Attention at SemEval-2023 Task 10: Explainable Detection of Online Sexism (EDOS)

Debashish Roy, Manish Shrivastava
IIIT Hyderabad, India
debashish.roy@students.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In this paper, we have worked on explainability and understanding of the decisions made by models in the form of classification tasks. The task is divided into 3 subtasks. The first task consists of determining Binary Sexism Detection. The second task describes the Category of Sexism. The third task describes a more Fine-grained Category of Sexism. Our work explores solving these tasks as a classification problem by fine-tuning transformer-based architecture. We have performed several experiments with our architecture, including combining multiple transformers, using domain adaptive pretraining on the unlabelled dataset provided by Reddit and Gab, Joint learning, and taking different layers of transformers as input to a classification head. Our system (with team name 'Attention') was able to achieve a macro F1 score of 0.839 for task A, 0.5835 macro F1 score for task B and 0.3356 macro F1 score for task C at the Codalab SemEval Competition. Later we improved the accuracy of Task B to 0.6228 and Task C to 0.3693 in the test set.

1 Introduction

Online sexism is a growing problem in this era, considering the large scale of online texts. Detecting online Sexism is an important NLP task, and has a lot of social impacts related to it. Sexism is defined as any abuse or negative sentiment that is directed towards women based on their gender or based on their gender combined with one or more other identity attributes (e.g. black women, trans women).

There are some prior works regarding the detection of sexist text (Rodríguez-Sánchez et al., 2022) at SemEval. Flagging what is sexist content and also explaining why it is sexist is very important aspects. We tried to focus on interpretability, trust, and understanding of the decisions made by models. Having an automated system that is capable of understanding, interpreting, and classifying sexist text will be a major step to make online sites safe.

Explainable Detection of Online Sexism (EDOS) at SemEval 2023 (Kirk et al., 2023), tries to solve this problem by exploring different systems that can categorize the dataset created with texts from Reddit and Gab.

The problem is divided into 3 subtasks. The first task is formulated as Binary Sexism Detection, where the main idea is to detect whether a text is sexist or not.

The second task describes the Category of Sexism. So, if the text is sexist, under which category it belongs? The categories are threats, derogation (treating someone as little worthy), animosity (a strong dislike or unfriendly feeling), and prejudiced discussions (an unfair feeling of dislike for a person or group because of race, sex, religion, etc). This makes this task a four-class classification problem given the text is sexist.

The third task describes the Fine-grained Vector of Sexist text. For posts that are sexist, systems were supposed to predict more fine-grained aspects of the category of sexism. Which can help in interpretability and understanding why the post is sexist on a much deeper level. The sexist posts are classified into an 11-class classification problem, given the text is sexist. The 11 classes are Descriptive attacks, Aggressive and emotive attacks, Casual use of gendered slurs, Profanities, and insults, Immutable gender differences and gender stereotypes, Supporting systemic discrimination against women as a group, Incitement, and encouragement of harm, Dehumanising attacks & overt sexual objectification, Supporting mistreatment of individual women, Backhanded gendered compliments, Threats of harm, Condescending explanations or unwelcome advice.

To solve this problem, we have fine-tuned transformer-based architecture. Our final submission is made up of two transformers: RoBERTa (Zhuang et al., 2021) and DeBERTa (He et al.,

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1972–1977
July 13-14, 2023 ©2023 Association for Computational Linguistics

2020). The embedding is then passed to the MLP. At the last layer, we have the number of classes. We have also explored Domain Adaptive Pre Training using masked language modeling on both of these transformers and used their weights for the classification task. The models remain the same for all the tasks except the last layer changes based on the number of classes.

