Detection, Classification and Quantification of HUrtful HUmor (HUHU) on Twitter Using Classical Models, Ensemble Models, and Transformers

Hugo Albert Bonet[1,†], Aina Magraner Rincón[1,†] and Alba Martínez López[1,†]

[1]*Universitat Politècnica de València, Spain.*

Abstract

Identifying hurtful comments in social media posts has high relevance in order to improve the common welfare. Nevertheless, sometimes hurtful messages are masked by humor, which may increase the difficulty of detecting and identifying this type of content. When making use of HUrtful HUmor (HUHU), the author feels free to spread prejudices without limits [1]. Because of the aforementioned reasons, the objective of this work is to propose a methodology to fasten the detection of harmful texts posted on social media by exploring the different machine and deep learning models for three different tasks in Spanish: HUrtful HUmor detection, target group identification, and prediction of the degree of prejudice. Different text representation together with classical models, ensemble models, and the Spanish transformers BETO [2] [3], and RoBERTa [4] [5] were evaluated on the dataset provided by the competition called "HUrtful HUmour (HUHU) Detection of humour spreading prejudice in Twitter" [6]. It was observed that: (i) transformers architectures highly outperform classical and ensemble models when it comes to detecting degree of prejudice and the target group, but have a serious problem with overfitting for the last one; (ii) oversampling was a key solution when dealing with imbalanced classes in a small data set; (iii) including extra features regarding the written style or the underlying intentions of the writer are of great utility when it comes to natural language tasks.

Keywords

Hurtful humor detection, Hurtful humor target classification, Degree of prejudice regression, Deep learning, Machine learning, Natural language processing (NLP)

1. Introduction

Natural language processing (NLP) [7] is currently one of the biggest and most promising fields regarding machine learning and deep learning. The complexity of language makes NLP a complicated and intriguing task. Some of the challenges faced when dealing with NLP tasks are that language has strict rules when it comes to structure, it has multiple significations depending on the context, or that minimal variations in some words completely change the meaning or comprehension of the message. In a nutshell, NLP includes a group of machine and

IberLEF 2023, September 2023, Jaén, Spain

[†]These authors contributed equally.

hugoalberthlu@gmail.com (H. A. Bonet); magraneraina@gmail.com (A. M. Rincón); albamartinez584@gmail.com (A. M. López)

https://www.linkedin.com/in/hugoalbert/ (H. A. Bonet); https://www.linkedin.com/in/ainamagranerrincon/ (A. M. Rincón); https://www.linkedin.com/in/albamartinezlopez/ (A. M. López)

deep learning techniques which deal with text input to perform different tasks like classification or regression.

In this work we mainly focus on applying NLP techniques together with the state-of-the-art transformer models for three tasks: (i) hurtful humor detection, which consists of distinguishing common harmful tweets from those which are masked with humor; (ii) target identification, where we classified if the tweet intends to spread sexism, prejudices against the LGBTIQ community, racism, or fatphobia; and (iii) degree of prejudice prediction, which consists of estimating how hurtful the tweets are in a scale from one to five.

It is necessary to highlight that several aspects of the process are going to be taken into account: (i) different representations of the text are going to be compared, such as bag of words, cleaning the text, or word embeddings; (ii) classical models –like SVM or Logistic Regression– are going to be compared with ensemble models –such as RandomForests or Stacking– and state-of-the-art transformers for Spanish –BETO or RoBERTa–; and (iii) the change of performance of the models when including extra features –like irony or emotions– is going to be considered.

The main research question is: Which combination of text treatment, extra features, and machine or deep learning model is better suited for each task? The above question is going to be answered based on the results of the dataset given by the "HUrtful HUmour (HUHU) Detection of humour spreading prejudice in Twitter" competition.

2. Methodology

The methodology followed in this work did not contain a data collection phase as the dataset was provided by the competition organizers. It was composed of six main steps:

2.1. Data processing

The data processing step corresponds to treat the text in order to feed the models properly. This section explains all the different techniques employed in this process, although later on we explain which of them where applied to each model.

