Off. A1-221, 210 Tupac Amaru Ave., Lima, Peru
lmauricioc@uni.pe

Abstract. Sentiment Analysis at Document Level is a well-known problem in Natural Language Processing (NLP), being considered as a reference in NLP, over which new architectures and models are tested in order to compare metrics that are also referents in other issues. This problem has been solved in good enough terms for English language, but its metrics are still quite low in other languages. In addition, architectures which are successful in a language do not necessarily works in another. In the case of Spanish, data quantity and quality become a problem during data preparation and architecture design, due to the few labeled data available including not-textual elements (like emoticons or expressions).

This work presents an approach to solve the sentiment analysis problem in Spanish tweets and compares it with the state of art. To do so, a preprocessing algorithm is performed based on interpretation of colloquial expressions and emoticons, and trivial words elimination. Processed sentences turn into matrices using the 3 most successful methods of word embeddings (GloVe, FastText and Word2Vec), then the 3 matrices merge into a 3-channels matrix which is used to feed our CNN-based model. The proposed architecture uses parallel convolution layers as k-grams, by this way the value of each word and their contexts are weighted, to predict the sentiment polarity among 4 possible classes. After several tests, the optimal tuple which improves the accuracy were <1, 2>. Finally, our model presents %61.58 and %71.14 of accuracy in InterTASS and General Corpus respectively.

Keywords: Convolutional neural network (CNN) · Sentiment analysis
Spanish tweets

The present work was supported by grant 234-2015-FONDECYT (Master Program) from Cienciactiva of the National Council for Science, Technology and Technological Innovation (CONCYTEC-PERU) and the Office Research of Universidad Nacional de Ingeniería (VRI - UNI).

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11141, pp. 622–629, 2018.
https://doi.org/10.1007/978-3-030-01424-7_61

1 Introduction

Semantic analysis has opened up several fields of research in NLP. In turn, these new fields have helped the development of comprehension systems, which include, as is explained in [1], cross- and multi-domain sentiment analysis, aspect-based sentiment analysis, fake news identification, classification of semantic relations, question answering of non-factoid questions among others. Liu [9] defines sentiment analysis or opinion mining as the computational study of people's opinions, sentiments, emotions, appraisals, and attitudes towards entities such as products, services, organizations, individuals, events, topics, and their attributes.

In sentiment analysis tasks, tweets analysis at document level is highlighted and long addressed one due to the large amounts of information about multiple topics generated in short time and its easy access (unlabeled data). Specific tasks linked to this problem has raised the interest of NLP community for several years [16]. The automatic sentiment detection in tweets is a powerful and useful tool for social networks analysis or advertising analysis and many other applications.

In this paper, we propose a CNN-based model that automatically processes short texts obtained from task 1 proposed in TASS 2017 [1] using tweets in Spanish and detects if a tweet expresses any polarity (positive, negative, neutral or none) about an specific topic. The next sections will be as follows. Section 2, covers related works in the area. Section 3, exposes our proposals (preprocessing method and architecture design) in detail. Section 4, includes final results and their analysis, and Sect. 5 presents our conclusions and future works.

2 Related Studies

Pang et al. [13] and Liu [8] provided an introduction to Sentiment Analysis area. Zhang et al. [20] have published a very complete state of art in Sentiment Analysis using deep learning approaches. They explained that sentiment analysis could be represented as a classification problem (classifying a text document on a bunch of predefined categories) and therefore addressed with different methods, Zhang et al. also mentions that black-box models such as neuronal networks and deep neuronal networks have become increasingly popular. About short texts analysis there are many papers which shows relevant results in real life applications using tweets in different languages.

Rodrigues Barbosa et al. [15] evaluates Twitter hashtags in sentiment analysis for Brazilian presidential elections in 2010. To do so, they analyzed 10,173,382 tweets labeled in 4 labels: Positive, Negative, Ambiguous and Neutral, for hashtags about candidates or events around the election day. They finally conclude that trends in Twitter over time were in accordance with the general feeling of the population. They also verified that information spreads on Twitter following a social graph model and people make their decisions consciously or not, depending on the feelings and choices of their contacts in Twitter.

