DetectiveReDASers at HSD-2Lang 2024: A New Pooling Strategy with Cross-lingual Augmentation and Ensembling for Hate Speech Detection in Low-resource Languages

Fatima Zahra Qachfar*
fqachfar@uh.edu
University of Houston
Houston, TX, USA

Bryan E. Tuck*
betuck@uh.edu
University of Houston
Houston, TX, USA

Rakesh M. Verma
rmverma2@central.uh.edu
University of Houston
Houston, TX, USA

Abstract

This paper addresses hate speech detection in Turkish and Arabic tweets, contributing to the HSD-2Lang Shared Task. We propose a specialized pooling strategy within a soft-voting ensemble framework to improve classification in Turkish and Arabic language models. Our approach also includes expanding the training sets through cross-lingual translation, introducing a broader spectrum of hate speech examples. Our method attains F1-Macro scores of 0.6964 for Turkish (Subtask A) and 0.7123 for Arabic (Subtask B). While achieving these results, we also consider the computational overhead, striking a balance between the effectiveness of our unique pooling strategy, data augmentation, and soft-voting ensemble. This approach advances the practical application of language models in low-resource languages for hate speech detection.

1 Introduction

Hate speech and offensive language on social media pose significant challenges, affecting individuals and communities globally. These concerns are exacerbated by the anonymity afforded by online platforms, leading to more aggressive behaviors (Fortuna and Nunes, 2018).

Addressing hate speech is crucial for protecting vulnerable and marginalized populations from discrimination and racism. The issue is particularly profound in low-resource languages like Arabic and Turkish, where cultural and linguistic diversity adds additional complexity to detection.

Conventional approaches in hate speech detection, which often rely on standard tooling libraries, may opt to remove emojis due to the unavailability of specific language support. This shortcoming is especially pronounced in a social media text, characterized by its brevity and unconventional language, where special characters like emojis have an influential impact on performance. In response to these challenges, we implemented support for Arabic and Turkish in the Emoji package, a functionality previously absent.

Hate speech detection research has traditionally focused on English (Mansur et al., 2023), with a recent shift towards multilingual contexts, including hate speech against immigrants and women (Basile et al., 2019). Current efforts are increasingly addressing the challenges in low-resource languages like Arabic and Turkish through new frameworks, datasets, and shared tasks (Mubarak et al., 2020; Beyhan et al., 2022; Hasanain et al., 2023). However, data scarcity and class imbalance in these languages still present considerable challenges, necessitating ongoing research and development.

We make the following key contributions and improvements over previous work: (1) A new pooling strategy that significantly improves classification of hate speech in Turkish and Arabic, contributing to higher Macro F1 scores, (2) An evaluation of a cross-lingual data augmentation technique to broaden and enrich the training datasets, enhancing the model's ability to generalize by focusing on language-specific challenges in hate speech contrary to (Ranasinghe and Zampieri, 2021) that solely relies on transfer learning from resource-rich to less-resourced language models, and (3) An implementation of a soft-voting ensemble framework to further boost model performance, as evidenced by the achieved Macro F1 scores.

2 Task and Dataset Description

In the HSD-2Lang shared task (Uludoğan et al., 2024), we focused on two main tasks: Subtask A for Hate Speech Detection in Turkish Tweets and Subtask B for limited Arabic Tweets. Subtask A involves analyzing a dataset of $9,140$ Turkish tweets, categorized across topics such as Anti-Refugee sentiment, the Israel-Palestine conflict, and Anti-Greek

*These authors contributed equally to this work.

Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024), pages 199–204

discourse, with both hateful and non-hateful tweets. Subtask B presented the challenge of detecting hate speech in a smaller, imbalanced dataset with 82 "hateful" and 778 "not hateful" Arabic tweets, primarily centered on anti-refugee sentiment.

3 Proposed Framework

The key elements of our approach to hate speech detection for subtasks A and B include emoji conversion and bidirectional translation between Turkish and Arabic datasets. We selected `ConvBERTurk`[1] (Schweter, 2020) and `AraBERTv02-Twitter`[2] (Antoun et al., 2020) as our baseline models for Turkish and Arabic texts, respectively.

