StanceCrafters at StanceEval2024: Multi-task Stance Detection using BERT Ensemble with Attention Based Aggregation

Ahmed Abul Hasanaath[1] and **Aisha Alansari**[2]
g202302610@kfupm.edu.sa[1]
aisha.ansari@kfupm.edu.sa[2]

Information and Computer Science Department
King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Abstract

Stance detection is a key NLP problem that classifies a writer's viewpoint on a topic based on their writing. This paper outlines our approach for Stance Detection in Arabic Language Shared Task (StanceEval2024), focusing on attitudes towards the COVID-19 vaccine, digital transformation, and women's empowerment. The proposed model uses parallel multi-task learning with two fine-tuned BERT-based models combined via an attention module. Results indicate this ensemble outperforms a single BERT model, demonstrating the benefits of using BERT architectures trained on diverse datasets. Specifically, Arabert-Twitterv2, trained on tweets, and Camel-Lab, trained on Modern Standard Arabic (MSA), Dialectal Arabic (DA), and Classical Arabic (CA), allowed us to leverage diverse Arabic dialects and styles. The code has been made open-source on the link https://github.com/gufranSabri/StanceDetection-MultiTaskLearning.

1 Introduction

Stance detection or identification is a growing research topic in the fields of Natural Language Processing (NLP), social media analysis, and information retrieval (IR). It seeks to classify a writer's viewpoint on a certain topic based on their writing towards a target (e.g., notion, idea, event) (AlDayel and Magdy, 2021). It is known to have several applications, including predicting election/referendum results, retrieving information, and classifying rumors (Küçük and Can, 2021). Stance detection is known to be closely related to sentiment analysis, with the target classes Favor, Against, and None (Feldman, 2013). It is also related to other tasks, such as sarcasm detection and irony detection. Despite its relevance, it has gotten little attention in the Arabic language.

Data-driven neural networks have demonstrated outstanding performance for a wide variety of tasks. Traditionally, neural networks are trained separately for each task. However, neural networks frequently require extensive labeled training samples to achieve superior results (Ahmed et al., 2023). Given the limited data for Arabic stance identification and the availability of related tasks, it is necessary to employ novel techniques to construct robust models.

Multi-task learning (MTL) is an approach that handles numerous tasks concurrently and can leverage knowledge from related tasks to address data shortages (Thung and Wee, 2018). It assumes that all learning processes, or at least a portion of them, are related to one another. Learning from multiple common tasks simultaneously allows models to collect generic information beyond task-specific characteristics (Zhang and Yang, 2021). MTL has been used in the field of Arabic stance identification with various weighting strategies, yielding better results than training a model using only the stance detection task (Alturayeif et al., 2023).

This paper details our work on the Stance Detection in Arabic Language Shared Task (StanceEval2024) (Alturayeif et al., 2024). The StanceEval2024 shared task identifies writers' attitudes (Favour, Against, or None) on COVID-19 vaccination, digital transformation, and women empowerment. The key challenge in this task is imbalance of importance between the tasks. Therefore, we apply parallel MTL combining sentiment, sarcasm, and stance tasks to share knowledge across multiple tasks. Moreover, we utilize diverse weighting loss techniques to give more importance to the primary task. The remaining sections of the paper are organized as follows: Section 2 describes the StanceEval2024 dataset, Section 3 describes the methodology, Section 4 demonstrates the experimental results, Section 5 discusses the key findings, and Section 6 concludes the work. The code has been made open-source and available on GitHub on the link

Target	#Tweets	%Favor	%Against	Split
COVID-19 Vaccine	1,167	43.62%	43.53%	Train
COVID-19 Vaccine	206	43.69%	43.69%	Test
Digital Transformation	1,145	76.77%	12.40%	Train
Digital Transformation	203	76.85%	12.32%	Test
Women Empowerment	1,190	63.87%	31.18%	Train
Women Empowerment	210	63.81%	30.95%	Test
All	3,502	61.34%	29.15%	Train
All	619	61.39%	29.08%	Test

Table 1: Summary of MAWQIF dataset: Train/test set distribution of tweets across classes (number and percentage).

https://github.com/gufranSabri/StanceDetection-MultiTaskLearning.

