I2C-Huelva at SemEval-2023 Task 10: Ensembling Transformers Models for the Detection of Online Sexism

Lavinia Felicia Fudulu, Alberto Rodríguez Tenorio, Victoria Pachón Álvarez, Jacinto Mata Vázquez
Escuela Técnica Superior de Ingeniería. Universidad de Huelva (Spain)
{laviniafelicia.fudulu912, alberto.rogriguez792}@alu.uhu.es,
{vpachon, mata}@uhu.es, mata@uhu.es

Abstract

This work details our approach for addressing Tasks A and B of the Semeval 2023 Task 10: Explainable Detection of Online Sexism (EDOS). For Task A a simple ensemble based of majority vote system was presented. To build our proposal, first a review of transformers was carried out and the 3 best performing models were selected to be part of the ensemble. Next, for these models, the best hyperpameters were searched using a reduced data set. Finally, we trained these models using more data. During the development phase, our ensemble system achieved an f1-score of 0.8403. For task B, we developed a model based on the deBERTa transformer, utilizing the hyperparameters identified for task A. During the development phase, our proposed model attained an f1-score of 0.6467. Overall, our methodology demonstrates an effective approach to the tasks, leveraging advanced machine learning techniques and hyperparameters searches to achieve high performance in detecting and classifying instances of sexism in online text.

1 Introduction

Sexism refers to any type of mistreatment or negative attitude that is specifically directed towards women based on their gender or based on their gender combined with one or more additional aspects of their identity (such as being a black woman, Muslim woman, or transgender woman). Allowing sexism to thrive on these platforms can normalize violence against women and other individuals who face discrimination based on gender or gender identity.

Therefore, it is crucial to take measures to eradicate sexism on social media and promote a more inclusive and respectful environment for all individuals. In this work, were used Transformer-based solutions (Wolf et al. 2020) and follow an ensemble strategy, specifically for the sexism detection in Task A. By completing these tasks, we became aware selecting the correct hyperparameters for each model also boosted the prediction rate for the model.

2 Background

A dataset developed for the SemEval 2023 Task - Explainable Detection of Online Sexism (Kirk et al. 2023) is used to train, validate, and test models. This dataset contains labelled data from the social network Twitter in English language. Labels associated to each tweet gives information used to classify if the tweet is sexist or not (for Task A), and in case that the tweet is sexist, categorize the sexism of the tweet (for Task B). The total size of the dataset is 13984 tweets (10602 not sexist and 3398 sexist).

The dataset, common to all tasks, contains the tweets along with the labels used for their classification. These labels are divided in two types: sexist label (used to detect if the tweet is sexist or not sexist) and category label (used to categorize the sexism of the tweet threats, derogation, animosity, and prejudiced discussion).

For Task A, based on the strategy described in (Vaca-Serrano 2022) for a similar task, a simple ensemble based of majority vote system was presented. First a review of transformers was carried out and the three best performing models were selected to be part of the ensemble. These models were trained in two phases. During the first phase, the best hyperparameters for each model were found. In the second phase these hyperparameters are used to learn with more training data. Finally, a simple assembly strategy is used.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 763–769

For task B, we developed a model based on the deBERTa transformer, utilizing the hyperparameters identified for Task A.

3 System Overview

In the approach proposed in this paper, we perform two tasks: detection of sexism (Task A) and classification of sexism (Task B).

3.1 Task A – Binary Sexism Detection

This task consists of creating a binary classification where the system must predict whether a tweet is sexist or not sexist. To accomplish this, we decided to select different pretrained models and train them with the dataset provided by the organizers.

To obtain the best models, preliminary experiments were conducted, where the data was split into two parts: training-validation and test, and models were trained using these parts. These experiments were conducted using three different versions of the dataset provided by the organizers. After these experiments, the three best models in terms of f1-score were selected to be part of the ensemble. Next, the appropriate hyperparameters for the models were obtained using a reduced version of the training-validation set. Subsequently, the model was trained with the optimal hyperparameters found using the entire dataset.