The first preprocessing employed consisted of a deep cleaning of the text. First, we got rid of all URLs, HTML tags, and punctuation symbols. After, all words were turned to lowercase, stop words were removed, the text was lemmatized, and words were stemmed using Porter Stemmer. Said type of cleaning significantly reduced the complexity of the text, getting rid of noise and other aspects that may or may not be important for the tasks.

With the deeply cleaned text, we created a representation of the tweets making use of Bag of Words (BoW) in both ways, just counting the appearences of each word and by applying the weighting scheme TF-IDF. However, we realized that, TF-IDF was not providing us with any significant advantage. The deeply cleaned text was also used to create a representation based on word embeddings, by using Fast Text [8] [9] [10] and also All-Mini-LM [11], a multilingual transformer which is known for being fast.

The second data processing method applied was really simple. The text was not treated and just used to extract the embeddings through All-Mini-LM.

The third and last method consisted of tokenizing the text with the corresponding Tokenizers for RoBERTa and BETO. The tokenized text was the input of the transformers when fine-tuning

[12] them. However, we also extracted the embeddings of RoBERTa to change the top model –the classifier– for other classical ones.

The last applied processing consisted of making use of pre-trained transformers [13] to extract extra features about the toxicity, hate speech (hate), context hate speech (context), irony, emotions, and sentiments, explained in detail in Appendix A.

2.2. Exploratory analysis

An exploratory analysis of both the tweets and the ground truths was conducted to picture a clear image of how the next steps needed to be developed.

When it comes to the first subtask, the detection of hurtful humor, the amount of not humorous tweets was more than twice the number of humorous ones, as seen in Appendix B. Besides, the analysis aimed also to show whether the written style should be used to tackle the task. Some aspects that were taken into account were the number of dashes (-) written, as a common structure of a joke includes a dialogue, the number of exclamation marks (!), or the number of uppercase letters, the last two because they represent emphasis. We discovered that humorous tweets had six times more dashes per tweet than not humorous ones. When counting the number of exclamation marks, the plots did not show a difference in quantity for humorous and not humorous tweets. However, when considering the number of exclamation marks per tweet, the plots showed that they are three times more frequent in tweets using humor. Last but not least, non-humorous tweets apparently used more uppercase than humorous ones, but again by normalizing the values we realized that there was no significant difference between both classes.

Regarding the second subtask, where the target of the comment had to be identified among the four groups mentioned in Section 1, we conducted a similar analysis. The vast majority of the tweets where sexist, while there was only a minority of tweets targeting fat people. The number of LGBTIQ and racist tweets where balanced. Whereas the number of uppercases per tweet was almost the same for every category, the number of dashes per tweet and the number of exclamation marks per tweet where clearly superior in tweets spreading fatphobia. Said discoveries regarding the punctuation of the text, led us to the idea of considering representations of the text that took that into account, like the embeddings without cleaning the texts.

For both classification tasks, an XGBoost classifier was applied using the BoW representation, as a first step to extract the subset of most important words in order to classify the samples. In both subtasks, the subsets where almost the same. Those most important words (referred as ”most” in the report) were also used as extra features.

The last subtask, the prediction of the degree of prejudice of the tweet, needed a different approach of the exploratory analysis, as the variable was numerical. The distribution of values followed a Gaussian Distribution, although it presented some negative asymmetry, being the mean approximately 3.

2.3. Model implementation and hyper-parameter analysis

For all three subtasks, several models were implemented, as well as a hyper-parameter analysis.

In the first subtask, which consisted of a binary classification, we implemented a series of models, most of which were based on a single-language model for Spanish: RoBERTa-base transformer. This was fine-tuned using its corresponding tokenized text. To this embeddings we added different task related features in order to observe if these were of any help in the given tasks. Also, a hyper-parameters analysis was conducted for all RoBERTa-base models, in which we studied the charts reflecting the loss for each epoch relating the training set and validation one. We finally considered the following as the best parameters: different learning rates (0.00001 and 0.000001 depending on the subtask) , 10 epochs as well as 16 or 32 batches. To all of these models a batch normalization was performed as well as a drop out of 0.3 after the concatenation of the embeddings and the extra features. Regarding the classical models implemented, a hyper-parameter analysis was conducted using a GridSearch strategy with 5-fold cross validation. Then, an ensemble method was used in which the estimators are the models with the best parameters obtained. The metrics used to evaluate them, in both Subtasks 1 and 2a, were the same as the ones used on the HUHU shared task: F1-Macro. The main models implemented were:

