Stanford MLab at SemEval-2023 Task 10: Exploring GloVe- and Transformer-Based Methods for the Explainable Detection of Online Sexism

Hee Jung Choi*, Trevor Chow*, Aaron Wan*, Hong Meng Yam*, Swetha Yogeswaran*, Beining Zhou*
Stanford University
{cheejung, tmychow, aaronwan, hongmeng, swethay, cathyzbn}@stanford.edu

Abstract

In this paper, we discuss the methods we applied at SemEval-2023 Task 10: Towards the Explainable Detection of Online Sexism. Given an input text, we perform three classification tasks to predict whether the text is sexist and classify the sexist text into subcategories in order to provide an additional explanation as to why the text is sexist. We explored many different types of models, including GloVe embeddings as the baseline approach, transformer-based deep learning models like BERT, RoBERTa, and DeBERTa, ensemble models, and model blending. We explored various data cleaning and augmentation methods to improve model performance. Pretraining transformer models yielded significant improvements in performance, and ensembles and blending slightly improved robustness in the F1 score.

1 Introduction

Online sexism has the potential to inflict significant harm on women (Ortiz, 2023), and it is a serious issue that must be addressed. With the increasing prevalence of social media, it has become easy for groups of people to spread sexist ideas and threaten the safety of others, with online social networks becoming increasingly inundated by sexist comments (Founta et al., 2018).

There have been numerous previous works on the detection of online sexism as a whole (Schütz et al., 2021; Aldana-Bobadilla et al., 2021; Rodríguez-Sánchez et al., 2020), including even applying these models to non-English datasets (Kumar et al., 2021; de Paula et al., 2021; Jiang et al., 2021). However, almost all of these models do not focus on precisely classifying why a certain text conveys sexist sentiments, and instead provide a binary classification for whether the text is sexist.

However, this is the task which is most likely to make these models useful for content moderation, since they provide both the moderator and the platform's users with a consistent explanation for why something was classified as sexist or not. In many cases, the difficulty of using machine classification when doing content moderation is the perception of it being decided by an arbitrary black box.

While the models may empirically have a high accuracy, due to the high social sensitivity of this task, human intervention is almost always required. With laws such as the General Data Protection Regulation (GDPR) established in Europe, which establishes a "right to explanation"(Hoofnagle et al., 2019), there is thus a huge need for model explainability on top of existing performance optimization (Mathew et al., 2020). With more detailed feedback about the categories of sexism, moderators can efficiently mitigate sexist sentiment online in a robust and rules-based manner, and therefore reduce gender-based violence.

As such, given the increasing importance of explainable detection in machine learning models, we propose and compare several natural language processing methods for doing so. We used GloVe- and transformer-based models, as well as various data cleaning and augmentation techniques, applying them on Reddit and Gab textual data to detect sexist messages and classify them into various categories of sexism.

2 Background and Task Setup

The data for this task was provided by SemEval Task 10 (Kirk et al., 2022). This labeled data set consisted of 10,000 entries extracted from Gab along with 10,000 entries from Reddit. The dataset is labeled according to the specifications of the required classifier for subtask A, subtask B, and subtask C. In addition to this labeled dataset, there were two unlabeled data sets which each contain 1 million entries from Gab and Reddit that were

[*] Authors are listed alphabetically. All authors contributed equally to this work.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1791–1797
July 13-14, 2023 ©2023 Association for Computational Linguistics

provided, which we used to improve our system's performance.

Subtask A requires a binary classifier to categorize posts into being sexist or non-sexist. Subtask B requires a four-class classification system which categorizes a *sexist* post according to one of the following categories: (1) threats, (2) derogation, (3) animosity, and (4) prejudiced discussions. Finally, for subtask C, of the posts which are *sexist*, an 11-class classification system categorizes the posts according to a more specific label of sexism. Subtasks B and C ensure that text which is labeled as sexist is given a specific label for why it is sexist, providing a degree of explainability for those using the model.

3 System Overview

3.1 Data Cleaning

The training data was taken from Reddit and Gab, and as such, it was essential to clean the data to get consistent formatting. Furthermore, all URL references were removed, hyphens and hashtags were replaced with spaces, all punctuation except apostrophes was removed, and all text was changed to lowercase. Finally, many slang abbreviations were replaced by their expanded forms using the mapping provided by the sms_slang_translator github repository[1].

