Lak_NLP at IberLEF-2022 DETESTS task: Automatic Classification of Stereotypes in Text

Fatima Laknani[1], Mercedes García Martinez[2]

[1]Escuela Técnica Superior de Ingeniería Informática(Universitat Politècnica de València), 46022 Valencia, Camí de Vera
[2]Pangeanic, 46015 Valencia, Avinguda de les Corts Valencianes, 26, Bloque 5

Abstract

This article briefly explains how the Lak_NLP team approached the problem of detecting racial stereotypes. This project involves two tasks. The first is to classify a text according to whether or not it presents racial stereotypes, the second subtask is to assign to those sentences ten categories of racial stereotypes which are: 1) "victims of xenophobia", 2) "suffering victims", 3) "economic resources", 4) a problem of "immigration control", 5) people with "cultural and religious differences", 6) people who benefit from our social policy, 7) a problem for "public health", 8) a threat to "security", 9) "dehumanization" and 10) "other" types of stereotypes.

By developing and studying various classifiers based on the Transformers architecture, namely BERT and BETO, we will address the task of detecting racial stereotypes.

Keywords

BERT, BETO, Racial Features, Transformers

1. Introduction

The use of the different tools provided by the Web has meant that millions of users can stay connected and thus, generate information continuously. These tools, such as messaging platforms, social networks, forums and others, are the main means by which Internet users sentence on what is of interest to them, even if this is offensive to other people. We speak about racial stereotypes when a sentence or sentence addresses an opinion with a certain prejudice about a group of people. Often these sentences are toxic and are written with bad intentions; automatically detecting these bad sentences is of great help to be able to control the content that is disseminated on the Web. Detecting the presence of stereotypes in online sentences can be a complicated task, since users can express themselves through sarcasm, insult, mockery, etc.

In this project, we present the systems that have been developed to cope with the tasks of the DETESTS (Detection and Classification of Racial Stereotypes in Spanish)[1] contest. The main objective of this task is the research and development of models to classify messages according to their presence or absence of racial stereotypes.

The DETESTS task is divided into two classification subtasks:

- Subtask 1: Detecting racial stereotypes, a binary classification problem where a particular sentence can be classified as either exhibiting racial stereotypes or not exhibiting racial

IberLEF 2022, September 2022, A Coruña, Spain.
falak@inf.upv.es (F. Laknani); m.garcia@pangeanic.com (M. G. Martinez)

stereotypes.

- Subtask 2: Assign sentences labeled as stereotypical to ten categories: 1) "victims of xenophobia", 2) "suffering victims", 3) "economic resources", 4) "immigration control" problem, 5) people with "cultural and religious differences", 6) people taking "benefits" from our social policy, 7) a problem for "public health", 8) a threat to "security", 9) "dehumanization" and 10) "other" types of stereotypes. Therefore, each sentence should be labeled as to whether or not it presents these categories.

2. Corpora

The dataset has been provided by the organizers of the DETESTS task at IberLEF 2022. It comprises sentences that belong to comments, the organizers segment the comments into sentences and therefore the detection of stereotypes is at the sentence level.

It is composed by a part of sentences belonging to NewsCom-TOX (Taulé et al., 2021) and another from the StereoCom corpus, sentences published in response to different Spanish online news (ABC, elDiario.es, El Mundo, NIUS, etc.) and forums (such as Menéame). A total of 5629 sentences are available, from which 3,306 sentences correspond to NewsCom-TOX and 2,323 sentences to StereoCom, 3817 of them correspond to the training set and the remaining sentences are the ones for test set that their predictions will be sent to the contest.

In the training set we have these variables:

- sentence_id: Refers to the id of each sentence.
- sentence_pos: Position of each sentence within a sentence.
- reply_to: Identifier of the sentence that starts a thread.
- sentence: sentence.
- racial_target: Presence or absence of discrimination that distinguishes between different groups, whether based on origin, race, ethnicity or religion.
- other_target: sentences that distinguish between any other minority or oppressed group.
- implicit: sentences that are expressed implicitly or directly.
- stereotype: Presence or absence of racial stereotypes.
- xenophobia: Presence or absence of xenophobia.
- suffering: sentences in which there is presence or absence that immigrants are suffering victims.
- economic: Racial sentences indicating whether they refer to immigrants seen as an economic resource or not.
- migration: Racial sentences indicating whether the problem is due to immigration control or not.
- culture: Racial sentences with presence or absence of cultural and religious differences .
- benefits: Racial sentences that do or do not relate the immigrant as a beneficiary of social policy.
- health: Racial sentences that do or do not associate immigrants with a public health problem

- security: Racial sentences in which the immigrant is or is not associated with a security threat.
- dehumanisation: Presence or absence of dehumanization.
- others: Presence or absence of other types of stereotypes.