Hyperparameters optimization was done with WandB (Biewald 2020), which simulates the training process of the models and mixes the possible values of the hyperparameters to find the combination that maximizes the desired score.

Finally, after the hyperparameter search, each model was trained, and prediction files were obtained. The results of the models were combined to achieve better results than those obtained with each separate model.

3.2 Task B – Category of Sexism

Our approach for this task was to select from the three models selected for the ensemble in the previous task, the one that maximized the f1-score, using the same hyperparameters and dataset than Task A. As the number of sexist labels is greater and the train data is smaller, the expected results in this task are worse than the previous one. As a tweet cannot be categorized as more than one type of sexism, the system is modelled as a multiclass classification, rather than a multilabel classification.

4 Experimental Setup

Overall, our experimental setup for an ensemble involved the following steps: experimental data preparation, model selection, hyperparameter tuning, ensemble creation and evaluation to ensure optimal performance.

- Experimental Data Preparation: EDOS dataset contains 13,984 tweets of which 80% was used for training the models, 16% for validation and the remaining 4%. The only preprocessing that was done at his level was to lowercase and remove urls, users, audio, and video links.

- Models Selection: To select the transformers to be part of the ensemble, several models were tested (see Section 4.2). These models were trained on the dataset described above and the 3 of them with the best macro f1-score were selected for the next step (hyperparameter search). Metrics can be seen in Table 1.

- Hyperparameter Search: A search for the best hyperparameters for these 3 models was then carried out. The hyperparameter space is described in Table 2. The method used was grid search.

- Ensemble Creation: Once the best hyperparameters have been found for each transformer, the three selected models were trained with these hyperparameters on 3 datasets. One of these datasets is the same as the one we have used previously. The other two are variants of the emoji preprocessing. Descriptions of these preprocessings are given in Section 4.1. In Table 4 the performance of these models trained with each dataset can be found. Those 3 with best macro f1-score were select to be part of the ensemble.

- Ensemble Evaluation: The final step is to evaluate the ensemble performance on the test set.

Model	Accuracy	Precision	Recall	AUC	Macro f1-score
bertweet-base-sentiment-analysis	0.50	0.50	1.00	0.50	0.67
roberta-large	0.76	0.77	0.76	0.76	0.76
nghuyong/ernie-2.0-base-en	**0.81**	**0.82**	**0.82**	**0.82**	**0.82**
bert-base-uncased	**0.81**	**0.82**	**0.81**	**0.81**	**0.81**
cardiffnlp/twitter-roberta-base-sentiment	0.78	0.78	0.78	0.78	0.78
roberta-base	0.76	0.76	0.76	0.76	0.76
microsoft/deberta-v3-base	**0.81**	**0.84**	**0.84**	**0.85**	**0.83**

Table 1. Preliminary experiments results to select the models for the ensemble

4.1 Data preprocessing

Our experiments have been carried out with 3 datasets, all derived from the dataset provided by the organisers, using different preprocessing techniques. The first dataset, which we will call 'original', was subjected to the following pre-processing techniques: convert all letters to lowercase and remove url, users, audio, and video links.

As it was said above, we created two more versions of the dataset, with the scope of study if the different treatments of the emojis in the datasets provided useful information and improved the score obtained:

- **Emojis_As_Text**: Replace all emojis with a text. To do this we used Python library *emoji* -https://pypi.org/project/emoji/

  Emojis_As_Tokens: Tokenize all emojis and add them to the model tokenizer. To do this, the tokenizer of the corresponding model was modified by adding the emojis appearing in the dataset. This was done so that the emojis would be treated as another token, rather than as an unknown word.

4.2 Models selection

As mentioned above, to find the models that would later form part of the ensemble, the performance of a set of transformers was checked, trained them with the experimental dataset (80% for training, 16% for validation and the remaining 4% for testing).