To tackle the limited and imbalanced data in Subtask B, we merged the translated Turkish dataset from Subtask A with Subtask B dataset. We applied a similar strategy for Subtask A, incorporating the translated Arabic tweets from Subtask B.

Our research introduces an innovative sequence representation technique, going beyond the conventional use of the `[CLS]` token. This method combines the mean and max values from the last hidden layer with the `[CLS]` token, each processed through separate linear layers with *tanh* activation and dropout. The outputs are then concatenated and fed into a final linear layer for classification as "hateful" or "not hateful".

Subtask A employed a soft-voting ensemble of five `ConvBERTurk` models in our application. In contrast, Subtask B utilized a single `AraBERTv02-Twitter` model. In the upcoming sections, we provide a comprehensive overview of the methodologies we employed in our project. These include a detailed description of how we preprocessed the data, consolidated the datasets, converted emojis, translated across Turkish and Arabic, pooled sequence representations, and finally, our training procedures.

3.1 Preprocessing and Dataset Consolidation

Data Preprocessing Our preprocessing approach rigorously standardizes text data, a vital step for reliable analysis. We use the ftfy[3] package to correct incorrectly encoded characters, resolving common encoding issues in text data. Next, we simplify whitespace by replacing excess newlines and tabs with a single space. Our method also uniquely addresses user mentions by substituting them with a standard term—"[مستخدم]" in Arabic and "[Kullanıcı]" in Turkish—to avoid skewing the language-specific processing. Likewise, we replace URLs and retweet indicators with consistent placeholders to minimize noise and point the focus on the textual content itself.

Emoji Conversion The emoji[4] package is updated as we implemented support for Arabic and Turkish languages. This update enables the conversion of emoji characters into their corresponding text descriptions in Arabic and Turkish. Emojis often carry significant emotional and contextual meanings (Hakami et al., 2022), and this conversion is vital for capturing these nuances.

Data Consolidation and Cross-Lingual Translation In our preprocessing workflow, we first address Subtask A by concatenating the three distinct datasets focusing on anti-refugee sentiment, the Israel-Palestine conflict, and anti-Greek discourse. Then, we split this unified dataset using an 80/20 train-test ratio. By adopting this unified approach, we can incorporate a broader range of data, thereby increasing the diversity of the dataset. Additionally, we translated Subtask B's Arabic dataset into Turkish using Google Translator[5] and merged this with Subtask A's training set. This step ensures linguistic consistency and enriches the training data's contextual scope.

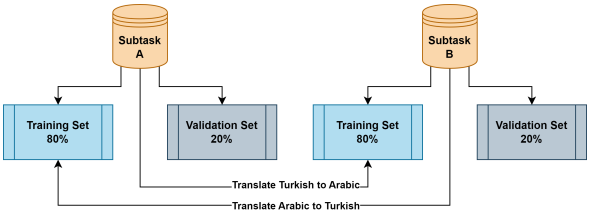

Figure 1: Data Augmentation Workflow

We tackle the challenges of limited and imbalanced data for Subtask B by leveraging thematic overlaps with Subtask A. We translate Subtask A's Turkish data into Arabic and integrate it into Subtask B's training set. This bidirectional translation strategy contributes to a more comprehensive and diverse training environment. We illustrate the similarity between subtasks in Appendix A.

[1] https://huggingface.co/dbmdz/convbert-base-turkish-cased

[2] https://huggingface.co/aubmindlab/bert-base-AraBERTv02-Twitter

[3] https://ftfy.readthedocs.io/

[4] https://github.com/carpedm20/emoji/

[5] https://deep-translator.readthedocs.io/

Throughout, we maintain a uniform preprocessing approach for both subtasks, adjusting slightly to accommodate the primary languages of Turkish for Subtask A and Arabic for Subtask B. This systematic data translation and consolidation approach is critical to our preprocessing strategy and aims to enhance our language models' overall quality and effectiveness.

3.2 Sequence Representation Pooling

We leverage a unique sequence representation technique for hate speech detection, termed "concat" pooling, which we apply in Bert-based models for both subtasks. Our method merges the `[CLS]` token with mean and max values from the last hidden layer's sequence dimension, aiming to enhance the comprehensiveness and diversity of sequence representation. This approach is in contrast to the Multi-CLS BERT method (Chang et al., 2023), which employs multiple `[CLS]` tokens in a singular BERT model, creating an ensemble-like effect without the substantial computational and memory costs typically associated with BERT ensembles.