3.2 Data Augmentation

3.2.1 Back Translation

For Subtask A, since the provided dataset contained far more "Not Sexist" samples than "Sexist" samples, we attempted to use back translation to generate augmented samples of the minority class. Specifically, we translated the minority samples in our training split to Dutch and then back to English, ensuring an even class distribution in our training split. We chose Dutch because it worked empirically (Beddiar et al., 2021) and since Dutch is one of the most similar languages to English, owing to their joint lineage in the West Germanic family. Results with back translation are specifically labeled in the results section.

3.2.2 Easy Data Augmentation

Since back translation did not improve our results in Subtask A, we attempted a different data augmentation approach for Subtask B: Easy Data Augmentation (EDA). We specifically followed a procedure similar to (Kalra and Zubiaga, 2021). We used three operations— synonym replacement, random insertion, and random swap with a rate of 0.05—to generate augmented samples of the three minority classes in Subtask B (namely "threats, plans to harm and incitement", "animosity", and "prejudiced discussions") in our training split. Much as with back translation, we generated enough samples for each minority class until the number of samples for each class was equal. Results with EDA are specifically labeled in the results section.

3.3 GloVe-Based Model

For our baseline model for Subtask A, we developed a GloVe-based logistic regression model. Although this is not a state-of-the-art model, it provided a useful benchmark for the performance of non-deep learning approaches. We used 50-dimensional GloVe vectors pre-trained on 2 billion tweets from (Pennington et al., 2014) to transform each word in the input text into its vector representation. For each sample's input text, we averaged the word vectors across the text to create a 50-dimensional input that we fit with a logistic regression model.

3.4 Transformer-based Models

3.4.1 BERT

BERT (Bidirectional Encoder Representations for Transformers) is a large-language model that has achieved impressive results in NLP experiments (Devlin et al., 2018). It uses a multi-layer, transformer-based encoder architecture and bidirectional self-attention to learn context from both preceding and following sentences. BERT was trained on a language modeling task as well as a next sentence prediction task.

We fine-tuned the BERT model to apply it to our specific Subtasks. We used the existing pre-trained bert-base-uncased tokenizer to preprocess the text for input the BERT model. We added one linear layer with ReLU activation as well as a second linear layer with the Sigmoid (Subtask A) or Softmax (Subtask B and C) activation function to generate the model output. The ultimate prediction was determined either by threshold (for Subtask A) or the argmax of the output vector (for Subtask B and C). For Subtask A, we found that the optimal threshold for a positive prediction was 0.35, as seen in Figure 1. If the model's output was 0.35 or greater, the text would be classified as sexist. We added dropout

[1] https://github.com/rishabhverma17/sms_slang_translator

with a rate of 0.5 to help reduce overfitting.

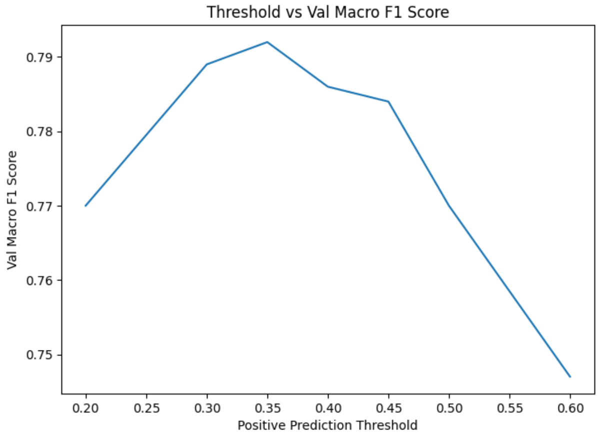

Figure 1: Positive Prediction Thresholds for BERT Classifier

3.4.2 RoBERTa

RoBERTa (Robustly optimized BERT approach), improves on BERT by using a larger-scale model trained on an even larger and cleaner corpus of text using a longer training schedule, larger batch sizes, and a more advanced masking strategy, resulting in improved performance on a wide range of natural language processing tasks (Liu et al., 2019).

We fine-tuned two separate RoBERTa models using the same architecture as our BERT model. We chose to use RoBERTa because it is trained more robustly using dynamic masking as compared to static masking in BERT, and was proven empirically to yield better results. This is because the generation of masks during training means that for each input, the number of different possible masks is much larger than with BERT.

The two separate RoBERTa models included: one existing RoBERTa model that was fine-tuned to classify sexist tweets (ft-RoBERTa)[2], and one RoBERTa model that was pre-trained on the provided unlabeled data (pt-RoBERTa)[3].

For Subtask A, we fine-tuned the ft-RoBERTa model. For Subtask B, we fine-tuned both the ft-RoBERTa and the pt-RoBERTa models. For Subtask C, we fine-tuned the pt-RoBERTa model. For all three Subtasks, we used the same architecture as the BERT model.

