ReBERT at HSD-2Lang 2024: Fine-Tuning BERT with AdamW for Hate Speech Detection in Arabic and Turkish

Utku Ugur Yagci
Middle East Technical University
utku.yagci@metu.edu.tr

Ahmet Emirhan Kolcak
Istanbul Technical University
kolcak20@itu.edu.tr

Egemen Iscan
King's Business School
egemen.iscan@kcl.ac.uk

Abstract

This research tackles the issue of detecting hate speech in Arabic and Turkish languages by utilizing pre-trained BERT models, namely TurkishBERTweet and Arabertv02-twitter. These models are enhanced through a comprehensive hyperparameter search to improve their performance. Our classifiers excelled in the HSD-2Lang 2024 contest, with the Turkish model placing second in Subtask A and the Arabic model first in Subtask B on the private leaderboard. Both models also ranked first on the public dataset. These results demonstrate the efficacy and adaptability of our approach in addressing the evolving challenges of hate speech detection in multilingual contexts.

1 Introduction

In this study, we have explored several fine-tuning strategies to establish BERT (Devlin et al., 2019) models and compared their performances on two separate datasets, one in Turkish and the other in Arabic. We aimed to outperform the competitor models in the HSD-2Lang Subtask A and Subtask B in detecting hate speech in tweets. The details of these subtasks are explained in the contest paper (Uludoğan et al., 2024).

BERT is a widely used and accepted approach in the field of natural language processing (NLP) due to its efficiency and high performance in detecting hate speech compared to most conventional model architectures. The original BERT paper proposed epoch numbers ranging from 2 to 4 and learning rates 5e-5, 3e-5, and 2e-5 with Adam optimizer for fine-tuning BERT (Devlin et al., 2019). However, during our experimentation, we extended the range of the proposed hyperparameters in the original study. By enlarging the range, we were able to try various combinations of hyperparameters to enhance our model's performance. Considering the competitive and limited nature of our task at hand, going beyond the suggested methods can be advantageous and provide a unique solution.

Our approach is insightful as it applies existing models and frameworks practically, and its competitive results offer valuable insights for future hate speech detection research in Arabic, Turkish, and other languages.

2 Related Work

The introduction of BERT (Bidirectional Encoder Representations from Transformers) by Devlin et al. (2019) has radically changed the field of detecting the nuances of languages contextually. The commonly adapted paradigm associated with BERT consists of a pre-training and a fine-tuning step (Xinxi, 2021). The fine-tuning step is intended for the model to specialize on a specific task. The fine-tuning of a pre-trained BERT model is proven to be significantly more robust compared to similar approaches offered in the past (Mosbach et al., 2020). In our case, this task was hate speech detection. Hence, we have chosen our pre-trained models accordingly. Mozafari et al. (2020) introduced a transfer learning approach where a pre-trained BERT model is used for detecting hate speech in social media. The body of past research has laid a solid foundation upon which our study is constructed, enabling us to train our models with novel insights and methodologies.

3 Methodology

We built and trained our models in Python using the PyTorch (Paszke et al., 2019) framework. We had access to Google Colab's A100 NVIDIA GPUs via subscription, which helped us experiment with several architectures efficiently. The performance of the GPU was crucial since we had time constraints for achieving both tasks. A powerful GPU creates a significant difference, especially when training with relatively high epoch numbers (e.g.,

Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024), pages 195–198

100,1000). The training data consists of 9140 observations for Subtask A and 960 observations for Subtask B. No training or test data was used besides the datasets provided to us as a part of the competition. We initially tested the base performance of pre-trained models from Hugging Face [1] on each subtask's training data without applying fine-tuning or preprocessing. Respectively, the initial models we decided not to proceed with were "AraBert Hate Speech Detecter" [2] developed by WidadAwane and "Bert Base Turkish Uncased" [3] developed by Dbmdz. The used models are then selected according to their performance. Afterward, we conducted a more structured hyperparameter search for the selected pre-trained models for fine-tuning. While training, we detected a data imbalance issue between the number of negatively (non hate speech) and positively (hate speech) labeled observations: in Subtask A, the initial ratio was approximately 70/30, whereas in Subtask B, it was even more skewed at 90/10. The fact that the imbalance is more apparent in Subtask B is particularly significant due to the limited size of the data, which has the potential to make its effects more impactful. The reduced dataset size in Subtask B amplifies the risk of model overfitting to the over-represented class, thus increasing the challenges in achieving a balanced and robust model performance. This situation was preventing the expected performance increase through fine-tuning. Due to this observation, we only used 80 percent of the negatively labeled (non-toxic) training data for Subtask B. The excluded negatively labeled observations were chosen randomly. We opted not to remove too many rows because the data is already very limited, which necessitated a careful balancing to avoid excessively diminishing our dataset's size.

