DetectiveRedasers at ArAIEval Shared Task: Leveraging Transformer Ensembles for Arabic Deception Detection

Bryan E. Tuck
University of Houston
betuck@uh.edu

Fatima Zahra Qachfar
University of Houston
fqachfar@uh.edu

Dainis Boumber
University of Houston
dboumber@uh.edu

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

Abstract

This paper outlines a methodology aimed at combating disinformation in Arabic social media, a strategy that secured a first-place finish in tasks 2A and 2B at the ArAIEval shared task during the ArabicNLP 2023 conference. Our team developed a hyperparameter-optimized pipeline centered around BERT-based models for the Arabic language, enhanced by a soft-voting ensemble strategy. Subsequent evaluation on the test dataset reveals that ensembles, although generally resilient, do not always outperform individual models. The primary contributions of this paper are its multifaceted strategy, which led to winning solutions for both binary (2A) and multiclass (2B) disinformation classification tasks.

1 Introduction

The spread of disinformation across social media platforms presents an omnipresent challenge that transcends modalities, manifesting in text, audio, and images (Shu et al., 2020). Within the sphere of text, disinformation is not exclusive to one language but spans many languages and dialects. Its impact permeates several topics, including politics, entertainment, sports, and finance. In our study, we direct our efforts to detecting disinformation in Arabic on Twitter as part of the Shared Task 2: Arabic Deception Detection, at ArAIEval, ArabicNLP 2023 (Hasanain et al., 2023).

From a research standpoint, Arabic has been receiving an increasing amount of attention addressing several key problems (Farghaly and Shaalan, 2009). Investigations into disinformation in Arabic can offer valuable insights into unique linguistic and cultural aspects that influence the dissemination and impact of false information within Arabic-speaking communities. In a social aspect, the consequences of disinformation across all global communities are profound. Specifically, Arabic is spoken by hundreds of millions of people worldwide and serves as a linguistic backbone for critical geopolitical regions. While also not exclusive to Arabic, disinformation can impact democratic processes and public health (Wolfsfeld et al., 2013). Research in this critical domain has real-world implications that can influence policy decisions, governance, and public well-being.

Arabic itself presents its own set of complexities; it is a rich language featuring intricate word formations and variations (Alzanin et al., 2022). This makes the language both highly derivational, meaning words can be formed from root words in various ways, and inflectional, indicating that the form of words can change to convey different meanings. These linguistic traits add an extra layer of difficulty to the already challenging task of disinformation detection.

Shared task 2 comprises two separate sub-tasks. Task 2A is a binary classification challenge requiring us to categorize whether a given tweet is disinformative. Task 2B, on the other hand, is a more nuanced multiclass classification task, where the objective is to identify fine-grained disinformation classes such as hate speech, offensive content, rumors, or spam (Mubarak et al., 2023b). With this task, there are several open problems due to phenomena including code-switching (Bentahila and Davies, 1983), short texts, and lack of grammatical structure in tweets. These issues lead to the deterioration of the effectiveness of conventional analytical tools. Code-switching refers to the practice of switching between languages within a conversation, or text. Tweets tend to mirror the linguistic styles and variations spoken by individuals hailing from a particular region. For example, Moroccan tweets contain Moroccan Darija mixed with French, English, or Spanish. This phenomenon can occur for various reasons, including cultural exchange, or historical factors such as colonization.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 494–501

In addressing disinformation detection in Arabic, our multi-faceted strategy begins with specialized preprocessing, including handling code-switching and incorporating tweet elements like hashtags and URLs, which previous literature often neglects (Bennessir et al., 2022). We then utilize large language models, specifically AraBERT (Antoun et al., 2020), and experiment with a soft-voting ensemble to improve performance. While effective, these large models are computationally expensive; we seek to mitigate this through optimization pipelines, which in turn add their own computational overhead.