The test set has only the variables:

- sentence_id
- sentence_pos
- sentence
- reply_to

3. System

In this section of the paper, we describe the systems developed for the detection of racial stereotypes. We propose a model based on Transformers [2], which can be understood as a model with encoder-decoder structure, employing attention mechanisms.

The diagram in the figure 1 refers to the architecture of the Transformer model, where the encoder can be differentiated on the left side and the decoder on the right side.

One of the main innovations of the transformer architecture was the positional encoding.

The sentence to be processed is passed as numerical vectors, in the case of Transformer the sentence is passed completely, so in this case the order of the words is given by a process of encoding each word thus generating a positional token and thus to inform the transformer model of what is the order of the different tokens of the different words that make up our sentence, this leads to the fact that in the transformer architecture the input sequence can be passed in parallel.

Therefore, given a sequence, the first step is to convert the string into a numerical representation using embedding, then a position encoding is added and the resulting vectors pass to the encoding stage, where the most relevant information is extracted from the sequence.

The transformer structure consists of a set of six encoders, all of them with the same distribution. Each of these encoders has a multi-head attention mechanism, which expresses numerically the possible relationships between the different words of the sequence, thus indicating which elements of the sentence to pay more attention to. On the other hand, we also distinguish a simple feed-forward network, which is applied to each of the attention vectors in order to transform these vectors into a format accepted by the next encoder or decoder block. On each of these two layers, a residual connection is used followed by a specific normalization of higher stability (Add & Norm). Thus, each block takes the input tokens, processes them in parallel and delivers as output a representation with attentional information. Once the encoding stage is finished, this output is connected to the decoder. This stage differs from the previous one in that the different blocks have an attentional masking block and an additional residual block. Like the encoder layers, residual connections are added on top of the layers along with a normalization layer. The first self-attention multi-head layer uses masked multi-head attention.

Currently for most natural language processing tasks, models based on the Transformer architecture are on trend.

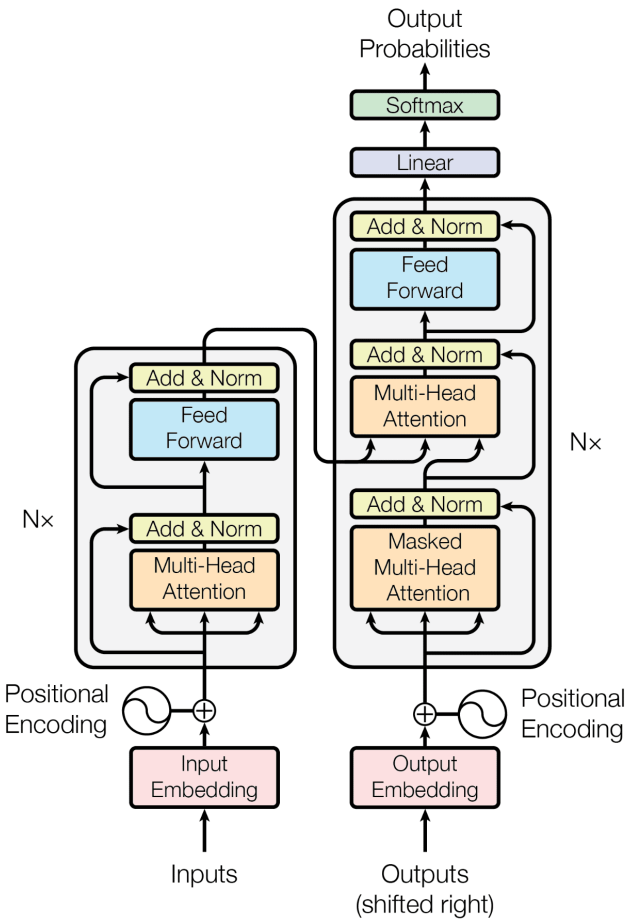

Figure 1: Transformer architecture[2].

The introduction of Bidirectional Encoder Representations from Transformers (BERT) [3], a model developed by Google in 2018, had a major impact on the world of natural language processing. The idea of the developers was to create a model that manages to learn language in general thus giving rise to a pre-trained model based on the Transformer architecture.