- RoBERTa-base transformer plus toxicity features.
- RoBERTa-base transformer with an addition of several task related features: irony, toxicity, hate, emotions, and sentiment.
- RoBERTa-base transformer plus the most important words extracted from the BoW analysis mentioned above and toxicity features.
- RandomForest with a number of estimators of 200 and a maximum depth of 30. Plus the most important words extracted from the BoW analysis mentioned above and toxicity features.
- Bagging Classifier with a Support Vector Classifier as the base estimator with the following parameters: C = 10, gamma = 0.1, kernel = rbf, and 50 estimators. This was trained with an embedding matrix obtained from the uncleaned data set through All-Mini-LM.

For the multi-label classification subtask we changed the number of batches for the RoBERTa-base transformer models to 32. The main trained models were:

- RoBERTa-base transformer plus toxicity features.
- RoBERTa-base transformer plus the most important words extracted from the BoW analysis mentioned above and context of hate speech features.
- RoBERTa-base transformer with an addition of several task related features: toxicity, hate, emotions and Sentiment.
- RoBERTa-base transformer with an addition of several task related features: toxicity, hate, irony, emotions, context and sentiments.
- Voting Classifier between two models: Random Forest with 50 estimators and a MLP Classifier using the identity activation, alpha: 0.001, 500 as the maximum iterations, two hidden layers with sizes 128 and 32, a constant learning rate, and a lbfgs solver. This was put through a Multi-Output Classifier.

Finally, for the regression subtask these were the most important trained models:

- RoBERTa-base transformer plus sentiments, emotions, hate, irony and toxicity features, as well as the most important words. The used hyper-parameters were: 10 epochs, 32 batches and a learning rate of 0.000001. This transformer is the only one without batch normalization.
- ExtraTreesRegressor with a number of estimators of 250 and a maximum depth of 30, to which we added sentiments, emotions, hate, irony and toxicity features.

For this last task, the metric used to evaluate the models was the RMSE.

2.4. Models comparison

The next step is to compare the results of the different models in order to rank them according to their performance. Due to the reduced amount of samples in the training set, a three-fold cross validation strategy was applied to calculate the F1-Macro score for each machine learning model and ensemble model. In the case of the transformers, which need high amounts of data to be trained, the strategy diverged to a mix of bagging and cross validation. We repeated three times the measurement of the model, randomly splitting each time into train, validation, and test, and randomly reordering the samples. The measure obtained was the average of the F1-Macro score obtained in the test set for the three runs.

The following tables show the results of the most relevant models according to their F1-Macro score from the whole amount of models tested, as mentioned in the previous section. Table 1 illustrates the results for the models for Subtask 1, Table 2 illustrates the results for the models for Subtask 2a, and Table 3 illustrates the results for the models for Subtask 2b.

Table 1
Model comparison on Subtask 1

	Used hyperparameters	F1-Macro
RoBERTa + toxicity	10 epochs, 16 batch	0.808
RoBERTa + toxicity + context + most	10 epochs, 16 batch	0.804
RoBERTa + toxicity + most	10 epochs, 16 batch	0.75
RF + toxicity + most	n_estim: 200, max_depth: 30	0.710
Bagging + SVC	C:10, gamma: 0.1, kernel: rbf, n_estim:5	0.732

Table 2
Model comparison on Subtask 2a

	Used hyperparameters	F1-Macro
RoBERTa + toxicity	10 epochs, 32 batch	0.824
RoBERTa + context + most	10 epochs, 32 batch	0.807
RoBERTa + toxicity + hate + emotions + sentiment	10 epochs, 32 batch	0.853
RoBERTa + toxicity + hate + emotions + context + sentiment	10 epochs, 32 batch	0.895
Voting + RF + MLP	best hyperparam for each	0.699

Table 3
Model comparison on Subtask 2b

	Used hyperparameters	F1-Macro
RoBERTa + sentiments + emotions + hate + irony + toxicity + most	10 epochs, 32 batc	0.821
ExtraTressRegressor + sentiments + emotions + hate + irony + toxicity	n_estim: 250, max_depth: 30	0.690

To see the comparison of a higher amount of models, see Appendix C.