In our implementation, we independently process the `[CLS]`, mean, and max outputs through separate linear layers, integrating *Tanh* activation and dropout before concatenation. This procedure ensures a robust and nuanced embedding, which we subsequently input into a final linear layer for classifying the inputs as "hateful" or "non-hateful". While inspired by Multi-CLS BERT's efficiency in managing multiple `[CLS]` embeddings, such an approach diverges by incorporating varied sequence elements to generate a more thorough representation for classification. Figure 2 illustrates our "concat" pooling architecture.

3.3 Soft-Voting Ensemble

Ensemble methods, rooted in collective decision-making, consistently demonstrate superior predictive accuracy and robustness over single-learner models (Jiang et al., 2023; Farooqi et al., 2021). For subtask A, we deploy a soft-voting ensemble consisting of five identical `ConvBERT-Turkish-Cased` models, differentiated only by their initializations. This strategy follows the methodology outlined by (Tuck et al., 2023) in Arabic deception detection, where we halt training at the two-epoch mark as soon as we reach the peak validation F1 Macro score. We

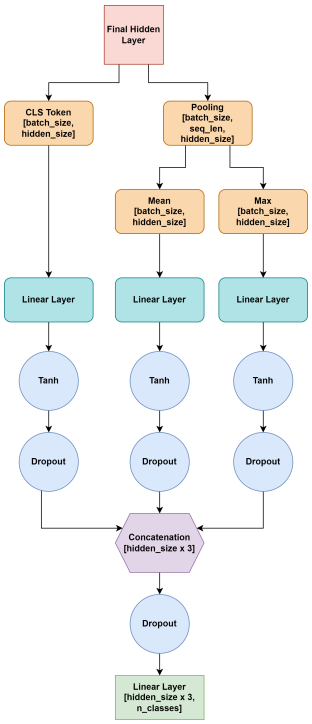

Figure 2: Concat Pooling Architecture

use the TorchEnsemble[6] library, an open-source, community-driven project, to facilitate the implementation of this ensemble technique, offering streamlined support for various ensemble methods.

3.4 Training Procedure

Our approach consistently applied the same hyperparameters across all experiments for both subtasks to ensure reliability and consistency. We chose the *AdamW* optimizer (Loshchilov and Hutter, 2019) for its efficiency in fine-tuning large language models, paired with the *Cross-Entropy* loss function, which is well-suited for binary classification tasks. This combination was selected to balance efficient learning with accurate performance.

We limited our training to a maximum of twenty epochs, incorporating an early stopping mechanism with a patience setting of five epochs. This strategy enhances computational efficiency and prevents overfitting by stopping the training when validation F1 Macro scores no longer improve. Although initial trials included a linear learning rate scheduler, we did not use it in our final experiments. Our observations indicated that maintaining a constant learning rate, combined with our chosen optimizer and early stopping, was the most effective. The static hyperparameters we used are as follows: Max

[6] https://github.com/TorchEnsemble-Community/Ensemble-Pytorch

Pooling Type	Data Aug.	Ensemble		Single Model	
		Val.	Test	Val.	Test
		Subtask A			
concat	Included	**0.7336**	**0.6964**	0.7130	0.6705
concat	Not Included	0.7203	0.6814	0.7272	0.6608
cls	Included	0.6794	0.6832	0.7368	0.6674
cls	Not Included	0.7348	0.6508	0.6929	0.6781
		Subtask B			
concat	Included	0.7826	0.6027	**0.8333**	**0.6000**
concat	Not Included	0.8461	0.7123	0.8148	0.6582
cls	Included	0.6956	0.5915	0.7333	0.6373
cls	Not Included	0.8148	0.7179	0.8148	0.6052

Table 1: Performance of `ConvBERT-Turkish-Cased` (Subtask A) and `AraBERTv02-Twitter` (Subtask B) models, using Macro F1 scores. 'Pooling Type' distinguishes between `[CLS]` token and concatenated embeddings. 'Data Aug.' indicates if augmentation was used ('Included') or not ('Not Included'). Bold results denote official submissions for each subtask.

Length – 128, Dropout – 0.075, Batch Size – 16, Learning Rate – $2e-05$, Random Seed – 42.

