AI-UPV at IberLEF-2021 DETOXIS task: Toxicity Detection in Immigration-Related Web News Comments Using Transformers and Statistical Models

Angel Felipe Magnossão de Paula[0000−0001−8575−5012] and Ipek Baris Schlicht[0000−0002−5037−2203]

Universitat Politècnica de València, Spain
{adepau, ibarsch}@doctor.upv.es

Abstract. This paper describes our participation in the DEtection of TOXicity in comments In Spanish (DETOXIS) shared task 2021 at the 3rd Workshop on Iberian Languages Evaluation Forum. The shared task is divided into two related classification tasks: (i) Task 1: toxicity detection and; (ii) Task 2: toxicity level detection. They focus on the xenophobic problem exacerbated by the spread of toxic comments posted in different online news articles related to immigration. One of the necessary efforts towards mitigating this problem is to detect toxicity in the comments. Our main objective was to implement an accurate model to detect xenophobia in comments about web news articles within the DETOXIS shared task 2021, based on the competition's official metrics: the F1-score for Task 1 and the Closeness Evaluation Metric (CEM) for Task 2. To solve the tasks, we worked with two types of machine learning models: (i) statistical models and (ii) Deep Bidirectional Transformers for Language Understanding (BERT) models. We obtained our best results in both tasks using BETO, a BERT model trained on a big Spanish corpus. We obtained the 3rd place in Task 1 official ranking with the F1-score of 0.5996, and we achieved the 6th place in Task 2 official ranking with the CEM of 0.7142. Our results suggest: (i) BERT models obtain better results than statistical models for toxicity detection in text comments; (ii) Monolingual BERT models have an advantage over multilingual BERT models in toxicity detection in text comments in their pre-trained language.

Keywords: Spanish text classification · Toxicity detection · Deep Learning · Transformers · BERT · Statistical models.

1 Introduction

The increase in the number of news pages where the reader can openly discuss the articles has driven the dissemination of internet users' opinions through so-

IberLEF 2021, September 2021, Málaga, Spain.

cial media [18, 10]. A survey carried out in the US by The Center for Media Engagement at the University of Texas at Austin states that most of the comments on news articles are posted by internet users who we call active-users or influencers [16]. They are highly active and generate huge amounts of data.

The imbalance in the amount of data generated by influencers and non-active users creates a distorted reality where influencers' opinions end up representing the opinion of all internet users to society [2]. This distorted reality can aggravate the existing social problems, as is the case with xenophobia, a heavy sense of aversion, or dread of people from other countries [19].

In recent years, the problem with xenophobia has been exacerbated by the increase in the spread of toxic comments posted in different online news articles related to immigration [3]. One of the first steps to mitigate the problem is to detect toxic comments regarding news articles [5]. For this reason, the Iberian Languages Evaluation Forum proposed the DEtection of TOXicity in comments In Spanish (DETOXIS) shared task 2021 [17].

The DETOXIS shared task comprises Task 1 and Task 2, which are respectively toxicity detection and toxicity level detection. The two tasks are performed on comments posted in Spanish in response to different online news articles related to immigration. Task 1 is a binary classification problem where the objective is to classify a Spanish text comment as 'toxic' or 'not toxic'. Task 2 aims to classify the same comment but among four classes: 'not toxic', 'mildly toxic', 'toxic', or 'very toxic'. Table 1 displays examples of comments classified across all classes.

Table 1. Comments examples

Toxicity	Toxicity_level	New's comment
not toxic	not toxic	Proximamente en su barrio
toxic	mildly toxic	Vienen a pagarnos las pensiones
	toxic	asi me gusta, que se maten entre ellos y en alta mar. Mas inmigrantes asi porfavor
	very toxic	A esosmoros hay que echarlos pero ya.O los politicos hacen algo o la gente tendra que ”actuar”

The detection of toxicity in comments is mostly done with Machine Learning (ML) models, especially deep learning models, which require large amounts of annotated datasets for robust predictions [8]. However, labeling toxicity is a challenging and time-consuming task that requires many annotators to avoid bias, and the annotators should be aware of social and cultural contexts [15, 11].

Our main goal was to implement an accurate model to detect xenophobic comments on web news articles within the DETOXIS shared task 2021, using the competition's official metrics. We decided to solve the problem by applying models that can learn using only a small amount of data, which can be done with statistical models and most advanced pre-trained deep learning models. Roughly speaking, there are two types of statistical models: Generative and Discriminative [9]. We chose to use one of each type. Thus, we tried a Naive Bayes (Generative) and a Maximum Entropy (Discriminative) model. Among the most advanced and highly effective deep learning models is Deep Bidirectional Transformers for Language Understanding (BERT), which comes with its parameters pre-trained in an unsupervised manner in a large corpus [7]. Therefore, it only needs a tuned train that can be run on a small set of data, which suits our problem. Our source code is publicly available[1]

The work's main contribution is to help in the effort to improve the results in the identification of toxic comments in news articles related to immigration. Unlike the vast majority of works [14], we use ML models that can tackle the xenophobia detection problem having only little data available. The second contribution is to build ML models and find their best configuration to deal not only with the classification of news articles as 'toxic' and 'not toxic', but also to infer the toxicity level of the comments into 'not toxic', 'mildly toxic', 'toxic', or 'very toxic'. As far as we know, there are few works in the literature in which the solution model tries to infer the toxicity level of the comments posted in the news related to immigration. On the DETOXIS official ranking, we obtained the 3rd place in Task 1 with the F1-score of 0.5996, and we achieved the 6th place in Task 2 with the CEM of 0.7142.

The article is organized as follows: Section 2 contains the methodology with fundamental concepts; Section 3 describes the experiments; Section 4 contains the results and discussions, and and Section 5 draws some conclusion and future work.

2 Methodology

This section explains the data structure, the evaluation metrics, and the ML models applied to solve our classification problems. In addition, the text representation used to encode the text comments.