2.5. Final models selection and submission

Once we have studied the performance of all the possibilities explained in Section 2.5, the final models selected to submit to the competition are shown in Table 4, so they could be tested on unknown test set to prove their capability of generalization. The criterion was choosing the models which presented a higher F1-Macro score in the case of the first two subtasks, and a lower RMSE in the last one.

Table 4
Final models

			Metrics		
Subtask	Submission	Model	F1-Macro	Accuracy	RMSE
1	1st	RoBERTa + toxicity	0.808	0.81	
	2nd	RoBERTa + toxicity + context + most	0.804	0.925	
2a	1st	RoBERTa + context + most	0.876	0.951	
	2nd	RoBERTa + toxicity	0.867	0.945	
2b	1st	RoBERTa + toxicity + emotions + hate + sentiments + irony			0.821
	2nd	ExtraTreesRegressor			0.69

2.6. Post-competition improvements

We are aware that our findings could have been enriched by the inclusion of other approaches or techniques. Once we had submitted our models, having received the labeled test set, we applied some of those ideas which we had came up with but that we could not include in the submissions, and we plan to continue thoroughly examining other methodologies, as we explain later in Section 5.

3. Results

In this section, we present the final results obtained in the HUHU competition and compare them with the results obtained during our experiments, shown in Table 5.

Table 5
Comparison of performance on training (3-fold cross validation) and test sets

Subtask	Submission	Metric	Test	Training	Position
1	1st	F1-Macro	0.399	0.808	51
	2nd	F1-Macro	0.339	0.804	53
2a	1st	F1-Macro	0.475	0.876	36
	2nd	F1-Macro	0.362	0.824	53
2b	1st	RMSE	0.887	0.821	4
	2nd	RMSE	0.985	0.69	32

As we can see, the models submitted for the classification tasks show a significant difference in performance compared to our previous results.

The results show that the solutions we had implemented to address a possible overfitting problem have not worked as we would have wanted. We used batch normalization for the Subtasks 1 and 2a plus a dropout technique for all subtasks, even though this last approach has successfully worked for Subtask 2b, it seems that the batch normalization has not influenced positively to avoid overfitting. Moreover, it has fuelled this problem.

Regarding the last subtask, where a regression model was presented, we have achieved outstanding results in the first submission, using a RoBERTa model and adding as extra features: toxicity, emotions, hate, sentiments, and irony. We have obtained a slight difference between the test RMSE value and the training one, accomplishing a fourth position in the HUHU ranking results. The ExtraTreesClassifier differed more when it comes to the performance in the submission, although its performance was not too far from the first submission. In spite of such small difference, said model is 28 positions below the transformer, representing the competitiveness of the models sent to the competition.

4. Conclusions

In this research, we have explored different methodologies to detect and identify harmful comments in social media posts, particularly on platforms like Twitter.

The findings of the study have yielded valuable insights and practical implications for selecting the optimal models. Additionally, the research outcomes have enhanced our comprehension of the topic, empowering us to make informed decisions, which have subsequently guided our development of additional models. These will be explained in Section 5.

Regarding the aspects we discovered that need to be taken into account in tasks related to HUHU, the following list remarks the most important ones:

- Implementing measures to mitigate the issue of imbalanced classes play an important role.
- State-of-the-art transformers usually outperform classical and ensembled models, although addressing the problem of overfitting is a serious issue when dealing with them. We observed a significant disparity between the results we obtained during our model evaluation and the actual results provided by the organizers, which suggests that more robust measures should be applied. As transformers are a complex type of deep learning models, a deeper investigation must be carried on before starting using them.
- Ensuring thorough data preparation and a comprehensive understanding of the variables that may be related to the study and the semantic meaning of the text, the optimal performance of the models.

5. Future work

Continuous improvement is a crucial aspect when working with natural language and artificial intelligence. Therefore, these working notes do not have an end and will be always open to new improvements. This section has the aim to reflect the additional aspects investigated after the submission and their influence in the result obtained once we were given the labeled test set. Moreover, this section also aims to state the ideas we came up with for future ideas not yet implemented.

