Toxicity Analysis in Twitter using some Machine Learning models

Álvaro Mazcuñán Herreros and Miquel Marín Colomé
Escuela Técnica Superior de Ingeniería Informática, ETSINF
Universitat Politècnica de València, Spain
almazher@inf.upv.es
mimaco1@inf.upv.es

Abstract: *This paper describes one of the key evaluation points in the Natural Language and Information Retrieval subject that was the participation in a natural language analysis competition and its corresponding task which corresponds to the DETOXIS IberLEF 2021 concourse. In this report we will explain briefly the procedure of how this team approached this project. Two models for the two sub-tasks have been submitted to the competition. The first one is the hybrid Stacking model in which SVM, Decision Tree, Random Forest and MLP models have been used with a logistic regression with the function of being metalearner. The second is the BETO model, which is a variant of the BERT model. All this is summarised in a table of results.*

1. Introduction

In our case, it was decided to choose the **DETOXIS** [1] contest in order to train Machine Learning models learnt during these courses and, in this way, to be able to analyse the different toxic tweets found in this database. In addition, it should be said that in the different models made, except for the BERT model, the sklearn [2] library has been used.

Two datasets were available for this task. The first was the train dataset, with a total of 3958 tweets. On the other hand, in order to validate the quality of the model, we had the test dataset, which consisted of 891 tweets. Overall, this dataset had a wide variety of variables such as: constructiveness, positive / negative stance, stereotype, sarcasm, aggressiveness among others.

However, for this competition, only the comment variable was used, i.e., the variable containing the different tweets from various Spanish newspapers such as ABC, elDiario.es, El Mundo, etc.

The aim of this competition was twofold. On the one hand, it was required to label the tweets in the test set with 0 or 1, i.e., whether these tweets were non-toxic or toxic, respectively. For this purpose, the variable toxicity was available in the training set to be able to evaluate the corresponding Machine Learning models. In addition, in the second subtask, the objective was a little more complicated, because in this case we were asked to label the same set of tests but, in this case, adding different levels of toxicity:

- 0 → Not toxic
- 1 → Mildly toxic
- 2 → Toxic
- 3 → Very toxic

As discussed above for the first subtask, in order to evaluate our models, the toxicity_level variable was available in the set of 3958 tweets.

2. Preprocessing

Before performing any separation of the data in order to train the corresponding models, it was decided to carry out a small amount of data cleaning/preprocessing.

To do this, the first thing that was done was to remove from the messages those characters that were emojis, hashtags (#), URLs and other special characters. Once these parts of the tweets had been eliminated, we began **tokenizing** the text, i.e. separating the tweet into words and, with this, obtaining a list of these words.

With this list of words, the next step was to eliminate **stopwords**, i.e. words that have no meaning in themselves. This group of words usually consists of articles, pronouns, prepositions, adverbs and some verbs in particular.
The next step is to apply the **stemming** and **lemmatization** algorithms. The former works by stemming the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. It has to be said that this approach can be successful on some occasions but not always.

Below is an example of how the stemming algorithm works:

studies → -es → studi
studying → -ing → study

On the other hand, the lemmatization algorithm takes into account the morphological analysis of words. To do this, it is necessary to have detailed dictionaries in which the algorithm can consult in order to link the form with its lemma. An example is shown as in the previous case:

studies → third person, present tense of the verb study → study
studying → gerund of the verb study → study

Once all the pre-processing part of the tweets has been done, we can move on to the text representation part that has been used for this contest. However, before going into this, a **training and test partition** has to be carried out. Specifically, in the training dataset (the one containing 3958 messages) a training and validation partition will be made (around 10-20% depending on the algorithm used).