3.4.3 HateBERT

HateBERT is a model that improves upon BERT for the task of abusive language detection (Caselli et al., 2021). More specifically, HateBERT uses training data with abusive language to better focus BERT's attention on relevant linguistic cues for this task. It employs a refined pre-training procedure using a large dataset of comments from banned subreddits containing offensive, abusive and hateful language to capture the nuances of this type of speech. Overall, these refinements on domain-specific data have allowed HateBERT to empirically achieve superior performance on various English hate speech detection datasets. We used this only for Subtask B.

3.4.4 DeBERTa

DeBERTa (Decoding-enhanced BERT with disentangled attention) is a model that improves upon both BERT and RoBERTa (He et al., 2020). More specifically, DeBERTa uses disentangled attention to better focus on relevant linguistic information, employs a more advanced decoding strategy that captures long-range dependencies, shares parameters between layers to improve efficiency and performance, and uses improved pre-training strategies to capture complex linguistic relationships. Overall, these improvements have led the DeBERTa model to empirically achieve better performance on various downstream NLP tasks.

For Subtasks B and C, we fine-tuned a DeBERTa model for the respective downstream tasks, using the same architecture as the BERT model.

3.4.5 Sentence-BERT

Sentence-BERT (SBERT) is a modification of BERT, with the key innovation being that SBERT is fine-tuned to encode sentences into fixed-length vectors that capture the semantic meaning of the sentence (Reimers and Gurevych, 2019). Since SBERT focuses on generating high-quality sentence embeddings while BERT generates embeddings on the word level, we experimented with the SBERT-based model in hopes of addressing the overfitting we saw when applying other BERT-based models. For Subtask B, we fine-tuned the SBERT model using the same architecture we used for the other BERT-based models.

3.4.6 Ensembles

In hopes of improving model performance, we experimented with ensemble models that involved

[2] https://huggingface.co/annahaz/xlm-roberta-base-misogyny-sexism-tweets

[3] https://huggingface.co/HPL/roberta-large-unlabeled-labeled-gab-reddit-task-semeval2023-t10-270000sample

concatenating embeddings from two different transformers into a single feature vector and then applying the linear and dropout layers on top of that feature vector to generate the ultimate prediction. We specifically experimented with two ensemble models: concatenating pt-RoBERTa embeddings with DeBERTa embeddings (referred to as Ensemble 1) and concatenating pt-RoBERTa embeddings with SBERT embeddings (referred to as Ensemble 2). Through concatenating the embeddings, we hoped to capture different aspects of the input text that may be better represented by one model over the other and develop a more comprehensive representation of the input text.

For Subtask B, we applied both Ensemble 1 and Ensemble 2, and for Subtask C, we applied Ensemble 1. For each ensemble model, both transformer embeddings were simultaneously fine-tuned on the corresponding downstream task during model training.

3.4.7 Model Blending

To further improve model performance, we experimented with a blending strategy that involved taking the weighted average between the predictions of two different models, giving slightly greater weight to the better-performing model. For Subtask A, we took the weighted average of the ft-RoBERTa and BERT models, multiplying ft-RoBERTa's prediction by 0.6 and BERT's prediction by 0.4 and summing the results for the ultimate prediction. For Subtasks B and C, we took the weighted average of Ensemble 1 and the pt-RoBERTa model. Similar to Subtask A's blended model, we multiplied Ensemble 1's predictions by 0.6 and pt-RoBERTa's predictions by 0.4 and summed them to generate the ultimate prediction.

4 Experimental Setup

To train and evaluate our models, we used an 80-20 split on the provided training dataset to create training and validation datasets for our models. We used a fixed train-val split so we could directly compare the performance of our models. During training, we monitored the Macro F1 score on the validation set, and our program saved the model with the best score. We trained our models with a maximum of 200 epochs, and we used early-stopping to stop training if the training loss did not decrease over 30 epochs. We trained all of our transformer-based models using the Adam optimizer and cross-entropy loss.

For our final submission, we trained our model on all the training data provided.

5 Results

5.1 Summary

As seen in Table 1, the best-performing model for Subtask A was the weighted-average blending strategy, though the difference in F1 scores between the three best-performing models (BERT, ft-RoBERTa, and Weighted Average) was small. ft-RoBERTa had a slightly better score than the plain BERT model. GloVe Vector-based model yielded the worse performance. The data augmentation we performed, such as back-translation, did not lead to any improvements.