2 Dataset and Tasks

In the ArAIEval shared task at ArabicNLP 2023, participants are presented with two main tasks: task 1 focuses on Persuasion Technique Detection, while task 2 aims at Disinformation Detection. Each of these primary tasks are further divided into two sub-tasks. Our research specifically concentrates on task 2, which consists of sub-task 2A and sub-task 2B. In sub-task 2A, the goal is to classify tweets as either disinformative or not, a binary classification problem. For sub-task 2B, we must identify specific types of disinformation within a tweet, which involves a multiclass classification framework. The fine-grained labels that we consider include hate speech, offensive language, rumors, and spam (Hasanain et al., 2023)(Mubarak et al., 2023a). Tables 1 and 2 represent the class distributions and total size of the training, validation, and testing sets, for task 2A and 2B, respectively.

	No Disinformation	Disinformation	Total
Training	11491	2656	14147
Dev	1718	397	2115
Test	2853	876	3729

Table 1: Class Distribution for Task 2A

	Hate-Speech	Offensive	Rumor	Spam	Total
Training	1512	500	191	453	2656
Dev	226	75	28	68	397
Test	442	160	33	241	876

Table 2: Class Distribution for Task 2B

3 System

For tasks 2A and 2B, our approach adopts a specialized methodology using comprehensive preprocessing which deals with code-switching and emoji conversion. After which an intensive search for optimal large language models and hyperparameters is performed. Our decisions of which models to utilize were based on performance on the validation set. The AraBERT-Covid19 model (Antoun et al., 2020) surfaced as the best fit for task 2A. This model, an enhancement of the original AraBERTv02, has been further refined through fine-tuning on 1.5 million multi-dialect Arabic tweets. These tweets, sourced from the extensive Arabic Twitter dataset (Alqurashi et al., 2020), specifically focused on Covid-19. Conversely, for task 2B, we utilize AraBERTv02-Twitter, which was pre-trained on approximately 60 million tweets spanning various Arabic dialects. Subsequently, we employ a soft voting ensemble method, integrating five AraBERTv02-Twitter models that have been optimized. While each model maintains identical hyperparameters and architecture, they differ solely in terms of random initialization. For this process, we utilized the TorchEnsemble library[1]. We optimize both AraBERTv02-Twitter and AraBERT-Covid19 models leveraging the optimization framework Optuna (Akiba et al., 2019). By the deadline for task 2, only two optimized models were evaluated: the AraBERT-Covid19 model for task 2A and the AraBERTv02-Twitter ensemble for task 2B.

To ensure the best performance in regards to our target metric, "micro f1", we explored a variety of models. Our initial model candidate list included the following: *a*) AraBERTv02-Twitter (Antoun et al., 2020) *b*) Arabert-Covid19 (Alqurashi et al., 2020) *c*) QCRI Arabic and Dialectal BERT (QARiB) (Abdelali et al., 2021) *d*) MARBERTV2 (Abdul-Mageed et al., 2021) *e*) and CAMeLBERT-DA SA (Inoue et al., 2021). Post-competition experimentation can be found in A.1.

3.1 Preprocessing

The Arab world has a rich and diverse history of languages, with many different dialects spoken across different regions. We have analyzed the provided data in both tasks using dialect identification, and we have found that most tweets in the dataset originated from the Kingdom of Saudi Arabia (KSA), Kuwait, and Egypt. We report these results in detail in Appendix A.

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

3.1.1 Code Switching

Arabic tweeters may use code-switching to express themselves more effectively or to communicate with a diverse audience. For example, users may start a tweet in Arabic, switch to English in the middle, and then finish it off in French. We now describe the preprocessing techniques we applied to the tweets to translate code-switched text to Arabic. For each tweet, we automatically detect code-switching fragments using "*Lingua*" [2] Python package, and we translate it to Arabic using Google's translation API.

3.1.2 Emoji Conversion

In tweets, emojis are typically used to convey emotions or ideas. Mubarak et al. (2022) showed the importance of emojis in the detection of Arabic offensive language and hateful speech.

