Sentiment Analysis at SEPLN (TASS)-2019: Sentiment Analysis at Tweet Level Using Deep Learning

Avishek Garain and Sainik Kumar Mahata

[1] Avishek Garain
Computer Science and Engineering
Jadavpur University, Kolkata
avishekgarain@gmail.com
[2] Sainik Kumar Mahata
Computer Science and Engineering
Jadavpur University, Kolkata
sainik.mahata@gmail.com

Abstract. This paper describes the system submitted to "Sentiment Analysis at SEPLN (TASS)-2019" shared task. The task includes sentiment analysis of Spanish tweets, where the tweets are in different dialects spoken in Spain, Peru, Costa Rica, Uruguay and Mexico. The tweets are short (up to 240 characters) and the language is informal, i.e., it contains misspellings, emojis, onomatopeias etc. Sentiment analysis includes classification of the tweets into 4 classes, viz., Positive, Negative, Neutral and None. For preparing the proposed system, we use Deep Learning networks like LSTMs.

Keywords: BiLSTM · Regularizers · Sentiment Analysis · CuDNNLSTM

1 Introduction

Sentiment **A**nalysis (SA) refers to the use of **N**atural **L**anguage **P**rocessing (NLP) to systematically identify, extract, quantify, and study affective states and subjective information. The Sentiment Analysis at SEPLN (TASS)-2019 [3] was a classification task where it was required to classify a Spanish tweet on basis of its sentiment,into various classes like, Positive, Negative, Neutral and None. It was further divided into two subtasks;the first subtask being monolingual testing of system while the second task being cross-lingual testing of system. However, the task threw some additional challenges. The given tweets involved lack of context, where the number of words were less than 240. Moreover, the tweets were in an

[3] https://competitions.codalab.org/competitions/21957

informal language and contained multi-linguality. Also, the classification system that would be prepared for the task, needed to be generalized for various test corpora as well.

To solve the task in hand, we built a bidirectional **L**ong **S**hort **T**erm **M**emory (LSTM) based neural network, for prediction of the sentiments present in the provided dataset. For both the subtasks, our system categorized the instances into P, N, NEU and NONE.

The rest of the paper has been organized as follows. Section 2 describes the data, on which, the task was performed. The methodology followed is described in Section 3. This is followed by the results and concluding remarks in Section 4 and 5 respectively.

2 Data

The dataset that was used to train the model was provided by InterTASS[1]. The data was collected from Twitter and it was retrieved using the Twitter API by searching for keywords and constructions that are often included in various texts of different sentiments. The dataset provided consisted of tweets in their original form along with the corresponding P, N, NEU and NONE labels, as shown in Table 1.

Table 1. Labels used in the dataset

Label	**Meaning**
P	Positive
N	Negative
NEU	Neutral
NONE	None

The dataset originally comprised of Spanish tweets of various dialects, namely ES(Spain), PE(Peru), CR(Costa Rica), UR(Uruguay) and MX(Mexico). The tweets were also tagged with their respective sentiments. We merged all this data and shuffled them. The resulting dataset had 7,265 sentiment tagged tweets, which were splitted into 5,086 instances of training data and 2,179 instances of development data. Our approach was to convert the tweets into a sequence of words and convert them into word embeddings. We then run a neural-network based algorithm on the processed tweet. Language and label based categorical division of data is given in Table 2, 3, 4 and 5.

The provided training and development data were merged and shuffled to create a bigger training set, and we refer to the same as training data in the methodology section.

Table 2. Training and Development data used for the system

Label	Train	Development
Spain	1125	581
Peru	966	498
Costa Rica	777	390
Uruguay	943	486
Mexico	989	510

Table 3. Distribution of the labels in the training dataset

Value	P	NEU	N	NONE
All	1994	710	1483	898

Table 4. Distribution of the labels in the development dataset

Value	P	NEU	N	NONE
All	850	297	615	418

Table 5. Distribution of the labels in the combined dataset

Value	P	NEU	N	NONE
All	2844	1007	2098	1316

3 Methodology

The first stage in our model was to preprocess the tweets. For the preprocessing steps, we took inspiration from the work on Hate Speech against immigrants in Twitter[4], part of SemEval2019. The steps used here are built as an advancement of this work. It consisted of the following steps:

1. Removing mentions
2. Removing URLs
3. Contracting whitespace
4. Extracting words from hashtags

The last step (step 5) consists of taking advantage of the Pascal Casing of hashtags (e.g. `#TheWallStreet`). A simple regex can extract all words; we ignore a few errors that arise in this procedure. This extraction results in better performance mainly because words in hashtags, to some extent, may convey sentiments of hate. They play an important role during the model-training stage.

We treat each tweet as a sequence of words with interdependence among various words contributing to its meaning. We convert the tweets into one-hot vectors. We also include certain manually extracted features listed below:

1. Counts of words with positive sentiment, negative sentiment and neutral sentiment in Spanish
2. Counts of words with positive sentiment, negative sentiment and neutral sentiment in English
3. Subjectivity score of the tweet
4. Number of question marks,Exclamations and full-stops in the tweet

For this, we used SenticNet5[2] for finding sentiment values of individual words after converting the sentences to English via GoogleTrans API. Apart from this, we also used a Spanish Sentiment lexicon for the same.

The use of BiLSTM networks is a key factor in our model. The work of [5] brought a revolutionary change by bringing the concept of memory into usage for sequence based problems. We were guided by the work of [6] who used a CNN+GRU based approach for a similar task. We use an approach which was influenced by this work to some extent.