Once this is done, the remaining 891 tweets from the other set can be used to tag the desired messages. In addition to performing the corresponding partitions, it has to be observed whether the classes are balanced or not, as this can lead to problems when evaluating the subsequent models. In the variable containing the binary classification, **toxicity**, 2316 tweets can be observed with class 0, i.e. non-toxic, and the rest (1147) with class 1, meaning that they are toxic. According to the criteria considered by the team, it was decided not to carry out any class balancing task. However, the situation varies in the **toxicity_level** variable. This variable contains 2317 tweets with class 0, 808 with class 1, 269 with class 2 and, finally, 69 with the most toxic class. In this case, it was decided to carry out a balancing task (the solution adopted will be briefly explained in the part on model evaluation).

3. Text representation

Having considered the issue of class balancing, we now move on to explain the text **representation techniques** used. The first of these is the **bag of words** [3]. It is a way of representing the vocabulary that we will use in our models and consists of creating a matrix in which each column is a token and the number of times that token appears in each sentence is counted.

Below is a small example of how this technique works:

We have 3 sentences which are as follows:

This movie is very scary and long
This movie is not scary and is slow
This movie is spooky and good

With these 3 sentences we get a vocabulary with all unique words as follows: "This", "movie", "is", "very", "scary", "and", "long", "not", "slow", "spooky", "good".

Applying bag-of-words we would be left with the following matrix:

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Table 1 - Matrix example to represent BOW

The problem with this technique is that it only considers unigrams. For this reason, the **n-grams** technique was also used, as word order could be considered in this way. It was decided to use between 2 and 3-grams (bigrams-trigrams). The procedure would be the same as before but taking into account the latter approach.

The third and last text representation technique used was **Term-Frequency - Inverse Document Frequency** (TF-IDF) [4]. This technique consists of measuring how important a word is within a text, i.e. each word will have an associated weight, depending on its importance.

4. Methods

Having explained the techniques used to represent tweets, we now move on to mention the different Machine Learning **models** that were used:

- SVC
- Decision Tree
- Logistic Regression
- MLP (Multilayer-Perceptron)
- Random Forest
- Stacking
- BETO

Since the Stacking and BETO models have been submitted as runs in the competition, the strategy used in both models will be briefly explained.

It must be said that the **Stacking** model [5] is a combination of some of the previous models, specifically we used SVC, Decision Tree, Random Forest and MLP as base models and, as a meta learner model, logistic regression. In addition, in some of the previous methods, in order to obtain the ideal parameters for each of them, fine tuning was used, specifically, the **Grid search** technique [6].

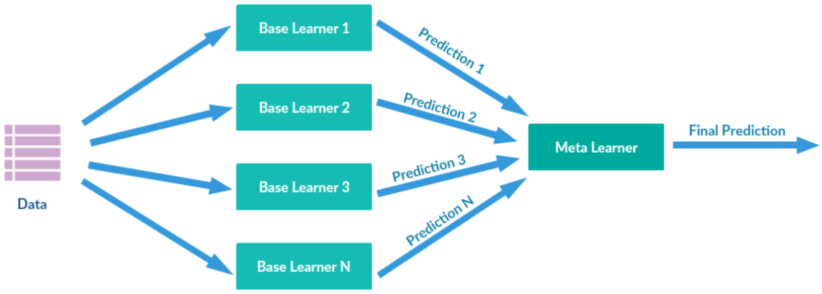

Fig. 1. General structure of the Stacking system1

As previously mentioned, the issue of imbalance is a problem to be taken into account when obtaining the labels for the different types of toxicity (toxicity_level). Therefore, for all the previous models, except for the BETO model, an extra parameter called *stratify* was used at the time of training these models, which allowed the classes to be balanced while performing the training task.

Finally, the **BERT** [7] technique was also used, specifically the dccuchile/bert-base-spanish-wwm-uncased model of Hugging Face for tweets in Spanish. [8] BERT uses Transformers, an attention mechanism that learns the contextual relationships between words in a text. In its basic form, Transformer includes two separate mechanisms: an encoder that reads the text input and a decoder that produces a task prediction. Since the goal of BERT is to generate a language model, only the encoder mechanism is needed. Because this technique was very new to all of us, it required more time to investigate how it worked and somehow implement it in Python with the help of experts who had already dealt with the subject before.

In this model, a maximum tweet length of 200 characters and a batch size of 16 were used for training.