For Subtask B, we see in Table 2 that pt-RoBERTa clearly sets itself from the pack out of all transformer-based models, achieving an F1 of 0.62. This is very close to the best results achieved by the models involving multiple transformers, as Ensemble 1 and the Weighted Average model achieved F1s of 0.622 and 0.624, respectively. Like with back translation in Subtask A, the EDA strategy we used for Subtask B failed to lead to any improvements. HateBERT, which is pre-trained on abusive text, beat out most of the benchmark BERT-based models except pre-trained RoBERTa, suggesting that the features it learns from abusive text do not perfectly translate across to sexism.

To prove the importance of data cleaning, we ran an experiment with our pt-RoBERTa model on the uncleaned input text, as seen in Table 2. The resulting F1 score of 0.571 was a significant decrease from the score on the cleaned input text.

Our results in Table 3 show that Ensemble 1 outperforms pt-RoBERTa and the Weighted Average model in Subtask C, which was a surprising difference given the results from Subtask B.

The performance of our final models used for submission on the Dev and Test sets can be seen in Table 4 and Table 5, respectively. Our models' performances on the Dev set were relatively consistent with our Val set results, but the performance on the Test set represented a noticeable decline, especially for Subtasks B and C.

5.2 Discussion

We see from Subtask A that ft-RoBERTa yielded slightly better results than the plain BERT model. However, since we did not test a plain RoBERTa model and because the difference in performance is very small, it is difficult to tell if this improvement was primarily due to the improvements of RoBERTa over BERT or the transfer-learning from fine-tuning on the Twitter task.

However, when it comes to pre-training on domain-specific data, we can clearly see that this is vital to improving results. In Subtask B, we see that pt-RoBERTa outperformed all single-transformer models, including DeBERTa. This shows that the success of pt-RoBERTa model can be primarily attributed to the more robust embeddings created after pre-training the RoBERTa embeddings on the unlabeled dataset.

For Subtasks B and C, creating ensemble-type models by concatenating embeddings from different transformer models produced slight improvements. Ensemble 2 was middle-of-the-pack in Subtask B since it involved concatenating embeddings from a poorer-performing transformers mode (SBERT). For Ensemble 1, while the improvement was only slight for Subtask B, there is a clear difference in performance between the pt-RoBERTa model's F1 score and Ensemble 1's F1 score for Subtask C. This demonstrates how concatenating embeddings from different transformer models can be an effective strategy for creating more robust representations of the input text.

For Subtasks A and B, blending the predictions of the best-performing models led to slight improvements in performance. However, the improvements were small, and for Subtask C, blending the models did not improve results. This indicates that model blending may not be the most optimal approach to improving model performance.

Table 1: Val Macro F1 scores of Subtask A Models

Model	Val F1
GloVe Vectors + Logistic Regression	0.623
BERT	0.792
ft-RoBERTa	0.798
Weighted Average: BERT & ft-Roberta	0.805
Augmentation: BERT & Back Translation	0.789

Table 2: Val Macro F1 scores of Subtask B Models

Model	Val F1
SBERT	0.534
BERT	0.521
DeBERTa	0.562
HateBERT	0.582
ft-RoBERTa	0.525
pt-RoBERTa	0.62
Ensemble 2	0.555
Ensemble 1	0.622
Weighted Average: En. 1 & pt-RoBERTa	0.624
Augmented: pt-RoBERTa & EDA	0.618
Uncleaned: pt-RoBERTa & Raw Data	0.571

Table 3: Val Macro F1 scores of Subtask C Models

Model	Val F1
pt-RoBERTa	0.393
Ensemble 1	0.416
Weighted Average: En. 1 & pt-RoBERTa	0.405

Table 4: Dev Set Results

Model	Dev F1
Weighted Average: BERT & ft-RoBERTa (A)	0.802
Weighted Average: Ensemble 1 & pt-RoBERTa (B)	0.628
Ensemble 1 (C)	0.382

Table 5: Test Set Results

Model	Test F1
Weighted Average: BERT & ft-RoBERTa (A)	0.798
Weighted Average: Ensemble 1 & pt-RoBERTa (B)	0.573
Ensemble 1 (C)	0.354

6 Conclusion

The three subtasks and classification models for online sexism were able to provide explanations for the model predictions by showing intermediate results of the classification. Although this does not provide a look inside a black box, it is nonetheless a useful explanation for the end user of the model, explaining the specific reason why a text might be sexist.

From our experiments, we saw that transformer-based models like BERT and RoBERTa worked the best to classify sexism in texts, as seen in Subtask A. Data cleaning was essential in improving our results. Furthermore, pre-training transformers models like RoBERTa on domain-specific text substantially improved the performance, rivaling the multi-transformer models in Subtask B.