Models used for these preliminary experiments were:

- **Bertweet-base-sentiment-analysis** (Pérez, et al. 2021): Deep learning model designed to analyze the sentiment expressed in short texts such as tweets. It uses the Transformers architecture and a pre-trained variant of BERT to classify the sentiment into positive, negative, or neutral.
- **Roberta-large** (Liu et al. 2019): Deep learning model based on the Transformers architecture, larger than BERT, and capable of handling large datasets. It uses the "masking retraining" technique to learn more effectively from data, which allows for a better understanding of language.
- **nghuyong/ernie-2.0-base-en** (Sun et al. 2020): Highly effective and accurate deep learning language model, pre-trained on natural language processing tasks in English. It is an enhanced version of BERT, which enables a better understanding of the context and meaning of words.
- **Bert-base-uncased** (Devlin et al. 2018): Deep learning language model pre-trained on a large amount of text data without distinguishing between uppercase and lowercase letters. It uses the Transformer architecture, which is highly efficient and effective in natural language processing.
- **cardiffnlp/twitter-roberta-base-sentiment** (Barbieri et al. 2020): Deep learning language model designed specifically for sentiment analysis in tweets. It is based on the transformers architecture and uses a pre-trained variant of BERT, called RoBERTa.
- **Roberta-base** (Liu et al. 2019): Deep learning language model based on the Transformer architecture, designed to process natural language text and perform various natural language processing tasks

such as text classification and sentiment analysis.

- **microsoft/deberta-v3-base** (He et al. 2022): Deep learning language model that uses the Transformer architecture and has been pre-trained on a large amount of text data. It focuses on understanding the syntactic and semantic structure of natural language and is designed for natural language processing tasks such as sentiment analysis, text classification, and text generation.

Hyperparameters used to train all these models were: 32 batch size, 5 epochs, 0.01 weight, 128 max length, decay and 2e-5 learning rate. These preliminary results are shown in Table 1. As it can be seen, ernie-2.0-base-en, bert-base-uncased and deberta-v3-base performed the best macro f1-score, so those models were selected to be part of the ensemble. For these models, the best combination of hyperparameters was then searched for. Details of the hyperparameter search are given in the next section.

4.3 Hyperparameter optimization

Once the three models were selected, we used a random reduced train and validation splits to find the best hyperparameters for each of the models. To do this, 500 sexist tweets and 500 non-sexist tweets were randomly selected from the training set.

Table 2 shows the hyperparameter space used for this first training step. Epochs number is set to 5 with early stopping patience 3 in all cases.

Hyperparameters	Values
Learning-Rate	2e-5, 3e-5, 5e-5,
Train Batch Size	16, 32
Weight decay	0.1, 0.01, 0.001
Max length	32, 64

Table 2. Hyperparameter space for Task A

The method used was grid search. Best hyperparameters found for each model are shown in Table 3.

Model	BS	LR	WD	ML
ernie-2.0-base-en	32	2e-5	0.1	64
bert-base-uncased	32	2e-5	0.1	64
deberta-v3-base	16	2e-5	0.01	64

Table 3. Models Best Hyperparameters (BS: Batch Size; LR: Learning Rate; WD: Weight Decay; ML: Max Length)

4.4 Ensemble creation

From the dataset provided by the organisers, three versions of the dataset were created as was explained in Section 4.1 and models were trained with them with the hyperparameters shown in Table 3. Table 4 shows the performance of these models with the three datsets. As it can be seen, combinations of *nghuyong/ernie-2.0-base-en* + *original*, *bert-base-uncased* + *original* and *deBERTa* + *original*, performed the best macro f1-score.

Figure 1 shows the flowchart of the methodology followed in the experimentation phase.