2.1 Dataset

The DETOXIS shared task organization granted its participants the NewsCom-TOX dataset [17] divided into train set and test set where text data are in Spanish. The train set consists of 3463 instances, and the test set consists of 891 instances. Both sets have as main labels: (i) 'Comment_id' and (ii) 'Comment'; but only the train set has the labels: (iii) 'Toxicity' and (iv) 'Toxicity_level', respectively for Task 1 and Task 2. The 'Comment_id' is a unique reference number assigned to each instance within the NewsCom-TOX dataset. The 'Comment' label is a text message posted in response to a Spanish online news article from different sources such as El Mundo, NIUS, ABC, etc., or discussion forums like

[1] https://github.com/AngelFelipeMP/Machine-Learning-Tweets-Classification

Menéame. Moreover, ‘Toxicity’ labels the comment for a particular instance between ‘toxic’ or ‘not toxic’ and the ‘Toxicity_level’ label classifies the same comment as ‘not toxic’, ‘mildly toxic’, ‘toxic’, or ‘very toxic’. Table 2 shows the label’s distribution for ‘Toxicity’ and ‘Toxicity_level’. We can see that the labels are unbalanced in both cases.

Table 2. Data distribution

Toxicity (Task 1)		Toxicity_level (Task 2)	
Label	Number of instances	Label	Number of instances
not toxic	2316	not toxic	2317
toxic	1147	mildly toxic	808
		toxic	269
		very toxic	69

The data annotation process was carried out by four annotators where two were linguists experts, and two were trained linguistic students. Three of them labeled all news article comments in parallel. Once they finished, an inter-annotator agreement test was executed. When a disagreement happens, the three annotators plus the senior annotator reviewed it in order to achieve accordance with the final label [17]. In Table 1, we can see examples from the DETOXIS train set of comments and its labels attributed by the annotators for ‘Toxicity’ and ‘Toxicity_level’.

Next we explain how we used the data during the project development in both tasks. First, we applied 10-fold cross-validation in the train set to find the best ML model. After that, we trained the selected model in the whole train set. Subsequently, we applied the selected model to make predictions on the official test set, as shown in Figure 1. These predictions were submitted to the DETOXIS shared task 2021.

2.2 Evaluation metrics

Because the train set is imbalanced, as we can see in Table 2, we selected evaluation metrics that are able to fairly evaluate ML models in this circumstance. For Task 1, we adopted Accuracy, Recall, Precision, and F1-score, which was the DETOXIS official evaluation metric for Task 1. For Task 2, we adopted Accuracy, F1-macro, F1-weighted, Recall, Precision, and CEM [1], the DETOXIS official evaluation metric for Task 2. We used the DETOXIS official metrics as performance measures to rank and select the best ML models during the cross-validation process for Task 1 and Task 2.

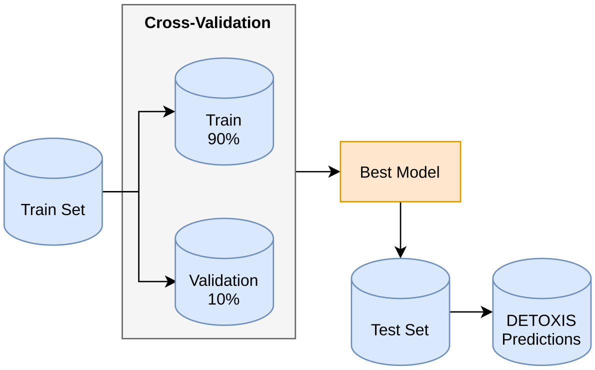

Fig. 1. Workflow.

2.3 Models

There are two types of statistical models: the Generative models and the Discriminative models [9]. We used one model from each kind. We adopted the Naive Bayes (Generative) model and the Maximum Entropy (Discriminative) model. Among the Transformers models, we decided to use the BERT models, one of the most advanced and highly effective Transformer models. They come with their parameters pre-trained in an unsupervised manner in a large corpus [7]. Therefore, they only need a supervised fine-tune train in the downstream task that can be run on a small set of data. We adopted: (i) the BETO model, a BERT model trained on a big unannotated Spanish corpus composed of three billion tokens [4]; and (ii) the mBERT, a BERT model pre-trained on the top 102 languages with the most extensive Wikipedia corpus. However, the balance among the language in the corpus was not perfect. For example, the English partition of the corpus was 1000 bigger than the Icelandic partition [6].

2.4 Text representation

To represent our text data in a way that the statistical models could handle it, we used two encode methods: (i) Bag of Words (BOW) [20]; and (ii) Term Frequency - Inverse Document Frequency (TF-IDF) [13]. The BOW represents a text comment by a unidimensional vector whose length is the size of the training vocabulary. In this case, each column of this vector contains the number of times a particular word from the vocabulary appears in the specific comment. The TF-IDF representation for each text comment is also a flat vector with the size of the training vocabulary. However, the value for each word on the vector follows the well-known TF-IDF calculation [13].

3 Experiments

This section explains the environment setup, the data preprocessing, and statistical models' feature extraction. Furthermore, the section also contains explanations for the 10-fold cross-validation process and how we selected the model to make predictions on the DETOXIS test set, which we submitted as our final results to the competition.

3.1 Environment setup

For code purposes, we used python 3.7.10. As a code editor/machine, we used Google collaborator[2]. The main python libraries that we used were: (i) NumPy 1.19.5 to work with matrix, (ii) Pandas 1.1.5 to handle and visualize data, (iii) Spacy 2.2.4 and (iv) the Natural Language Toolkit (NLTK) 3.2.5 for natural language transformations, (v) Pytorch 1.6.0, and (vi) Transformers 3.0.0 to actually implement the BERT models. In addition, we used (vii) Sklearn 0.22.2 to implement the statistical models.

3.2 Preprocessing

For both tasks, we only preprocess the data for the statistical models. The preprocessing step was carried out on the text data from the train and test sets. We used the built-in python model for Regular Expression (RegEx) and the NLTK python library. Applying RegEx, we removed stock market tickers, old-style retweet text, hashtags, hyperlinks and changed the numbers to the tag "<number>". We employed the NLTK on the text comments to remove stop-words, stem and tokenize the words.