When it comes to aspects already added to the project, we created extra features with the frequency of dashes and exclamation marks for each tweet, instead of just including them in the tokenized input for the models. This change leaded to a better performance of almost all the models. For example, the Random Forest went from an F1-Macro score of 0.6 to 0.69 by including those variables.

However, the greatest improvement was caused by tackling the problem of unbalanced classes. The first approach consisted of assigning weights to the categories while training the transformers, which did not solve the problem. The second try was based on oversampling [14] –as undersamplig, which was the idea proposed by other teams, did not seem appropriate with such an small database–. We proposed two options:

- Duplicating the rows corresponding with the minority class: This method was used both in Subtasks 1 and 2a, improving the performance from 0.399 and 0.475 to 0.413 and 0.492 respectively. By discarding the batch normalization in Subtask 2a, we reached 0.725. This method was also applied to a Random Forest Classifier for Subtask 1, obtaining an F1-Macro score of 0.826 in the test set, superior to the 0.820 obtained by the winner of the HUHU competition.
- Using SVMSMOTE method for oversampling [15]: The second method was just applied to the first subtask. It provides the new set with more variability, which reflects in the models as an improve in performance. The Random Forests Classifier obtained in this case a value of 0.831 in the test set.

For the future, we propose different changes in order to seek for the best model. The first idea is to change the way of training the transformers by adding automatic functions included

in Python libraries to ensure the correct learning of the neural network. By adding this, we can focus our efforts on changing the architecture in order to avoid overfitting, as well as improving the model by adding decay to the learning rate or more complex ways to combine the embeddings of the different words such as convolutional layers.

The second proposal is to adapt the SVMSMOTE oversampling strategy to the multi-label task, in order to avoid the problem of imbalanced classes. For this task, we also propose to solve the problem with chain classifiers, which would allow the models to extract relations between the predicted categories.

A. Extra features

The aim of this appendix is to offer an explanation of the extra features obtained with pre-trained models:

- Toxicity: Classifies the sentence according to different levels and ways of expressing toxic statements. The features returned are the following:
  - Toxicity
  - Severe toxicity
  - Obscene
  - Identity attack
  - Insult
  - Threat
  - Sexual explicit
- Hate Speech: Describes how hateful a sentence is, according to the following features:
  - Hateful
  - Targeted
  - Aggressive
- Context Hate Speech: Focuses on the target of the hateful statement. As our data set only showed hateful tweets, it seemed perfect for these extra features. The features where:
  - CALLS
  - WOMEN
  - LGBTI
  - RACISM
  - CLASS
  - POLITICS
  - DISABLED
  - APPEARANCE
  - CRIMINAL
- Irony: Tells whether a tweet is expressing irony or not.
- Emotions: States if the sentence is expressing different emotions, which are the following:

  - Joy
  - Sadness
  - Anger
  - Surprise
  - Disgust
  - Fear
  - Others
- Sentiments: Shows the degree of positivity, negativity, or neutrality of the text.

B. Exploratory analysis. Graphics

This appendix aims to show the plots created during the exploratory analysis.

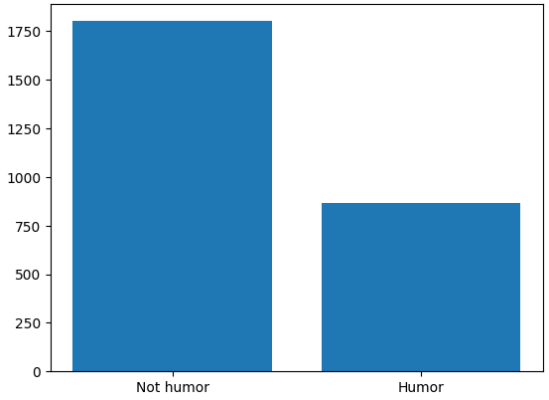

Figure 1: Distribution of labels for Subtask 1.