BERT is the result of taking the transformer network and keeping the encoding block. BERT base is consisting of 12 encoder layers, 12 attention heads, and 110M parameters.

This model employs a masking method so that certain tokens are randomly masked and then predicted based on context alone. The base model can be pre-trained with large datasets and then specialized layers can be added, after which the model can be retrained with a smaller dataset to perform more specific tasks.

For the network to be able to analyze the sequences it is necessary to adjust the text, for this purpose classification and separation tokens are used.

Given a sequence, the classification token [CLS] is used at the beginning of the sequence,

and the separation token, [SEP], is used to separate one string from another.

Subsequently for each token an embedding is obtained, which passes the given token to a vector and which will present a positional encoding.

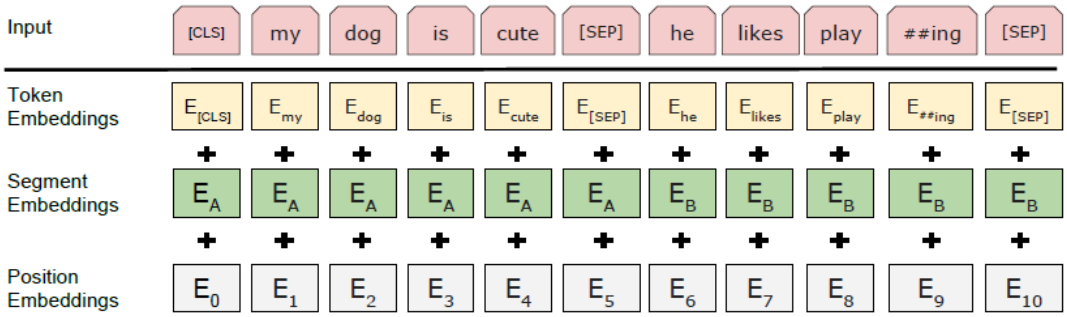

Figure 2: Generation of BERT input[3]

If more than one string is to be passed to the model as input, as shown in figure 2, in addition to the embedding vector and the position vector, a vector is added to indicate to which segment the sentence belongs, since the first sentence, which is before the separation token, is encoded with a different embedding than the subsequent string, in short, the segment vector is used to distinguish the different strings.

Finally, these resulting vectors are added together generating a vector as an output, which is passed as input to the BERT model. BERT has been previously trained using two large datasets, Wikipedia and Google Books. This model has the advantage of being able to analyze the text in a bidirectional way, which means that when encoding a certain word it takes into account all its context from the beginning and the end of the sentence. BETO [4] has proven to be very successful in many natural language processing (NLP) tasks for Spanish.

BETO is a Spanish language model created in 2019 based on Transformers which is basically the Spanish version of BERT, it takes into consideration the accentuation and the Spanish letter "ñ".

This model allows carrying out natural language processing tasks in Spanish and can be used for several tasks, such as sentiment analysis[5], classification, etc.

Like BERT, this model consists of a first pre-training stage and then fine-tuning, which refers to a specialized training in a particular task to be carried out.

BETO-uncased has been used as a base model with a previous preprocessing. This preprocessing is responsible for removing punctuation marks and URLs from the different sentences.

4. Experiments

As mentioned above, in this contest we have used BETO-uncased as a base model with prior preprocessing.

