# DAP-LeR-DAug: Techniques for enhanced Online Sexism Detection

**Jayant Panwar**
LTRC, International Institute of
Information Technology, Hyderabad
jayant.panwar@research.iiit.ac.in

**Radhika Mamidi**
LTRC, International Institute of
Information Technology, Hyderabad
radhika.mamidi@iiit.ac.in

## Abstract

The swift surge of digital communication on social media platforms has brought about an increase in hate speech online, especially sexism. Such content can have devastating effects on the psychological well-being of the users, and it becomes imperative to design automated systems that can identify and flag such harmful content. Human moderation alone is inadequate to manage the volume of content, necessitating efficient technological solutions. In this study, we explore the performance of different modern techniques on Bert-based models for detecting sexist text. We explore four such techniques, namely, Domain Adaptive Pre-training (DAP), Learning Rate Scheduling (LeR), Data Augmentation (DAug), and an ensemble of all three. The results show that each technique improves performance differently on each task due to their different approaches, which may be suited to a certain problem more. The ensemble model performs the best in all three subtasks. These models are trained on a Semeval'23 shared task dataset, which includes both sexist and non-sexist texts. All in all, this study explores the potential of DAP-LeR-DAug techniques in detecting sexist content. The results of this study highlight the strengths and weaknesses of the three different techniques with respect to each subtask. The results of this study will be useful for researchers and developers interested in developing systems for identifying and flagging online hate speech.

## 1 Introduction

Text classification tasks have been around for a long time, and so has online hate speech. Posting without any consequences is stimulus enough for people to be overly hurtful in their comments and be ignorant of others' feelings. Some might just do it to "troll" someone, some out of pure hatred, and some for channelling their inner frustration. With time, the presence of hate speech prevalent online increases too, and all the major social platforms nowadays are trying to find ways to flag and curb it. Sexism has been present since before the Internet, and thus, there is no surprise that it is one of the most used forms of hate speech online today.

In our study, we aim to develop an automated system that can detect and classify sexism using different techniques, namely, Domain Adaptive Pre-training (DAP), Learning Rate Scheduling (LeR), and Data Augmentation (DAug). For the same, we use the dataset shared by the task organizers of Task-10 of SemEval-2023 (Kirk et al., 2023). The dataset contains data for the following three subtasks:

- **Subtask-A**: binary classification task in which systems must figure out whether a certain piece of text is sexist or not
- **Subtask-B**: systems must classify the sexist piece of content into its appropriate class from the given 4 classes
- **Subtask-C**: systems must accurately classify the sexist text into one of the listed 11 classes

Further details regarding sexism category names can be seen in Figure-1. As visible from the definitions discussed above, the complexity of the task increases with each level. We go from dealing with a simple binary classification task to an 11-class multi-classification problem. This is precisely why we tackle the task with three unique techniques and an ensemble of all three combined techniques. For implementing the these techniques we use three BERT-based models, namely, RoBERTa, HateBERT, and BERTweet. The best model for each task is the ensemble model. This is because each of the three techniques is beneficial in its own way and using an ensemble model makes sure that the advantages of all three techniques are utilized simultaneously.

DAP boosts the scores most for Task-A, LeR for Task-B, and DAug for Task-C thanks to their