Instead of removing all emojis from tweets like (Bennessir et al., 2022), we choose to convert them to Arabic descriptive text since emojis might hold meaning in the context of a short deceptive tweet representing positive or negative sentiment. For this we add Arabic language support to the "*emoji*" [3] Python package using normalized representations from the latest release of Unicode Common Locale Data Repository (CLDR) [4] to avoid broken Unicode. We create a dictionary of Arabic emoji representation based on the *emojiterra* website.[5]

3.2 Hyperparameter Optimization

We use the Optuna framework (Akiba et al., 2019) for hyperparameter optimization, primarily due to its straightforward setup, versatility, and choices of efficient sampling and pruning algorithms. For tasks 2A and 2B, we opted for the Tree-Structured Parzen Estimator (TPE) (Bergstra et al., 2011) as our sampling method, as it offers superior efficiency compared to traditional grid search techniques. We began the optimization process with multivariate and grouping settings, integrating a Hyperband pruner (Li et al., 2018), stopping unpromising trials early. This setup allowed each trial to run for a duration ranging from two to twelve epochs. The optimization process encompassed 100 trials aimed at maximizing the "micro-f1" metric, the search space is detailed in Table 3, with the addition of the five candidate models outlined in Section 3. Post-competition, we continued to fine-tune individual models under the same conditions, and these results, along with original task hyperparameters, are located in tables 5, 6, 7, and 8 in Appendix A.1.

Parameter	Value
Learning Rate	1e-05 - 5e-05
Batch Size	8, 16, 32, 64
Dropout	0.0 - 0.5
Max Length	32 - 128

Table 3: Optuna Search Space

3.3 Voting Ensemble

Ensembling techniques, like hyperparameter optimization, come with computational expenses and tuning complexities. The success of ensemble methods hinges on several factors, including the training process of the baseline models (Mohammed and Kora, 2023). Our ensemble employs a "soft voting" scheme, guided by the performance of our top individual model identified through hyperparameter optimization. In this configuration, we employ five AraBERTv02-Twitter models for task 2A and five AraBERTv02-Covid19 models for task 2B, each optimized according to the parameters specified in Table 3. The ensemble is trained for two epochs, which was found to be the point of peak validation performance.

In the soft voting mechanism (Zhou, 2012), each individual classifier, denoted as h_i, generates a l-dimensional vector $(h_i^1(x)..., h_i^l(\mathbf{x}))^T$ for a given instance $\mathbf{x}$. Here $h_i^j(\mathbf{x})$ represents the estimated posterior probability $P(c_j|\mathbf{x})$ and falls within the range of $[0, 1]$. The final output for class c_j is the average of all individual outputs, represented as follows:

$$H^j(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} h_j^i(\mathbf{x})$$

3.4 Training Procedure

Our optimization and fine-tuning pipeline uses the AdamW optimizer for effective parameter updates and Cross Entropy as the loss function, given its efficacy in classification problems. We use early stopping with five epochs as a stopping criteria, saving the model best last state. To expedite training without compromising model quality, we uti-

[2] https://github.com/pemistahl/lingua

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

[4] https://github.com/unicode-org/cldr/raw/release-43/common/annotations/ar.xml

[5] https://emojiterra.com/copypaste/ar/

Task	Model	Validation		Test	
		Macro F1	Micro F1	Macro F1	Micro F1
Task 2A	AraBERT-Covid19	**84.73%**	**91.06%**	**86.26%**	**90.48%**
	AraBERT-Covid19 Ensemble	84.31%	90.58%	85.84%	90.02%
Task 2B	AraBERTv02-Twitter	81.12%	84.89%	**75.51%**	**84.36%**
	AraBERTv02-Twitter-Ensemble	**82.19%**	**85.14%**	75.41%	83.56%

Table 4: Micro F1 and Macro F1 scores are presented for each task, covering both validation and testing sets. The highest values are highlighted in **bold**.

lize automatic mixed precision (AMP)[6], reducing both memory usage and training time. Notably, we choose not to employ a learning rate scheduler, deviating from some traditional approaches. As a safety measure, we also implement gradient clipping with a maximum norm of 1.0 to ensure numerical stability and avoid issues like the exploding gradient problem.