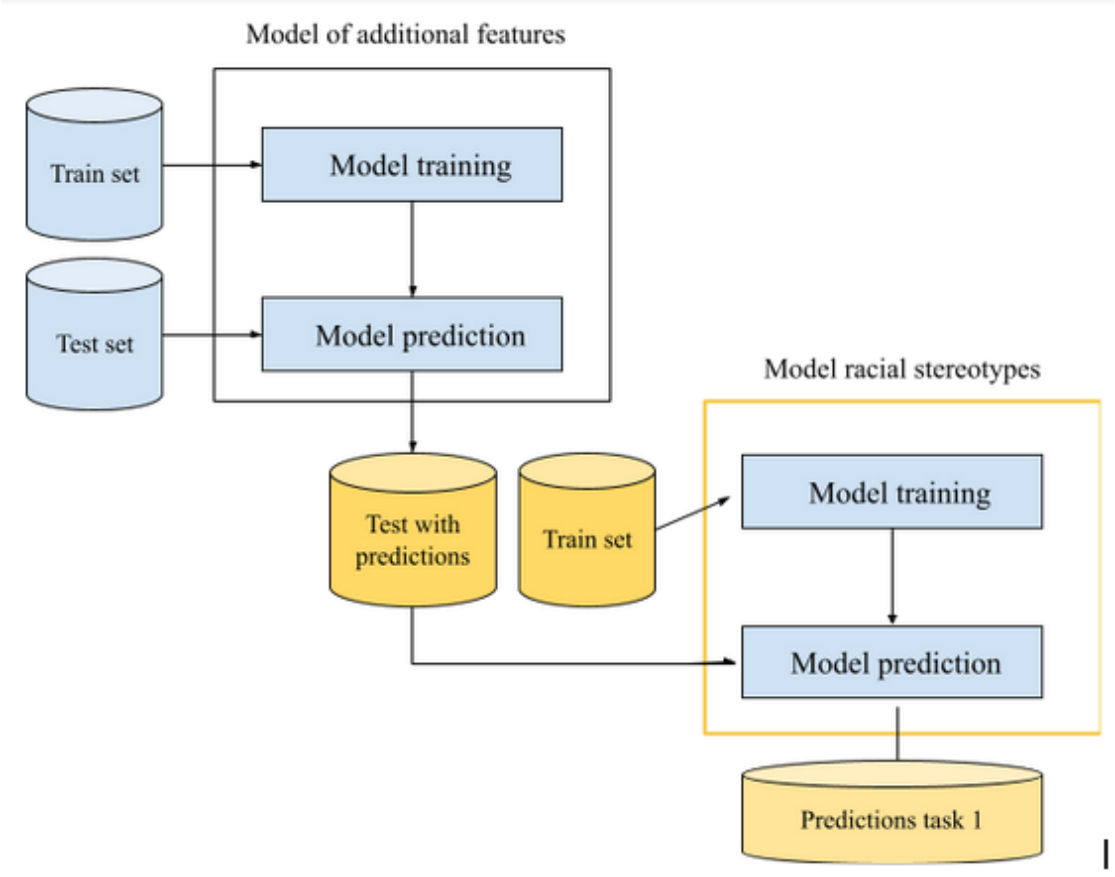

Figure 3: Process to be followed to obtain predictions for task 1.

Each of the models used has certain hyperparameters for training. The learning rate has been set to 6e-6, the batch size to 34 and the number of epochs to 4.

On the other hand, Adam has been used as optimizer.

On the other hand, using linguistic or racial features can be helpful to improve the base model, so it has been decided to create alternative models with the different features available in the training set in order to make the prediction on the test set and pass it to the model of interest.

Figure 3 shows the procedure performed to obtain the predictions for the first task. Firstly, we train a model to automatically detect the features that we would like to use. For example, if we want to predict the values of "racial_target", we will use the model trained to detect this feature, which it is not available in the test set. For the training of the model, the training dataset is divided into five random groups of equal size. Four groups are used to train the model and the remaining one to test it.

Figure 4, shows how the cross validation works [6], where as we have mentioned, our training data are divided into five parts, once we obtain the score for each block, at the end of