Our system ranked 35th in sub-task 1, 50th in sub-task 2, and 44th in sub-task 3, out of 583 participants. The detailed comparison with the top team in the leaderboard is given below

Rank	Team Name	Task A F1-accuracy
1	PingAnLifeInsurance	0.8746
35	Attention (Our Team)	0.839

Table 1: Our System performance in comparison to the top team for task A

Rank	Team Name	Task B F1-accuracy
1	JUAGE	0.7326
50	Attention (Our Team)	0.5835

Table 2: Our System performance in comparison to the top team for task B

Rank	Team Name	Task C F1-accuracy
1	PALI	0.5606
44	Attention (Our Team)	0.3356

Table 3: Our System performance in comparison to the top team for task C

All of our code is made publicly available on Github[1]. The domain adaptive pretrained versions of RoBERTa and DeBERTa are also made publicly available at Huggingface [2]

2 Background

Transformer-based models such as RoBERTa and DeBERTa has shown outstanding performance on wide domains of NLP, including text classification. These models can capture a sort of structure in the hateful speech as well (Basile et al., 2019).

Domain Adaptive Pre Training is used in many domains of NLP, to make the model focus on the specific domain rather than a large variety of text it is trained on. Don't Stop Pre Training: Adapt Language Models to Domains and Tasks (Gururangan et al., 2020) talk about the increase in the performance of a model if it is pretrained on similar types of data.

SemEval-2021 Task 7: HaHackathon, Detecting and Rating Humor and Offense (Meaney et al., 2021) talks about the importance of pretraining and using ensembling methods to achieve high accuracy. It also described the importance of lexical features.

This SemEval task itself provided the dataset. The dataset contains text in English. The dataset contains 20,000 texts. Half of the text was taken from Gab and the other half from Reddit. The entries are then manually annotated by three trained annotator women. For each text, we have (a) label-sexist, describing whether the label is sexist or not, (b) label-category: given the text is sexist what is the category of sexism, in the case of non-sexist text it is none. (c) label-vector: it is more fine-grained details about the category of sexism.

The data is divided into 70% as training data, 10% as validation data, and 20% as test data. The distribution of labels for all three tasks is described in Table 1, Table 2, and Table 3. The major issue with the dataset for task C is less number of samples. All of the text is less than 64 words, so while generating tokens for transformers we have kept the max length to 64.

Classes	Instances
Sexist Text	3398
Non-Sexist Text	10602

Table 4: Class distribution for sub-task 1 train dataset.

Classes	Instances
Threats plan to harm, and incitement	310
Derogation	1590
Animosity	1165
Prejudiced discussions	333

Table 5: Class distribution for sub-task 2 train dataset.

3 System Overview

Mainly we have used Transformers, but for a baseline, we have used LSTM (Hochreiter and Schmidhuber, 1997). We trained the embeddings from scratch using the Pytorch embeddings layer. Then LSTM is followed by MLP at last, with a number of classes at the last layer.

Our main focus was on using Transformer. We have used Transformers with several variations. Based on this variation we try to come up with the best model.

[1] https://github.com/debashish05/Explainable_Detection_of_Online_Sexism

[2] https://huggingface.co/debashish-roy/

Classes	Instances
Threats of harm	56
Incitement and encouragement of harm	254
Descriptive attacks	717
Aggressive and emotive attacks	673
Dehumanising attacks & overt sexual objectification	200
Casual use of gendered slurs, profanities, and insults	637
Immutable gender differences and gender stereotypes	417
Backhanded gendered compliments	64
Condescending explanations or unwelcome advice	47
Supporting mistreatment of individual women	75
Supporting systemic discrimination against women as a group	258

Table 6: Class distribution for sub-task 3 train dataset.