3.3 Feature extraction

The feature extraction process was executed to focus on achieving good results with the statistical models. These models' performance is susceptible to their input features [12]. Hence, after preprocessing the datasets for the statistical models, we executed the feature extraction process to create good input features. We encode the text comments in two different manners: (i) BOW [20]; and (ii) TF-IDF [13].

The two proposed encode methods are based on word occurrences, and unfortunately, they completely ignore the relative position information of the words in the comments. Therefore, we lose the information about the local ordering of the words. In order to mitigate this problem and preserve some of the local word ordering information, we increase the vocabulary by extracting 2-grams and 3-grams from the text comments despite dimensional increasing.

[2] `https://colab.research.google.com/`

3.4 Cross-validation

The cross-validation process was performed on the train set aiming to find the best ML model to make the prediction on the DETOXIS test set. We can see the summary of our cross-validation process in Figure 2. During the cross-validation, each statistical model received different input features, and the BERT models tried different hyper-parameters.

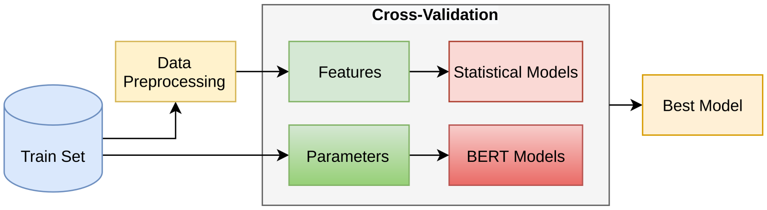

Fig. 2. General diagram ML models cross-validation.

In Figure 3, we can see the cross-validation process that focuses on the BERT models. We tried different combinations for the Output BERT, Learning Rate, Batch Size, and Epochs. The BERT models are composed of a pre-trained model plus a linear layer at the top which receives the output of the pre-trained BERT model. We have two different options for the Output BERT: (i) the sequence of hidden states at the output of the last layer which we performed a mean pooling and max pooling operation and concatenated them into a unified unidimensional vector that we called 'hidden'; (ii) the pooler of the last layer's hidden state of the first token of the sequence further processed by a linear layer and a tanh activation function that we called 'pooler'. For the Learning Rate, we tried 1E-5, 3E-5, and 5E-5. For the Batch Size, we tried 8, 16, 32, and 64. The number of Epochs was from 1 to 20.

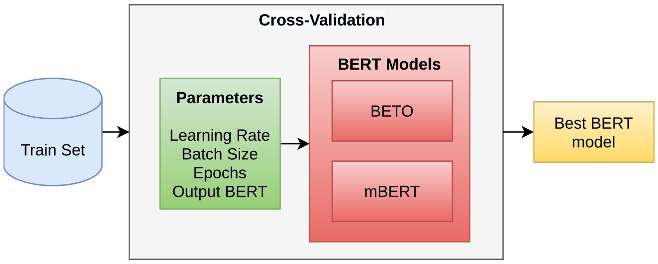

Fig. 3. Cross-validation BERT models.

Figure 4 illustrates the cross-validation process for the statistical models. We can see that we tried four algorithm versions of the Naive Bayes (NB) model:

the Multinomial, the Bernoulli, the Gaussian, and the Complement ones. On the other hand, we tried only the original version of the Maximum Entropy (ME) model but with different solvers: the liblinear, newton, sag, saga, and lbfgs. We call solvers the algorithms used in the optimization problem. We tried different vocabulary sizes for all statistical models using different n-grams combinations and the two encode methods: BOW and TF-IDF.

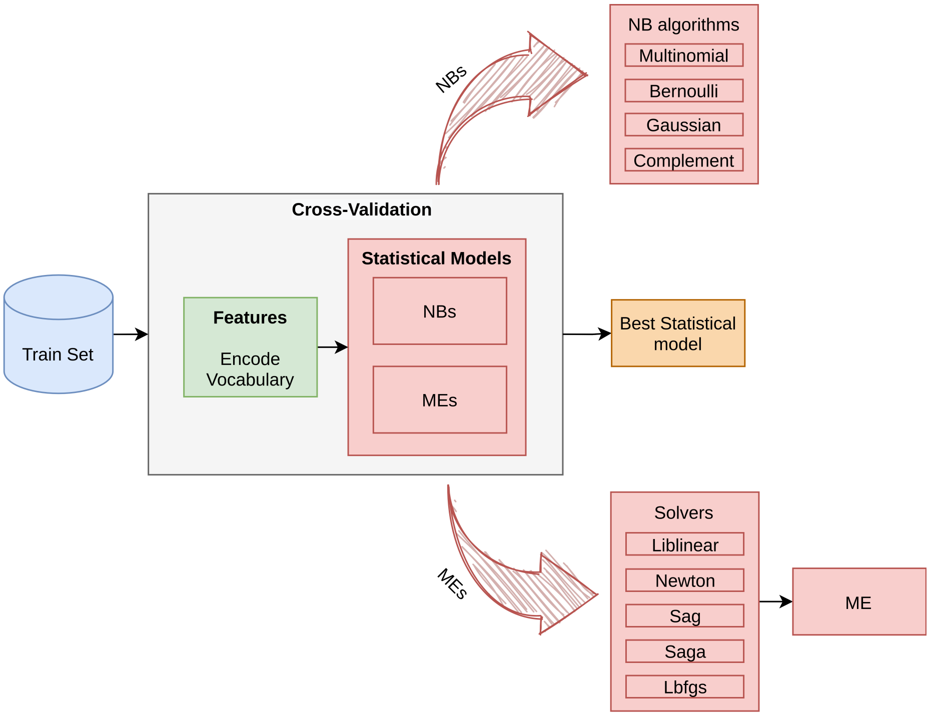

Fig. 4. Cross-validation statistical models.

Tables 3, 4, 5, and 6 show the results of the 10-fold cross-validation process for the statistical models. Tables 3 and 4, in this sequence, show the results of the ME model and the NB model for Task 1. Tables 5 and 6 respectively show the results of the ME model and the NB model for Task 2. The first column in Tables 3 and 4 shows the solver, and the first column in Tables 5 and 6 shows the NB algorithm. All the other columns have the same meaning in the four tables.

