# Online Sexism detection with ELECTRA

**Mahdi Amiri** **Behrooz Nikandish** **Lynne Zhang**

University of Groningen
Groningen, The Netherlands
{m.amiri.2, b.nikandish, l.zhang.50}@student.rug.nl

## Abstract

This paper describes the methods that we used to submit the system for the shared task SemEval 2023, task 10: Explainable Detection of Online Sexism (EDOS). To address subtask A, we developed six models in three different paradigms, namely further pretraining, typical finetuning, and prompt-based learning. Firstly, we built three classifiers by finetuning ELECTRA-small, ELECTRA-base, and ELECTRA-large from the standard pretrained language models. Then, we further pretrained ELECTRA-small using 2 million entries of domain-specific unlabeled data. Finally, we employed T5-base and T5-large models in the prompting setting. During the development phase, the fine-tuned ELECTRA-large achieved the best performance with an F1-score of 0.84. The T5-base model performed the second best with an F1-score of 0.82. Additionally, we discovered that utilizing techniques such as data augmentation, preprocessing, oversampling, and extension with the CallMeSexistBut dataset does not improve performance.

## 1 Introduction

Social media has grown to be a significant and influential medium of communication for most people. Many violent, hateful, and sexist words and sentences mainly target women in the massive volume of texts and messages posted daily on social media. It significantly restricts women's ability to participate equally in public and political life and impacts their freedom of expression (Barker and Jurasz, 2019). Although the internet was created to promote free expression, social media platforms today have a responsibility to safeguard their users from harmful content by monitoring and regulating it. Consequently, removing such hateful content is becoming a higher priority for social media networks.

Although it is possible to determine the key features to spot hate speech on social media (Clarke and Grieve, 2017), discrimination against women is a complicated issue that incorporates cultural or traditional practices. Previous studies have attempted to detect misogyny and sexism in social media using different approaches. While some studies utilized classical machine learning techniques (Nina-Alcocer, 2018, Saleem et al., 2017), other attempts have used deep neural networks, such as Convolutional Neural Networks (CNNs) (Zhang and Luo, 2018, Park and Fung, 2017), Long Short-Term Memory Networks (LSTM) (Badjatiya et al., 2017), and bidirectional LSTM (Anzovino et al., 2018). More recently, transformer-based (Vaswani et al., 2017) pre-trained language models (PLM) have been used to address the task. While some studies used BERT (Butt et al., 2021), mBERT (Schütz et al., 2021), RoBERTa (Paula et al., 2021), or GPT2 (Paula et al., 2021), some attempts have been made to combine transformer-based models and classical machine learning approaches (Lopez-Lopez et al., 2021) to classify sexism texts.

This work is inspired by the SemEval 2023 shared task - task 10, Explainable Detection of Online Sexism (EDOS), which is proposed at Codalab[1]. In this paper, we detail the different methods employed to tackle sub-task A of the shared task to identify sexist hate speech on social media networks. Specifically, we concentrate on a labeled dataset consisting of 14,000 entries from Gab and Reddit. The unique aspect of our approach is the utilization of various state-of-the-art transformer-based (Vaswani et al., 2017) PLMs in different paradigms including further pretraining, finetuning, and prompt-based learning. Initially, we use ELECTRA PLM (Clark et al., 2020) for typical finetuning and further pre-training using 2 million unlabeled data from Gab and Reddit. Afterward, we utilize T5 PLM (Raffel et al., 2019) to investi-

[1]https://codalab.lisn.upsaclay.fr/competitions/7124

gate how encoder-decoder models can tackle the task in a prompting setting. In addition, we extend the training data using CallMeSexistBut (Samory et al., 2021) dataset and conduct extensive data augmentation and data preprocessing to enhance the robustness of the classification models.

In this study we would like to shed light on the following research questions:

- **RQ1**: Does further pre-training using 2 million unlabeled data results in better performance compared to the fine-tuning approach?
- **RQ2**: Does utilizing an encoder-decoder model in the prompting paradigm lead to improved performance?
- **RQ3**: Does enhancing the data through expansion, augmentation, and pre-processing result in better performance?

