Applying Sentiment Analysis on Spanish Tweets Using BETO

Ariadna de Arriba, Marc Oriol and Xavier Franch

Universitat Politècnica de Catalunya, Barcelona, Spain
{ariadna.de.arriba, marc.oriol, xavier.franch}@upc.edu

Abstract. Emotion analysis of messages using machine learning techniques is a difficult and cumbersome task requiring a major effort to obtain reliable results. This challenge is even more pronounced when the target language is not English, but Spanish. To overcome this challenge, this paper describes how UPC Team applied sentiment analysis on social media messages (in particular, on Twitter) written in Spanish and, related to events that took place in April 2019 from different domains. To this aim, we present a machine learning model based on BERT and describe the results obtained to reach an accuracy of 65% approx. and the 12th position in the ranking, for this second edition of the contest for emotion detection of Spanish tweets EmoEva-lEs@IberLEF2021.

Keywords: Sentiment Analysis · Machine Learning · Social Media · Natural Language Processing · Twitter · Tweets · BERT

1 Introduction

Sentiment analysis in Spanish is a challenging task that has not been as much addressed as in the English context. Even though the Spanish language is spoken by more than 500 million speakers (being one of the most spoken languages in the world, just behind English, Chinese and Hindi), sentiment analysis in Spanish remains, comparatively, not sufficiently explored.

Sentiment analysis requires much time and effort to succeed in developing a good enough machine learning model. It embraces several critical steps, from data preprocessing (e.g., lemmatization, stemming, tokenization) to model customization (e.g., fine-tuning hyperparameters). But it is even more difficult when, instead of classifying the sentiment through a polarity spectrum (from positive to negative), aims at classifying the emotions of the messages (e.g., joy, sadness, fear, surprise).

To overcome this issue, the Iberian Languages Evaluation Forum (IberLEF) and the Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN) have organized since 2012 a series of competitions and workshops to attract the attention of re-

IberLEF 2021, September 2021, Málaga, Spain.

searchers to develop sentiment analysis tools and techniques for the Spanish language [1]. They have always organized sentiment analysis within the polarity spectrum analysis until past year when they introduced emotion detection. This year, they decided to repeat the competition with emotion analysis which they named EmoEvalEs [2].

In this paper, we describe the system we have developed to deal with emotion classification in Spanish tweets applying Natural Language Processing (NLP) techniques and developing a machine learning model based on BETO [3], a Spanish version of BERT [4].

In the following section, we explain the task presented and the dataset provided. Then, in section 3, we describe the system developed and the steps executed to overcome the task. In the results section, we expose the metrics obtained relative to other participants, including an analysis of our results. Finally, we close the paper with conclusions and references.

2 Task description

The task presented on this edition of IberLEF 2021 consists in classifying tweets into the emotion expressed in that text.

The dataset [5] was provided by the EmoEvalEs organizer and it is composed of tweets written in Spanish and based on several events that took place in April 2019 related to different domains: entertainment, catastrophe, political, global commemoration, and global strike. The corpus consists of 8223 tweets distributed in three subsets: dev (844 tweets), train (5723 tweets) and test (1656 tweets). Dev and train corpus are labelled with seven distinct emotions (see Table 1) and they have been used to develop and train the machine learning model, respectively. The test subset has been used to test the model, and its emotion distribution has been made public days after the competition ended.

Table 1. Emotion distribution

	# samples		
Emotion	(dev)	(train)	(test)
anger (also includes annoyance and rage)	85	589	168
disgust (also includes disinterest, dislike, and loathing)	16	111	33
fear (also includes apprehension, anxiety, concern, and terror)	9	65	21
joy (also includes serenity and ecstasy)	181	1227	354
sadness (also includes pensiveness and grief)	104	693	199
surprise (also includes distraction and amazement)	35	238	67

others: the emotion expressed in a tweet as 'neutral or no emotion'	414	2800	814

The metrics used to evaluate the performance of the task are its accuracy and the macro averages of its precision, recall and F_1 score.