Thus, the tables' third column shows the n-grams used as a vocabulary where the numbers within the parentheses are the lower and upper limits of the n-gram word range used. The n-gram range of (1, 1) means only 1-grams, (1, 2) means 1-grams and 2-grams, and (1, 3) means 1-grams, 2-grams, and 3-grams. The rest of the columns have the evaluation metrics for each group of the selected parameters. For Task 1, the evaluation metrics are Accuracy, F1-score, Recall, and Precision, and for Task 2, the evaluation metrics are Accuracy, F1-macro, F1-weighted, Recall, Precision, and CEM.

Table 3. Cross-validation ME models for Task 1

Solver	Encoder	Vocabulary	Accuracy	F1-score	Recall	Precision
Liblinear	TF-IDF	(1,1)-grams	0.7112	0.3031	0.2021	0.6882
		(1,2)-grams	0.6976	0.1546	0.0949	0.8786
		(1,3)-grams	0.6893	0.1036	0.0635	0.6500
	BOW	(1,1)-grams	0.6994	0.4652	0.3993	0.5649
		(1,2)-grams	0.7118	0.4353	0.3434	0.6060
		(1,3)-grams	**0.7126**	0.4101	0.3128	0.6188
Newton	TF-IDF	(1,1)-grams	0.7115	0.3043	0.2030	0.6890
		(1,2)-grams	0.6979	0.1547	0.0949	**0.8928**
		(1,3)-grams	0.6893	0.1036	0.0635	0.6500
	BOW	(1,1)-grams	0.6991	0.4645	0.3984	0.5647
		(1,2)-grams	0.7106	0.4331	0.3416	0.6030
		(1,3)-grams	0.7132	0.4106	0.3128	0.6212
Sag	TF-IDF	(1,1)-grams	0.7115	0.3043	0.2030	0.6890
		(1,2)-grams	0.6979	0.1547	0.0949	**0.8928**
		(1,3)-grams	0.6893	0.1036	0.0635	0.6500
	BOW	(1,1)-grams	0.7002	**0.4679**	**0.4019**	0.5670
		(1,2)-grams	0.7141	0.4427	0.3512	0.6110
		(1,3)-grams	0.7155	0.4168	0.3189	0.6248
Saga	TF-IDF	(1,1)-grams	0.7112	0.3031	0.2021	0.6882
		(1,2)-grams	0.6979	0.1562	0.0957	0.8800
		(1,3)-grams	0.6893	0.1036	0.0635	0.6500
	BOW	(1,1)-grams	0.6985	0.4652	0.4002	0.5629
		(1,2)-grams	0.7135	0.4432	0.3521	0.6102
		(1,3)-grams	0.7181	0.4246	0.3242	0.6337
Lbfgs	TF-IDF	(1,1)-grams	0.7115	0.3043	0.2030	0.6890
		(1,2)-grams	0.6979	0.1547	0.0949	**0.8928**
		(1,3)-grams	0.6893	0.1036	0.0635	0.6500
	BOW	(1,1)-grams	0.6991	0.4645	0.3984	0.5647
		(1,2)-grams	0.7106	0.4331	0.3416	0.6030
		(1,3)-grams	0.7132	0.4106	0.3128	0.6212

Table 4. Cross-validation NB models for Task 1

NB Algorithm	Encoder	Vocabulary	Accuracy	F1-score	Recall	Precision
Multinomial	TF-IDF	(1,1)-grams	**0.6933**	0.1703	0.1062	**0.7282**
		(1,2)-grams	0.6878	0.0995	0.0609	0.7167
		(1,3)-grams	0.6843	0.0805	0.0487	0.5500
	BOW	(1,1)-grams	0.6685	0.4868	0.4821	0.4960
		(1,2)-grams	0.6480	0.5232	0.5868	0.4736
		(1,3)-grams	0.5795	**0.5355**	0.7289	0.4238
Bernoulli	TF-IDF	(1,1)-grams	0.6674	0.3344	0.2574	0.4979
		(1,2)-grams	0.6524	0.1910	0.1274	0.4297
		(1,3)-grams	0.6529	0.1765	0.1160	0.4168
	BOW	(1,1)-grams	0.6674	0.3344	0.2574	0.4979
		(1,2)-grams	0.6524	0.1910	0.1274	0.4297
		(1,3)-grams	0.6529	0.1765	0.1160	0.4168
Gaussian	TF-IDF	(1,1)-grams	0.4730	0.4192	0.5831	0.3294
		(1,2)-grams	0.5287	0.3918	0.4733	0.3386
		(1,3)-grams	0.5307	0.3961	0.4794	0.3418
	BOW	(1,1)-grams	0.4675	0.4282	0.6102	0.3317
		(1,2)-grams	0.5223	0.4068	0.5090	0.3425
		(1,3)-grams	0.5249	0.4084	0.5099	0.3442
Complement	TF-IDF	(1,1)-grams	0.6604	0.4083	0.3800	0.4633
		(1,2)-grams	0.6785	0.3255	0.2648	0.4845
		(1,3)-grams	0.6835	0.3378	0.2727	0.5071
	BOW	(1,1)-grams	0.6234	0.5165	0.6156	0.4472
		(1,2)-grams	0.5928	0.5216	0.6749	0.4263
		(1,3)-grams	0.5215	0.5256	**0.8004**	0.3915

