Sexism identification in Social Networks EXIST Task proposal [1]

Álvaro Faubel Sanchis[2] and Clara Martí Torregrosa[2]

[2] Polytechnic University of Valencia, Valencia 46022, Spain
{alfausa1, clamarto}@inf.upv.es

Abstract. Twitter is a well-known microblogging social site where users express their views and opinions. Consequently, tweets tend to be a clear view of society. With the use of different techniques like pretrained models, this paper describes the process to try some approaches such as from feature extraction to deep learning models (BERT), in order to achieve a system that allows classifying tweets as sexist or non-sexist and disjoin them into levels. Finally, we will conclude with the best system, which is BERT pre-trained model.

Keywords: Sexism, BERT, Ensemble Methods, IDF and Word Embeddings.

1 Introduction

Nowadays, the society is still sexist, and it can be seen on social media, that is a clear reflection of opinions, knowledge, and culture population. Among them, Twitter stands out as social platform is a bidirectional communication service, which is perfectly structured to share from personal experiences to opinions on current news as fast as possible.

Nevertheless, these opinions could perturb other users, for example the case of sexist, aggressive or toxic tweets. These comments could have serious consequences in real life, and legal issues towards social platforms. As a result, a need for language models specific to social media domain arises.

To address these types of problems, it has been created a bunch of international competitions. In this case, we participated in EXISTS 2021 [1], the first shared task on Sexism Identification in Social networks at IberLEF 2021 [2].

In this work, based on the main idea of previous studies, we understand that the best techniques for this type of problem use: deep learning models [3], feature vectors and word embeddings [4] [5]. We address three techniques, selecting and sending those that achieve the best results.

[1] *IberLEF 2021, September 2021, Málaga, Spain*.

2 Experimental Setting

2.1 Data Collection

Once we enrolled in the task, we received two sets of tweets and gabs messages, both with text in Spanish and English. The first one refers to the train set with 6977 tweets and others 3386 for testing in the second set.

There are two tasks, the first one on sexism identification in a binary classification (sexist or non-sexists). The following task aims to categorize the sexism into five different types: ideological and inequality, stereotyping and dominance, objectification, sexual violence, misogyny, and non-sexual violence.

2.2 Preprocessing

The first step for any text analysis is the preprocessing of the text. Text in tweets is characterized by its informal or colloquial language. In order to address it, initially we started by the Python package tweet-preprocessor.

Also, it was necessary to make an exhaustive cleaning, which it goes from deleting all non-alphanumeric characters, convert all text to lowercase to delete stop-words, with the help of the Regex library and Nltk package.

Following, we performed a tokenization process and additionally for Spanish a text stemming and for English tweets a lemmatization.

This process was done for our approaches based on Words Embeddings and IDF Matrix.

In case of pre-trained BERT model, we only replace URLs, Mentions, Reserve words, Emojis and Smileys with special tokens.

2.3 Text Representation & Models

We proposed different strategies in order to represent the tweets and then we used different models in each representation.

The first one uses Word Embeddings to represent text, with the use of the pre-trained model GloVe Twitter 200 from the Python library *Gensim*. GloVe is an unsupervised learning algorithm for obtaining weight vector for words. In this case, we used a weight vector of size 200 for each term of our tweets. For each tweet we add the vectors of their terms, dividing the sum by the number of words, and finally obtaining an average weight vector. Here there was the possibility that terms in our tweets do not have their weight vector in the pre-trained model, so first we fitted another model only with our data. It is obvious that weights of this new model will not be as robust as the previous ones. However, it is a better way than filling a zero vector.

For the second strategy instead of using Word Embedding, we focus on IDF matrix using the Python sklearn package for tweet representation. As a rule, there are two metrics that evaluate the term's weight based on its occurrences, these are Term Frequency (TF) and Inverse Document Frequency (IDF). While TF considers all term equally important, IDF reduces weight based on the number of occurrences.

$$IDF(t) = ln\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}\right) \tag{1}$$

Deep neural network models have a significant value in this context, one example of these models is BERT, which we use in our last approach.

Bidirectional Encoder Representations from Transformers (BERT) is a technique created in 2018 for researchers at Google. Since the release of this model, many users have been using this type of neural networks in several tasks. The pre-trained models are useful to solve one biggest challenge in NLP, that is the lack of enough training data. Because starting with one of these pre-trained models, with a little fine-tuning, it could help us in our task, we decided to use it.

We choose BETO for Spanish tweets which is a BERT model trained on a big Spanish corpus [6] and Twitter-roBERTa-base for English text, which was trained on more than 58M tweets. It is necessary to mention that, based on the results presented in the *Tweet-Eval benchmark* paper [7], other models for English (i.e. BERTweet [8]).

To implement BERT in this project, it was necessary to use Pytorch and HuggingFace Tranformers.

2.4 Results

Task 1 has been organized with a typical binary classification, while task 2 is a multi-label problem. In order to get better results in our second task we decided to use only the tweets classified as sexist, leaving aside those not. It was done because non-sexist tweets are irrelevant to determine the level of sexism.