We use a bidirectional LSTM based approach to capture information from both the past and future context followed by an Attention layer consisting of initializers and regularizers.

Our model is a neural-network based model. Initially, the manual feature vectors are appended with the feature vector obtained after converting the processed tweet to one-hot encoding. It is then passed through an embedding layer which transforms the tweet into a 128 length vector. The embedding layer learns the word embeddings from the input tweets. We pass the embeddings through a Batch Normalization layer of dimensions 10 X 128. This is followed by one bidirectional LSTM layer containing 128 units with its dropout and regular dropout set to 0.4 and activation being a sigmoid activation. This is followed by a Bidirectional CuDNNLSTM layer with 64 units for better GPU usage. This is followed by the final output layer of neurons with softmax activation, where, each neuron predicts a label as present in the dataset.

For both subtasks 1 and 2, we train a model containing 4 neurons for predicting `P(0/1)`, `N(0/1)` and `NEU(0/1)` and `NONE(0/1)` respectively. After the CuDNNLSTM layer we have added a regularizing layer which is initialized with glorot_uniform initializer. This layer has both W_regularizers and b_regularizers to prevent the model from overfitting. This provides better validation and test results by generalizing the feature learning process. The model is compiled using the Adam optimization algorithm with a learning rate of 0.0005. Categorical-crossentropy is used as the loss function. The working is depicted in Figure 1.

We note that the dataset is highly skewed in nature. If trained on the entire training dataset without any validation, the model tends to completely overfit to the class with higher frequency as it leads to a higher accuracy score.

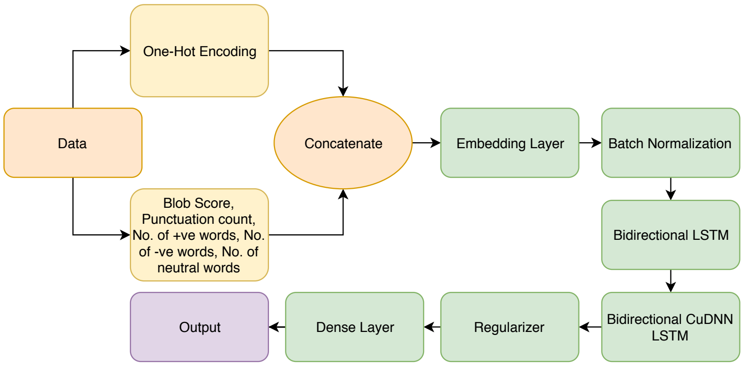

Fig. 1. Flowchart of working model

To overcome this problem, we took some measures. Firstly, the training data was split into two parts — one for training and one for validation comprising 70 % and 30 % of the dataset respectively. The training was stopped when two consecutive epochs increased the measured loss function value for the validation set.

Secondly, class weights were assigned to the different classes present in the data. The weights were approximately chosen to be proportional to the inverse of the respective frequencies of the classes. Intuitively, the model now gives equal weight to the skewed classes and this penalizes tendencies to overfit to the data.

4 Results

We participated in subtasks 1 and 2 of Sentiment Analysis at SEPLN (TASS)-2019 and our system ranks first among the competing participants.

We have included the automatically generated tables with our results. The results(rounded off to 3 decimal places) are depicted in Table 6, 7 and 8.

Table 6. Comparison of development phase accuracies with and without hashtag preprocessing

System	Train (%)	Validation (%)
Without	63.34	48.86
With	67.18	51.98

Table 7. Task 1 Statistics

Metric	System	F1	Precision	Recall
CR	BiLSTM	0.250	0.245	0.256
ES	BiLSTM	0.261	0.265	0.258
MX	BiLSTM	0.384	0.393	0.376
PE	BiLSTM	0.263	0.272	0.254
UY	BiLSTM	0.218	0.240	0.201

Table 8. Task 2 Statistics

Metric	System	F1	Precision	Recall
CR	BiLSTM	0.250	0.245	0.256
ES	BiLSTM	0.261	0.265	0.258
MX	BiLSTM	0.384	0.393	0.376
PE	BiLSTM	0.263	0.272	0.254
UY	BiLSTM	0.218	0.240	0.201

5 Conclusion

In this system report, we have presented a model which performs satisfactorily in the given tasks. The model is based on a simple architecture. There is scope for improvement by including more manually extracted features (like those removed in the preprocessing step) to increase the performance. Another fact is that the model is a constrained system, which may lead to poor results based on the modest size of the data. Related domain knowledge may be exploited to obtain better results. Use of regularizers led to proper generalization of model, henceforth increasing our task submission score.

References

1. et Al., V.R.: Tass - workshop on sentiment analysis at sepln (2013-03)
2. Cambria, E., Poria, S., Hazarika, D., Kwok, K.: Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In: AAAI (2018)
3. Díaz-Galiano, M.C., et al.: Overview of tass 2019. CEUR-WS, Bilbao, Spain (2019)
4. Garain, A., Basu, A.: The titans at SemEval-2019 task 5: Detection of hate speech against immigrants and women in twitter. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 494–497. Association for Computational Linguistics, Minneapolis, Minnesota, USA (Jun 2019), https://www.aclweb.org/anthology/S19-2088
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, http://dx.doi.org/10.1162/neco.1997.9.8.1735
6. Zhang, Z., Robinson, D., Tepper, J.: Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In: Lecture Notes in Computer Science. Springer Verlag (2018)