All the code, data, and experiment results are available on the GitHub repository[2].

## 2 Data

To train our models we employ various datasets in this study. First, we use the original data which is provided by the shared task organizers. Then, we increase the size of the training data by incorporating the CallMeSexistBut dataset (Samory et al., 2021) and applied various data augmentation and preprocessing techniques to improve the classification models' robustness. In the following sections, we provide a detailed description of the various data sets that are used in our experiments.

### 2.1 Training Datasets

Table 1 demonstrates the distribution of sexist/not sexist classes in the following training datasets:

**EDOS Data** The main training data set, which is provided by the EDOS organizers comprises 14,000 entries, which is quite imbalanced. As shown in table 1, 75.73% of data has been labeled as not sexist (10602) and 24.27% as sexist (3398). In the remaining of this paper, we refer to this data set as the "EDOS dataset".

**Augmeneted data** We attempt to extend the training dataset using data augmentation techniques. We augmented our training dataset by replacing some words in a sentence and adding them to the original data. The augmentation process resulted in a dataset of 32948 data points, of which 18.15% are sexist and 81.85% are not sexist (see table 1).

**Combined data** We enlarged the training data by merging the existing EDOS dataset with an external dataset, named "CallMeSexistBut" (Samory et al., 2021), that was compiled by GESIS - Leibniz Institutes for Social Sciences. This dataset contains three different types of 'short-text' content: 1. posts on social media (tweets) 2. psychological survey items, and 3. synthetic adversarial variants of the first two. The dataset consists of English texts labeled as "sexist" (positive class) or "not sexist"(negative class). It includes 13631 entries, and 1809 among them are sexist and 11822 of them are not sexist. Having combined the two datasets (EDOS and CallMeSexistBut), as shown in table 1, we have 24831 data points, 18.13% of which are positive samples (sexist) and 81.87% are negative samples (not sexist). Before combining these two datasets, we split 20% of the EDOS dataset as the validation set for our internal evaluation.

**Over-Sampled data** The main idea with oversampling is to reduce the unbalancing in the data by adding some records related to the low-resolution classes to achieve a rather balanced dataset as much as possible. For this purpose, we create an oversampled dataset containing 20328 (56.23%) not sexist samples and 15828 (43.77%) sexist samples.

**Preprocessed data** To see the effects of messy data on the results, we performed some data preprocessing on the Combined data to create another dataset. The preprocessing steps are replacing emojis with a constant, removing emojis completely, changing the casing of the tweets, and removing URLs and html tags.

Table 1: The distribution of binary classes in different Datasets

| Dataset | Records | Not Sexist(%) | Sexist(%) |
|---|---|---|---|
| EDOS | 14000 | 75.73 | 24.27 |
| Augmented | 32948 | 81.85 | 18.15 |
| Combined | 24831 | 81.87 | 18.13 |
| Preprocessed | 24831 | 81.87 | 18.13 |
| OverSampled | 36153 | 56.23 | 43.77 |

### 2.2 Unlabeled Dataset

For further pretraining purposes, we employ two million entries from the Reddit and GAB unlabeled

[2]https://github.com/bnanik/Shared_Task_SemEval2023

data that have been cleaned and prepared under the same procedure as the labeled data.

### 2.3 Validation Set

We use 80% of the EDOS data for training purposes and 20% for the unbiased internal evaluation of the model's performance. To maintain the distribution of classes across all subsets, we stratify the data based on classes.

### 2.4 Development and test sets

The development data consists of 2,000 entries, that can be used in the development phase to score submissions. The test set contains 4000 entries that are used for the final evaluation of the system.

## 3 System overview

We approach the task by various transfer learning methodologies. Our primary contribution is the investigation of two PLMs, namely ELECTRA (Clark et al., 2020) and T5 (Raffel et al., 2019), in three different paradigms: 1) Further pre-training, 2) Typical fine-tuning, and 3) Prompt-based learning. In the following subsections, we explain our baseline and the three approaches in more detail.