Table 5. Cross-validation ME models for Task 2

Solver	Encoder	Vocabulary	Accuracy	F1-macro	F1-weighted	Recall	Precision	CEM
Liblinear	TF-IDF	(1,1)-grams	0.6826	0.2363	0.5753	0.2691	0.2798	0.7070
		(1,2)-grams	0.6809	0.2233	0.5610	0.2629	0.2693	0.6972
		(1,3)-grams	0.6797	0.2214	0.5585	0.2616	0.2649	0.6923
	BOW	(1,1)-grams	0.6526	0.3236	0.6034	0.3129	0.4473	0.6831
		(1,2)-grams	0.6722	0.3070	0.6038	0.3059	0.4539	0.7018
		(1,3)-grams	0.6780	0.2944	0.5998	0.2995	0.4418	**0.7080**
Newton	TF-IDF	(1,1)-grams	0.6826	0.2509	0.5870	0.2767	0.3199	0.7067
		(1,2)-grams	**0.6846**	0.2302	0.5691	0.2675	0.2934	0.7041
		(1,3)-grams	0.6829	0.2267	0.5643	0.2652	0.2649	0.6977
	BOW	(1,1)-grams	0.6465	**0.3587**	**0.6125**	**0.3367**	**0.4942**	0.6827
		(1,2)-grams	0.6682	0.3176	0.6079	0.3117	0.4504	0.6984
		(1,3)-grams	0.6740	0.2997	0.6028	0.3019	0.4303	0.7022
Sag	TF-IDF	(1,1)-grams	0.6826	0.2509	0.5870	0.2767	0.3199	0.7067
		(1,2)-grams	**0.6846**	0.2302	0.5691	0.2675	0.2934	0.7041
		(1,3)-grams	0.6829	0.2267	0.5643	0.2652	0.2649	0.6977
	BOW	(1,1)-grams	0.6460	0.3393	0.6107	0.3251	0.4386	0.6824
		(1,2)-grams	0.6690	0.3101	0.6077	0.3073	0.4306	0.6997
		(1,3)-grams	0.6742	0.2919	0.6021	0.2977	0.3976	0.7039
Saga	TF-IDF	(1,1)-grams	0.6826	0.2509	0.5870	0.2767	0.3199	0.7067
		(1,2)-grams	**0.6846**	0.2302	0.5691	0.2675	0.2934	0.7041
		(1,3)-grams	0.6829	0.2267	0.5643	0.2652	0.2649	0.6977
	BOW	(1,1)-grams	0.6459	0.3380	0.6102	0.3241	0.4387	0.6825
		(1,2)-grams	0.6699	0.3145	0.6077	0.3101	0.4526	0.7015
		(1,3)-grams	0.6763	0.2902	0.6027	0.2974	0.4028	0.7053
Lbfgs	TF-IDF	(1,1)-grams	0.6826	0.2509	0.5870	0.2767	0.3199	0.7067
		(1,2)-grams	**0.6846**	0.2302	0.5691	0.2675	0.2934	0.7041
		(1,3)-grams	0.6829	0.2267	0.5643	0.2652	0.2649	0.6977
	BOW	(1,1)-grams	0.6465	0.3587	**0.6125**	**0.3367**	**0.4942**	0.6827
		(1,2)-grams	0.6682	0.3176	0.6079	0.3117	0.4504	0.6984
		(1,3)-grams	0.6740	0.2997	0.6028	0.3019	0.4303	0.7022

Table 6. Cross-validation NB models for Task 2

NB Algorithm	Encoder	Vocabulary	Accuracy	F1-macro	F1-weighted	Recall	Precision	CEM
Multinomial	TF-IDF	(1,1)-grams	0.6743	0.2160	0.5523	0.2576	0.2393	0.6808
		(1,2)-grams	**0.6769**	0.2161	0.5528	0.2583	0.2417	**0.6882**
		(1,3)-grams	0.6766	0.2158	0.5523	0.2580	0.2686	0.6881
	BOW	(1,1)-grams	0.6151	0.2736	0.5806	0.2807	0.2796	0.6384
		(1,2)-grams	0.6061	0.2747	0.5798	0.2858	0.2858	0.6473
		(1,3)-grams	0.5137	0.2657	0.5250	0.2878	0.2810	0.6133
Bernoulli	TF-IDF	(1,1)-grams	0.6220	0.2472	0.5491	0.2631	0.2782	0.6210
		(1,2)-grams	0.6396	0.2212	0.5451	0.2504	0.2302	0.6286
		(1,3)-grams	0.6396	0.2175	0.5426	0.2485	0.2252	0.6268
	BOW	(1,1)-grams	0.6220	0.2472	0.5491	0.2631	0.2782	0.6210
		(1,2)-grams	0.6396	0.2212	0.5451	0.2504	0.2302	0.6286
		(1,3)-grams	0.6396	0.2175	0.5426	0.2485	0.2252	0.6268
Gaussian	TF-IDF	(1,1)-grams	0.4031	0.2311	0.4429	0.2345	0.2471	0.5075
		(1,2)-grams	0.4923	0.2530	0.5056	0.2540	0.2586	0.5376
		(1,3)-grams	0.4915	0.2532	0.5058	0.2541	0.2592	0.5386
	BOW	(1,1)-grams	0.4000	0.2333	0.4418	0.2398	0.2504	0.5081
		(1,2)-grams	0.4834	0.2534	0.5009	0.2566	0.2590	0.5361
		(1,3)-grams	0.4834	0.2538	0.5015	0.2570	0.2597	0.5370
Complement	TF-IDF	(1,1)-grams	0.5911	0.2746	0.5648	0.2837	0.2893	0.5948
		(1,2)-grams	0.6483	0.2749	0.5811	0.2880	**0.3171**	0.6342
		(1,3)-grams	0.6497	0.2800	**0.5845**	0.2936	0.3162	0.6377
	BOW	(1,1)-grams	0.4932	**0.2916**	0.5289	0.3211	0.3002	0.5751
		(1,2)-grams	0.3647	0.2509	0.4386	**0.3365**	0.3035	0.5459
		(1,3)-grams	0.2010	0.1749	0.2712	0.3261	0.3086	0.4923

Tables 7, 8, 9, and 10 show the top 5 results obtained in the 10-fold cross-validation process for the BERT models. We used the DETOXIS official metrics to rank the models, which are the F1-score for Task 1 and the CEM for Task 2. Tables 7 and 8, in this sequence, show the top 5 results of the mBERT model and the BETO model for Task 1. Tables 9 and 10 respectively show the top 5 results of the mBERT model and the BETO model for Task 2. In all four tables, the first column shows the BERT model, and the second column displays the type of Output BERT. The third column shows Learning Rate, the fourth column shows the Batch Size, and the fifth column indicates the number of Epochs. The rest of the columns have the evaluation metrics for each group of the selected parameters. For Task 1, the evaluation metrics are Accuracy, F1-score, Recall, and Precision, and for Task 2, the evaluation metrics are Accuracy, F1-macro, F1-weighted, Recall, Precision, and CEM.