4 Results and Discussion

The top two candidate models, identified through hyperparameter optimization, were AraBERT-Covid19 for task 2A and AraBERTv02-Twitter for task 2B. These selections represented the only results submitted by the task deadline. Our results presented in Table 4 reveal some compelling patterns and anomalies. Specifically, task 2A favored the single AraBERT-Covid19 model over its ensemble counterpart. This approach led by a noticeable margin of 0.48% macro f1 and 0.43% micro f1 with the validation set.

Task 2B presents a more intricate challenge, which utilizes AraBERTv02-Twitter as the primary model. While the AraBERTv02-twitter ensemble performed better during the validation phase, it was ultimately outperformed by the single AraBERTv02-Twitter model in the test set by 0.1% macro f1 and 0.8% micro f1. The drop in macro f1 scores from the validation to the test set in task 2B suggests an issue with model generalization. This might be attributed to the inherent complexity of multiclass problems, which often require capturing more nuanced relationships in the data. This presents a challenging task compared to a binary classification task like task 2A. Another challenge for task 2B is the smaller dataset in comparison to task 2A, which can be seen in Section 2, Table 2 and Table 1 respectively. With the unbalanced nature of task 2 as a whole, the small dataset size, and a more intricate class balancing issue, our approach may have failed to learn minority classes, overfitting to the majority classes.

It's also important to highlight that we did not fine-tune the ensemble's hyperparameters, which could have contributed to its less-than-optimal performance against the single models. This supports the idea that ensemble methods, while often robust, require task-specific validation. In future work, optimization techniques specifically for ensembles and not just the individual models may prove to be beneficial, such as an varied amount of classifiers in the ensemble or different weighting techniques. The exploration of additional preprocessing techniques to better handle code-switching could also be a beneficial avenue.

Our results reiterate the importance of nuanced model selection, especially given the challenges posed by binary and multiclass classification tasks. Our findings also pave the way for future work focused on improving computational efficiency and generalization capabilities of disinformation detection models.

5 Conclusion

In this study, we tackled the nuanced problem of disinformation detection in Arabic, a language fraught with complexities like code-switching and dialectal variations. We combined meticulous preprocessing with hyperparameter-optimized AraBERT models, effectively achieving first-place performance in both binary and multiclass deception detection tasks at ArAIEval 2023. A notable insight from our empirical analysis is that individual models occasionally outperform ensembles, indicating the need for careful model selection. Our results not only validate our comprehensive approach but also invite further research into optimizing ensemble methods and addressing the challenges associated with code-switching and dialectal variations in Arabic text. Future work should look at refining these ensemble strategies and explor-

[6] https://pytorch.org/docs/stable/amp.html

ing additional preprocessing techniques, as we aim to create universally effective tools for countering disinformation.

Limitations

While our methodology is proven to work well for the Arabic language and the disinformation detection task, it may not transfer as well to other languages or other domains. Further experimentation on other languages and domains would be required to evaluate the overall efficacy of our pipelines. Lack of time with respect to the task did not allow us to delve into ensemble optimization or explore other possible ensembling techniques. The computational complexity of hyperparameter optimization with additional overhead from transformer architectures and ensemble methods may lead to scaling issues with larger datasets and other domains.

Ethics Statement

Our work complies with the ACL Ethics Policy. We report details of the hyperparameters and architectures for reproducibility. We plan to make the pipeline available in the near future for the benefit of other researchers.

Acknowledgements

We thank the ArAIEval reviewers for their constructive suggestions. This research was partly supported by NSF grants 1433817, 1950297, and 2210198, ARO grants W911NF-20-1-0254 and W911NF-23-1-0191, and ONR award N00014-19-S-F009. Verma is the founder of Everest Cyber Security and Analytics, Inc.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *ArXiv preprint*, abs/2102.10684.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2623–2631. ACM.

Sarah Alqurashi, Ahmad Alhindi, and Eisa Alanazi. 2020. Large arabic twitter dataset on covid-19. *ArXiv preprint*, abs/2004.04315.