### 3.1 Baseline

We utilize the Support Vector Machine (SVM) algorithm (Cortes and Vapnik, 1995) as the baseline in which we employ the TF-IDF vectorization to create features by extracting information from the data. We use the TF-IDF vectorizer with a limit of 10,000 words capturing uni-grams and bi-grams.

### 3.2 Further Pretraining

Further pretraining is the process of continuing the pretraining of a pretrained model with additional data in order for the model to get more knowledge from that data. The idea is to first initialize a standard pretrained model and then feed new data to the tokenizer and finally start the pretraining which resulted in better achievements in both high- and low-resource settings (Gururangan et al., 2020, Lin et al., 2022, Baevski et al., 2019, Lee et al., 2019).

We employ the ELECTRA[3] as a promising model that outperforms many other transformers of the same size but with less amount of computation (Clark et al., 2020).

[3]Efficiently Learning an Encoder that Classifies Token Replacements Accurately

The generator component performs the state-of-the-art replaced token detection pretraining task. Unlike BERT (Devlin et al., 2019) that uses a Masked Language Model to train the data with a 15% masking rate on the input tokens, the ELECTRA uses a corruption mechanism that tries to replace the tokens with plausible ones by the generator component. The discriminator, then, learns how to identify which masked token is real and which one has been replaced. ELECTRA has three versions as shown in table 2 in which the large model achieved the best performance among others on the GLUE benchmarks (Wang et al., 2018).

Table 2: Characteristics of the ELECTRA models

| Model | Layers | Hidden Size | Params | GLUE score |
|---|---|---|---|---|
| Small | 12 | 256 | 14M | 77.4 |
| Base | 12 | 768 | 110M | 82.7 |
| Large | 24 | 1024 | 335M | 85.2 |

In this approach, we continue the pre-training process of the ELECTRA-small model with the two million unlabeled datasets. As a result, the model will learn more about the syntactic roles, and the relation between sentences and terminologies hence updating parameters and weights. However, unlike (Krishna et al., 2022), our data for further pretraining and finetuning are not the same but in the same domain. As the ELECTRA models include both generator and discriminator, we extract the discriminator part for further use.

### 3.3 Typical Fine-tuning

To adapt a pre-trained model to the sexism detection task, we fine-tune both the further pre-trained and the standard ELECTRA models.

The architecture of the proposed model consists of an ELECTRA encoder with a 3-layer neural network classifier on top, which gets the output of the last ELECTRA layer, pass it through a hidden layer of size 20, and finally classifies them into 2 classes namely "sexist" and "not sexist". To avoid overfitting we apply a dropout layer that drops the tensors with a probability of 0.5 randomly to the output layer which has been shown to be an effective technique for regularization and preventing the co-adaptation of neurons(Hinton et al., 2012).

We also consider early stopping as a strategy to decide if the model needs to stop training sooner or not.

### 3.4 Prompt-based learning

Although fine-tuning PLMs has been proven to achieve superior results over other conventional approach, it requires a substantial amount of data and computational resources for each downstream task. Recently, prompt-based learning has experienced a resurgence in NLP, and it has been shown that developing very large PLMs and prompting them alleviates the need for additional data for fine-tuning (Brown et al., 2020). In this paradigm, the input text is modified using a textual template and is fed to a particular PLM to conduct the task.

The basic prompting processes take place in three steps (Liu et al., 2021):

**Prompt Addition:** In this step a *prompting function* is defined to pre-process the input text. This step consists of two processes: 1) Creating a *template*, which consists of some fixed extra tokens and two slots: *input slot* [X] for input text and *answer slot* [Z] for predicted output that will be used in the *answer mapping* step. 2) Filling input slot [X] with the input text.

**Answer Search:** In this step, the output slot [Z] in the prompt will be filled by a potential answer, which is the highest-scoring answer.