It is essential to mention that our initial goal was to have the best score in the competition rather than have a more general model that can detect hate speech on a wide variety of datasets. Our only performance benchmark during training was the unlabeled public test dataset. The public test data covers only 20 percent of the total test data. Therefore, we delve into finding a configuration for fine-tuning that performs better than the other competitors in the public dataset—assuming that the performance on the public dataset will carry over to the private test dataset (80 percent of the unlabeled test data).

[1] https://huggingface.co/

[2] https://huggingface.co/WidadAwane/AraBert_Hate_Speech_Detecter/

[3] https://huggingface.co/dbmdz/bert-base-turkish-uncased/

4 Experimental Setup

For the preprocessing step, we didn't alter the grammatical attributes through processes such as lemmatization or stemmization. We therefore have not carried out any Part-of-Speech (POS) tagging, or similar analyses. This is due to the way that BERT and transformer (Vaswani et al., 2023) models function in general. From our literature review, we decided that it is best to keep the data as raw as possible, with the presence of punctuation as well as any expressions of tone, except only slight modifications to make it cleaner. We used the original preprocessing function by the TurkishBERTweet (Najafi and Varol, 2023) authors, which we then used as our main model for Subtask A. This preprocessing function only converts URL and emoticons into tags similar to HTML format, and doesn't apply any further modifications. For Subtask B, no preprocessing was applied.

We fine-tuned the submitted TurkishBERTweet model on NVIDIA A100 GPU and set the batch size equal to 32. After that, we set the max sequence length to 256, base learning rate to 5e-5, epsilon to 1e-8, and warm-up proportion to 10 percent of the total steps. We use the AdamW optimizer, introduced by Loshchilov and Hutter (2017), with the default parameters and set the scheduler to polynomial weight decay. Table 1 shows the results of the hyperparameter search conducted on the TurkishBERTweet model.

For the second Subtask, we used the pre-trained Arabertv02-twitter model (Antoun et al.). We fine-tuned the model on NVIDIA GTX 1070 and set the batch size equal to 8. We set the max sequence length to 250, base learning rate to 5e-4, epsilon to 1e-8, output attentions = False, output hidden states = False, and correct bias = False. Table 2 shows the results of the hyperparameter search conducted on the Arabertv02-twitter model.

5 Results and Discussion

Model configurations can be seen in Table 1 with corresponding F1 scores in public and private datasets. Best submitted model configurations are written in bold. The public scores for Subtask A and B were 0.89 and 0.74, respectively, placing our

Subtask	Pre-trained Model	Optimizer	Learning Rate	Scheduler	Batch	Epoch	F1 Score	
							Public	Private
A	TurkishBERTweet	**AdamW**	**5e-5**	**Polynomial Decay with 10% Warmup**	**32**	**100**	**0.74**	**0.69**
		AdamW	5e-4	Linear Decay with No Warmup	32	100	0.74	0.69
		AdamW	5e-5	Linear Decay with 10% Warmup	32	100	0.73	0.69
		AdamW	5e-5	Linear Decay with No Warmup	32	1000	0.73	0.70
		AdamW	5e-5	Linear Decay with 10% Warmup	32	5	0.72	0.66
		Adafactor	-	Adafactor Schedule	128	15	0.70	0.66
B	Arabertv02-Twitter	**AdamW**	**5e-4**	**Linear Decay with No Warmup**	**8**	**8**	**0.89**	**0.74**
		AdamW	5e-5	Linear Decay with No Warmup	8	4	0.85	0.66
		AdamW	5e-5	Linear Decay with No Warmup	8	8	0.85	0.76
		AdamW	5e-5	Linear Decay with No Warmup	16	4	0.80	0.69

Table 1: Fine-Tuned model results for Private and Public datasets.

models at the top of the public leaderboard on both subtasks. With the release of the private leaderboard scores, our models ranked 1st in the Arabic Subtask with an F1 score of 0.74 and 2nd in the Turkish Subtask with an F1 score of 0.69.

During the hyperparameter search for Subtask A, we could only achieve minor performance differences up to -2 to +2 percent in terms of F1 score and accuracy by altering the base learning rate, epoch size, and batch size. One of the main findings for Subtask A is that we could enhance the performance only by implementing unconventional epoch lengths such as 100 and 1000. While the proposed range for epoch numbers is [2,4], we have achieved better-performing models up to 4 percent by using a relatively high number of epochs during the training phase. Furthermore, we also achieved better results by applying a 10 percent warmup during training. The best-performing score for Subtask A was achieved by implementing the polynomial weight decay scheduler with a 10 percent warmup. In addition to the hyperparameter search, we trained another model with a different optimizer named Adafactor (Shazeer and Stern, 2018). We used a batch size of 128 and an epoch number of 15. However, since the resulting F1 score was far below compared to the scores of models trained with the AdamW optimizer, we did not experiment any further.