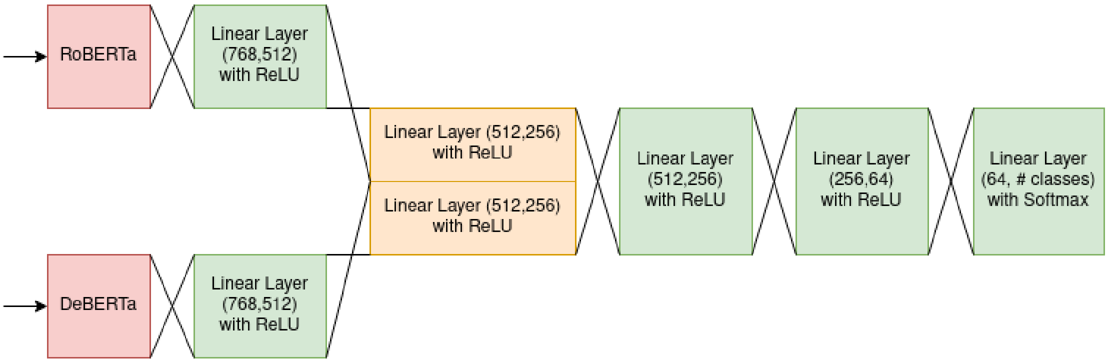
RoBERTa
Linear Layer (768,512) with ReLU
DeBERTa
Linear Layer (768,512) with ReLU
Linear Layer (512,256) with ReLU
Linear Layer (512,256) with ReLU
Linear Layer (512,256) with ReLU
Linear Layer (256,64) with ReLU
Linear Layer (64, # classes) with Softmax

Figure 1: Architecture of our system.

- **RoBERTa:** It is built on BERT masked language strategy, but excludes BERT's next-sentence pretraining objective, and training with much larger mini-batches and learning rates. RoBERTa BASE model consists of 12 transformer layers, 12 self-attention heads per layer, and a hidden size of 768.
- **DeBERTa:** We have used the BASE model which consists of 12 transformer layers, 12 self-attention heads per layer, and a hidden size of 768. It tries to improve RoBERTa by using two techniques: a disentangled attention mechanism and an enhanced mask decoder.
- **Domain Adaptive Pretraining:** We try to tailor a pretrained model to the domain of a target task. To do so we do masked language modeling over the text related to the domain.

With these two transformers, we have experimented with the architecture. The different architecture is as follows:

1. **RoBERTa/DeBERTa (last layer) + MLP:** We have taken the last layer of the RoBERTa the base model followed by some linear layers and fine-tuned over the training data. The same setup has been used with DeBERTa as well.
2. **RoBERTa/DeBERTa (average of all layers) + MLP:** Instead of taking the last layer only, here we have taken the average of all the layers in the RoBERTa base model. Which are then followed by some linear layers and fine-tuned over the training data. The same setup has been used with DeBERTa as well.
3. **RoBERTa and DeBERTa concatenation of the last layer + MLP:** Output from the last layers of RoBERTa and DeBERTa were concatenated. Then this is followed by MLP. The last layer consists of neurons with a number of classes.
4. **RoBERTa and DeBERTa combined (before concatenation of two transformers they are passed through a linear layer):** Output from the last layers of RoBERTa and DeBERTa were passed through the different linear layers and then concatenated with each other. Then this is followed by some more fully connected layers. The last layer consists of neurons with a number of classes. The model is presented in Fig 1.
5. **Joint Learning for Task B:** For task B, earlier we were only taking the text that is only sexist. But now we will use all the data. And instead

		Task A		Task B		Task C	
S.No	Experiment	Val	Test	Val	Test	Val	Test
1	LSTM + MLP	0.70	0.70	0.34	0.38	0.17	0.17
2	RoBERTa last layer + MLP	82.90	81.47	59.74	58.29	59.70	33.48
3	DeBERTa last layer + MLP	83.24	81.99	61.05	59.07	29.85	30.09
4	RoBERTa avg of all layers + MLP	79.85	78.65	34.83	45.71	26.76	21.58
5	DeBERTa avg of all layer + MLP	80.40	78.62	59.79	55.15	22.11	21.61
6	RoBERTa+DeBERTa + Embeddings of these two are concatenated. Before concatenating these are passed through one linear layer & then followed by MLP	82.96	82.26	62.47	**62.28**	33.99	31.61
7	Experiment 5 + Domain Adaptive Pre Training with unlabelled text	**84.27**	82.66	63.48	60.85	**35.30**	**36.93**
8	Joint Learning for task B using task (A and B's data), the last layer is of 5 neurons with labels from task B and one non-sexist text	NA	NA	54.57	53.29	NA	NA
9	RoBERTa+DeBERTa + Domain Adaptive Pretraining. Embedding of RoBERTa and DeBERTa are concatenated, no linear layer used before concatenation here (unlike experiment 5) and then passed through MLP)	84.13	**83.9**	**67.00**	58.35	34.33	33.56