4 Results and Discussion

Table 1 outlines the performance of our models in Subtasks A and B, with the official submissions in bold, achieving 1st place in Subtask A and 3rd place in Subtask B. We offer a systematic view, examining the effects of pooling strategies, comparing ensemble and single-model configurations, and augmenting training data.

For Subtask A, our ensemble model utilizing concatenated pooling—synthesizing the `[CLS]` token, mean, and max embeddings—demonstrated substantial dominance on the test set with a Macro F1 score of 0.6964. This superior performance is attributed to our novel sequence representation, which provides a holistic comprehension of the input data, as opposed to the `[CLS]` token-based approach that achieved a lower score of 0.6832 with data augmentation.

In Subtask B, the ensemble models exhibited a pronounced sensitivity to data augmentation. The ensemble with `[CLS]` token pooling and no data augmentation achieved the highest test score of 0.7179. Conversely, when data augmentation was introduced, the same ensemble approach reduced test performance to 0.5915. Similarly, the ensemble model with concatenated pooling reflected this trend, where the non-augmented approach yielded a robust score of 0.7123 on the test set, compared to a lower 0.6027 with data augmentation.

For single models in Subtask B, the concatenated pooling type with data augmentation resulted in a test score of 0.6000, indicating that the single models were less affected by augmentation. However, this score was still outperformed by the non-augmented ensemble model, highlighting the nuanced impact of augmentation strategies on model performance. The intricate dynamics of the impact of data augmentation are underscored in Subtask B, where its application does not enhance model effectiveness. This difference is particularly notable when comparing the performance of single models against ensemble configurations.

The test scores suggest that ensemble models, especially with non-augmented concatenated pooling, are robust across both subtasks. The discrepancy in performance between the concat and `[CLS]` methods within ensemble configurations highlights the effectiveness of our pooling strategy. These findings emphasize the need for careful consideration when applying data augmentation, as it may not always be beneficial and depends on the specific task and model architecture.

5 Conclusion

In conclusion, our paper introduces an innovative approach combining data augmentation, pooling strategy, and a soft-voting ensemble framework for effective hate speech detection in Turkish and Arabic, languages typically underrepresented in computational linguistics. We successfully enriched the training sets with a broader spectrum of examples by leveraging cross-lingual translation through Google Translator. This approach yielded impressive F1-Macro scores of 0.6964 and 0.7123 in Turkish and Arabic, respectively, demonstrating broad potential in diverse linguistic contexts. The effectiveness of our strategy in low-resource languages opens new avenues for future research, potentially addressing more nuanced aspects of hate speech detection and expanding to other underrepresented languages.

Acknowledgments.

Research partly supported by NSF grants 2210198 and 2244279, and ARO grants W911NF-20-1-0254 and W911NF-23-1-0191. Verma is the founder of Everest Cyber Security and Analytics, Inc.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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 *International Workshop on Semantic Evaluation*.

Fatih Beyhan, Buse Çarik, Inanç Arin, Aysecan Terzioglu, Berrin A. Yanikoglu, and Reyyan Yeniterzi. 2022. A turkish hate speech dataset and detection system. In *International Conference on Language Resources and Evaluation*.

Haw-Shiuan Chang, Ruei-Yao Sun, Kathryn Ricci, and Andrew McCallum. 2023. Multi-CLS BERT: An efficient alternative to traditional ensembling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–854, Toronto, Canada. Association for Computational Linguistics.

Zaki Mustafa Farooqi, Sreyan Ghosh, and Rajiv Ratn Shah. 2021. Leveraging transformers for hate speech detection in conversational code-mixed tweets.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Shatha Ali A. Hakami, Robert Hendley, and Phillip Smith. 2022. Emoji sentiment roles for sentiment analysis: A case study in Arabic texts. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 346–355, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. Araieval shared task: Persuasion techniques and disinformation detection in arabic text. In *ARABICNLP*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

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

Zainab Mansur, Nazlia Omar, and Sabrina Tiun. 2023. Twitter hate speech detection: A systematic review of methods, taxonomy analysis, challenges, and opportunities. *IEEE Access*, 11:16226–16249.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments. In *Workshop on Arabic Natural Language Processing*.