Table 7. Top 5 mBERT models cross-validation for Task 1

Model	Output BERT	Learning Rate	Batch Size	Epochs	Accuracy	F1-score	Recall	Precision
mBERT	pooler	3E-05	32	11	0.6972	**0.6010**	0.6842	0.5594
	hidden	5E-05	32	8	0.7094	0.5865	0.6167	0.5759
	hidden	5E-05	32	9	0.7102	0.5838	0.6202	0.5713
	hidden	3E-05	64	16	0.7259	0.5819	0.5778	0.6083
	pooler	3E-05	16	8	0.6838	0.5798	0.6715	0.5319

Table 8. Top 5 BETO models cross-validation for Task 1

Model	Output BERT	Learning Rate	Batch Size	Epochs	Accuracy	F1-score	Recall	Precision
BETO	pooler	1E-05	32	4	0.7446	**0.6314**	0.6514	0.6338
	pooler	1E-05	64	7	0.7415	0.6276	0.6578	0.6184
	pooler	1E-05	16	7	0.7267	0.6265	0.6829	0.6016
	pooler	5E-05	64	14	0.7554	0.6245	0.6203	0.6485
	pooler	3E-05	64	16	0.7565	0.6237	0.6117	0.6568

Table 9. Top 5 mBERT models cross-validation for Task 2

Model	Output BERT	Learning Rate	Batch Size	Epochs	Accuracy	F1 macro	F1 weighted	Recall	Precision	CEM
mBERT	pooler	1E-05	16	12	0.7031	0.4165	0.7477	0.4206	0.4483	**0.7599**
	hidden	1E-05	16	14	0.6955	0.4158	0.7486	0.4252	0.4344	0.7588
	hidden	3E-05	16	4	0.7006	0.3839	0.7475	0.3970	0.4054	0.7581
	hidden	1E-05	16	10	0.7011	0.3832	0.7515	0.3917	0.4210	0.7580
	pooler	1E-05	16	4	0.6974	0.3984	0.7496	0.4067	0.4182	0.7580

Table 10. Top 5 BETO models cross-validation for Task 2

Model	Output BERT	Learning Rate	Batch Size	Epochs	Accuracy	F1 macro	F1 weighted	Recall	Precision	CEM
BETO	hidden	1E-05	16	4	0.7170	0.4035	0.7678	0.4091	0.4469	**0.7769**
	hidden	1E-05	8	3	0.7165	0.4138	0.7696	0.4151	0.4611	0.7747
	hidden	3E-05	32	6	0.7188	0.4096	0.7483	0.4173	0.4355	0.7746
	hidden	3E-05	64	5	0.7148	0.4178	0.7632	0.4235	0.4461	0.7746
	hidden	1E-05	8	5	0.7153	0.4168	0.7649	0.4219	0.4592	0.7739

3.5 Best model

At the end of the cross-validation, we selected the best model for each task accordingly with the DETOXIS official metric for the specified task, as shown in Figure 5.

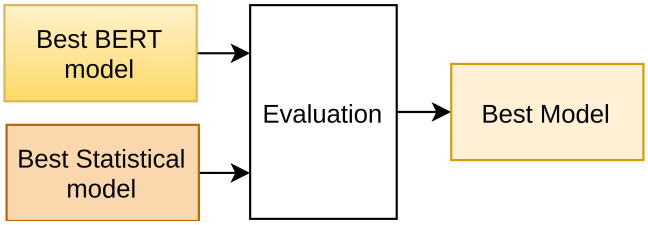

Fig. 5. Selection of the best ML model.

Table 11 shows the best results for each ML model tried on the cross-validation process for Task 1, which are mBERT, BETO, NB, and ME. Table 12 displays the top 5 best model for the Task 1 during the whole cross-validation process.

Table 11. The best result of each model in the cross-validation for Task 1

Model	Accuracy	F1-score	Recall	Precision
BETO	0.7446	**0.6314**	0.6514	0.6338
mBERT	0.6972	0.6010	0.6842	0.5594
NB	0.5795	0.5355	0.7289	0.4238
ME	0.7002	0.4679	0.4019	0.5670

Table 12. Top 5 models cross-validation for Task 1

Model	Output BERT	Learning Rate	Batch Size	Epochs	Accuracy	F1-score	Recall	Precision
BETO	pooler	1E-05	32	4	0.7446	**0.6314**	0.6514	0.6338
BETO	pooler	1E-05	64	7	0.7415	0.6276	0.6578	0.6184
BETO	pooler	1E-05	16	7	0.7267	0.6265	0.6829	0.6016
BETO	pooler	5E-05	64	14	0.7554	0.6245	0.6203	0.6485
BETO	pooler	3E-05	64	16	0.7565	0.6237	0.6117	0.6568

Table 13 shows the best models performace for each ML model on the cross-validation process for Task 2, which are mBERT, BETO, ME, and NB. Table 14 displays the top 5 best model for the Task 2 whole cross-validation process.