Go et al. [3] introduced a method to classify Twitter messages. Positive and negative tweets are separated using emoticons labels: ":) /:-)" or ":(/:-(.". They collected 80,000 positive and 80,000 negative tweets as a training set. In preprocessing step, emoticons were removed on training process because the negative impact on precisions on the SVM and Maximum Entropy (ME) classifiers, but has insignificant effects on Naive-Bayes based classifier. Then, they segmented sentences by unigrams (word by word), bigrams (two words), unigram-bigram, and the Speech features extracted by well-known descriptors. Their results in accuracy using SVM and unigrams were 82.9%, while using unigram-bigram in ME and Naive-Bayes were 82.7%, being considered in both cases the best results for each method.

Kin [7] and Wang et al. [19] presented respectively their attempts to use convolutional (CNN) and recursive (RNN) neural networks for polarity classification in short texts, achieving quite inspiring results that define standard architectures to solve the problem. CNN architecture allows to get a fast convergence and presents, in most of cases, a remarkable performance on sentence classification. By other hand, RNN usually converges slow but it can interpret sequences of words better, that is more useful applied to text due to it could capture the context in a sentence. Lost memory or vanishing gradient is a problem for RNN. So a residual network or recurrent Long-Short Term Memory network (LSTM) [19] is capable of capturing the special functions of words avoiding lost memory problem.

In sentiment analysis of tweets at document level, Hassan et al. [5] pro- posed to merge CNN and LSTM-RNN models for shorts texts due LSTM avoid vanishing gradient problem but depending on the text size while CNN works better for very short texts, which are normally the tweets size. For IMDB opinions database, they achieved 88.3% using a single word embedding channel in binary classification. While Severyn et al. [17] explored CNN solutions using Twitter database, getting 84.79% in accuracy for phases and 64.59% in message level.

As can be seen most of works come from English datasets. In Spanish there are few works which define the state of art on TASS datasets. Navas-Loro et al. [12], and Martínez-Cámara et al. [10] resume most of works and methods developed during TASS 2017 competition. In TASS 2017, best results were obtained by neural network models. Hurtado Oliver et al. [6] obtained 60.70% in accuracy InterTASS corpus and 72.50% in General Corpus using a fully connected neural network with ReLU functions, dropout layer (p = 0.3) and polarity-specific embeddings.

3 The Problem and Data Description

Sentiment analysis task can be summarized as multi-class classification problem, considering the polarities as classes (none, neutral, positive or negative attitude expressed in a tweet). We have used the TASS 2017 database [1] in our experiments. This database was employed in the ‘Workshop on Semantic Analysis’ during the International Conference of the Spanish Society for Natural Language Processing (SEPLN). The competition goal were to classify four types of tweets polarities in task 1, which are: N - negative; P - positive; NEU - neutral and NONE - none classified. Training data is composed as follows: InterTASS (1008 tweets), General TASS (7219 tweets) and InterTASS development corpus (506 tweets), while testing data contains InterTASS test (1899 tweets) and General-TASS test (60798 tweets).

4 Methodology

In this section, we present the pre-processing methodology realized and the architecture designed.

4.1 Preprocessing

Based on Severyn et al. [17] and Navas-Loro and Rodríguez-Doncel [12], we create a tokenizer to handle trivial terms and repeated words following this steps:

- Delete URLs, extra blank spaces, special characters and repeated words.
- Change words to lowercase.
- Replace laugh expressions (like ‘jajaja’, ‘haha’, ‘LOL’, etc.) by ‘ja’.
- Replace colloquialisms by formal expressions (e.g. ‘por’ instead of ‘x’).
- Create a stop words dictionary to delete trivial words.

In addition, we replaced emoticons by words based on emoticons-clusters model proposed by Wang and Castanon [18], which statistically represents the meaning of emoticons. Table 1 plots the statistical representation of emoticons.

Table 1. Emoticons clusters and its statistical meaning from [18]

Cluster	Emoticons	Statistical meaning
A	:) :D =)	Good thanks happy fantastic lovely wonderful amazing…
B	;) :-) ;-) :-D = D ; P =] XD	Smile friends face music favorite pic kind coffee pleasure positive exciting healthy …
C	:(:/ :’) :’(:-(D: ;(:-/ :— :/	Miss sorry bad hate sad omg sick late mad ugh ugly broke
D	:P ;D :-P :] :p	What lol don’t no know think can’t why ever never look…
E	(:	Love follow please hey wish goodnight…
F	XP	Stuck shoot fatally
H	8)	Best fun coming week playing top happiness weekend…

4.2 Word Vectors

Word vectors are the numerical representation of words, which are encoded using different criteria. The most successful criteria are based on training of networks using a corpus. For our case, the corpus for embedding training is composed by “General Corpus”, “Social TV”, “STOMPOL”, and “InterTASS” datasets [1], and encoded using GloVe [14], Word2Vec [11], and FastText [2] models. The three models we selected are considered as top representations which means that related words are close at vector level.