Samah M. Alzanin, Aqil M. Azmi, and Hatim A. Aboalsamh. 2022. Short text classification for Arabic social media tweets. *Journal of King Saud University - Computer and Information Sciences*, 34(9):6595–6604.

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.

Mohamed Aziz Bennessir, Malek Rhouma, Hatem Haddad, and Chayma Fourati. 2022. iCompass at Arabic hate speech 2022: Detect hate speech using QRNN and transformers. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 176–180, Marseille, France. European Language Resources Association.

Abdelali Bentahila and Eirlys E Davies. 1983. The syntax of arabic-french code-switching. *Lingua*, 59(4):301–330.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2546–2554.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8:1–.

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 *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, 18(185):1–52.

Ammar Mohammed and Rania Kora. 2023. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35(2):757–774.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023a. Detecting and identifying the reasons for deleted tweets before they are posted. *Frontiers in Artificial Intelligence*, 6.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023b. Detecting and reasoning of deleted tweets before they are posted. *ArXiv preprint*, abs/2305.04927.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2022. Emojis as anchors to detect arabic offensive language and hate speech.

Kai Shu, Amrita Bhattacharjee, Faisal Hammad Alatawi, Tahora H. Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. 2020. Combating disinformation in a social media age. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10.

Gadi Wolfsfeld, Elad Segev, and Tamir Sheafer. 2013. Social Media and the Arab Spring: Politics Comes First. *The International Journal of Press/Politics*, 18(2):115–137. Publisher: SAGE Publications Inc.

Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms*, 1st edition. Chapman & Hall/CRC.

A Appendix

A.1 Experimental Results

In this section, we present our continued experimentation. Instead of including all the models in the search space, individual hyperparameter optimizations were conducted on each model. This resulted in hyperparameters that differed from those in our original experiments. These results are displayed below in tables 5 and 6. Macro precision, recall, F1-score, and accuracy are reported.

In task 2A, QARIB secures the second-highest precision on the validation set and the highest on the test set, suggesting its proficiency in accurately identifying positive classes and minimizing false positives. AraBERTv02-Twitter leads recall for both validation and test sets, indicating its strength in identifying actual positive instances. Both QARIB and AraBERTv02-Twitter demonstrate robust performance, leading in various metrics.

For task 2B, AraBERTv02-Twitter continues its strong performance, showing the highest precision on the test set. Meanwhile, AraBERT-Covid19 achieves the highest recall and F1-score across both sets, indicating a balanced strength in precision and recall, closely followed by MARBERTv2.

The results underscore that no single model consistently outperforms across all metrics, suggesting that model selection should consider the specific performance metrics of interest. The varied leadership in different metrics across both tasks implies a lack of a universally superior model.

Ultimately, our findings revealed a distinct set of optimal parameters divergent from those in our original search space, which encompassed all candidate models. The specifics of these parameters are detailed in tables 7 and 8. Interestingly, for task 2A, the AraBERT-Covid19 model exhibited superior performance with parameters derived from our initial, more generalized search space, as opposed to those obtained from a model-specific search. In contrast, for task 2B, the AraBERTv02-Twitter model demonstrated enhanced performance when employing parameters from a search space tailored for that specific model.

A.2 Dialect Language Identification

For Arabic dialect language detection, we used the "bert-base-arabic" model (Inoue et al., 2021) provided by CAMel (Computational Approaches to Modeling Language) Laboratory on the HuggingFace Hub [7] trained on MADAR (Bouamor et al., 2018) Twitter dataset which contains Arabic dialect tweets originating from 25 regions. We show in Figure 1 and Figure 2 the distribution of dialects in the Training and Development Sets for tasks 2A and 2B.