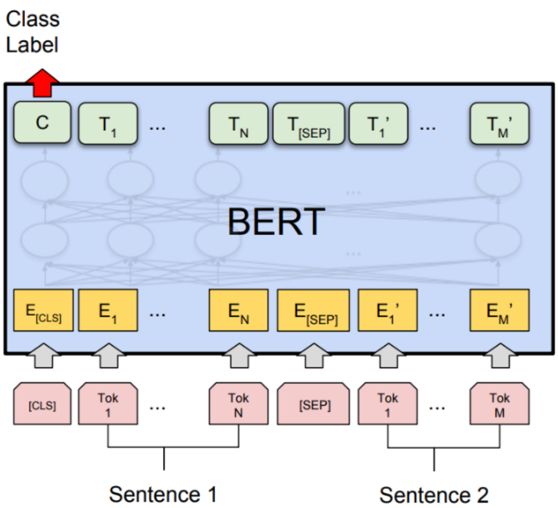

Fig. 2. BERT Model

5. Results

Once a brief description of the models used by this team has been given, we move on to the results. However, it should first be noted that in order to evaluate the quality of these models, two evaluation measures have been used: F1-Score and Accuracy

It should be noted that the contest organizers did not use accuracy as an evaluation metric in the DETOXIS contest. For the DETOXIS competition, due to the fact that only a maximum of 5 runs could be sent, we decided to send the Stacking and BETO models (although in this case the results were not very good, but we decided to send it anyway, as it was a new technique that we had never learnt and we wanted to try it).
The results for the BETO model were as follows:

System	Toxicity	Levels
Accuracy	0.7839	0.265
Loss	0.9899	1.406

Table 2. BETO performance on the testing set

On the other hand, the results for the Stacking hybrid model were as follows:

System	Toxicity	Levels
Accuracy	0.7716	0.6949
F1-Score	0.7568	0.6347

Table 3. Stacking performance on the testing set

6. Conclusion and Future Work

In the DETOXIS **ranking** [9], we have finished in the following ranks:
- Predicting ‘toxicity’: **8th** with an accuracy of 0.6936
- Predicting ‘toxicity_level’: **11th** with an accuracy of 0.4632

Throughout this project, different approaches could be adopted in order to obtain the best possible results in labelling tweets with their corresponding toxicity values. However, due to lack of time, some of the improvements we had in mind could not be implemented. Therefore, knowing this, the legacy/possible improvements of this project are as follows:

1. Because in the case of BETO the accuracy leaves much to be desired, what could be done is the following: because that class is very unbalanced, around 2000 samples and, in this case, you want to predict the level of toxicity of a tweet according to whether the model has previously predicted it as whether that tweet is toxic or not. Therefore, you could first make a prediction on the toxicity variable and, once you have obtained the predictions, you would obtain a new *new_predictions* column and with this you will only work with those tweets that the model has predicted as toxic.

2. Perform balancing tasks before training the models.

3. Use more variables such as sarcasm, aggressiveness, etc. and not just the variables of the tweet comment itself.

References

[1] «DETOXIS-IberLEF,» 2021. [En línea]. Available: https://detoxisiberlef.wixsite.com/website .

[2] «scikit-learn Machine Learning in Python,» [En línea]. Available: https://scikit-learn.org/stable/ .

[3] R. J. Z.-H. Z. Yin Zhang, Understanding Bag-of-Words Model: A Statistical Framework.

[4] J. G. Sang-Woon Kim, Research paper classifcation systems based on TF-IDF and LDA schemes.

[5] A. Alves, Stacking machine learning classifiers to identify Higgs bosons at the LHC.

[6] B. Siji George C G, Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction.

[7] J. Devlin, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,» [En línea]. Available: https://nlp.stanford.edu/seminar/details/jdevlin.pdf.

[8] «Hugging Face,» [En línea]. Available: https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased .

[9] «DETOXIS IberLFEF 2021 Results,» [En línea]. Available: https://detoxisiberlef.wixsite.com/website/evaluation-results.