The models involving concatenating transformer embeddings produced slightly (Subtask B) to significantly better (Subtask C) results, illustrating how combining information from different transformer models produces better representations. Blending model outputs led to slight performance improvements, but these improvements were small in comparison to the improvements seen from pre-training and concatenation.

7 Limitations and Future Work

In this paper, we chose the pre-trained RoBERTa model as a consistent benchmark across all three tasks. Beyond RoBERTa, we used differing models for different tasks by iterating on their performances in previous stages, introducing new models (e.g. SBERT, DeBERTa, HateBERT etc.) for Task B due to the poor performance of some models in Task A and picking the better-performing models in Task B for Task C.

Given the significant improvements shown in the pre-trained RoBERTa model on the unlabeled data, our system could be further improved if we pre-trained more robust models, such as DeBERTa, on the unlabeled data. We would like to explore this direction in the future if we had more computational resources.

Ensembles and model blending led to slight performance improvements. However, there are still many combinations and methods of transformer ensembles and model blending we were unable to experiment with due to time constraints. To build a more robust model, we would systematically experiment with other techniques and combinations of ensembles and model blending.

The data augmentation approaches we attempted failed to lead to better results. This could be due to the fact that the augmented data is not in the distribution of the dataset, or it could show that our model is slightly overfitted. We should like to explore other augmentation strategies since dealing with minority classes would be key in further improving the macro F1 score of our system. To navigate class imbalance, we would like to experiment with more generative models in addition to data augmentation and weighting methods.

Nonetheless, we believe these results show the potential of using pre-trained transformer models coupled with concatenating embeddings in explainable textual detection.

Acknowledgements

This research effort would not have been possible without the support of Stanford ACMLab. We would also like to thank Hannah Rose Kirk, Wenjie Yin, Paul Röttger, and Dr. Bertie Vidgen for organizing SemEval 2023 Task 10: Towards the Explainable Detection of Online Sexism. Furthermore, we would like to thank the reviewers for their insightful comments which strengthened the quality of our paper. Finally, we would like to acknowledge Google Colaboratory for its free computing services.

References

Edwin Aldana-Bobadilla, Alejandro Molina-Villegas, Yuridia Montelongo-Padilla, I. Lopez-Arevalo, and Oscar S. Sordia. 2021. A language model for misogyny detection in latin american spanish driven by multisource feature extraction and transformers. *Applied Sciences*.

Djamila Romaissa Beddiar, Md Saroar Jahan, and Mourad Oussalah. 2021. Data expansion using back translation and paraphrasing for hate speech detection. *CoRR*, abs/2106.04681.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Angel Felipe Magnossão de Paula, Roberto Fray da Silva, and I. Baris Schlicht. 2021. Sexism prediction in spanish and english tweets using monolingual and multilingual bert and ensemble models. *ArXiv*, abs/2111.04551.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the international AAAI conference on web and social media (ICWSM)*, abs/1802.00393.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Chris Jay Hoofnagle, Bart van der Sloot, and Frederik J. Zuiderveen Borgesius. 2019. The european union general data protection regulation: what it is and what it means*. *Information & Communications Technology Law*, 28:65 – 98.

Aiqi Jiang, Xiaohan Yang, Yang Liu, and Arkaitz Zubiaga. 2021. Swsr: A chinese dataset and lexicon for online sexism detection. *Online Soc. Networks Media*, 27:100182.

Amikul Kalra and Arkaitz Zubiaga. 2021. Sexism identification in tweets and gabs using deep neural networks. *CoRR*, abs/2111.03612.

Hannah Rose Kirk, Wenjie Yin, Paul Röttger, and Bertie Vidgen. 2022. Semeval 2023 task 10: Towards the explainable detection of online sexism (edos).

Ritesh Kumar, Soumya Pal, and Rajendra Pamula. 2021. Sexism detection in english and spanish tweets. In *IberLEF@SEPLN*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. In *AAAI Conference on Artificial Intelligence*.

Stephanie Ortiz. 2023. “if something ever happened, i’d have no one to tell:” how online sexism perpetuates young women’s silence. *Feminist Media Studies*, 1-16.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, and Laura Plaza. 2020. Automatic classification of sexism in social networks: An empirical study on twitter data. *IEEE Access*, 8:219563–219576.

Mina Schütz, Jaqueline Boeck, Daria Liakhovets, Djordje Slijepcevic, Armin Kirchknopf, Manuel Hecht, Johannes Bogensperger, Sven Schlarb, Alexander Schindler, and Matthias Zeppelzauer. 2021. Automatic sexism detection with multilingual transformer models. *ArXiv*, abs/2106.04908.