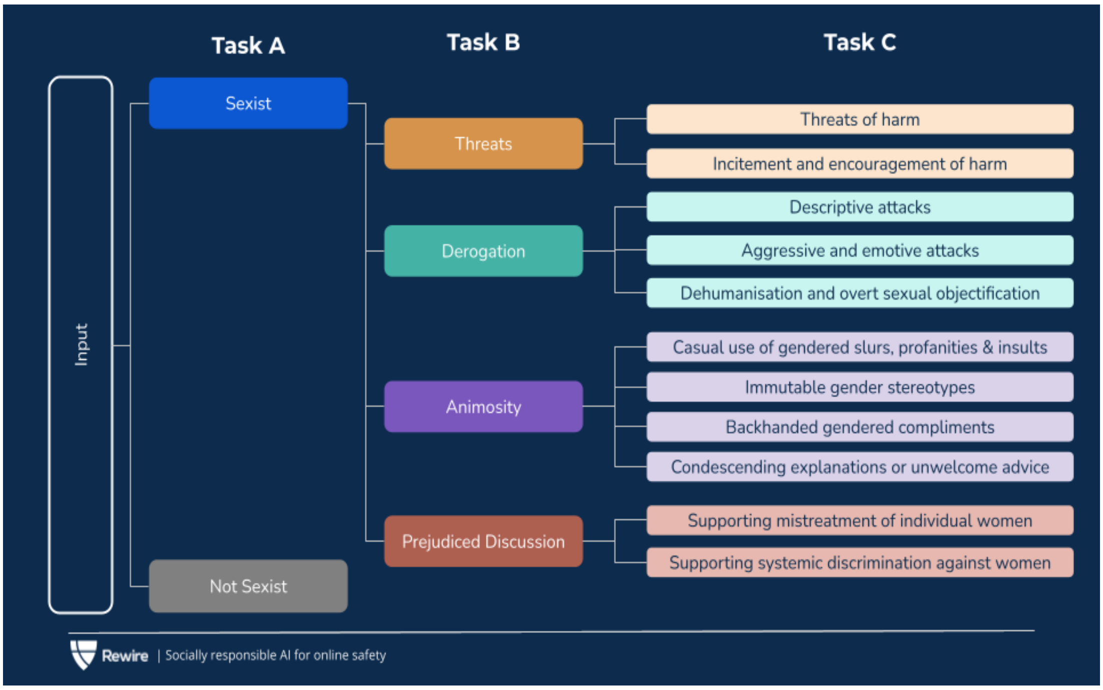
Task A
Task B
Task C
Input
Sexist
Not Sexist
Threats
Derogation
Animosity
Prejudiced Discussion
Threats of harm
Incitement and encouragement of harm
Descriptive attacks
Aggressive and emotive attacks
Dehumanisation and overt sexual objectification
Casual use of gendered slurs, profanities & insults
Immutable gender stereotypes
Backhanded gendered compliments
Condescending explanations or unwelcome advice
Supporting mistreatment of individual women
Supporting systemic discrimination against women
Rewire | Socially responsible AI for online safety

Figure 1: The shared task categories. Image adopted from the organizers of Kirk et al. (2023)

unique approach which caters to the respective subtasks. As a result of their ensemble model, our system is comfortably able to beat the best baseline model from the original task paper (Kirk et al., 2023).

## 2 Related Work

Detection of online sexism has been a task that many researchers have worked on over the past many years. Some showed how we can use both conventional and deep learning approaches to identify various forms of sexism in a multi-lingual setting (Rodríguez-Sánchez et al., 2020) while others have created their own datasets to examine different forms of sexist content prevalent nowadays (see Parikh et al. (2019), Samory et al. (2021)). In our study, however, we stick to the dataset for the EDOS task, so we can compare the performance of our systems with other major baselines and top-ranked systems.

There have also been important efforts when it comes to adapting the models to a certain domain. In our case that is adapting BERT-based models (for BERT see Devlin et al. (2019)) to hate speech, sexism to be specific. The authors of Gururangan et al. (2020) have shown how models can improve in performance by adapting a certain domain. For this, first, the model is trained on a large unlabelled dataset and then fine-tuned on the smaller labelled dataset, which fits in line with our case. This is where the motivation of the DAP technique comes from.

Zhao et al. (2022) showcased how important it is for the learning rates to adapt to the task so as to achieve best performance in classification tasks. This helps in faster convergence while training which ultimately leads to better results. Similarly, Data augmentation has always been shown to improve performance generally in text classification tasks. For instance, the EDA framework (Wei et al., 2019), where simple updates like synonym replacement, random insertion, random swap, and random deletion improved classification performance by a good extent. Likewise, there are other data augmentation approaches such as stochastic replacement of words in the sentence (Kobayashi, 2018), and using Pre-trained Language Models to get diverse and semantically correct text samples (Anaby-Tavor et al., 2019). In our study, we choose to stick with the simpler EDA approach.

## 3 System Overview

The system for our study can be broken up into 5 different parts. Firstly, we have the Bert-based models as it is, i.e., we do not employ any techniques on them. Then, we have got our DAP-LeR-DAug individual models to understand which technique works best in which scenarios. Finally, we wrap it all up by having an X model, which is basically an ensemble of all the three techniques discussed.

### 3.1 Baselines

As mentioned earlier, we will have three BERT-based models as our baselines, namely, RoBERTa, HateBERT, and BERTweet. RoBERTa (Liu et al., 2019) is an advanced BERT-based pretraining approach that optimizes and enhances performance on various natural language understanding tasks through extensive training with larger batches and more data, resulting in improved language representations. HateBERT (Caselli et al., 2021), on the other hand, is a specialized transformer-based model tailored for detecting hate speech in text, designed to provide accurate identification of offensive content through fine-tuned representations and focused training on hate speech data. Finally, BERTweet (Nguyen et al., 2020) is an adaptation of the BERT model specifically designed for processing and understanding text from social media platforms like Twitter, offering improved performance on tasks involving informal language, hashtags, mentions, and other characteristics unique to Twitter discourse.