4.3 Convolutional Neural Network

Convolutional neural networks (CNN) are networks divided into two sections: convolution and fully connected section. According to Goodfellow et al. [4], the convolutional section trains to obtain best features which represent input using linear and non-linear activation functions in ReLU or pooling layers, the last output layer in the convolution stage is the feature map (a set of complex and hard to interpret descriptors) which is used by the classifier (the fully connected section). Normally, the fully connected layer is build using a Multilayer Perceptron (MLP).

Preprocessed data is composed by keywords which implies the unigram representation (1 × N convolution) of each keyword in global polarity evaluation. Word vector models are merged into a 3-channels matrix <GloVe, FastText, Word2Vec>, which is the input of our model.

Figure 1 shows the architecture implemented. The input has <D × E> dimension, where D is the dictionary size and E is the encoding size. To obtain the k-gram analysis we apply 4 convolution layers (<k × E> dimension) in parallel. Each convolution layer needs 100 kernels to train and generates 400 feature maps. MaxPooling layer returns the maximum value per each feature map, then all outputs are flattening into a 400 × 1 vector, which is used as input for the fully connected layer. In the fully connected layer, we used a MLP with 200 neurons and ReLU activation function in the hidden layer, 4 neurons with logistic activation function in the output layer. For training, we apply a categorical cross-entropy loss to maximize the separation between classes, the ADAM training and a dropout layer (p = 0.25) to reduce complexity and avoid over-fitting.

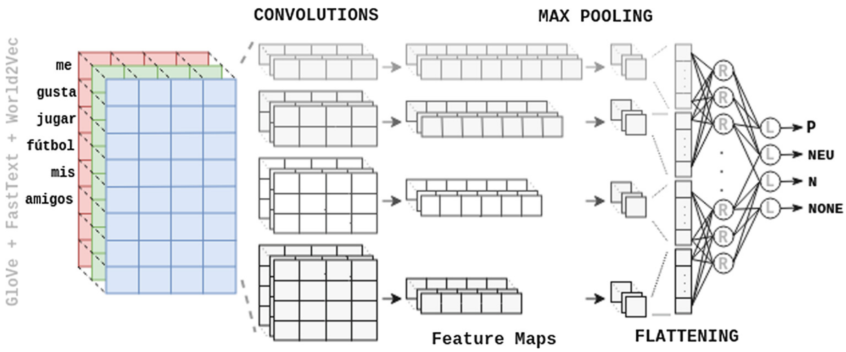

Fig. 1. Assuming that the dictionary size (D) is 8, the encoding size (E) is 4 and the four convolution layers are <<1, 2, 3, 4> × E>. Then, the preprocessed tweet 'me gusta jugar fútbol mis amigos' is classified following the pipeline

5 Experiments and Results

To run experiments, we used a PC with the following settings: 3,6 GHz Intel Core i7 processor, 16 GB 3000 MHz DDR4 memory and NVIDIA GTX 1070 and for implementation we used TensorFlow-1.5 Framework.

5.1 Filters Setting

To define the convolution filters size we performed 2 experiments. In the first one, we combine parallel filters (<<1, 2, 3, 4, 5> × E> dimensions) without repetition into groups of different sizes. To test all combinations we executed each tuple of filters with same conditions. The first experiment results are listed in Table 2.

Table 2. Best three accuracies per run using InterTASS corpus

Run	First		Second		Third	
	Combination	Result	Combination	Result	Combination	Result
1	<1, 2>	0.6182	<2, 4>	0.6131	<1, 2, 3, 4>	0.6125
2	<1, 2, 3>	0.6124	<1, 2, 4>	0.6112	<1, 2>	0.6099
3	<1, 2>	0.6163	<1, 4>	0.6128	<1, 3>	0.6118
4	<1, 2>	0.6156	<1, 3>	0.6120	<1, 2, 3, 5>	0.6114
5	<1, 2>	0.6214	<1, 2, 3, 4>	0.6144	<1, 2>	0.6134

On the second one, we selected the best tuples based on Table 2, then we tuned parameters per each tuple to get optimal results. We run ten times each tuple in order to obtain the best, worst and average accuracies. The second experiment results are showed in Table 3.

