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

Hong Meng Yam, Aaron Wan, Swetha Yogeswaran, Beining Zhou, Hee Jung Choi, Trevor Chow
Stanford University
{hongmeng, aaronwan, swethay, cathyzbn, cheejung, tmychow}@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, our goal was to not only classify whether the text is sexist, but also to assign the text to specific subcategories of sexism in order to provide additional information as to why the text is sexist. We explored many types of models, including classical machine learning methods using GloVe embeddings as well as transformer-based deep learning models like BERT, RoBERTa, and DeBERTa. We also explored various data cleaning and data augmentation methods to improve model performance. We showed that transformer-based models performed significantly better than GloVe-based models. Pre-training transformer models also yielded significant improvements in performance, and further optimizations include concatenation and blending model outputs.

1 Introduction

Online sexism is a serious issue that must be addressed. Especially with the increasing use of social media, it has become easy for groups of people to spread sexist ideas and threaten the safety of others. While there have been numerous studies on online sexism detection as a whole, few models have been developed with a focus on precisely classifying sexist sentiments, yet it is precisely such models that will greatly improve the effectiveness of these tools in content moderation for both the moderator and the platform's users. With more detailed feedback about the categories of sexism, moderators can more efficiently mitigate sexist sentiment online and reduce gender-based violence.

Given the increasing importance put on explainable detection AI, we propose and compare between several natural language processing methods, which we hope will be useful in guiding future development of explainable textual models. Herein, we used GloVe- and transformer-based models, and various data cleaning and augmentation techniques, applied on Reddit and Gab textual data to detect sexist messages and classify them into various categories of sexism.

2 Background

2.1 Task Setup

The data for this task was provided by SemEval Task 10 (Kirk et al., 2022). This labelled data set consists of 10,000 entries extracted from Gab along with 10,000 entries from Reddit. The dataset is labelled according to the specifications of the required classifier for subtask A, subtask B, and subtask C.

Subtask A requires a binary classifier to categorize sexist/non-sexist posts. Subtask B requires a four-class classification system that 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.

Additionally, two unlabelled data sets that each contains 1 million entries from Gab and Reddit were also provided, which we ultimately used to improve our system performance.

3 System Overview

3.1 Data Cleaning

Especially because the data was taken from Reddit and Gab, it was essential to clean the data to get consistent formatting. Specifically, all URL references were removed, hyphens and hashtags were replaced with spaces, all punctuation except apostrophes were removed, and all text was changed to lowercase. Also, many slang abbreviations were replaced by their expanded forms using the mapping provided the sms_slang_translator github reposi-

tory[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, doubling the number of minority samples in our training split. Results with back translation are specifically labelled 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. We generated enough samples for each minority class until the number of samples of each class was equal. Results with EDA are specifically labelled in the results section.

3.3 GloVe-Based Model

For our baseline model for Subtask A, we developed a GloVe-based logistic regression model. 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, so if the model's output was 0.35 or greater, the text would be classified as sexist. We also added dropout with a rate of 0.5 to help reduce overfitting.

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

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 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 relation-

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

ships. 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.

3.4.4 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 a SBERT-based model in hopes of addressing the overfitting we saw when applying other BERT-based models. For Subtask B, we fine-tuned a SBERT model using the same architecture we used for the other BERT-based models.

3.4.5 Ensembles

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

3.4.6 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 the 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 also used early-stopping to stop training if the training loss did not decrease over 30 epochs. We trained all our transformer-based models using the Adam optimizer and cross-entropy loss.

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

5 Results

5.1 Summary

As seen in Table 1, the best performing model for Subtask A was the model involving the weighted-average blending strategy, though the difference in F1 scores between the three best-performing models (BERT, ft-RoBERTa, and Weighted Average) was not very large. ft-RoBERTa had a slightly better score than the plain BERT model. However, it is clear that the worst-performing model was the GloVe Vector-based model. Additionally, while we hoped that back translation would improve the model's accuracy on the minority class and increase the F1 score, the data augmentation we performed did not lead to any improvements.

For Subtask B, we see in Table 2 that out of all the models only involving one transformer, pt-RoBERTa clearly sets itself from the pack, 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.

To prove the importance of data cleaning, we also 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 cleaned input text.

Our results in Table 3 show us that Ensemble 1 outperforms pt-RoBERTa as well as the Weighted Average model when it comes to 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 realtively 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 even DeBERTa. DeBERTa typically has stronger performance that RoBERTa on benchmark tasks, so we would argue that the significantly better results achieved by the pt-RoBERTa model can be primarily attributed to the more robust embeddings created after pre-training the RoBERTa embeddings on the unlabelled dataset.

For Subtasks B and C, creating ensemble-type models by concatenating embeddings from different transformer models also led to slight improvements. Ensemble 2 was middle-of-the-pack in Subtask B, but this was expected because Ensemble 2 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 also 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
ft-RoBERTa	0.525
pt-RoBERTa	0.62
Ensemble 2	0.555
Ensemble 1	0.622
Weighted Average: Ensemble 1 & pt-RoBERTa	0.624
Augmented Data: pt-RoBERTa & EDA	0.618
Uncleaned Data: 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: Ensemble 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

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 also essential in improving our results. Furthermore, we observed the importance of pre-training transformers models like RoBERTa on domain-specific text, as this led to substantial improvements in performance that even rivaled 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 is a promising approach to developing better representations of language, especially for the explainable detection of online sexism. Blending model outputs also led to slight performance improvements, but these improvements were small in comparison to the improvements seen from pre-training and concatenating.

In terms of future work, we believe that given the significant improvements shown the pre-trained RoBERTa model on the unlabelled data, our system could be further improved if we pre-trained more robust models, such as DeBERTa, on the unlabelled data. Additionally, while the data augmentation approaches we attempted failed to lead to better results, it could be beneficial to explore other augmentation strategies, since dealing with minority classes would be key in further improving the macro F1 score of our system.

Nonetheless, we believe these results show the huge potential of using pretrained transformer models coupled with concatenating embeddings in explainable textual detection, and would be of use for the future development of such models in the field.

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. We would also like to acknowledge Google Colaboratory for their free compute services.

References

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.

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

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).

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.

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.