**Answer Mapping:** Each potential answer has a corresponding output to be mapped to the actual classes.

## 4 Experiments

In this section, we explain the experimental setup of several models that we explained in the previous section.

#### Further pretraining ELECTRA

We continue pretraining on the small model with 88000 more steps to cover new unlabeled data entries gathered from GAB and Reddit in one epoch. This procedure is started by selecting the ELECTRA-small model checkpoint with 12 layers and a hidden size of 256. This model has been already trained on the OpenWebTextCorpus[4] which is based on Reddit posts. All the ELECTRA models use the exact same vocabulary as English uncased BERT[5]. Table 7.16 shows the hyperparameters that we use for further pretraining the ELECTERA-small model. In order to use the pre-trained encoder in the finetuning phase, we extract the discriminator from the whole pre-trained model. Moreover, as the official pretraining code[6] utilizes TensorFlow while our model uses the PyTorch library, we convert the pre-trained model to the PyTorch one for further use.

#### Finetuning ELECTRA

To train the whole model on the downstream sexism detection task, we do experiments with a set of hyperparameters against the ELECTRA encoder as table 7.17

We fine-tune the ELECTRA model on both the further pre-trained model(small) and the standard huggingface models (small, base, large)[7]. We also examine the small model 1) with the patience value 1 which means that the training process will not tolerate more than one loss increasing. It will stop the training after reaching the patience threshold, and 2) without any early stopping strategy to see what would be the effect of patience on the performance.

#### Prompting T5

In this approach, we employ T5 PLM (Raffel et al., 2019) in the prompting paradigm. T5 is an encoder-decoder PLM that considers all NLP tasks in a text-to-text format. Its implementation configuration is similar to the original Transformer in (Vaswani et al., 2017).

**OpenPrompt** We implement our model in OpenPrompt[8] (Ding et al., 2021), an easy-to-use toolkit that allows users to create prompt-based learning pipelines. It supports loading transformer-based models directly from the *huggingface*. First we define the task by determining the `classes` and transferring input text data into `InputExample` format. Then, we load the `"T5-base"` or `"T5-large"` models from the *transformers* library. Next, we establish a "Template" for the prompt. Below are the templates we defined for our classifiers. The variables `"text_a"` and `"mask"` will be replaced by the input text and the predicted label, respectively:

Template 1: {"placeholder":"text_a"}. Is the text sexist? {"mask"}.

Template 2: {"placeholder":"text_a"}. The text is {"mask"}.

[4]https://skylion007.github.io/OpenWebTextCorpus

[5]https://storage.googleapis.com/electra-data/vocab.txt

[6]https://github.com/google-research/electra

[7]https://huggingface.co/google/electra-small-discriminator

[8]https://github.com/thunlp/OpenPrompt

Table 3: The results of three approaches, further pretraining, finetuning, and prompting paradigms using ELECTRA and T5 PLMs trained over the EDOS dataset. (EL:ELECTRA, F1-dev: F1-score of development set submissions)

| Model | Prec. | Rec. | F1 | F1-dev |
|---|---|---|---|---|
| Baseline | **0.845** | 0.685 | 0.719 | 0.700 |
| **Pretraining** | | | | |
| EL-Small | 0.743 | 0.771 | 0.755 | 0.747 |
| **Finetuning** | | | | |
| EL-Small | 0.796 | 0.801 | 0.799 | 0.781 |
| EL-Base | 0.825 | 0.831 | 0.828 | 0.807 |
| EL-Large | 0.829 | **0.861** | **0.843** | **0.842** |
| **Prompting** | | | | |
| T5-base | 0.829 | 0.802 | 0.814 | 0.815 |

We define a verbalizer to map the encoded dataset labels (0 and 1) to their corresponding actual labels. We set the maximum sequence length to 512, and a batch size of 4. We use *cross entropy loss* as a loss function. Using *Adam* optimizer with a learning rate of 1e-4 we train the model over a training dataset in 4 epochs.