It is evident from the description of the selected BERT-based models as to why they are apt for our experiment which is heavily focused on natural language understanding and dealing with sexism, a form of hate speech. For the baseline stage, we use them as they are and fine-tune them on our shared task dataset. Then we evaluate how they perform.

### 3.2 DAP

DAP refers to Domain Adaptive Pre-training. The organizers of the task (Kirk et al., 2023) had also provided a dataset of 2 million unlabelled posts from Gab and Reddit. We utilize this enormous dataset with the Masked Language Modelling (MLM) objective as we believe this pairing would hold the most promise for enhancing the performance of our BERT-based models in classifying sexist content. By being subjected to diverse and extensive linguistic contexts from the unlabelled dataset during MLM pretraining, the models gain a robust understanding of general language patterns and nuances. This enriched linguistic foundation forms the cornerstone for improved comprehension of text, enabling the models to capture subtle linguistic cues and contextual variations inherent in sexist content.

During fine-tuning with labelled data, the models' already adept language representations are seamlessly adapted to the specific domain of sexism detection. This dual-stage process harmonizes its universal language understanding with domain-specific features, resulting in heightened discriminatory power to accurately identify and classify sexist text instances. The fusion of pretraining's broad language expertise and fine-tuning's task-specific tailoring equips the models with a well-rounded ability to identify and categorize nuanced and varied forms of sexist content across the different classes of sexist content.

### 3.3 LeR

LeR refers to Learning Rate Scheduling. Learning rate scheduling enhances model performance by dynamically adjusting the step size during training. This technique accelerates convergence by initially allowing larger parameter updates, ensuring quicker progress towards the optimal solution. As training advances, the learning rate is reduced, stabilizing optimization and preventing overshooting. By navigating the loss landscape more effectively, learning rate scheduling helps evade local minima and improves generalization by mitigating noise fitting. Although this technique does not contribute linguistically in terms of word embeddings, contextual understanding of the domain, etc., it can still prove to be very important.

This technique is particularly valuable for stabilizing training with large batch sizes, adapting to data characteristics, and achieving fine-tuned results in transfer learning scenarios. In essence, learning rate scheduling fine-tunes the learning process itself, fostering quicker convergence, robustness, and overall improved model performance.

### 3.4 DAug

DAug implies Data Augmentation. The dataset we have is highly imbalanced for each subtask. For example, the majority class in tasks A and B has more than 3 times the number of data instances as compared to the minority class. For task C, the case is even worse. There are minority classes with not even 100 instances while some majority classes have more than 700 instances. A dataset like this can make the best of classifying models biased towards the majority class. There are various different techniques to counter that, and Data augmentation is certainly one of them. It concerns itself with creating new data for classes with lim-