2 Data

The MAWQIF dataset is a multi-label Arabic dataset designed for stance identification (Alturayeif et al., 2022). It includes 4121 tweets around three topics: COVID-19 vaccine, digital transformation, and women's empowerment. Each tweet is annotated with stance, sentiment, and sarcasm. The dataset is separated into two sets: training and testing. Training accounts for 85% of the data, while testing accounts for 15%. Table 1 provides a summary of the number of tweets belonging to each category.

In this study, we only considered the stance labels: Against and Favour. The dataset is pre-processed using the ArabertPreprocessor using the bert-base-arabertv02-twitter model. The dataset was further pre-processed by removing all the hashtags, URLs, mentions, punctuation, and repeating characters, as well as applying normalization.

3 System

This study explored using multiple BERT models for stance detection in Arabic text. We began with a single BERT model in the shared layers and then, inspired by the work in (Dang et al., 2020) and (Ganaie et al., 2022), we experimented with combining two models (Arabert, Arabert-twitter, Camel-bert) to leverage their diverse training data. We evaluated different weighting and aggregation techniques (averaging, attention) to find the optimal combination (Figure 1). Finally, we explored various loss weighting methods (static, relative) to further improve performance.

During training, the following configurations were used: 20 epochs, learning rate of 2e-5, weight decay of 1e-5, dropout of 0.1, batch size of 4, seed value of 42, and max token length of 128. The models were validated based on the validation F1-score.

3.1 MTL

In MTL, parameter sharing techniques determine how models leverage information across tasks. This study employs hard parameter sharing, where the initial layers of a model are shared by all tasks, promoting the learning of generalizable features. Each task then has its own output layer to capture task-specific information. Additionally, we utilize joint training, a technique where all tasks are trained simultaneously. This allows the model to learn shared representations that benefit all tasks while also enabling each task to adapt to its specific requirements. This combination of hard parameter sharing and joint training fosters knowledge transfer and improves the model's overall performance (Zhang and Yang, 2021).

3.2 Aggregation techniques

Ensemble aggregation techniques are crucial in aggregating predictions from several models to improve a system's overall performance and robustness. In this study, we experimented with mean aggregation and attention-based aggregation.

3.2.1 Mean Aggregation

The mean aggregation technique is the simplest ensemble aggregation technique. It operates by averaging multiple models' predictions to produce the final prediction. Although it is simple and straightforward, this technique assumes that all models are equally useful, which is not always the case (Briskilal and Subalalitha, 2022).

3.2.2 Attention based aggregation

The attention aggregation technique overcomes the limitation of the averaging technique by assigning different weights to each model's predictions based on their relevance and considering the context of an input. It can dynamically choose the most suitable

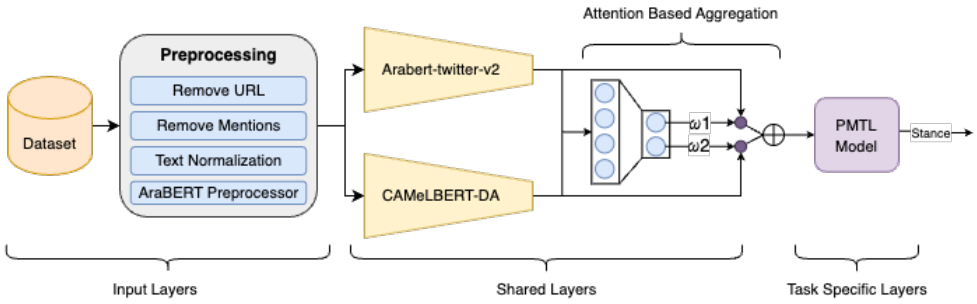

Figure 1: Stance Detection Ensemble Framework

models for each input, resulting in more accurate predictions (Mnassri et al., 2022).

3.3 Task weighting

In the MTL paradigm, the total loss for all tasks can be defined in various ways, including combining all loss terms with manual weights per loss (static) or using a weighted sum with learnable weights (dynamic) (Gong et al., 2019).