In both tasks, we have tried several models from sklearn and have validated them with the test set by Cross Validation 5 K-fold. Initially we started with baseline models, then with grid parameters and finally we combined these models with hybrid approximations and ensemble methods, the obtained results shown in Table 1 and Table 2:

Table 1. Word Embeddings results.

Model (Accuracy Metric)	Task 1	Task 2
SVM	0.716	0.558
Random Forest	0.702	0.508

Logistic Regression	0.684	0.547
Decision Tree	0.587	0.355
SVM {‘C’: 1, ‘gamma’: 0,1, ‘kernel’: ‘rbf’} Task 1 {‘C’: 10, ‘gamma’: 0,01, ‘kernel’: ‘rbf’} Task 2	0.716	0.572
Random Forest {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 100} task 1 {'bootstrap': False, 'criterion': 'gini', 'max_depth': None} task 2	0.696	0.534
Logistic Regression {'C': 0.1, 'penalty': 'l2'} task 1 & task 2	0.686	0.558
Bagging SVM	0.715	0.566
Bagging LR	0.690	0.561
AdaBoost LR	0.651	0.424
Stack LR & SVM	0.718	0.574

Table 2. IDF results.

Model (Accuracy Metric)	Task 1	Task 2
SVM	0.725	0.581
Random Forest	0.713	0.591
Logistic Regression	0.719	0.602
Decision Tree	0.667	0.534
SVM {‘C’: 1, ‘gamma’: 1, ‘kernel’: ‘sigmoid’} Task 1 {‘C’: 10, ‘gamma’: 0.1, ‘kernel’: ‘rbf’} Task 2	0.715	0.602
Random Forest {'bootstrap': True, 'criterion': 'gini', 'max_depth': 100} task 1 {'bootstrap': True, 'criterion': 'entropy', 'max_depth': 100} task 2	0.708	0.590
Logistic Regression {'C': 0.1, 'penalty': 'l2'} task 1 & task 2	0.714	0.601
Bagging SVM	0.715	0.594
Bagging LR	0.722	0.592
AdaBoost LR	0.565	0.240
Stack LR & SVM	0.717	0.602

To sum up, for the feature extraction proposals, the best model when Word Embedding is used Stacking Classifier with SVM and Logistic Regression, and with IDF matrix, Bagging with Logistic Regression.

In our last proposal, we used BERT’s pretrained tokenizers for each language. These tokenizers basically add special tokens to the text and transform it into numerical vectors that serve as input to the neural network (see Fig. 1).

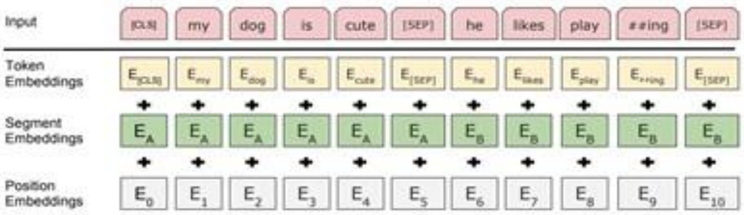
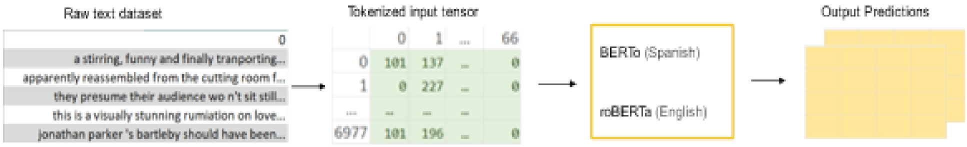

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
E[CLS] Emy Edog Eis Ecute E[SEP] Ehe Elikes Eplay E##ing E[SEP]
Segment Embeddings
EA EA EA EA EA EA EB EB EB EB EB
Position Embeddings
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10

Fig. 1. BERT Tokenizer.

It is relevant to say that the max length selected for the Spanish language was 155 for task 1 and 120 for task 2, and for English texts, 170 in both cases. Thus, the columns of those tweets that are less than the maximum length will be filled to 0 in the creation of the tokenized matrix. In addition, those texts of greater length will be truncated to that size. This is called padding and truncation.

After having all the tweets tokenized, we pass them to the pretrained neural network, that will give us outputs on which we can extract the final predictions. Fig. 2 shows how it works:



0
a stirring, funny and finally tranporting...
apparently reassembled from the cutting room f...
they presume their audience wo n't sit still...
this is a visually stunning rumiation on love...
jonathan parker 's bartleby should have been...

	0	1	...	66
0	101	137	...	0
1	0	227	...	0
...	...	...	...	
6977	101	196	...	0

Fig. 2. Pipeline of the BERT model.

In this approach we tried two classifiers. The first only considers the first vector for each layer which is called CLS Vector (see Fig. 3).