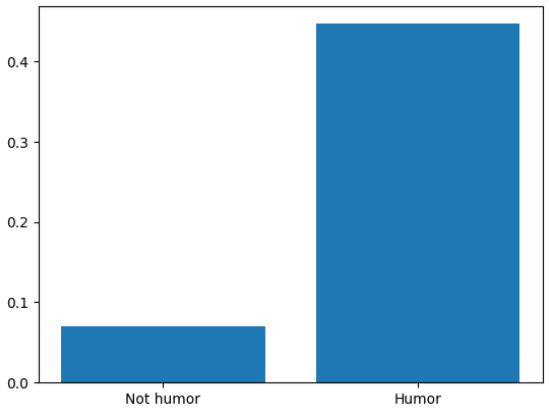

Figure 2: Distribution of dashes per tweet depending on the label for Subtask 1.

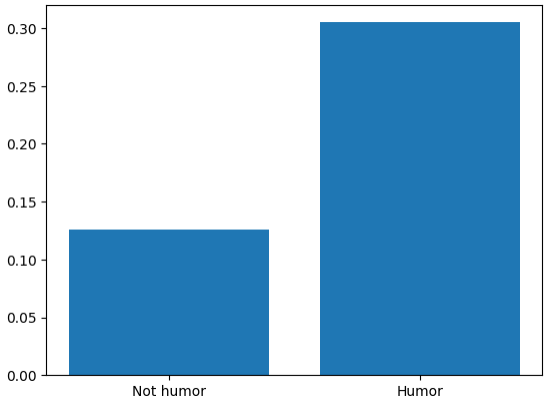

Figure 3: Distribution of exclamation marks per tweet depending on the label for Subtask 1.

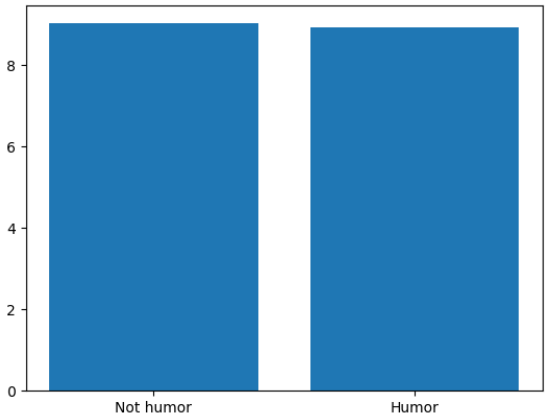

Figure 4: Distribution of upper letters per tweet depending on the label for Subtask 1.

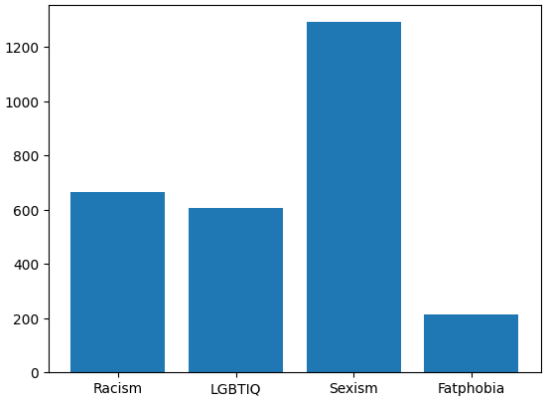

Figure 5: Distribution of labels for Subtask 2a.

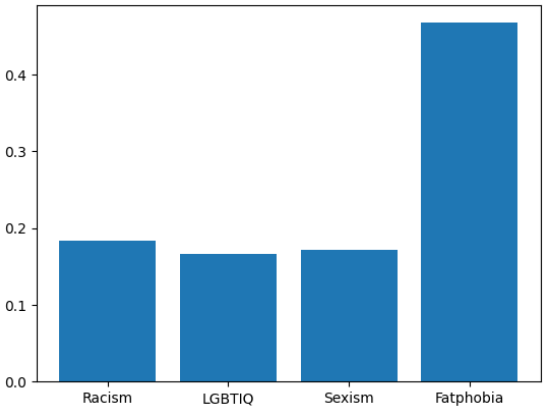

Figure 6: Distribution of dashes per tweet depending on the label for Subtask 2a.

