Detecting Aggressiveness in Mexican Spanish Tweets with LSTM + GRU and LSTM + CNN Architectures

Victor Peñaloza[a]

[a]RLICT: Research Laboratory in Information and Communication Technologies, Universidad Galileo, 7a. Avenida, calle Dr. Eduardo Suger Cofiño, Zona 10, Ciudad de Guatemala, Guatemala

Abstract

This paper presents a description of our participation in MEX-A3T 2020 aggressiveness detection on the Spanish Mexican tweets track. The goal of this task is to analyze a corpus comprised of Spanish Mexican tweets and identify its aggressiveness level (aggressive or not). For this task, we proposed two architectures; the first one is a BiLSTM + GRU based, and the second is a BiLSTM + CNN based architecture. After experimenting and evaluating, our BiLSTM + CNN model achieves 63.88% on aggressive class F1-Score, and our BiLSTM + CNN model achieves 63.87% on aggressive class F1-Score.

Keywords

Aggressiveness, Long Short Term Memory, Gated Recurrent Unit, Convolutional Neural Network, Twitter, Mexican Spanish text classification.

1. Introduction

The use of social communication tools on the Internet is being an essential side of daily human life. These social communications tools are generating a large amount of data that has sparked analysis interest among natural language and data science experts.

Although diverse models have been proposed to analyze social media data, there are still many challenges and ample space to improve research. One of these challenges is the multi-language content generated in these social networks. To push the improvement of research and to promote research in Mexican Spanish data, MEX-A3T 2020 proposed a track to identify aggressiveness on Mexican Spanish Tweets.

This study proposed two architectures that use LSTM, GRU, and Convolutional Networks as a block to be evaluated on the MEX-A3T 2020 aggressiveness detection track.

This paper is comprised of five sections: the first one presents an introduction to this task and study. The second section describes the corpus preprocessing phase. The third section describes the proposed architectures. The fourth section presents the results achieved in competition and the testing phase. The last section presents some conclusions and future work to continue the experiment with this task and architectures.

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: victorsergio@galileo.edu (V. Peñaloza)
ORCID: 0000-0001-7335-8255 (V. Peñaloza)

2. Data Preprocessing

Although supervised deep learning models can learn the main features from a dataset, the performance of such models depends on the quality of input data [1]. Previous sentiment analysis research on twitter-based corpus shows that various corpus-preprocessing techniques provide a significant improvement in model performance. Some techniques merely remove noise data, and others reduce terms and expressions to basic meaning [2].

2.1. Basic Data Preprocessing

For models described in this paper, the next steps were performed on the training data set [3]:

1. Lower case input text.
2. Remove URLs: URLs were encoded on the training data set as <URL>.
3. Remove accents, diaresis and tilde characters: Input text to NFKD to ASCII.
4. Remove numeric characters.
5. Remove single character and two-character elements.
6. Remove punctuation symbols.

2.2. Text Sequences Length

LSTM [4] and GRU [5] architectures are a proposal to learn long term dependencies. Despite the success of these architectures, there are concerns about the ability of these networks to manage such dependencies [6]. Considering those, we decided to limit the length of text sequences looking to get a sequence length that preserves the relevant information about the tweet and reduces the model training time. Trimming was done by shortening at the end of each text sequence.

2.3. Lemmatization

Lemmatization makes a morphological analysis of words and tries to remove inflectional endings, returning words to their dictionary word. In previous research, the use of lemmatization outperforms base algorithms on language modeling [7].The pipeline used was:

1. Tokenization.
2. Multiword tokens expansion.
3. POS labeling.
4. Lemmatization.

For the previous pipeline, we used AnCora treebank, Spanish models, from Python Stanford NLP package [8].

2.4. Stop Words

We remove stop words using the Spanish corpus from open-source Natural Language Toolkit (NLTK) [9].

2.5. Word Vectors

As a word-level representation, we used pre-trained embedding vectors with FastText [10] library. Embedding vectors used were pre-trained on external Mexican Spanish tweets. This pre-trained file contains 1,247.3M tokens with 100 dimensions each. These vectors were provided by the last MEX-A3T 2019 organizers [11].

2.6. Balance Dataset

On un-balanced data sets, different categories were represented unequally. So the output model is not biased to learn features of the majority class in classification task use of over-sampling techniques on minority class was proposed previously to get a better classifier performance. SMOTE is an oversampling method, in which the minority class is over-sampled creating "synthetic" samples rather than by over-sampling with replacement [12].

MEX-A3T 2020 training corpus was not balanced; we applied the SMOTE method to get a corpus with aggressive and not-aggressive equally represented classes.