Table 7: Macro F1 scores of experiments.

of treating only the label category only, we are adding another class which implies the text is non-sexist. This makes the problem a 5-class classification problem, rather than a 4-class with the increase in a large amount of data. The idea is to generalize the model and give more text to train on. At the inference time since we are supposed to output only from the 4 classes, we will give the class with the largest probability as output. But if the predicted class is not sexist text, then return the second largest probability class, since we already know the text is sexist text.

6. **Domain Adaptive Pretrained RoBERTa and DeBERTa combined (before concatenation of two transformers they are passed through a linear layer):** Output from the last layers of Domain Adaptive Pretrained RoBERTa and DeBERTa were passed through the different linear layers and then concatenated with each other. The transformers are trained on the 2M unlabelled texts provided in the task itself. Then this is followed by some more fully connected layers. The last layer consists of neurons with a number of classes. The model is presented in Fig 1 with the change of Domain Adaptive Transformers.

All these models are used for all the tasks with a change in the last layer with the number of classes.

4 Experimental setup

The dataset provided was already divided into train, dev, and test. We have experimented with some preprocessing of the text, which includes converting emoji to text, converting all the characters to lowercase, converting the link to <link> tag, removing punctuations, removing words with numbers, removing stop words, and using WordNet Lemmatizer. But after the cleaning step accuracy was low compared to the non-cleaned text. A probable reason is with the transformer tokenization mechanism is more useful than the naive tokenization mentioned.

For the hyperparameter, we have taken the number of epochs on the training set as 20, the Learning rate is 1e-5, the loss function used is Cross Entropy Loss and the optimizer used in AdamW (Loshchilov and Hutter, 2019). We have used the Macro F1 score as a measure of evaluation for all three tasks.

5 Results

Results for different experiments are mentioned in Table 7. Our official submission is experiment number 8. Later we experimented with 5 and 6 after the deadline. We have achieved a rank of 35th in task A, 50th in task B and 44th in task C.

With Domain Adaptive Pretraining we have seen reduced perplexity scores of these language models, which measure the amount of randomness in our model or how confused the model is, so the less the better.

The model has been tested across two datasets, EXIST2021 (Rodríguez-Sánchez et al., 2022), Automatic Detection of Sexist Statements Commonly Used at the Workplace(Grosz and Conde-Cespedes, 2020). Both datasets consist of sexist and non-

Predicted ->	Threats & plans to harm	Derogation	Animosity	Prejudiced
Threats & plans to harm (Actual)	62	11	12	4
Derogation (Actual)	16	303	**107**	28
Animosity (Actual)	11	**112**	199	11
Prejudiced discussions (Actual)	9	27	12	46

Table 8: Confusion Matrix for Task B

Predicted->	Not sexist	Sexist
Not sexist (Actual)	2909	121
Sexist (Actual)	**346**	624

Table 9: Confusion Matrix for Task A

sexist tweets and the data is taken from Twitter and Gab. We have used the testing split of these two datasets without training on the train split and monitored the performance. The results are mentioned in Table 10.

Dataset Name	**Paper**	**Our Model**
EXIST 2021	0.83	0.56
Sexist-workplace statement	0.88	0.57

Table 10: Comparison of F1 score of Model (Without Training on the train split of the mentioned dataset) with other Dataset

6 Error Analysis

These error analyses are performed on experiment no 6.