Hyperparameter search for Subtask B involves different combinations of epoch numbers and batch sizes. Since the amount of training data for Subtask B is limited to 960, fine-tuning with limited data involves the risk of overfitting. With limited training data, there is a higher risk that the model will memorize the training examples rather than learn generalizable patterns. This situation can lead to poor performance on unseen data for hate speech detection. Furthermore, Arabertv02-twitter is pre-trained on various tasks with more than 60 million tweets. Therefore, training on a small dataset may not effectively adapt the model's ability for hate speech detection. Taking this condition into account, we trained our models with a linear decay scheduler with no warmup. Our best-submitted model exceeded expectations performance-wise by scoring 0.89 in the public test set and 0.74 in the private test set.

When comparing our top-ranking models with each other, we observed that the fine-tuned TurkishBERTweet model performed worse than the fine-tuned Arabertv02-tweet model if our sole consideration as a benchmark was the F1 score. However, there may be other factors to discuss before reaching such a conclusion. One potential factor is the limited data in the Arabic Subtask. It is unclear whether the F1 score is a sufficient metric by itself to compare models when at least one of those models is trained or evaluated on limited data. We also noticed that our submissions had lower variation in F1 score for Subtask A, compared to Subtask B, which may also be due to the limited data constraint in Subtask B. Hence, it may be misleading and out of scope to compare these two types of models to each other.

6 Conclusion

In this paper, we conducted experiments to fine-tune pre-trained BERT models for the hate speech detection task. We explore various approaches to maximize the performance of each algorithm by adjusting the hyperparameters. This paper focuses on the three primary hyperparameters: learning rate, batch size, and epoch length. Our final leaderboard rankings in Arabic and Turkish Subtasks turned out to be 1 and 2, respectively. This is a demonstration of consistent success, indicating that fine-tuning BERT is a practical and effective approach for detecting hate speech in various aspects of a particular language. Several factors, such as the

choice of a pre-trained model and hyperparameters used in fine-tuning, contribute to obtaining a satisfactory result. The selection of the pre-trained model should be done under consideration of the format of the data. Models pre-trained with data from Twitter can help achieve better results. Furthermore, the hate speech detection for this task requires a pre-trained model along with an effective tokenizer that is capable of tokenizing specific features for Twitter, such as hashtags, URLs, and emojis. Such particular features can enhance the performance of the classification task by expanding the perception of toxicity in our model.

7 Future Work

We plan to delve further into hate speech detection literature to improve the performance of our models. We believe that the performance of our models can be enhanced by implementing more advanced fine-tuning methods discussed by Sun et al. (2019). In addition, we aim to compare our findings with those from established machine learning algorithms, as well as more contemporary approaches such as GPT. This study will serve as a benchmark in our future studies of similar tasks.

Limitations

The use of specific GPU resources in this project might limit the reproducibility of our results for researchers with different setups. Our models are tailored for contest datasets. Hence, they may not perform well when applied to a wider variety of datasets for detecting hate speech. The hyperparameter optimization was constrained by the availability of only the public dataset for validation. This study does not prioritize the efficiency during training. Additionally, our reliance on pre-trained models restricts the adaptability of future research to modify initial model parameters.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based Model for Arabic Language Understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

I. Loshchilov and F. Hutter. 2017. Fixing Weight Decay Regularization in Adam. *ArXiv*, abs/1711.05101.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Marius Mosbach, Maksym Andriushchenko, and D. Klakow. 2020. On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. *ArXiv*, abs/2006.04884.

Marzieh Mozafari, R. Farahbakhsh, and N. Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. *PLoS ONE*, 15.

Ali Najafi and Onur Varol. 2023. TurkishBERTweet: Fast and Reliable Large Language Model for Social Media Analysis. *arXiv preprint arXiv:2311.18063*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. `https://pytorch.org/`.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? pages 194–206.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Gökçe Uludoğan, Somaiyeh Dehghan, İnanç Arın, Elif Erol, Berrin Yanikoglu, and Arzucan Özgür. 2024. Overview of the Hate Speech Detection in Turkish and Arabic Tweets (HSD-2Lang) Shared Task at CASE 2024. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, Malta. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Z. Xinxi. 2021. Single task fine-tune BERT for text classification. 11911:119111Z – 119111Z–6.