Table 3. Statistical results per tuple using InterTASS corpus

Filters	Best run	Worst run	Average
<1, 2>	**0.6219**	**0.6124**	**0.6158**
<1, 3>	0.6163	0.6035	0.6094
<1, 2, 3>	0.6175	0.6029	0.6126
<1, 2, 3, 4>	0.6118	0.5908	0.6008

5.2 Sentiment Analysis

Table 4 expose results for InterTASS and General corpus. In the contest, testing and training data were available in different packages, so results presented in Table 4 refers the testing precision, then we compare our results (CNN-EMOTIC) before the state of art (*).

Table 4. Comparative results in TASS-2017 for sentiment analysis from [1], (*) are best results in the contest and our results are in bold

Proposed system	Corpus	
	InterTASS	General
CNN-EMOTIC	**0.618**	**0.743**
ELiRF-UPV-run1	0.607	0.666
RETUYT-svm cnn	0.596	0.674
ELiRF-UPV-run3	0.597	0.725*
jacerong-run-2	0.602	0.701
jacerong-run-1	0.608*	0.706
INGEOTECevodag-001	0.507	0.514

6 Conclusions and Future Works

The results presented in this paper show that the proposed approach is efficient in sentiment analysis of tweets at document level in Spanish. Based on experiments, our CNN-based model presents an accuracy of 61.82% and 73.22% in testing for InterTASS and General Corpus. During architecture design, we used a well-known CNN-based model of the state of the art but setting a different convolutional tuples. After many runs we concluded that <1, 2> tuple is the best combination, this could be explained if we consider unigram (<1>) representation as the weight of each word and bi-gram (<2>) representation as the weight of context for short texts. The 3-channels input allows a more accurate word- vector representation of the tweet. Also this improvement was possible importing the emoticons statistical meaning [18] to our preprocessing step. During tests, those factors meant a slight but important improvement (from 59.3%–70.7% to 61.58%–74.14% in InterTASS and General corpus respectively). To improve our current results we have to integrate a semantic windows and entropy-based model for large texts, considering to break the words/emoticons according to context (not just for sentiment analysis but aspect-based sentiment analysis).

References

1. Tass 2017 homepage. http://www.sepln.org/workshops/tass/. Accessed 20 May 2018
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
3. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford **1**(12) (2009)
4. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)
5. Hassan, A., Mahmood, A.: Deep learning approach for sentiment analysis of short texts. In: 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), pp. 705–710. IEEE (2017)

6. Hurtado Oliver, L., Pla, F., González Barba, J.: Elirf-upv en tass 2017: Análisis de sentimientos en twitter basado en aprendizaje profundo, p. 6, September 2017
7. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
8. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
9. Liu, B.: Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, Cambridge (2015)
10. Martınez-Cámara, E., Díaz-Galiano, M., García-Cumbreras, M., Garcıa-Vega, M., Villena-Román, J.: Overview of TASS 2017. In: Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN (TASS 2017), vol. 1896 (2017)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Navas-Loro, M., Rodríguez-Doncel, V.: OEG at TASS 2017: Spanish sentiment analysis of tweets at document level
13. Pang, B., et al.: Opinion mining and sentiment analysis. Foundations and Trends ®. Inf. Retrieval **2**(1–2), 1–135 (2008)
14. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
15. Rodrigues Barbosa, G.A., Silva, I.S., Zaki, M., Meira Jr., W., Prates, R.O., Veloso, A.: Characterizing the effectiveness of twitter hashtags to detect and track online population sentiment. In: CHI 2012 Extended Abstracts on Human Factors in Computing Systems, pp. 2621–2626. ACM (2012)
16. Rosá, A., Chiruzzo, L., Etcheverry, M., Castro, S.: Retuyt en tass 2017: Análisis de sentimientos de tweets en español utilizando svm y cnn. In: Proceedings of TASS (2017)
17. Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 959–962. ACM (2015)
18. Wang, H., Castanon, J.A.: Sentiment expression via emoticons on social media. arXiv preprint arXiv:1511.02556 (2015)
19. Wang, X., Liu, Y., Chengjie, S., Wang, B., Wang, X.: Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1343–1353 (2015)
20. Zhang, L., Wang, S., Liu, B.: Deep learning for sentiment analysis: a survey. Wiley Interdisc. Rev. Data Min. Knowl. Discov., e1253 (2018)