3. Systems Description

Recurrent networks have proven to be useful in natural language processing tasks for their ability to carry information from the past [13]. On the other hand, convolutional neural networks have been used and showed promising results in diverse applications of natural language processing [14]. Additionally, the architecture used has proven to be effective on previous NLP classification tasks[15] and was altered to be adapted to this specific domain task.

This paper discussed two model's performance with slightly different approaches. The first model (Fig. 1) is comprised of an embedding input layer, followed by a spatial dropout that feeds a BiLSTM layer and a BiGRU layer respectively. Each of BiLSTM and BiGRU individual blocks feeds an independent global average polling layer and global max-pooling layer. The polling layers outputs are merged and followed by a dense layer with a ReLU activation function. Next batch normalization and dropout are applied. The last layer is dense with a SoftMax activation function.

The first model (BiLSTM + BiGRU) was trained using an Adam optimizer (learning rate = 3e-5, epsilon = 1e-8, norm clipping = 1.0), with sparse categorical loss entropy as a loss function, and was trained for 13 epochs.

The second model (Fig. 2) is a slightly different version of the first model, but the BiGRU layer was replaced for a 1D convolutional layer, and was trained for 15 epochs. Table 1 shows in detail the values of the parameters used for each model.

4. Results

The official competition metric was the F1 score on aggressive class. Table 2 shows our results on MEX-A3T 2020 on the test dataset and results on an own test data set used to experiment on the modeling phase. Own test data set was created, taking 20% of content provided official training set. Additionally, Table 2 shows two baselines used by organizers to compare with

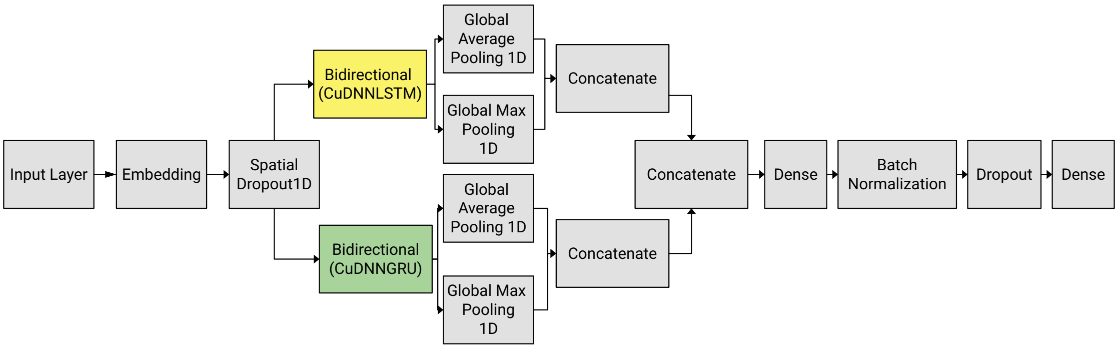

Figure 1: BiLSTM + BiGRU architecture.

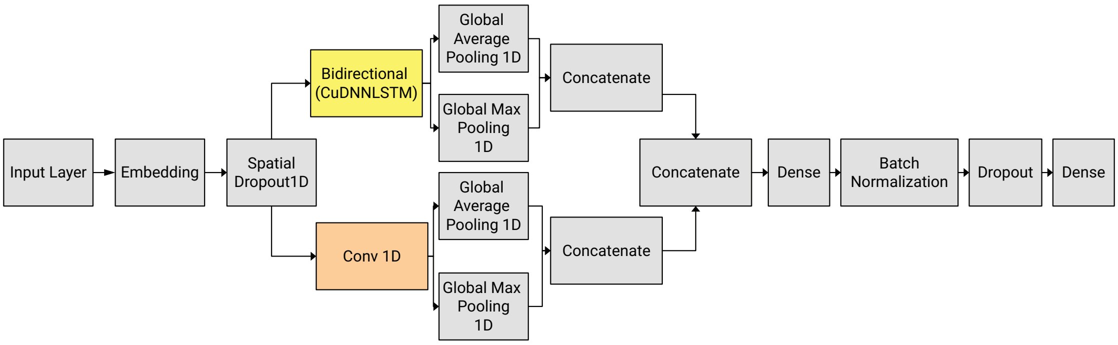

Figure 2: BiLSTM + CNN architecture

participating models, and some results from other participants ranked by a place on competition are shown too.

Based on the results, it should be noted that the two proposed architectures achieved similar performance. It can be observed that achieved results on the official test set not differ so much from results achieved on own test set. This indicates that chosen test data for the modeling phase represents well the proposed task dataset, and proposed models are not overfitting the training set.