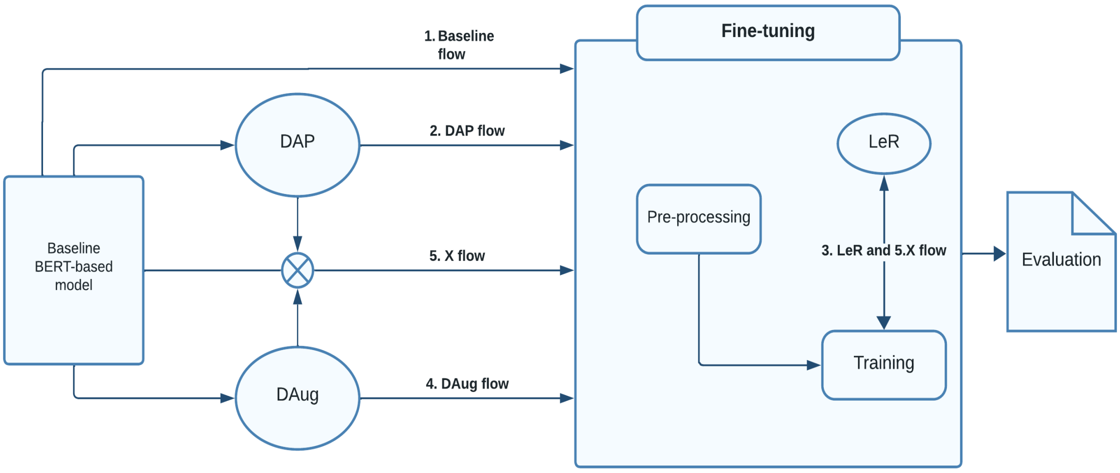

Figure 2: System Architecture

ited data available. It can significantly enhance a dataset with limited sexist posts by generating diverse variations of existing examples. Techniques such as synonym replacement, paraphrasing, and introducing minor textual perturbations help create additional instances of sexist content. By simulating different linguistic expressions and contexts, data augmentation enriches the dataset, which can, in turn, improve the model generalization and performance, even when original sexist instances are sparse.

As discussed in earlier sections, we make use of a similar approach as taken by authors of Easy Data Augmentation (EDA) (Wei et al., 2019). By introducing synonym replacement, random insertion, random swap, and random deletion to the text, the EDA framework generates diverse instances of the original text. This augmented data enriches the training dataset, improving model generalization and performance. EDA is demonstrated to be remarkably effective across various text classification tasks, showcasing its ability to alleviate the challenges posed by limited training data and contributing to more robust and accurate text classification models. This is why the EDA approach will be helpful for us, for all three subtasks. We discuss the exact setup details in the coming section.

### 3.5 X

The final or the X part of our system is basically a combination or an ensemble of all the three unique techniques we have discussed thus far. The ensemble capitalizes on the complementary strengths of each technique, effectively navigating linguistic complexities through pre-trained domain understanding, fine-tuning with task-specific context, and enriched data diversity. This holistic approach promotes greater robustness to nuances in sexist content and addresses challenges posed by limited labelled data. Ideally, this should outperform the individual technique models and ultimately lead to the best performance when it comes to classifying sexist content.

## 4 Experimental Setup

We discuss our experimental setup (see Figure-2) in two forms: technique and fine-tuning specific. Fine-tuning specific setup is applied to all the five models irrespective of the technique being used. We discuss the LeR setup in technique specific section, but we must remember that it is applied only while fine-tuning.

### 4.1 Technique specific

As discussed beforehand, one of the major problems we have is the class imbalance in the dataset. For that, we use the Data Augmentation technique. But, in order to do justice to other techniques so as not to make their classifiers biased toward the majority class, we had to consider other approaches for them like Undersampling and Oversampling. In Undersampling, we remove a certain number of data instances from the majority class to make sure the classes are more or less balanced. However, in Oversampling, we do the opposite. We replicate data instances of the minority class until we have achieved balance among all the classes in the dataset. Undersampling has been shown to perform better for this shared task (Panwar and Mamidi, 2023), while Oversampling has been shown to perform worse than using the dataset as it is, i.e., imbalanced. Therefore, for the model variations that do not include data augmentation, i.e., Baseline, DAP, and LeR, we use Undersampling to balance the dataset.

Regarding the setup for Data Augmentation models, we implement the EDA framework (Wei et al., 2019), as explained earlier. For generating data, we decided to choose RoBERTa as it gave semantically closer data to the actual data when compared with the data generated by HateBERT and BERTweet. We limit the augmentation probability to 0.3 as above this threshold, the system generates very noisy data, which can lead to loss of semantics and an overall reduction in the performance of the models.

For the Domain Adaptive Pre-training technique, we use the Masked Language Modelling objective. The first and foremost step is to obviously use the correct tokenizers and pre-process tokens that may not contribute semantically a lot to the sentence. For example, tokens like [USER], [HASHTAGS], [URLS], [MENTIONS], etc can be removed to improve efficiency and accuracy. Then we create the masked sentences, and we do so by randomly masking a certain percentage of the sentence. Then, the model learns by predicting the masked tokens based on the surrounding context. The goal is to minimize the loss between the actual masked and predicted tokens. By gaining a better idea of the contextual relationships from posts on sexist forums, the model should ideally perform better than without DAP.

For the Learning Rate scheduling models, we experimented primarily with four different types of LRs: Step decay, Exponential decay, Cosine annealing, and One-Cycle LR. They performed more or less similarly, with the only difference being when it came to the X or the ensemble model. In that case, cosine annealing edges out other approaches and this may be due to the fact that the X model has a lot going underneath the layers. Not only does it have more contextual embeddings thanks to DAP, but it also has more data to work with because of DAug. These rising complexities require complex learning rate scheduling policies like that of Cosine Annealing.