3.3.1 Static weighted sum

The static weighted sum calculates the total loss for each sentiment, sarcasm, and stance task while multiplying each loss by a given weight. The static weighted sum is formulated as shown in Equation 1.

$$\mathcal{L} = \sum_{i=1}^{n} w_i x_i \quad (1)$$

Where n refers to the number of tasks (stance, sentiment, and sarcasm), x_i refers to a specific loss value, and w_i refers to the weight multiplied by the corresponding loss. In this study, w_1 was set to 0.6, w_2 was set to 0.3, and w_3 was set to 0.1 (Alturayeif et al., 2023).

3.3.2 Relative weighted sum

The relative weighted sum is a dynamic weighting based on the intuition that tasks with higher losses should receive more attention. This method dynamically adjusts weights during optimization to allocate a larger weight to the stance loss, as shown in Equation **??**

$$\mathcal{L} = \omega x_{stance} + \frac{\omega}{2} x_{sentiment} + \frac{\omega}{3} x_{sarcasm} \quad (2)$$

Where ω is the learnable weight.

4 Results

In this section, the results of the developed models are reported during the development and test phases.

4.1 Development phase results

Table 2 summarizes the validation F1-score of fine-tuning Arabert-Twitter, Arabert, and CamelBert as single models and ensemble combinations. AraBERT-twitter has the highest F1-score when used alone, with a score of 89.8937, while CAMeL-Lab has the lowest F1-score among the single models, with a score of 84.9208. Interestingly, combining the best performing and weakest performing model resulted in the highest F1-score of 90.2293. Accordingly, we can conclude that combined models yield higher validation F1-score than solo models, likely because the combination of models can capture more diverse features and generalize better.

Table 3 indicates that using attention instead of averaging slightly improves the Arabert-twitter and Camel-bert ensemble. The attention ensemble method, when paired with a static weighting mechanism, has the best validation F1-score (91.3668). It demonstrates that employing a static weighting technique can be very effective in such a scenario.

4.2 Test phase results

Table 4 shows the Macro F1 score of the submitted model (Arabert-Twitter + CamelBert aggregated using attention mechanism with static weighting). The suggested model excelled at the Woman Empowerment and Digital Transformation tasks, but struggled with the COVID vaccination. Our model significantly surpassed the results obtained in the Digital Transformation task when compared to the Mawqif dataset results. Nonetheless, a minor decline is observed in the results related to COVID Vaccination and Woman Empowerment.

5 Discussion

Our suggested model exceeded the findings from the Mawqif dataset by 2.79%. We discovered that pairing Arabert-Twitter with CamelBert yielded

AraBERT-twitter	AraBERT	CAMeLBert	Ensemble method	Overall F1 Score
✓	✗	✗	-	**89.8937**
✗	✓	✗	-	88.9128
✗	✗	✓	-	84.9208
✓	✓	✗	average	88.2169
✓	✗	✓	average	**90.2293**
✗	✓	✓	average	89.1754
✓	✓	✓	average	89.1331

Table 2: F1 scores for all possible two BERT ensemble combinations on validation set

AraBERT-twitter	AraBERT	CAMeL-Lab	Ensemble method	Weighting Method	Overall F1 Score
✓	✗	✓	attention	equal	90.8483
✓	✗	✓	attention	static	**91.3668**
✓	✗	✓	attention	relative	90.7109

Table 3: F1 scores for aggregation and loss weighting methods on the best BERT ensemble on validation set

Women Empowerment	COVID Vaccine	Digital Transformation	Overall Score
85.06	79.84	80.14	81.68

Table 4: Macro F1 scores on the Blind Test Set (StanceEval2024) for each category and overall.

the greatest results. Arabert's success can be attributed to the Mawqif dataset, which was acquired via Twitter. Arabert-twitter was trained on tweets that were not Modern Standard Arabic (MSA), whereas CamelBert was trained on MSA, dialectal Arabic (DA), and classical Arabic (CA), consequently combining them enhanced the results. We also observed that aggregating the models using the attention mechanism performed better than the mean mechanism. The attention mechanism can outperform the mean mechanism in MTL since it dynamically weights model contributions based on their relevance. Moreover, we found that using static weighting leads to attaining the best performance. We assume that the reason behind this is the low confidence present in the dataset. Static weighting is less sensitive to noise and outliers in the loss values. the proposed framework performed relatively better on the Women's Empowerment target and relatively worse on the COVID Vaccine target. We believe the cause is that we fine-tuned the models using Against and Favour labels. Woman Empowerment has the fewest None labels, whereas COVID Vaccine has most None labels.