Table 13. The best result of each model in the cross-validation for Task 2

Model	Accuracy	F1-macro	F1-weighted	Recall	Precision	CEM
BETO	0.7170	0.4035	0.7678	0.4091	0.4469	**0.7769**
mBERT	0.7031	0.4165	0.7477	0.4206	0.4483	0.7599
ME	0.6780	0.2944	0.5998	0.2995	0.4418	0.7080
NB	0.6769	0.2161	0.5528	0.2583	0.2417	0.6882

Table 14. Top 5 models cross-validation for Task 2

Model	Output BERT	Learning Rate	Batch Size	Epochs	Accuracy	F1 macro	F1 weighted	Recall	Precision	CEM
BETO	hidden	1E-05	16	4	0.7170	0.4035	0.7678	0.4091	0.4469	**0.7769**
BETO	hidden	1E-05	8	3	0.7165	0.4138	0.7696	0.4151	0.4611	0.7747
BETO	hidden	3E-05	32	6	0.7188	0.4096	0.7483	0.4173	0.4355	0.7746
BETO	hidden	3E-05	64	5	0.7148	0.4178	0.7632	0.4235	0.4461	0.7746
BETO	hidden	1E-05	8	5	0.7153	0.4168	0.7649	0.4219	0.4592	0.7739

After the cross-validation, we chose the best model for Task 1, which following Table 12 is BETO with the respective parameters: (i) pooler as Output BERT; (ii) 1E-05 Learning Rate; (iii) Batch Size equal 32; and (iv) 4 training Epochs. We also selected the best model for Task 2 that following Table 14 is BETO with the respective parameters: (i) hidden Output BERT; (ii) 1E-05 Learning Rate; (iii) Batch Size equal 16; and (iv) 4 training Epochs. Having the best models and their parameters, we trained the models on the train set.

Once the best models are trained, we use those models to make the predictions on the DETOXIS test set. These predictions afterward were submitted to the DETOXIS shared task organization as our final results.

4 Results and Discussion

We discovered important information on the cross-validation results. Looking at Table 3, we can see that the ME model achieves its best results on Task 1 with the BOW encode based on the F1-score evaluation metric, which is 0.4679. The highest Accuracy 0.7126 and Recall 0.4019 are also performed with the BOW encode. The only performance metric in which the TF-IDF encode obtains a higher score is Precision that is 0.8928. Thus, we can conclude that BOW is the best encoding for the ME model on Task 1 in the DETOXIS training set. Moreover, employing the Sag solver, the ME model achieved a higher F1-score, Recall, and Precision. Hence, it seems to us that Sag was the best solver for the ME model on Task 1 in the DETOXIS training set. We do not have a definitive conclusion about the vocabulary size because the ME model achieved its highest results with different numbers of n-grams for each metric.

Observing Table 4, we see that the NB model achieves its best results on Task 1 with the BOW encode based on the F1-score evaluation metric, which is 0.5355. The NB model also obtained the higest Recall 0.8004 with the BOW encode, but its highest results for Accuracy 0.6933 and Precision 0.7282 were with the TF-IDF encode. Therefore, we can not conclude which encode method is the best for the NB model on Task 1 in the DETOXIS training set. A similar case occurs with the vocabulary size where the NB model that employed 1-grams, 2-grams, and 3-grams achieved the highest F1-score and Recall. However, the NB model with a 1-grams vocabulary size obtained the highest Accuracy and Precision. The different NB algorithms obtained a similar performance based on the F1-score. In most cases, they achieved their best results with the BOW encode.

We can see in Table 5 that the ME model achieved its best results on Task 2 with the BOW encode based on the CEM evaluation metric, which is 0.7080. The highest F1-macro 0.3587, F1-weighted 0.6125, Recall 0.3367, and Precision 0.4942 are also obtained with the BOW encode. The only performance metric in which the TF-IDF encode obtains a higher score is the Accuracy, which is 0.6846. Thus, we can conclude that BOW is the best encoding for the ME model on Task 2 in the DETOXIS training set. Moreover, employing the Newton solver, the ME model achieved a higher Accuracy, F1-macro, F1-weighted, Recall, and Precision.

Hence, it seems to us that Newton was the best solver for the ME model on Task 2 in the DETOXIS training set. We concluded that the vocabulary size of 1-grams is the best for the ME model on Task 2 in the DETOXIS training set because the ME model achieved its highest Accuracy, F1-macro, F1-weighted, Recall, and Precision.

Table 6 shows that the NB model achieved its best results on Task 2 with the TF-IDF encode based on the CEM evaluation metric, which is 0.6882. The NB model also obtained its highest Accuracy 0.6769, F1-weighted 0.5845, and Precision 0.3171, with the TF-IDF encode, but its highest results for F1-macro 0.2916 and Recall 0.3365 were obtained with the BOW encode. Therefore, we can conclude that the TF-IDF encode best suits the NB model on Task 2 in the DETOXIS training set. We see indications in Table 6 that the ideal vocabulary for the NB model on Task 2 in the DETOXIS training set is composed of 1-grams and 2 grams. Once with this vocabulary, the model obtained its highest Accuracy, Recall, Precision, and CEM results. The different NB algorithms obtained similar performance based on the CEM ranged from 0.49 to 0.68.

Based on the F1-score, the mBERT model achieved its best performance on Task 1 with a value of 0.6010, as we can see in Table 7. The model parameters are the following: (i) pooler as Output BERT; (ii) 3E-05 Learning Rate; (iii) Batch Size equal 32; and (iv) 11 training Epochs. Table 8 shows that the BETO model obtained its best performance on Task 1 also based on the F1-score with the following parameters: (i) pooler as Output BERT; (ii) 1E-05 Learning Rate; (iii) Batch Size equal 32; and (iv) 4 training Epochs. The BETO model obtained a F1-score value of 0.6314, which was also the highest among all the ML models in the cross-validation process. For this reason, the BETO model with the mentioned parameters was used for our Task 1 official prediction on the DETOXIS test set. These predictions afterward were submitted as our official Task 1 results.

Observing Table 9, we can conclude that based on the CEM, the mBERT model achieved its best performance on Task 1 with the following parameters: (i) pooler as Output BERT; (ii) 1E-05 Learning Rate; (iii) Batch Size equal to 16; and (iv) 12 training Epochs. This model achieved the CEM of 0.7599. Table 10 shows that the BETO model obtained its best performance on Task 2 also based on the CEM with the following parameters: (i) hidden as Output BERT; (ii) 1E-05 Learning Rate; (iii) Batch Size equal to 16; and (iv) 4 training Epochs. The BETO model obtained CEM value of 0.7769, which was also the highest among all the ML models in the cross-validation process. For this reason, the BETO model with the mentioned parameters was used for our Task 2 official prediction on the DETOXIS test set. These predictions afterward were submitted as our official Task 2 results.