### 4.2 Fine-tuning specific

This part is very intuitive. We split the dataset into 85:15 ratio with the former used for training and the latter for validation. The authors of the task have provided separate data for testing and we believe it would be better to test our models on that to compare how we stand with task paper baselines and other top-ranked teams. During the training phase, first, we do simple pre-processing. Most of the pre-processing is handled comfortably with the appropriate tokenizers of the different models we have considered. However, we take care extra care on our own end to remove tokens that do not contribute semantically to the system. For example, hashtags, emojis, noisy tokens like "heyyyyyy", "yolooooooo", etc. For training our classifiers, we set epochs as 10 and batch size as 16. After training the classifiers, we proceed to evaluate them.

## 5 Results

For evaluation, we make use of macro average F-1 scores. This helps us to compare the performance of our approach with that of the task paper baselines and other top-ranked teams. A major reason for adopting macro average F-1 scores could be that during evaluation it treats each class of the dataset appropriately. This is very beneficial in cases, where the dataset is highly imbalanced, like in our case.

From the results in Table-1, we can see that the ensemble model with RoBERTa as baseline performed the best on the evaluation test. There can be different reasons for that, but the primary reason has to be the architecture of RoBERTa and the fact that we have used RoBERTa-base in our Data Augmentation phase. Models like HateBERT and BERTweet have a good understanding of hate speech beforehand, thanks to their architecture and pre-training. It is possible that techniques like DAP and DAug did not help these models as much as they helped RoBERTa since they have been exposed to a wide variety of hate speech data and our techniques did not increase their contextual understanding or vocabulary a whole lot.

Another important point to note is that the X model performs the best for each baseline. All three techniques that we decided upon, when employed together, can cause the model to perform best. It is also intuitive as the X model is one which has been pre-trained heavily on about 2 million posts for adapting the sexist content domain,

| *Model* | *Task-A: 2 class* | *Task-B: 4 class* | *Task-C: 11 class* |
|---|---|---|---|
| RoBERTa-base | 83.22 | 59.77 | 34.01 |
| RoBERTa-DAP | 84.67 | 62.23 | 36.44 |
| RoBERTa-LeR | 82.45 | 63.97 | 38.21 |
| RoBERTa-DAug | 83.59 | 63.87 | 39.89 |
| **RoBERTa-X** | **85.09** | **65.89** | **40.23** |
| HateBERT-base | 82.13 | 60.56 | 33.84 |
| HateBERT-DAP | 82.55 | 62.23 | 35.19 |
| HateBERT-LeR | 82.14 | 64.12 | 37.77 |
| HateBERT-DAug | 82.34 | 64.01 | 38.09 |
| HateBERT-X | 82.78 | 64.53 | 38.34 |
| BERTweet-base | 84.01 | 61.12 | 30.01 |
| BERTweet-DAP | 84.33 | 62.01 | 32.71 |
| BERTweet-LeR | 84.03 | 62.89 | 34.66 |
| BERTweet-DAug | 84.12 | 62.88 | 34.81 |
| BERTweet-X | 84.39 | 63.45 | 35.22 |

Table 1: Macro Avg. F-1 Scores of Classifiers on all subtasks

has got augmented data with minority classes also being represented adequately, and finally, can train optimally thanks to the learning rate scheduling technique. The three techniques complement each other and bring out the best when used together.

We really notice the impact of individual techniques when we look at the results task-wise. For task A, we can see that the DAP technique improves the score the most on the baseline. This is intuitive as well because for a simple binary classification subtask, having more embeddings and wider vocabulary to work with makes it even easier for the model to figure out if the content is plain sexist or not. The LeR approach works best with increasing complexities of the task. It works better for Tasks B and C than it does for Task A. The effect of optimal convergence is noticed more easily when there are more classes involved in the task. It performs the best for task B and is also good for task C. It is not that its performance drops in task C but that Data augmentation works too well for task C and it outshines the LeR technique. We have established multiple times in this study that the dataset is imbalanced, and this imbalance increases with the increasing complexity of the task. Undersampling can only work so well when we have to deal with 11 classes in task C, and the majority of them are very under-represented. This is where Data Augmentation comes in handy. By creating more data instances for the minority classes, we are able to give the model more data to work with and thus increase its performance in classification.

| *Model* | *Task A* | *Task B* | *Task C* |
|---|---|---|---|
| Best Baseline | 82.35 | 59.26 | 31.71 |
| Top-ranked | 87.46 | 73.26 | 56.06 |
| RoBERTa-X | 85.09 | 65.89 | 40.23 |