3 System description

We propose a system for the classification of emotions of Tweets written in Spanish based on a BERT variant model named BETO.

To build the machine learning model, we applied the following process pipeline (see Fig.1):

- **Pre-processing tweets:** In this stage we applied NLP techniques to clean and normalize the messages in the tweets.
- **Training machine learning model:** Using the pre-processed tweets, we trained a model based on BETO.
- **Fine-tuning model:** To improve the accuracy of our machine learning model, we iteratively fine-tuned several hyperparameters of the BETO model and retrained the model to obtain better results.

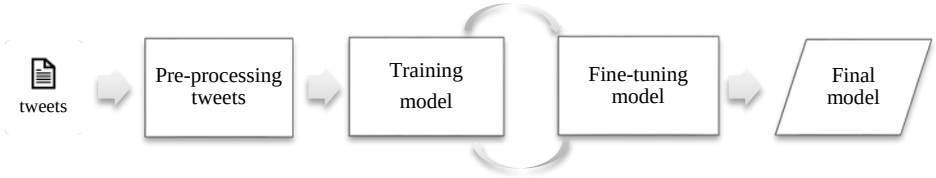

Fig. 1. Process pipeline

3.1 Preprocessing

Pre-processing is a critical step of all NLP systems [6]. We applied some general NLP pre-processing techniques and specific methods for emotion processing to obtain a "cleaner" text. Some examples of tweets with their preprocessing results are shown in Table 2. The methods and techniques applied are:

- **Remove URLs:** Many tweets contain URLs to websites that provide more information but that are not relevant to the sentiment analysis process. We removed these URLs as they only make the text noisier.
- **Remove hashtags:** We removed all hashtags in the text.

- **Remove numbers:** Numbers are not providing useful information for emotion analysis.
- **Replace emojis and emoticons:** Emojis and emoticons are remarkably significant in sentiment analysis. A text can be apparently neutral but can express any emotion by adding emojis. To facilitate the machine learning process, we replaced the emojis and emoticons with a text that represents the emotion they evoked.
- **Replace abbreviations:** Social media users express themselves in a colloquial mode. These users tend to use abbreviated words and expressions that are not usually present in dictionaries for NLP but are known to everybody. To this aim, we built a list of fifty common abbreviations and we replaced them with their correct form. Some of them are: *q (que), bn (bien) o mñn (mañana), tqm (te quiero mucho), pti (para tu información).*
- **Replace laughs:** Laughs may be ambiguous but they are mostly used to express the 'happy' emotion. Nevertheless, users tend to write laughs in multiple forms. So we replaced all words that start with 'ja', 'je', 'ji', 'jo' with 'jajaja' (after checking that the term does not exist in the dictionary to avoid replacing existing words), to obtain a general form for laughing.
- **Remove punctuation marks:** We removed punctuation marks such as commas, question marks, or quotation marks because they do not provide any emotion in the text by themselves. In several cases, some punctuation marks (e.g., exclamation marks) can emphasize the emotion that the text expresses. As there is no generalized rule for making the machine detect these cases in a clear way, we decided to remove them prior to producing more confusion to the machine.
- **Remove repeated characters:** In social media, most people are not following spelling rules and it is common to repeat some characters especially at the end of a word (e.g. *holaa* or *graciaas!*). We removed these additional characters.
- **Lemmatization:** Common technique in NLP. It consists in transforming all verb conjugations into its infinitive form. For example, we replaced *vivían* by *vivir*.
- **Remove stopwords:** Stopwords are words that are not providing any useful information (e.g., *y, aunque, con*). Removing them is very common in NLP. In Python there are many libraries to obtain a list of stopwords in different languages. In this case, we used the nltk library [7].
- **Remove blank spaces:** The last stage is removing extra white spaces that tweets could have or the ones created when we replaced or deleted words in previous steps.

Some of these rules have not been tested in this dataset (e.g., repeated characters removal or laughs removal) due to the messages are not containing any of them. Even so, we have included it as it is very common in social media users.