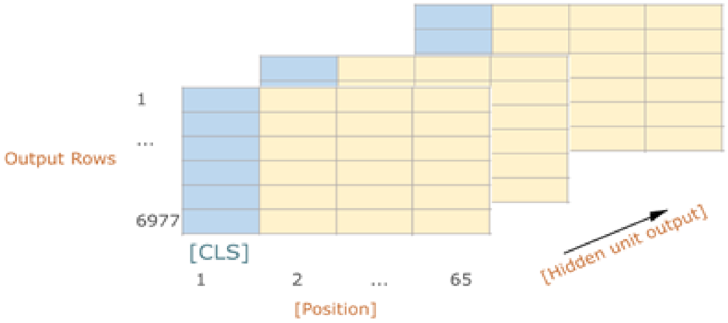


Fig. 3. Layers output from BERT model.

At first, we fit a linear layer with only one neuron per class with the first output vector CLS, on which is applied a *Softmax* activation function.

Nevertheless, another classifier returns us better results. In lieu of only selecting the first array, this methodology considers all the outputs, creating an interconnected dense layer to which the *Relu activation function* is applied. Then, another dense layer is created with less input vectors and with as many neurons as classes, in which the activation function applied is the *LogSoftmax*.

Focusing on technical details, we have used the Cross Entropy as loss function, a learning rate of $1e{-}5$ and an epsilon of $1e{-}8$. Finally, the dropout to avoid the overfitting over forward step was 0.1.

In addition, we choose 10 epochs and a batch size of 16 for the training loop. So, in order to train and validate our model, we split the data in three sets, 90% for training, 5% for validation over the epoch's iterations and 5% for testing the performance once finished the training.

Next tables show the BERT's results in each task and language over the test subset.

Table 3. BERT results over training data.

	Task 1 Accuracy	Task 2 F1-Score
BERT Spanish	0.84	0.73
BERT English	0.83	0.67

2.5 Competition Results

Since with BERT's modelling we obtained better results, this was our main proposal in the competition. However, we also sent the other two.

Table 4. Final results for all runs.

	Task 1 Accuracy	Task 2 F1-Score
BERT	0.7637	0.5578
IDF (Bagging LR)	0.6944	0.4673
Word Embeddings (Stack LR & SVM)	0.6962	0.1585

Finally, we got 13th position in task 1 and 7th in task 2.

3 Conclusions and Future work

We proposed three different approaches for the Exits task on sexism detection in Twitter. The first two models use feature extraction techniques: IDF and pre-trained word embeddings. And the last based on BERT pre-trained model.

The best results were obtained with the last model, since such a model has been trained with a corpus large enough to handle one of the biggest issues in this field, which is the lack of input data.

However, the amount of data is never enough, so an improvement would be, to train with more data and to improve model parameters. Furthermore, we could test the combination of neural network models, as well as LSTM techniques.

IberLEF proceedings [9].

References

1. F. Rodriguez Sanchez, J. Carillo de Albornoz, L. Plaza, J. Gonzalo, P. Rosso, M. Comet and T. Donoso, "Overview of EXIST 2021: sEXism Identification in Social neTworks," *Procesamiento del Lenguaje Natural*, vol. 67, (2021).
2. "IberLEF 2021: Iberian Languages Evaluation Forum.", https://sites.google.com/view/iberlef2021/, (2021).
3. J. Carrillo de Albornoz, L. Plaza and F.Rodrígez Sanchéz, "Automatic Classification of Sexism in Social," NLP & IR Group, UNED, December 17, (2020).
4. S. Frenda, B. Ghanem, M. Montes y Gómez and P. Rosso, "Special Section: Intelligent and Fuzzy Systems applied to Language & Knowledge Engineering.", vol. 34, no. 5, pp. 2959-2969, (2018).
5. P. Chiril, V. Moriceau, F. Benamara, A. Mari, G. Origgi and M. Coulomb-Gully, "He said "who's gonna take care of your children when you are at ACL?", Reported Sexist Acts are Not Sexist. Association for Computational Linguistic, pp. 4055-4066 (2020).
6. D. UChile, "BETO: Spanish BERT.", https://github.com/dccuchile/beto.
7. J.Camacho Collados, L. Neves, L.Espinosa Anke and F. Barbereri, "TweetEval: Unified Benchmarck and Comparative Evaluation for Tweet Classification.", School of Computer Science and Informatics, Cardiff University, United Kingdom (2020).
8. V. Research, "BERTweet: A pre-trained language model for English Tweets (EMNLP-2020).", https://github.com/VinAIResearch/BERTweet, last accessed 04/19/2020.
9. IberLEF 2021 proceedings: Manuel Montes, Paolo Rosso, Julio Gonzalo, Ezra Aragón, Rodrigo Agerri, Miguel Ángel Álvarez-Carmona, Elena Álvarez Mellado, Jorge Carrillo-de-Albornoz, Luis Chiruzzo, Larissa Freitas, Helena Gómez Adorno, Yoan Gutiérrez, Salud María Jiménez Zafra, Salvador Lima, Flor Miriam Plaza-de-Arco and Mariona Taulé (eds.): *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021), CEUR Workshop Proceedings*, 2021