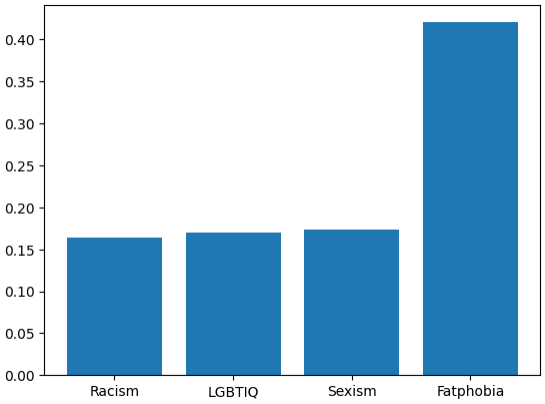

Figure 7: Distribution of exclamation marks per tweet depending on the label for Subtask 2a.

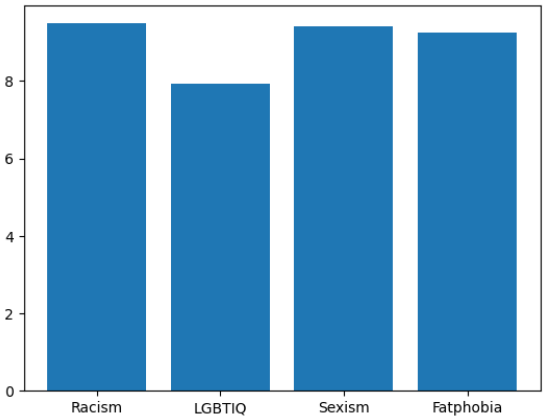

Figure 8: Distribution of upper letters per tweet depending on the label for Subtask 2a.

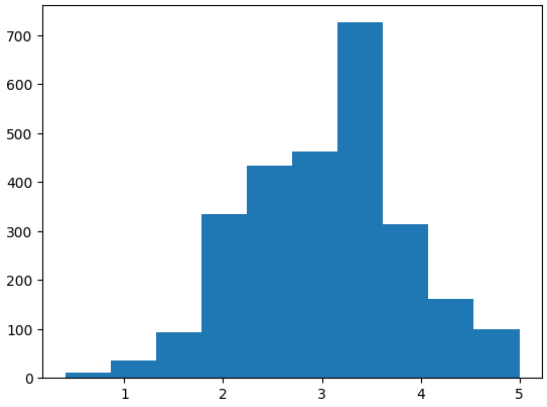

Figure 9: Distribution for Subtask 2b.

C. Extra models

Table 6
Extended model comparison on Subtask 1. Transformers

Model	Features	Used hyperparameters	F1-Macro
RoBERTa	toxicity	10 epochs, 16 batch	**0.808**
	toxicity + context + most	10 epochs, 16 batch	**0.804**
	irony	10 epochs, 32 batch	**0.786**
	emotions	10 epochs, 32 batch	0.768
	toxicity + most	10 epochs, 16 batch	0.749
RoBERTa + CNN	toxicity	10 epochs, 16 batch	0.766
BETO		10 epochs, 16 batch	0.780

Table 7
Extended model comparison on Subtask 1. Classical models

Model	Text rep	Used hyper-parameters	F1-Macro
RF + toxicity + most	All-MiniLM	n_estim: 200, max_depth: 30	0.710
RF +irony	All-MiniLM	n_estimators: 300, max_depth: 20	0.696
RF	All-MiniLM	n_estimators: 300, max_depth: 10	0.685
LR	BoW	C:10, penalty: l1, solver: saga	0.705
SVC	BoW	C:0.1, gamma: scale, kernel: linear	0.700
LR	All-MiniLM*	C:10, penalty: l2, solver: liblinear	0.696
LR	All-MiniLM	C:10, penalty: l2, solver: saga	**0.726**
SVC	All-MiniLM*	C:10, gamma: 0.1, kernel: rbf	0.705
SVC	All-MiniLM	C:10, gamma: 0.1, kernel: lbf	**0.776**
Voting + LR + SVC	BoW	best hyperparam for each	0.704
Bagging + LR	BoW	C:10, penalty: l1, solver: saga	0.703
AdaBoost + SVC	BoW	C:10, gamma: scale, kernel: linear	0.488
XGBoost	BoW	max_depth: 3, eta: 0.1	0.641
Voting + LR + SVC + DTC	All-MiniLM	best hyperparam for each	0.691
Bagging + LR	All-MiniLM	C:10, penalty: l2, solver: saga	0.721
Bagging + SVC	All-MiniLM	C:10, gamma: 0.1, kernel: rbf	**0.732**
AdaBoost + SVC	All-MiniLM	C:10, gamma: 0.1, kernel: rbf	0.591
Stacking + DTC + SVC + KNN + LR	All-MiniLM	best hyperparam for each	0.718

* indicates the data has been processed before the text representation, if the technique chosen is embedding representation.