Table 2. Examples of tweets preprocessing

Tweet	Preprocessed tweet
Que devastador lo de #NotreDame 😢 no parece que va a quedar mucho...	devastador tristeza no parecer ir quedar mucho
La mejor manera de entender mi idioma #DiaDelLibro https://t.co/W9wwoyW8qk	mejor manera entender idioma
Espero que todos ya estén listos para gritar los goles del Barça!! porque hay que creer y confiar que mañana ellos van a ganar😁😁 ♥💪⚽⚽#ChampionsLeague #Barça	esperar listo gritar gol haber creer confiar mañana ir ganar felicidad felicidad

3.2 Training

We trained the machine learning model using a Spanish variant of the BERT model named BETO. BERT is a machine learning technique pre-trained with the Toronto Book Corpus and Wikipedia [4] and developed by Google [8]. BERT is a transformer-based deep learning model able to deal with multiple NLP problems. It uses attention masks to encode each word during the training stage and predict up to 15% of masked words using NSP (Next Sentence Prediction) and to understand the context of a sentence [9].

BETO is a variant of BERT, which has been pre-trained exclusively on Spanish data, with a dataset of similar size as BERT. We have chosen BETO over BERT, as BETO has been able to outperform other BERT-based models in several NLP-based activities using Spanish as language [10].

During the training stage, some configuration parameters are fixed meanwhile others are adjusted as detailed in the next section. The parameters set before starting training the model are the batch size and the maximum sequence length. The batch size is the number of samples used in one iteration in the training stage and was set to 64 . The maximum sequence length was set to 256 as tweets are short texts up to 240 characters maximum.

3.3 Fine-tuning

In the fine-tuning stage, we adjusted several hyperparameters (see Table 3) to improve our trained model and obtain better results.

The hyperparameters we adjusted are:

- **Learning rate**: Hyperparameter that controls how much the model changes in response to the estimated error each time the model weights are updated. Choosing the optimal learning rate is a difficult task, as a small learning rate may result in a slow training process and a value that is too large can cause the model to diverge instead of to converge to the solution (see Fig. 2) [11].

Fig. 2. Adjusting learning rate hyperparameter [12]

- **Epsilon:** This hyperparameter is a very small number to prevent any division by zero in the implementation [13].
- **Number of epochs:** Hyperparameter that indicates the number of times that the model visits the entire training dataset. We adjust it to control the weight decay as it uses the following formula:

$$weight\ decay = learning\ rate\ /\ number\ of\ epochs \quad (1)$$

In this regard, we should be careful choosing the value for epochs as weight decay is used to prevent overfitting and to keep a weight small to avoid exploding gradients.

Table 3. Hyperparameters fine-tuning for each submission

#	Learning rate	Epsilon	Number of epochs
1	$1x10^{-5}$	$1x10^{-4}$	15
2	$1x10^{-5}$	$1x10^{-5}$	15
3	$1x10^{-4}$	$1x10^{-4}$	10

4 Results

The results extracted from the evaluation phase are shown in Table 4. We submitted three different versions trained with different hyperparameters tuned. The first submission was trained with 15 epochs, a learning rate of 10^{-5}, and epsilon of 10^{-4}. In the second one, we decided to reduce the epsilon to 10^{-5} but keep the learning rate and the number of epochs to check if model performance improves changing only the epsilon value. Finally, in the third submission, epsilon and learning rate were set to 10^{-4} meanwhile the number of epochs was reduced to 10.

Table 4. Results from evaluation phase

#	Accuracy	Macro avg F1 Score	Macro avg precision	Macro avg recall
1	0.644324	0.612514	0.594360	0.644324
2	0.652778	0.622223	0.600479	0.652778
3	0.641908	0.626723	0.625913	0.641908

As the best performance results correspond to the second submission, that is the one we submitted to the official leaderboard. In relation with other participants, we obtained an accuracy of 65% approximately which placed us in the 12th position in the ranking. Despite this position, the accuracy obtained is not so far from the winner which has got around a 73% of this metric.