### 4.1 Evaluation

In order to assess the methods, we employ the Validation Dataset to calculate precision, recall, and f1-score as internal evaluation measures. Since the dataset is imbalanced, we consider the macro average of these metrics. We also measure the performance of our models through the F1-score by submitting the predictions of the provided dev set to the Codalab portal.

## 5 Results and Discussion

In this section, we present the results that we achieve in the experiments.

**EDOS Data** Table 3 compares the models' performance trained over EDOS data. Our findings show that all the proposed models outperform the baseline. This demonstrates the effectiveness of transformer-based models compared to other types of machine learning models, as confirmed by previous research (González-Carvajal and Garrido-Merchán, 2020, Kumar and Ojha, 2019). Interestingly, the baseline has a high precision, making it a strong competitor among the other models, coming in second best.

The fine-tuned ELECTRA-large model outperforms all the other models with an F1-score of 0.842 over the submitted dev set. It surpasses the ELECTRA-small and ELECTRA-base models by 3.5% and 6.1%, respectively, which is in line with (Clark et al., 2020) findings. This superiority is on all metrics such as precision, recall, and accuracy which means the ELECTRA Large model was able to figure out the input representation more precisely than others. The T5-base model in the prompting paradigm performed the second best with an F1-score of 0.815.

The results also illustrate that the finetuning of the ELECTRA small model has a better performance compared to the pre-trained model with the same size which is not in line with the previous findings (Gururangan et al., 2020). Considering the fact that the ELECTRA has been trained on OpenWebTxtCorpus which is already based on the Reddit posts, performing further pretraining using a combination of Reddit and Gab might not help the model to learn the knowledge behind the data more than the original samples or even damage it. However, conducting the pretraining process in more epochs might improve the performance and let the model learn the representations precisely.

Table 4: Experiment results with different prompt templates training T5-base PLM over EDOS dataset

| Prompt template | Acc. | F1 |
|---|---|---|
| (1) [X]; This text is [Z] | **0.869** | **0.814** |
| (2) [X]; Is the text sexist? [Z] | 0.849 | 0.779 |
| (3) [X]; [Z] | 0.850 | 0.797 |

**Prompt Templates** To verify that using prompts is effective in this task, we conduct two experiments using different prompt templates and one without any prompt. The results, shown in Table 4, indicate that template 1 is the best template to be used for this task.

**Data manipulation** To investigate the impact of data expansion, data augmentation, and data preprocessing, we conduct more experiments in our prompting approach. we trained the T5-base and T5-large PLMs on the Combined dataset. The results in table 5 show that the T5-base model in the prompting approach performed the best, achieving an F1-score of 0.809 on the dev set. It reveals that using the T5-large model resulted in a lower F1-score of 0.805.

The table shows that data augmentation and preprocessing resulted in a significant decrease in

performance. The T5-base model trained on augmented and preprocessed datasets had F1-scores of 0.761 and 0.767 respectively. However, when we only used lowercasing and skipped other preprocessing steps, the model performed better with an F1-score of 0.805. In addition, compared to augmentation and preprocessing, the oversampling technique showed better performance with an F1-score of 0.780 on the development set. Overall, our findings indicate that expanding, augmenting, preprocessing, and oversampling data does not improve the performance of our model. Thus we stick to the original data without any data manipulation in other approaches (shown in table 3) and for our submission.

Table 5: Experiment results of prompting T5 PLM using Combined, Augmented, Preprocessed, and oversampled data.

| Model | Prec. | Rec. | F1 | F1-dev |
|---|---|---|---|---|
| **Combined** | | | | |
| T5-base | 0.791 | 0.774 | 0.782 | 0.809 |
| T5-large | 0.791 | 0.797 | 0.794 | 0.805 |
| **Augmented** | | | | |
| T5-base | 0.799 | 0.719 | 0.744 | 0.761 |
| **Preprocessed** | | | | |
| T5-base | 0.746 | 0.774 | 0.757 | 0.767 |
| T5-base (lowercasing) | 0.799 | 0.774 | 0.785 | 0.805 |
| **Oversampled** | | | | |
| T5-base | 0.744 | 0.765 | 0.753 | 0.780 |