Table 8
Extended model comparison on Subtask 2a. Transformers

Model	Features	Used hyperparameters	F1-Macro
RoBERTa	toxicity	10 epochs, 32 batch	0.824
	context + most	10 epochs, 32 batch	0.807
	toxicity + hate + emotions + sentiment	10 epochs, 32 batch	**0.853**
	toxicity + hate + emotions + context + sentiment	10 epochs, 32 batch	**0.895**
BETO	sentiments + emotions + hate + toxicity + irony + context + most	10 epochs, 32 batch	**0.876**

Table 9
Extended model comparison on Subtask 2a. Classical models

Model	Text rep	Used hyper-parameters	F1-Macro
MLPC	BoW	act: identity, alpha: 0.0001, h_l_s: (128, 32), solver:lbfgs	0.510
MLPC	All-MiniLM	act: relu, alpha: 0.0001, h_l_s: (128, 32), solver:adam	0.550
Voting + RFC + MLP	All-MiniLM	best hyperparam for each	0.699

Table 10
Extended model comparison on Subtask 2b. Transformers

Model	Features	Used hyperparameters	RMSE
RoBERTa	sentiments + emotions + hate + irony + toxicity + most	10 epochs, 32 batc	0.821
BETO	sentiments + emotions + hate + toxicity + irony	10 epochs, 32 batch	0.851

Table 11
Extended model comparison on Subtask 2b. Classical models

Model	Text rep	Used hyper-parameters	RMSE
ExtraTressRegressor + sentiments + emotions + hate + irony + toxicity	All-MiniLM	n_estim: 250, max_depth: 30	0.690

References

[1] L. I. Merlo, When humour Hurts: A Computational Linguistic Approach, Bachelor's thesis, Universitat Politècnica de València, 2022. URL: http://hdl.handle.net/10251/188166.

[2] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, V. Araujo, Albeto and distilbeto: Lightweight spanish language models, 2023. arXiv:2204.09145.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. arXiv:1907.11692.

[5] J. M. Pérez, D. A. Furman, L. A. Alemany, F. Luque, Robertuito: a pre-trained language model for social media text in spanish, 2022. arXiv:2111.09453.

[6] R. Labadie-Tamayo, B. Chulvi, P. Rosso, Everybody hurts, sometimes. overview of hurtful humour at iberlef 2023: Detection of humour spreading prejudice in twitter, in: Procesamiento del Lenguaje Natural (SEPLN), 2023.

[7] E. D. Liddy, Natural language processing, in: Encyclopedia of Library and Information Science, 2nd ed., Marcel Decker, Inc., New York, 2001.

[8] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov, Fasttext.zip: Compress-

ing text classification models, 2016. arXiv:1612.03651.

[9] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, 2016. arXiv:1607.01759.

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, 2017. arXiv:1607.04606.

[11] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2020. URL: https://arxiv.org/abs/2004.09813.

[12] I. Goyal, P. Bhandia, S. Dulam, Finetuning for sarcasm detection with a pruned dataset, 2022. arXiv:2212.12213.

[13] J. M. Pérez, J. C. Giudici, F. Luque, pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks, 2021. arXiv:2106.09462.

[14] A. T. Handoyo, H. rahman, C. J. Setiadi, D. Suhartono, Sarcasm detection in twitter - performance impact while using data augmentation: Word embeddings, INTERNATIONAL JOURNAL of FUZZY LOGIC and INTELLIGENT SYSTEMS 22 (2022) 401–413. URL: https://doi.org/10.5391%2Fijfis.2022.22.4.401. doi:10.5391/ijfis.2022.22.4.401.

[15] Q. Wang, Z. Luo, J. Huang, Y. Feng, Z. Liu, A novel ensemble method for imbalanced data learning: Bagging of extrapolation-smote svm, Computational Intelligence and Neuroscience 2017 (2017) Article ID 1827016. URL: https://doi.org/10.1155/2017/1827016.