Tharindu Ranasinghe and Marcos Zampieri. 2021. Multilingual offensive language identification for low-resource languages. *Transactions on Asian and Low-Resource Language Information Processing*, 21:1 – 13.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Bryan Tuck, Fatima Qachfar, Dainis Boumber, and Rakesh Verma. 2023. DetectiveRedasers at ArAIEval shared task: Leveraging transformer ensembles for Arabic deception detection. In *Proceedings of ArabicNLP 2023*, pages 494–501, Singapore (Hybrid). Association for Computational Linguistics.

Gökçe Uludoğan, Somaiyeh Dehghan, İnanç Arın, Elif Erol, Berrin Yanıkoğlu, 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.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

A Appendix

Subtask Data Similarity

Figure 3 and Figure 4 represent the text embeddings of subtask A and B training sets from the models `ConvBERT-Turkish-Cased` and `AraBERTv02-Twitter` in the Turkish and Arabic embedding spaces respectively using the dimensionality reduction algorithm T-SNE (Van der Maaten and Hinton, 2008). The T-SNE algorithm draws the similarities between neighbors using the student t-distribution. As illustrated in Figure 3, we have plotted 10.000 samples consisting of 860 Arabic tweets translated to Turkish and 9.140 Turkish Original tweets from Subtask A training set.

According to Figure 3, the Arabic tweets that were translated into Turkish from Subtask B closely resemble the original Turkish tweets found in the

training data for Subtask A. This observation is further supported by the findings presented in Table 1 Subtask A, which indicates that incorporating the additional translated data into the training process leads to an improvement in the F1-score.

In Figure 4, the Turkish-translated tweets are not as close to subtask B's original Arabic tweets and are in another cluster. This discrepancy has resulted in decreased performance in Subtask B when incorporated as additional translated training data, as shown in Table 1 Subtask B.

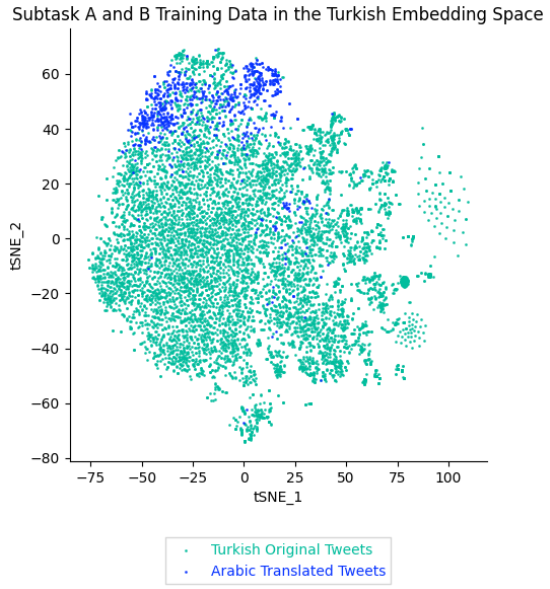

Figure 3: Training Data in Turkish Embedding Space using `ConvBERT-Turkish-Cased`

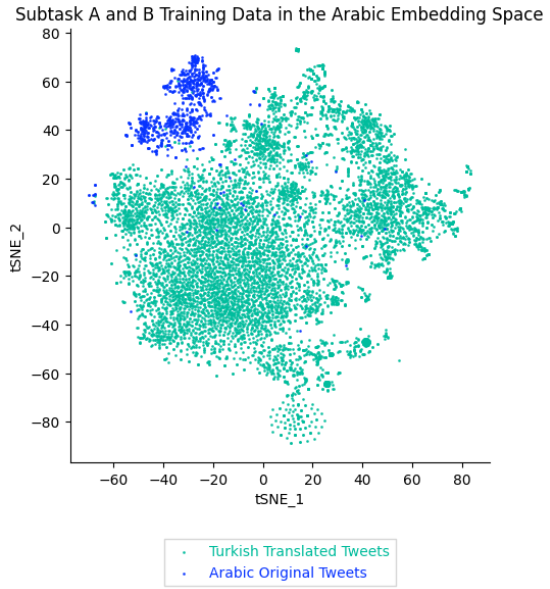

Figure 4: Training Data in Arabic Embedding Space using `AraBERTv02-Twitter`