Model	Dataset	Accuracy	Precision	Recall	AUC	Macro F1-score
nghuyong/ernie-2.0-base-en	Original	0.81	0.82	0.76	0.82	**0.82**
	Emojis_As_Text	0.81	0.82	0.76	0.81	0.81
	Emojis_As_Tokens	0.80	0.80	0.73	0.80	0.81
bert-base-uncased	Original	0.81	0.82	0.76	0.82	**0.82**
	Emojis_As_Text	0.82	0.82	0.77	0.81	0.82
	Emojis_As_Tokens	0.82	0.82	0.77	0.81	0.82
Microsoft/deberta-v3-base	Original	0.88	0.84	0.84	0.84	**0.84**
	Emojis_As_Text	0.88	0.85	0.82	0.82	0.83
	Emojis_As_Tokens	0.88	0.83	0.83	0.83	0.84

Table 4: Model performance

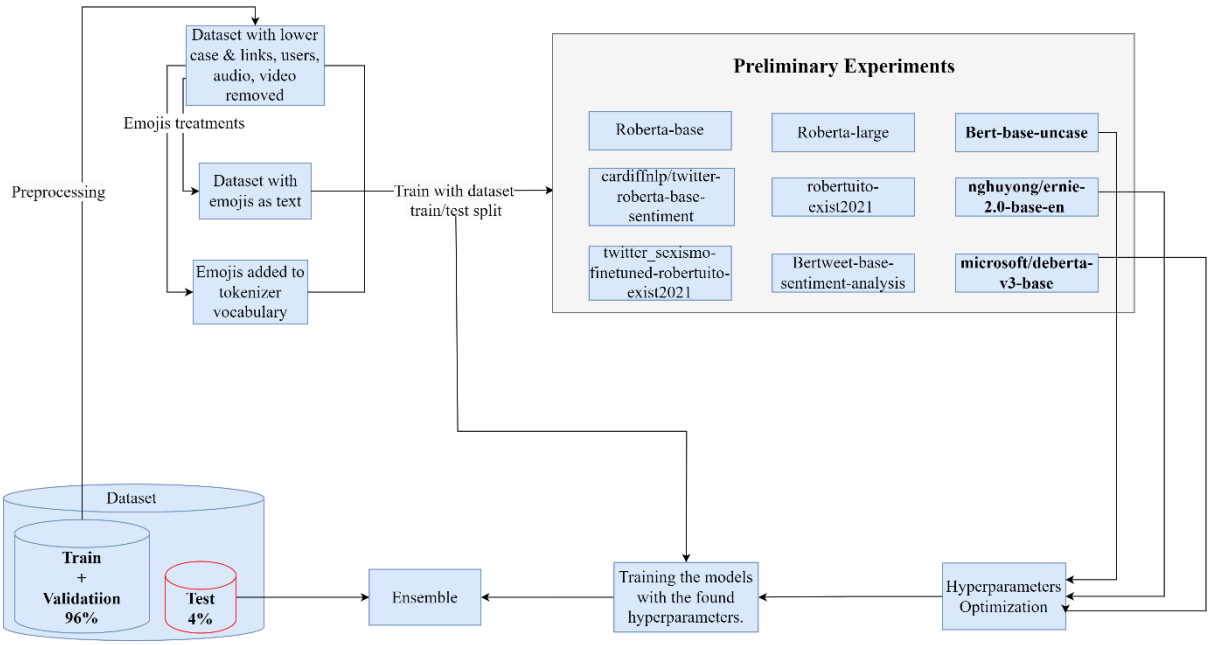

Figure 1. Experimental Flow Diagram.

Using these three models thus trained, the ensemble presented as our proposal for Task A was created. An ensemble is a machine learning technique that combines the results of different models to obtain a better result. When given a text input, each model generates an output, and these outputs are combined using an aggregation technique, in our case, a majority vote model, to generate the ensemble output. Table 5 shows the performance of the ensemble and the individual models over the development dataset. These results were obtained once the labels of the development dataset were released. As can be seen, the results of the ensemble are better than those obtained by the individual models. In the test phase, it achieved a 0.83, reaching the 33 position.