We achieved 16th place with run 2 (BiLSTM + CNN). Although our results are lower than baselines models, this work shows a comparison between two proposed models on aggressiveness detection on Mexican Spanish tweets and leave possibilities open for architecture improvement with further research.

5. Conclusions and Future Work

In this work, we describe our participation in MEX-A3T@IberLEF2020, Aggressiveness Identification on Spanish Mexican Tweets Track [3].

We have shown two proposed architectures, first uses a BiLSTM + BiGRU combination as the base and second are BiLSTM + CNN combination based.

Table 1
Model architecture parameters. Parameters marked with * are parameters of the convolutional layer used only in the LSTM + CNN model.

Parameter	Value	Description
spatial dropout rate	0.2	Fraction of the input embedding layer output to drop.
biLSTM layer units	600	The dimensionality of bidirectional LSTM output space.
biGRU layer units	600	The dimensionality of bidirectional GRU output space.
filters*	332	The number of output filters in the convolution.
kernel size*	2	Length of the 1D convolution window.
activation function*	ReLU	Convolutional layer activation function.
dense layer units	144	The dimensionality of the intermediate dense layer output space.
dropout rate	0.2	The probability that each element of intermediate dense layer output is dropped.
last dense layer units	2	The dimensionality of the last dense layer output space. Binary classification with SoftMax activation function.

Table 2
Official results of aggressive detection on organizer test data and own evaluation results on own test data set.

Rank	Team Name	Official F1 aggressive	Own test F1 aggressive
1	CIMAT-1	0.7998	-
7	Baseline (Bi-GRU)	0.7124	-
12	Baseline (BoW-SVM)	0.6760	-
16	**UGalileo-2 (BiLSTM + CNN)**	**0.6388**	**0.6650**
17	**UGalileo-1 (BiLSTM + BiGRU)**	**0.6387**	**0.6333**
21	Intensos-2	0.2515	-

According to our experiment results, these two architectures show similar results on the aggressiveness detection task. Although proposed architectures achieved lower results compared to baseline models, it is possible to continue improving them, especially working on the corpus-preprocessing phase. We think that we have lost task-relevant information on tweets preprocessing phase that did not allow us to obtain better models performance.

Additionally, it would be worth to try other embedding vectors and dictionaries that represent better particular features of Mexican Spanish.

Acknowledgments

This work was supported by Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación (FISICC) and Research Laboratory in Information and Communication Technolo-

gies (RLICT), both part of Universidad Galileo from Guatemala.

References

[1] S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas, Data preprocessing for supervised leaning, World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering 1 (2007) 4104–4109.

[2] G. Angiani, L. Ferrari, T. Fontanini, P. Fornacciari, E. Iotti, F. Magliani, S. Manicardi, A comparison between preprocessing techniques for sentiment analysis in twitter, in: KDWeb, 2016.

[3] M. E. Aragón, H. Jarquín, M. Montes-y Gómez, H. J. Escalante, L. Villaseñor-Pineda, H. Gómez-Adorno, G. Bel-Enguix, J.-P. Posadas-Durán, Overview of mex-a3t at iberlef 2020: Fake news and aggressiveness analysis in mexican spanish, in: Notebook Papers of 2nd SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF), Malaga, Spain, September, 2020.

[4] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

[5] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734. URL: https://www.aclweb.org/anthology/D14-1179. doi:10.3115/v1/D14-1179.

[6] J. Zhao, F. Huang, J. Lv, Y. Duan, Z. Qin, G. Li, G. Tian, Do rnn and lstm have long memory?, 2020. arXiv:2006.03860.

[7] V. Balakrishnan, E. Lloyd-Yemoh, Stemming and lemmatization: A comparison of retrieval performances, in: Lecture Notes on Software Engineering, volume 2, 2014, pp. 262–267.

[8] P. Qi, T. Dozat, Y. Zhang, C. D. Manning, Universal dependency parsing from scratch, in: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 160–170. URL: https://nlp.stanford.edu/pubs/qi2018universal.pdf.

[9] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, 1st ed., O'Reilly Media, Inc., 2009.

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606 (2016).

[11] INGEOTEC, FastText Word Embeddings for Spanish Language Variations, 2019 (accessed June 10, 2020). URL: https://github.com/INGEOTEC/RegionalEmbeddings.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[13] T. Mikolov, M. Karafiát, L. Burget, Jan, H. . Černocký, S. Khudanpur, Recurrent neural network based language model., in: In INTERSPEECH 2010„ 2010, pp. 1045–1048.

[14] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),

Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751. URL: https://www.aclweb.org/anthology/D14-1181. doi:10.3115/v1/D14-1181.

[15] E. Garcia, Mercado libre data challenge, https://github.com/eduagarcia/meli-challenge-2019, 2019.