To sum up the comments about the cross-validation results, looking at Tables 12 and 14, we can see that the BETO model with different combinations of parameters obtained the five first positions on the ranking for the best ML model for Task 1 and Task 2.

The DETOXIS organization provided us with the results of the test set. Table 15 shows our result on Task 1 plus the three official DETOXIS baselines:

Random Classifier, Chain BOW, and BOW Classifier. Our model obtained an F1-score around 59% greater than the results obtained by the best baseline on Task 1.

Table 15. Test set results for Task 1

Model	F1-score
BETO	**0.5996**
Random Classifier	0.3761
Chain BOW	0.3747
BOW Classifier	0.1837

Table 16 shows the results of our model and the three DETOXIS baselines on Task 2. Our BETO model was able to achieve a CEM of 9% higher than the best DETOXIS baseline result obtained by the Random Classifier.

Table 16. Test set results for Task 2

Model	CEM
BETO	**0.7142**
Chain BOW	0.6535
BOW Classifier	0.6318
Random Classifier	0.4382

On the DETOXIS official ranking, we obtained 3rd place on Task 1 with F1-score 0.5996, and we achieved 6th place on Task 2 with CEM 0.7142.

5 Conclusion and Future Work

Xenophobia is a problem which is aggravated by the increase in the spread of toxic comments posted in different online news articles related to immigration. In this paper, to address this problem within the DETOXIS 2021 shared task, we tried two types of ML models: (i) statistical models and (ii) BERT models. We obtained the best results in both tasks using BETO, a BERT model pre-trained with a big Spanish corpus. Our contributions are as follows: (i) help in the effort to improve the results in the identification of toxic comments in news articles related to immigration. Unlike the vast majority of works, we use ML models that can tackle the xenophobia detection problem having only little data available; (ii) We build an ML model and find its best configuration to deal not only with the classification of news articles as ‘toxic’ and ‘not toxic’, but also to infer the toxicity level of the comments into ‘not toxic’, ‘mildly toxic’, ‘toxic’, or ‘very toxic’.

Based on the DETOXIS official metrics, we concluded that our results indicate that: (i) BERT models obtain better results than statistical models for toxicity and toxicity level detection in text comments; and (ii) Monolingual BERT models achieve higher results in comparison with the multilingual BERT models in toxicity detection and toxicity level detection in their pre-trained language.

After all, our BETO model obtained the 3rd position on Task 1 official ranking with the F1-score of 0.5996, and it achieved the 6th position on Task 2 official ranking with the CEM of 0.7142. As future work, we aim to include sentiment lexicons on the model's input to boost its performance.

References

1. Amigó, E., Gonzalo, J., Mizzaro, S., Carrillo-de Albornoz, J.: An effectiveness metric for ordinal classification: Formal properties and experimental results. arXiv preprint arXiv:2006.01245 (2020)
2. Baeza-Yates, R.: Biases on social media data: (keynote extended abstract). In: Companion Proceedings of the Web Conference 2020. p. 782–783. WWW '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3366424.3383564
3. Blaya, C., Audrin, C.: Toward an understanding of the characteristics of secondary school cyberhate perpetrators. In: Frontiers in Education. vol. 4, p. 46. Frontiers (2019)
4. Canete, J., Chaperon, G., Fuentes, R., Pérez, J.: Spanish pre-trained bert model and evaluation data. PML4DC at ICLR **2020** (2020)
5. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 11 (2017)
6. Devlin, J.: Multilingual bert readme document. `https://github.com/google-research/bert/blob/a9ba4b8d7704c1ae18d1b28c56c0430d41407eb1/multilingual.md` (2018)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Gheisari, M., Wang, G., Bhuiyan, M.Z.A.: A survey on deep learning in big data. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). vol. 2, pp. 173–180 (2017). https://doi.org/10.1109/CSE-EUC.2017.215
9. Jebara, T.: Machine learning: discriminative and generative, vol. 755. Springer Science & Business Media (2012)
10. Kim, J.W., Chen, G.M.: Exploring the influence of comment tone and content in response to misinformation in social media news. Journalism Practice **15**(4), 456–470 (2021). https://doi.org/10.1080/17512786.2020.1739550
11. Korencic, D., Baris, I., Fernandez, E., Leuschel, K., Sánchez Salido, E.: To block or not to block: Experiments with machine learning for news comment moderation. In: Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation. pp. 127–133. Association for Computational Linguistics, Online (Apr 2021), `https://www.aclweb.org/anthology/2021.hackashop-1.18`
12. Nelson, W.B.: Accelerated testing: statistical models, test plans, and data analysis, vol. 344. John Wiley & Sons (2009)

13. Pimpalkar, A.P., Raj, R.J.R.: Influence of pre-processing strategies on the performance of ml classifiers exploiting tf-idf and bow features. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal **9**(2), 49–68 (2020)
14. Plaza-Del-Arco, F.M., Molina-González, M.D., Ureña López, L.A., Martín-Valdivia, M.T.: Detecting misogyny and xenophobia in spanish tweets using language technologies. ACM Trans. Internet Technol. **20**(2) (Mar 2020). https://doi.org/10.1145/3369869
15. Risch, J., Krestel, R.: Delete or not delete? semi-automatic comment moderation for the newsroom. In: Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018). pp. 166–176 (2018)
16. Stroud, N.J., Van Duyn, E., Peacock, C.: News commenters and news comment readers. Engaging News Project pp. 1–21 (2016)
17. Taulé, M., Ariza, A., Nofre, M., Amigó, E., Rosso, P.: Overview of the detoxis task at iberlef-2021: Detection of toxicity in comments in spanish. Procesamiento del Lenguaje Natural **67** (2021)
18. Winter, S., Brückner, C., Krämer, N.C.: They came, they liked, they commented: Social influence on facebook news channels. Cyberpsychology, Behavior, and Social Networking **18**(8), 431–436 (2015)
19. Xenophobia: Retrieved from https://en.oxforddictionaries.com/definition/money. Oxford Online Dictionary (2021)
20. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics **1**(1-4), 43–52 (2010)