**Early Stopping** We also examined the effect of the early stopping strategy on the ELECTRA large model during the training. We achieved 0.842 f1-score when we complete the training with 20 epochs without any early stopping setup while it gained an f1-score of 0.833 in the setting with early stopping with a patience value of 1 which the model gets stuck in the local minimum. This highlights the volatility of the results and suggests that input might lead to false positive/false negative results in some cases.

## 6 Conclusion

In this project, to address the SemEval 2023 shared task - task 10, sub-task A, we built six transformer-based models in three different paradigms namely typical finetuning, further pretraining, and prompt-based learning. To address the first research question we developed four classifiers. We start with finetuning approach in a pipeline of ELECTRA encoder with a head of neural network classifier. In this approach, we employ ELECTRA-small, ELECTRA-base, and ELECTRA-large models to build different encoders. Then, we conduct a further pretraining process on the ELECTRA-small model using 2 million unlabeled data from Reddit and GAB datasets provided by the organizers. We replace the previous encoder in finetuning step with the new pretrained one. Finally, we finetune the pretrained ELECTRA discriminator as another method to adapt the model to the downstream task. The results of our experiments on the ELECTRA-small model show that continuing pretraining with domain-specific data does not perform better than finetuning the same model. However, finetuning the ELECTRA-large model outperforms all the other approaches including further pretrained ELECTRA-small.

In regards to the second research question, we used both T5-base and T5-large models in a prompting paradigm, and our results indicate that both models have similar performance at least in this task. However, we found that the finetuned ELECTRA-large model performed better than these encoder-decoder models.

To address the last research question, we used an external dataset (CallMeSexistBut) to expand our data. In addition, we manipulate the data in different settings such as augmentation, preprocessing, and oversampling. We found that our top-performing models achieved the best performance using the original data without any expansion, augmentation, preprocessing, or oversampling.

Due to time and computational limitations, we were unable to conduct more experiments to further explore the prompting approach using ELECTRA, T5-large, or even larger models. Based on the results of our study, we suggest that further research should investigate the use of the ELECTRA model with prompting. Additionally, further pre-training of the ELECTRA-large model may provide more meaningful comparisons with our models.

## Acknowledgement

We would like to thank Dr. Tomasso Caselli and Lucas Edman from the Faculty of Arts at the University of Groningen, for their support and assistance during this project.

## References


Anzovino, M., E. Fersini, and P. Rosso (2018). Automatic identification and classification of misogynistic language on twitter. In M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, and F. Meziane (Eds.), *Natural Language Processing and Information Systems*, Cham, pp. 57–64. Springer International Publishing.

Badjatiya, P., S. Gupta, M. Gupta, and V. Varma (2017). Deep learning for hate speech detection in tweets. *CoRR abs/1706.00188*.

Baevski, A., S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli (2019, November). Cloze-driven pretraining of self-attention networks. 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)*, Hong Kong, China, pp. 5360–5369. Association for Computational Linguistics.

Barker, K. and O. Jurasz (2019). Online misogyny: A challenge for digital feminism? *Journal of International Affairs 72*(2), 95–114.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Eds.), *Advances in Neural Information Processing Systems*, Volume 33, pp. 1877–1901. Curran Associates, Inc.

Butt, S., N. Ashraf, G. Sidorov, and A. Gelbukh (2021). Sexism identification using bert and data augmentation - exist2021. *CEUR Workshop Proceedings 2943*, 381–389. Publisher Copyright: © 2021 CEUR-WS. All rights reserved.; null ; Conference date: 21-09-2021.

Clark, K., M. Luong, Q. V. Le, and C. D. Manning (2020). ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR abs/2003.10555*.