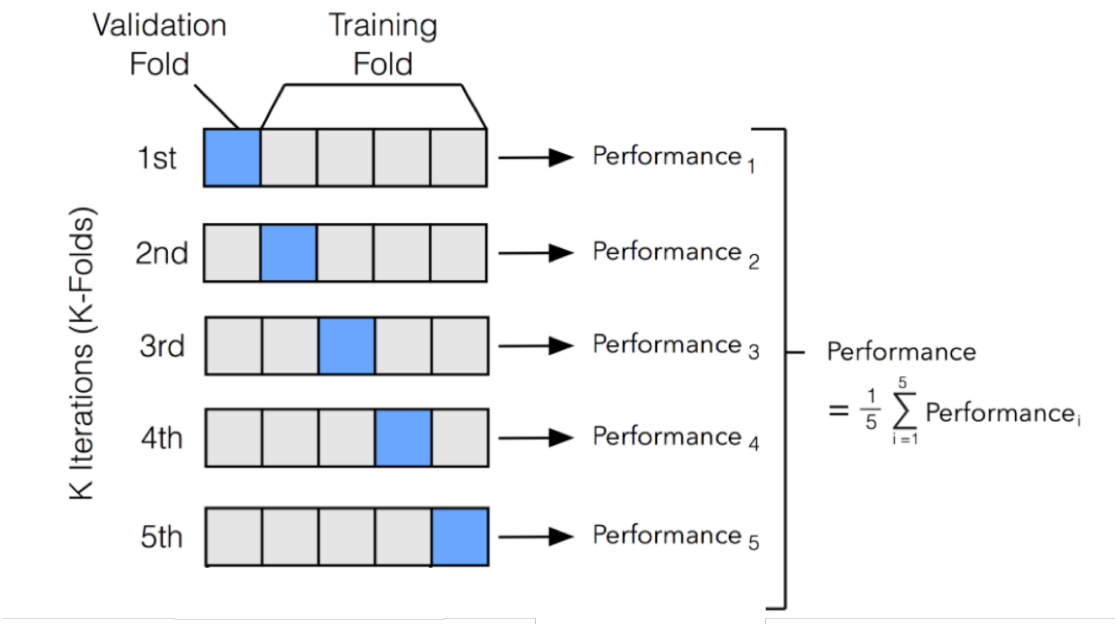

Figure 4: Cross validation process

the procedure an average is made and an overall score is obtained, this score is what will allow us to compare between different models and determine the quality of the predictions of each one.

Once we have chosen the best model to predict our variable "racial_target", we predict the values of this variable over the test set. This will provide us the additional variable that will be used for the predictions.

Finally, we will pass our training dataset to our model of interest of racial stereotypes, and to make the predictions we will pass the test set with the newly created variable that refers to the predictions of "racial_target".

4.1. Analysis of training results

In this section, we will analyze the results obtained for the DETESTS task. For the first subtask, we first perform a training of the base model, the performed process is the one explained in the previous section, where the training set is divided using the cross-validation procedure and a final score is obtained for that model.

Table 1 shows the results for the BETO-base model without additional features, which obtains an F1-score of 0.701 and accuracy of 0.818.

The next step is to find out if different linguistic features can improve this result by passing these features to the base model.

The table 2 shows the scores obtained using as input to the model features such as "implicit",

Table 1
Training results task 1 DETESTS

BETO-base-uncased	
F1-score	0.701
Accuracy	0.818

Table 2
Training results task 1 DETESTS with additional features

BETO-base-uncased with preprocessing		
	racial_target	implicit
F1-score	0.899	0.919
Accuracy	0.923	0.945

which refers to whether the stereotype is expressed implicitly.

The model that gives us the best result is the BETO-uncased model with preprocessing using the feature "implicit" where we go from an F1-score of 0.701 to 0.919.

Table 3
Training results task 1 DETESTS with several additional features

BETO-base-uncased with preprocessing	
	racial_target+implicit
F1-score	0.950
Accuracy	0.965

We can use more than one feature as input to the model, as shown in table 3, such as the combination of "implicit + racial_target" which provides an F1-score of 0.950.

These results obtained above have been performed on the training set, of which we already have the real corresponding labels. In order to obtain the predictions on the test set with these features that are not available, we have built new models that generate a prediction of these features. Thus, being able to subsequently pass the values of these variables to the model of interest.

Table 4
Model training results for "racial_target" and "implicit".

BETO-base-uncased with preprocessing		
	racial_target	implicit
F1-score	0.793	0.579
Accuracy	0.872	0.842

Table 4 shows the results of training a separate model for both the "racial_target" feature

and the "implicit" variable. We see that the model for predicting the "racial_target" label scores better, with an F1 score of 0.793 and an accuracy of 0.872, while the model for predicting whether a sentence is implicit or not generates an F1 score of 0.579.

Once these models have been trained, the test set is passed to them and the predictions of this corpus are obtained for each of the variables, in order to subsequently pass them to our model of interest of racial stereotypes, and to study whether the predictions of the base model are improved.

Table 5
Results of racial stereotyping model passing "racial_target" predictions.

BETO-base-uncased with preprocessing	
	racial_target
F1-score	0.759
Accuracy	0.841

The table 5 shows the results obtained for our racial stereotyping model by passing as input also the "racial_target" feature. We can see that we went from a score of 0.899 when the training was performed knowing the actual labels to a score of 0.759, which implies a significant decrease, but compared to the base model a certain improvement has been obtained. The procedure performed for the second task is similar to the first one, in this case this task consists of classifying those sentences labeled as presenting racial stereotypes in different categories, namely: xenophobia, suffering victims, economic resources, a problem of migratory control, people with cultural and religious differences, people who benefit from our social policy, a problem for public health, a threat to security, dehumanization and other types of stereotypes. The base model in this case remains BETO-uncased with a prior process of preprocessing the sentences. First, the base model has been trained without adding additional features.