Model	Accuracy	Precision	Recall	AUC	Macro f1-Score
nghuyong/ernie-2.0-base-en	0.89	0.85	0.83	0.828	0.84
bert-base-uncased	0.86	0.82	0.82	0.819	0.82
Microsoft/deberta-v3-base	0.87	0.82	0.81	0.808	0.82
Ensemble	0.88	0.85	0.83	0.829	0.84

Table 5: Performance Comparison

4.5 Task B

For this task, a similar approach to the previous task was taken. The first step in this approach was to select the best-performing model from the previous task. To do this, we examined the results of the three models that were used in the previous task and selected the one that had the highest macro f1-score. This model was then chosen for use in the current task. Then we used the same hyperparameters and dataset preprocessed that were used in the previous task for the selected model.

As can be seen in Table 1, the best performing model was deberta-v3-base, with the original dataset version and the hyperparameters shown in Table 3. This model was trained again with the dataset distribution for train and test described in Section 4. As result of the evaluation deberta-v3-base model got macro f1-score 0.64 over the development dataset.

5 Results

Preliminary experiments results are presented in Table 1. The optimal hyperparameters for fine-tuning the final models are listed in Table 3. The results of final testing on different versions of the dataset are presented in Table 4. Table 5 shows that the ensemble performed better than the individual models of which it was composed. The agreement percentage (that is, the number of cases in which the three models coincided in their prediction) was 86%.

Finally, the results of the ensemble of models described in previous sections during the development and testing phases of the competition are presented in Table 6, along with their ranking.

Task	Development Score	Test Score	Ranking
A	0.8403	0.8396	33
B	0.6467	0.5794	51

Table 6: Competition results

5.1 Errors Analysis

This section will present a brief study of errors detected when making a prediction through the model. The purpose of this section is to understand why the prediction was not accurate and determine the causes of the errors found.

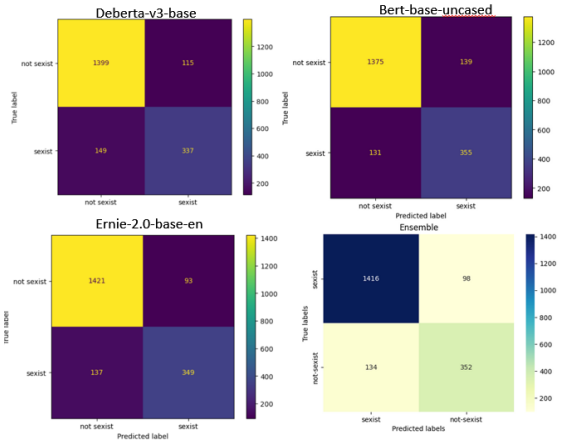

Figure 2. Confussion Matrix – Task A

Figure 2 shows the confusion matrix (calulated from the development dataset once the labels were released) of the ensemble and its models. Overall the ensemble behaved better than its components.

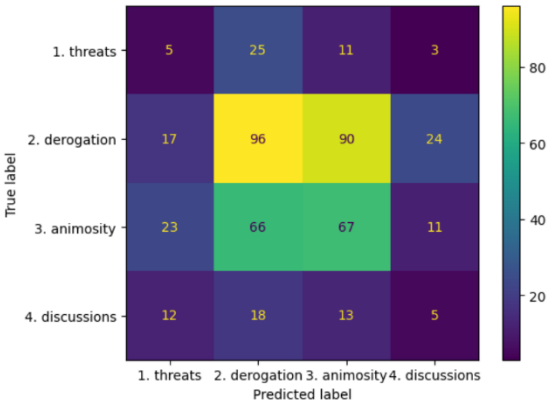

Figure 3. Confusion Matrix – Task B

Tweet	Class	Prediction
looks like bit the hand that fed her.	Not sexist	Not sexist
i predict joe will brow beat her into self flagellation.	Sexist	Not sexist

Table 7: Error Analysis Example

Figure 3 shows the confusion matrix relative to the proposed model for Task B. As can be seen, most cases of error occur between the derogation class and the animosity class.