Clarke, I. and J. Grieve (2017, August). Dimensions of abusive language on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*, Vancouver, BC, Canada, pp. 1–10. Association for Computational Linguistics.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019, June). 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)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.

Ding, N., S. Hu, W. Zhao, Y. Chen, Z. Liu, H.-T. Zheng, and M. Sun (2021). Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

González-Carvajal, S. and E. C. Garrido-Merchán (2020). Comparing BERT against traditional machine learning text classification. *CoRR abs/2005.13012*.

Gururangan, S., A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith (2020). Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR abs/1207.0580*.

Krishna, K., S. Garg, J. P. Bigham, and Z. C. Lipton (2022). Downstream datasets make surprisingly good pretraining corpora. *arXiv preprint arXiv:2209.14389*.

Kumar, R. and A. K. Ojha (2019). Kmi-panlingua at hasoc 2019: Svm vs bert for hate speech and offensive content detection. In *FIRE (Working Notes)*, pp. 285–292.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang (2019). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR abs/1901.08746*.

Lin, T.-H., T.-C. Chi, and A. Rumshisky (2022). On task-adaptive pretraining for dialogue response selection.

Liu, P., W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ArXiv abs/2107.13586*.

Lopez-Lopez, E., J. Carrillo-de Albornoz, and L. Plaza (2021). Combining transformer-based models with traditional machine learning approaches for sexism identification in social networks at exist 2021. In M. Montes, P. Rosso, J. Gonzalo, M. E. Aragón, R. Agerri, M. Carmona, E. Mellado, J. Carrillo-de Albornoz, L. Chiruzzo, L. A. d. Freitas, H. Gómez-Adorno, Y. Gutiérrez, S. M. J. Zafra, S. Lima, F. M. P. d. Arco, and M. Taulé (Eds.), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021*, Volume 2943 of *CEUR Workshop Proceedings*, pp. 431–441. CEUR-WS.org.

Nina-Alcocer, V. (2018). Ami at ibereval2018 automatic misogyny identification in spanish and english tweets. In *IberEval@SEPLN*.

Park, J. H. and P. Fung (2017). One-step and two-step classification for abusive language detection on twitter. *CoRR abs/1706.01206*.

Paula, A., R. Silva, and Barış (2021, 11). Sexism prediction in spanish and english tweets using monolingual and multilingual bert and ensemble models.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR abs/1910.10683*.

Saleem, H. M., K. P. Dillon, S. Benesch, and D. Ruths (2017). A web of hate: Tackling hateful speech in online social spaces. *CoRR abs/1709.10159*.

Samory, M., I. Sen, J. Kohne, F. Flöck, and C. Wagner (2021). " call me sexist, but...": Revisiting sexism detection using psychological scales and adversarial samples. In *ICWSM*, pp. 573–584.

Schütz, M., J. Boeck, D. Liakhovets, D. Slijepcevic, A. Kirchknopf, M. Hecht, J. Bogensperger, S. Schlarb, A. Schindler, and M. Zeppelzauer (2021). Automatic sexism detection with multilingual transformer models. *CoRR abs/2106.04908*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Wang, A., A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Zhang, Z. and L. Luo (2018). Hate speech detection: A solved problem? the challenging case of long tail on twitter. *CoRR abs/1803.03662*.

# 7 Apendix

## 7.1 A. Further pretraining

Table 6: Hyper parameters for further pretraining the ELECTRA small model

| Hyperparameter | Value |
|---|---|
| Embedding size | 128 |
| Hidden size | 256 |
| Num attention heads | 4 |
| Num hidden layers | 12 |
| Pad token id | 0 |
| Vocab size | 30522 |

## 7.2 B. Finetuning

Table 7: Hyperparameters of ELECTRA fine-tuning

| HyperParameter | Value |
|---|---|
| Number of epochs | 20 |
| Batch size | 32 |
| Learning rate | 2e-5 |
| Weight decay | 0.01 |
| Epsilon | 1e-8 |
| Seed value | 42 |
| Early stopping patience | 3, None |
| Optimizer | AdamW |