Table 2: Comparison of the performances of the Best Baseline model in Task paper, the top-ranked systems for each subtask, and our best performing model: RoBERTa-X

Lastly, we compare our best-performing model, i.e., RoBERTa-X, with the best baseline model of the task paper (Kirk et al., 2023) and the top-ranked systems for each subtask. We are able to comfortably beat the best baseline model in each of the subtasks, thanks to the ensemble of our effective techniques. We were not able to beat the top-ranked system in any subtask, even though we came close. However, we must note that for this shared task, no single approach was the top-ranked among all the three subtasks. The top-ranked system score for each subtask in Table-2 is from a different team. We were able to create a single approach that at least beat the best baseline. Comparing our scores with the task leaderboard, we would stand in the top 30% submissions in task A, top 25% submissions in task B, and top 40% submissions in task C.

## 6 Conclusion

Through this study, we were able to explore the effectiveness of the DAP-LeR-DAug techniques when it comes to classifying hate speech in the form of sexism. We were able to demonstrate that each technique works well with a specific subtask, and when employed together in the form of an ensemble, they perform the best, irrespective of the BERT-based model being used. This goes on to show that the scores achieved were not coincidental, and the techniques indeed complement each other in a good way.

Although the DAP-LeR-DAug techniques do not perform the best for any specific subtask when compared with top-ranked systems, it should be pointed out that they do surpass the scores achieved by the best baseline model in the original task paper quite comfortably. Nevertheless, there are a lot of ways to improve upon the scores achieved, which we discuss in the next section.

## Limitations

Like any other research study, ours, too, is filled with limitations. Overcoming some of these would directly result in better scores for each subtask while some others may increase the training time but nonetheless will improve the performance of the models.

First of all, we have used only the base versions of the BERT-based models. If not for the restraint of computational resources, we could have used the large, extra-large, versions of the baseline models. The larger vocabulary and increased number of parameters would directly help to achieve better scores in all three subtasks.

Another way to improve our performance could be using more data for DAP. The suggestion is indeed greedy but will improve the performance nonetheless. Similarly, we could experiment with other forms of hyperparameter tuning apart from LeR alone. Some of them could be optimizing the dropout rate, loss functions, weight decay, and activation functions. The impact of tuning these may not be very large but it will optimize our performance.

We can also try to use different data augmentation approaches. In our study, we have only used the EDA approach but there are more complex ways to augment data. For example, Backtranslation, in which we translate the English sentence to a certain language and then back to English. This is an easy and effective way to generate more samples for under-represented classes and ultimately balance the dataset.

Lastly, we can try to improve our pre-processing stage as well. In our pre-processing stage, we get rid of all the emojis and hashtags but they have been shown to improve the performance of classification tasks (Eisner et al., 2016). They can be converted to vector embeddings and then combined with our word embeddings to form custom vector embeddings. This will directly improve the performance of our model as emojis are used a lot on social platforms nowadays and they contribute to the context and semantics of the text.

## References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2019. Not enough data? deep learning to the rescue! *Computing Research Repository*, arXiv:1911.03118. Version 2.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. In *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media*, pages 48–54, Austin, TX, USA. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasovi'c, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *Computing Research Repository*, arXiv:2004.10964. Version 3.

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.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:1907.11692. Version 1.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Jayant Panwar and Radhika Mamidi. 2023. PanwarJayant at SemEval-2023 task 10: Exploring the effectiveness of conventional machine learning techniques for online sexism detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1531–1536, Toronto, Canada. Association for Computational Linguistics.

Pulkit Parikh, Harika Abburi, Pinkesh Badjatiya, Radhika Krishnan, Niyati Chhaya, Manish Gupta, and Vasudeva Varma. 2019. Multi-label categorization of accounts of sexism using a neural framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1642–1652, Hong Kong, China. Association for Computational Linguistics.

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.

Mattia Samory, Indira Sen, Julian Kohne, Fabian Floeck, and Claudia Wagner. 2021. "call me sexist, but...": Revisiting sexism detection using psychological scales and adversarial samples. *Computing Research Repository*, arXiv:2004.12764. Version 2.

Jason Wei, Kaiqing Zou, Mingxuan Chen, and Lei Li. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *Computing Research Repository*, arXiv:1901.11196. Version 2.

Kailin Zhao, Xiaolong Jin, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2022. MetaSLRCL: A self-adaptive learning rate and curriculum learning based framework for few-shot text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2065–2074, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.