In Table 7, an example of prediction error and a correct prediction are shown (column class express the class given in the dataset for that tweet, while column prediction shows the prediction of our model). The tweet "I predict Joe will brow beat her into self flagellation" is considered sexist maybe due to the use of the term "brow beat", which implies an attitude of superiority and dominance on Joe's part towards an unidentified woman ("her"). Additionally, the phrase suggests that the woman will be forced to punish herself ("self flagellation") as a result of Joe's alleged intimidation. But the model predicted “not-sexist” maybe because of the use of “I predict”, which implies that someone is talking about the attitude of someone else. Even though Joe’s behavior can be considered sexist, the comment about him can be either a criticism or a call for attention.

The phrase "looks like she bit the hand that fed her" is not inherently sexist. The expression "bit the hand that fed her" refers to someone who has been ungrateful or disloyal to someone who has provided them with help or support in the past.

Although the exact cause of model errors has not been identified in some cases, they may be due to common reasons why transformers can fail in their

predictions, such as insufficient or inadequate data, model overfitting, noise in the data, bias in the data, or incorrect use of hyperparameters.

6 Conclusions

In this work we present our approach for addressing Tasks A and B of the Semeval 2023 Task 10: Explainable Detection of Online Sexism (EDOS). During the development phase, our overall system got a macro f1-score of 0.8403 for Task A and 0.6467 for Task B. Overall, our methodology demonstrates an effective approach to tasks, leveraging advanced machine learning techniques and hyperparameter searches to achieve high performance in detecting and classifying instances of online text sexism.

Acknowledgments

This paper is part of the I+D+i Project titled "Conspiracy Theories and hate speech online: Comparison of patterns in narratives and social networks about COVID-19, immigrants, refugees and LGBTI people [NON-CONSPIRA-HATE!]", PID2021-123983OB-I00, funded by MCIN/AEI/10.13039/501100011033/ and by "ERDF A way of making Europe".

References

Barbieri, Francesco, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification." *Findings of the Association for Computational Linguistics: EMNLP 2020*: 1644. doi:10.18653/v1/2020.findings-emnlp.148.

Biewald, Lukas. 2020. "Experiment Tracking with Weights and Biases." . https://www.wandb.com/.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *CoRR* abs/1810.04805.

He, Peng, Gang Zhou, Mengli Zhang, Jianghong Wei, and Jing Chen. 2022. "Improving Temporal Knowledge Graph Embedding using Tensor Factorization." *Applied Intelligence (Dordrecht, Netherlands)*. doi:10.1007/s10489-021-03149-w.

Kirk, Hannah Rose, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. "SemEval-2023 Task 10: Explainable Detection of Online Sexism.". doi:10.48550/arxiv.2303.04222. https://arxiv.org/abs/2303.04222.

Liu, Zhuang, Wayne Lin, Ya Shi, and Jun Zhao. 2019. *A Robustly Optimized BERT Pre-Training Approach with Post-Training* Springer International Publishing. doi:10.1007/978-3-030-84186-7_31.

Pérez, Juan Manuel, Juan Carlos Giudici, and Franco Luque. 2021. "Pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP Tasks." . doi:10.48550/arxiv.2106.09462. https://arxiv.org/abs/2106.09462.

Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. "ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding." *Proceedings of the ... AAAI Conference on Artificial Intelligence* 34 (5): 8968-8975. doi:10.1609/aaai.v34i05.6428.

Vaca-Serrano, Alejandro. Detecting and Classifying Sexism by Ensembling Transformers Models. *IberLEF 2022.*

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, et al. 2020. "Transformers: State-of-the-Art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. doi:10.18653/v1/2020.emnlp-demos.6.