As we can observe, the F1 score is very close to accuracy, which indicates that the dataset is sufficiently balanced. Metrics results are very similar in all submissions which may indicate that model cannot be improved so much only by fine-tuning the hyperparameters chosen with the data provided.

The main errors we have detected in tweets classification in the validation stage are mainly due to pre-processing issues. For instance, we identified that tweets that may be predicted as 'others' are always classified as 'sadness' or 'joy', probably by the influence of replacing emojis in the pre-processing phase. Another issue is that the system is less accurate for detecting emotions that have a small number of samples, as it is the case for 'fear', 'disgust' and 'surprise'.

For future work, we could obtain more data from Twitter to increase the performance of the model by training it with a bigger size corpus and a little bit more balanced. In this case, we only trained the model with the train dataset but we could have trained it with train and dev dataset to have more data for the machine. Another improvement for the future could be trying other values and combinations for the hyperparameters that cannot be tested for this competition or adjusting other parameters as the batch size.

5 Conclusions

Developing a machine learning model to classify a text into emotions is a challenging task. The way human beings express themselves is very ambiguous and detecting which emotions they want to express can be extremely difficult even for them, because many times they do not even know how they feel. For this reason, it is important to have good quality data with labelled emotions well remarked and above all be patient as finding an optimal model is a slow and hard task.

In this paper, we have presented a BETO-based machine learning model to classify into emotions Spanish tweets. The results of the evaluation show that the model is able to identify the correct emotion on approximately 2 out of 3 occasions (accuracy=0.65). As future work, we plan to improve the overall system by enhancing the preprocessing phase (e.g., taking into account capital letters that could emphasize the text emotion or

keeping some hashtags that could be valuable for our aim), fine-tuning further the hyperparameters of the machine learning model and make an in-depth quantitative error analysis to improve our results in emotion classification.

References

1. Montes, M., Rosso, P., Gonzalo, J., Aragón, E., Agerri, R., Álvarez-Carmona, M. Á., … Taulé, M. (Eds.). (2021). *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021)*. *CEUR Workshop Proceedings*.
2. Plaza-del-Arco, F. M., Jiménez-Zafra, S. M., Montejo-Ráez, A., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2021). Overview of the EmoEvalEs task on emotion detection for Spanish at IberLEF 2021. *Procesamiento del Lenguaje Natural*, *67*(0).
3. Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Pérez, J. (2020). Spanish Pre-Trained BERT Model and Evaluation Data. In *PML4DC at ICLR 2020*.
4. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. Retrieved from http://arxiv.org/abs/1810.04805
5. Plaza-del-Arco, F. M., Strapparava, C., Ureña-López, L. A., & Martín-Valdivia, M. T. (2020). EmoEvent: A Multilingual Emotion Corpus based on different Events. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 1492–1498). Marseille, France: European Language Resources Association. Retrieved from https://www.aclweb.org/anthology/2020.lrec-1.186
6. Uysal, A., & Günal, S. (2014). The impact of preprocessing on text classification. *Inf. Process. Manag.*, *50*, 104–112.
7. Natural Language Toolkit — NLTK 3.6.2 documentation. (n.d.). Retrieved May 28, 2021, from https://www.nltk.org/
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [cs]*. Retrieved from http://arxiv.org/abs/1706.03762
9. Horev, R. (2018, November 17). BERT Explained: State of the art language model for NLP. *Medium*. Retrieved February 12, 2021, from https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
10. Plaza-del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2021). Comparing pre-trained language models for Spanish hate speech detection. *Expert Systems with Applications*, *166*, 114120. https://doi.org/10.1016/j.eswa.2020.114120
11. Brownlee, J. (2019, January 22). How to Configure the Learning Rate When Training Deep Learning Neural Networks. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/
12. Setting the learning rate of your neural network. (2018, March 2). *Jeremy Jordan*. Retrieved May 31, 2021, from https://www.jeremyjordan.me/nn-learning-rate/
13. Brownlee, J. (2017, December 19). A Gentle Introduction to Transfer Learning for Deep Learning. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/transfer-learning-for-deep-learning/