6.1 Task A

With the given confusion matrix in Table 9 test data, we can infer that the model will predict the right label 96% of the time if the text is not sexist. Issues come when the tweet is sexist. Which is a major problem. Approximately 35% of the time the model is classifying sexist text as non-sexist, which is a dangerous thing.

For example, this text: “Don’t run away like a little jewish girl.” and “woman here Opinion discarded.” are treated as not sexist., but they are sexist.

There is a total of 346 texts that are sexist but classified as not sexist. Applying some sort of rule-based system where if this word comes then it will be sexist will not work. Because we need to understand the whole context. Most of the time there is no hidden meaning in the sentence.

One of the main reasons for poor performance in predicting text is sexist, which is less number of samples that are sexist. If we can get more samples that are sexist, our model can try to learn the pattern among them and perform better. We should have tried using distil version of these transformers, as they have less number of weights, which can be trained with fewer samples. And the different distil versions of the transformer can be used, as the number of samples to train on keeps on getting smaller.

6.2 Task B

The confusion matrix for task B is provided in Table 8. The model is confused many times between Derogation and Animosity. Approximately 25% of the time when the text is derogation it is predicting animosity. And nearly 33% time when the text is animosity it is treating it as derogation. So the model is not able to clearly distinguish between these classes. Adding more data can help us solve this issue.

7 Conclusion

Domain Adaptive Pre-Training helps us get some extra accuracy numbers in all three sub-tasks. Combining RoBERTa and DeBERTa makes the model big but also brings more variety of aspects to the model.

Although the macro F1 accuracy is quite good, the data instances where it is failing are very costly in the case of task A. In the case of binary classification of sexism, a false negative is high, i.e. a sexist text is sometimes treated as non-sexist. Considering the case of online social media, taking a post down for some time even if it is not sexist is less harmful than keeping a sexist text on social platforms. In general, the model is failing to predict sexist texts. A manual inspection of the failed text of this type suggested that there is a large scope for improvement in the model.

Another weakness of the transformer model is to deploy it in real-world case scenarios where the data is real-time and the volume is high. The texts need to process at high speed. Although transformers can parallelize the computation, the still amount of computation needed is high and we may need to devise some hybrid approach for that.

Despite that, the model shows very promising results, where the number of samples to train on is large. To further improve the accuracy we can use more annotated data from people. Although this will be costly, it will be a one-time effort.

The model that we submitted for Task B was not able to generalize well. The macro F1 dropped from 67% to 58.35%. This can be due to overfitting or dissimilarity between the distribution of the test and train set.

One of the ideas that we look forward to working with is to predict pseudo labels for unlabelled tasks. For example, if our best model is sure that the label belongs to a particular task with more than 99% probability then use that data instance as training data for the model. In this way, we can increase the data to train for our model.

Another idea that can be looked for is to generate text of a similar type (Bhatt and Shrivastava, 2022) using decoder-based transformers. For example, we can take text which is less in number and can train on causal language objectives over the labeled samples. This results in more data samples to train on.

8 Acknowledgments

I want to thank Sagar Joshi for helping me understand the task. I would also like to thank Tathagata Raha and Vijayasaradhi Indurthi for giving possible directions to explore the problem. We also want to thank the organizers of this task

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Sahil Bhatt and Manish Shrivastava. 2022. Tesla at SemEval-2022 task 4: Patronizing and condescending language detection using transformer-based models with data augmentation. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 394–399, Seattle, United States. Association for Computational Linguistics.

Dylan Grosz and Patricia Conde-Cespedes. 2020. Automatic detection of sexist statements commonly used at the workplace.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Hannah Rose Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. SemEval-2023 Task 10: Explainable Detection of Online Sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, Laura Plaza, Adrián Mendieta-Aragón, Guillermo Marco-Remón, Maryna Makeienko, María Plaza, Julio Gonzalo, Damiano Spina, and Paolo Rosso. 2022. Overview of exist 2022: sexism identification in social networks. *Procesamiento de Lenguaje Natural*, 69:229–240.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.