Table 6
Results task 2 DETESTS using BETO-base-uncased with preprocessing

	xenophobia	suffering	economic	culture	benefits	security	dehumanisation	health	other
F1-score	0.152	0.142	0.286	0.454	0.575	0.537	0.121	0.163	0.211

We can see in table 6 that there are variables such as xenophobia or victims of suffering where the score is lower than in others such as culture, this may be due to the fact that there is an imbalance in the data and therefore there are not too many sentences labeled as xenophobia and victims of suffering, which leads to better predictions of some features than others.
After analyzing the results with the base model, it has been decided to use other features, which are "racial_target" and "implicit", to study if this result is improved.

Table 7 shows the results obtained when training two models, one of them with the racial_target feature as an additional input and the other with the implicit feature. As we can see, the results improve with respect to the base model, especially in the model with the racial_target feature,

Table 7
F1-score task 2 DETESTS using BETO-base-uncased with preprocessing with additional features

	xenophobia	suffering	economic	culture	benefits	security	dehumanisation	health	other
Model + racial_target	0.186	0.168	0.329	0.536	0.693	0.625	0.246	0.0.228	0.276
Model + implicit	0.138	0.218	0.276	0.398	0.693	0.436	0.233	0.212	0.229

where the score of all the features improves.

5. Contest results

The objective of the DETESTS task, as mentioned in previous sections, is to detect the presence of racial stereotypes in sentences.

Different models have been developed that allow the correct detection of these.

We can highlight that the use of additional features can improve the predictions of our base model, as it has occurred in the first task where the additional use of the feature "racial target" means an increase in the score of our model, thus generating better predictions regarding the detection of racial stereotypes.

Once the different models have been trained and those that provide the best results have been selected, the labels for our test set are predicted, and the test set is sent to the contest where it is evaluated by the contest organizers, since they have the real labels of the test set.

Table 8
Results of the competition task 1 DETESTS

Ranking	Team Name	F-score
1	I2C_III Run_1	0.7042
2	I2C_III Run_2	0.7038
...	...	...
10	Lak_NLP Run_4	0.6627

The table 8 shows the results of the competition for the first task. Out of a total of 141 models submitted, one of the models developed in this work has been ranked in tenth position. This model corresponds to BETO-uncased with preprocessing using the predictions of the "racial_target + implicit" features, and as we can see it has obtained a score of 0.6627. Therefore, we can conclude that it is an adequate model since it has been classified in a good position.

Table 9 shows the ranking obtained for the second task, out of a total of 18 models submitted, in our case an eighth position has been obtained with the BETO-uncased model with preprocessing using the predictions of the "racial target + implicit" features. It can therefore be concluded that the use of features together with a given model can considerably improve the results of the model and it has been possible to create models that provide good results, in particular for the first task.

Table 9
Results of the competition task 2 DETESTS

Ranking	Team Name	ICM	Hierarchical-F	Propensity-F
1	MALNIS Run_2	-0.2379	0.8813	0.8716
2	MALNIS Run_1	-0.2830	0.8807	0.8703
...	...	...	...	...
8	Lak_NLP Run_4	-0.4242	0.8606	0.8469

6. Conclusions and future work

The present work was aimed at developing models that allow classifying sentences according to the presence of racial stereotypes. This has been done by participating in the DESTEST task. The development of different models has been carried out and also the use of different additional features as input to these models.

It has been possible to conclude that the use of these additional features can considerably improve the predictions of the models, thus obtaining better results and therefore better classifying the different sentences published by their presence of stereotypes.

The first future work could be to increase the database, achieving a more balanced dataset, since there is a significant imbalance of sentences belonging to the different racial characteristics.

On the other hand, another future line would be to create models using other features different from those already used in this project and achieve better results. Finally, the models can be also used to other types of texts such as news

References

[1] A. Ariza, W. S. Schmeisser-Nieto, M. Nofre, M. Taulé, E. Amigó, B. Chulvi, P. Rosso, Overview of the DETESTS Task at IberLEF-2022: DETEction and classification of racial STereotypes in Spanish, Procesamiento del Lenguaje Natural 69 (2022).

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[4] J. Canete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, Pml4dc at iclr 2020 (2020) 2020.

[5] A. de Arriba Serra, M. Oriol Hilari, J. Franch Gutiérrez, Applying sentiment analysis on Spanish tweets using BETO, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021): co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing: Málaga, Spain, September, 2021, CEUR-WS. org, 2021, pp. 1–8.

[6] P. Refaeilzadeh, L. Tang, H. Liu, Cross-validation., Encyclopedia of database systems 5 (2009) 532–538.