Encouraged by the success of MTL with attention-based combination, a future direction can be in regards to exploring Graph Neural Networks (GNNs). This would involve constructing a graph where each node represents a BERT model's output, and then combining them using techniques like graph convolution, similar to work in (Liu et al., 2018), (Du and Wang, 2022). Moreover, incorporating more weighting techniques, such as Hierarchical and Uncertainty weighting could be a direction to improve the results. Employing uncertainty sampling is another way to improve generalization with less data as suggested in (Pilault et al., 2020).

6 Conclusion

This study explored an MTL approach for stance detection in Arabic text, focusing on attitudes towards COVID-19 vaccines, digital transformation, and women's empowerment. The proposed model architecture included a shared task layer comprising Arabert-Twitter and CamelBert, which were combined via an attention mechanism, followed by a task-specific layer for each task. This ensemble approach, combined with static loss weighting, achieved a strong validation F1-score of 91.3668, outperforming architectures with a single BERT model in the shared layers. The results suggest that leveraging diverse training data through multi-task learning with attention-based combinations can improve the model's ability to capture various features and generalize better. On the test set, while the model excelled at the Women's Empowerment task (85.06 F1 score), it highlights the need for further development to improve performance on topics like COVID-19 vaccination (79.84 F1 score). Overall, the model achieved a promising macro F1 score of 81.68 on the blind test set.

Acknowledgments

We are deeply grateful to Dr. Hamzah Luqman for his support and guidance. We also thank KFUPM for providing essential resources, including GPUs.

References

Shams Forruque Ahmed, Md Sakib Bin Alam, Maruf Hassan, Mahtabin Rodela Rozbu, Taoseef Ishtiak, Nazifa Rafa, M Mofijur, ABM Shawkat Ali, and Amir H Gandomi. 2023. Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, 56(11):13521–13617.

Abeer AlDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4):102597.

Nora Alturayeif, Hamzah Luqman, and Moataz Ahmed. 2023. Enhancing stance detection through sequential weighted multi-task learning. *Social Network Analysis and Mining*, 14(1):7.

Nora Alturayeif, Hamzah Luqman, Zaid Alyafeai, and Asma Yamani. 2024. Stanceeval 2024: The first arabic stance detection shared task. In *Proceedings of The Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*.

Nora Saleh Alturayeif, Hamzah Abdullah Luqman, and Moataz Aly Kamaleldin Ahmed. 2022. Mawqif: A multi-label arabic dataset for target-specific stance detection. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 174–184.

J Briskilal and CN Subalalitha. 2022. An ensemble model for classifying idioms and literal texts using bert and roberta. *Information Processing & Management*, 59(1):102756.

Huong Dang, Kahyun Lee, Sam Henry, and Ozlem Uzuner. 2020. Ensemble bert for classifying medication-mentioning tweets. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 37–41.

Hang-Yuan Du and Wen-Jian Wang. 2022. A clustering ensemble framework with integration of data characteristics and structure information: a graph neural networks approach. *Mathematics*, 10(11):1834.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. 2019. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632.

Dilek Küçük and Fazli Can. 2021. Stance detection: Concepts, approaches, resources, and outstanding issues. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. *arXiv preprint arXiv:1809.09078*.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2022. Bert-based ensemble approaches for hate speech detection. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 4649–4654. IEEE.

Jonathan Pilault, Amine Elhattami, and Christopher Pal. 2020. Conditionally adaptive multi-task learning: Improving transfer learning in nlp using fewer parameters & less data. *arXiv preprint arXiv:2009.09139*.

Kim-Han Thung and Chong-Yaw Wee. 2018. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22):29705–29725.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.