The top three dialects used in the provided data are from Saudi Arabia, Kuwait and Egypt. While

[7] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-msa-did-madar-twitter5

Model	Validation				Test			
	Prec.	Recall	F1	Acc.	Prec.	Recall	F1	Acc.
AraBERT-Covid19	85.73%	83.31%	84.44%	**90.83%**	86.94%	84.10%	85.39%	89.89%
AraBERTv02-Twitter	83.72%	**85.32%**	**84.48%**	90.31%	86.99%	**86.96%**	**86.98%**	**90.64%**
QARIB	85.01%	82.79%	83.83%	90.45%	**87.81%**	84.77%	86.14%	90.43%
MARBERTv2	**86.14%**	81.52%	83.54%	90.59%	86.68%	82.66%	84.40%	89.38%
CAMeLBERT-DA SA	84.63%	81.37%	82.85%	90.02%	86.48%	83.51%	84.85%	89.54%

Table 5: Task 2A hyperparameter optimized models post-hoc comparison of macro validation and test metrics. Highest values are in **bold**.

Model	Validation				Test			
	Prec.	Recall	F1	Acc.	Prec.	Recall	F1	Acc.
AraBERT-Covid19	83.15%	**78.91%**	**80.79%**	84.13%	73.36%	**74.29%**	**73.55%**	81.85%
AraBERTv02-Twitter	**85.99%**	75.30%	79.15%	83.88%	**75.84%**	71.39%	73.21%	**82.65%**
QARIB	85.40%	75.45%	78.49%	**84.38%**	73.97%	72.65%	71.92%	81.85%
MARBERTv2	83.51%	75.53%	78.27%	83.38%	74.79%	73.57%	73.54%	82.08%
CAMeLBERT-DA SA	78.78%	76.31%	76.90%	83.12%	70.14%	73.14%	70.28%	80.02%

Table 6: Task 2B hyperparameter optimized models post-hoc comparison of macro validation and test metrics. Highest values are in **bold**.

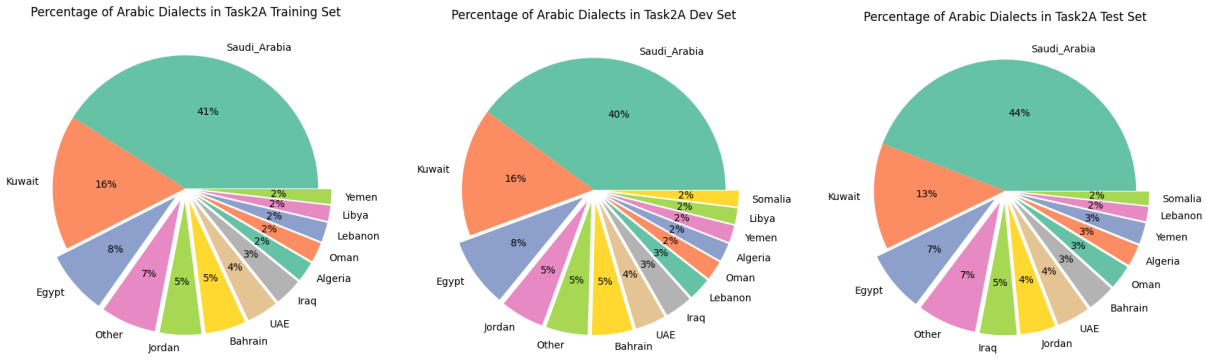
Model	Learning Rate	Batch Size	Dropout	Max Length
AraBERT-Covid19	1.38e-05	32	0.325	115
AraBERT-Covid19 *	1.0e-05	8	0.375	78
AraBERTv02-Twitter	1.74e-05	64	0.0	79
QARIB	1.73e-05	32	0.15	94
MARBERTv2	1.03e-05	64	0.5	99
CAMeLBERT-DA SA	1.62e-05	16	0.0	67

Table 7: Task 2A best hyperparameters for each model, determined post-hoc. Models marked with an asterisk (*) indicate the hyperparameters of the task submitted model.

Model	Learning Rate	Batch Size	Dropout	Max Length
AraBERT-Covid19	1.14e-05	8	0.25	88
AraBERTv02-Twitter	2.82e-05	32	0.2	100
AraBERTv02-Twitter*	5.0e-05	64	0.4	57
QARIB	2.00e-05	32	0.4	93
MARBERTv2	1.17e-05	8	0.1	60
CAMeLBERT-DA SA	1.32e-05	32	0.125	91

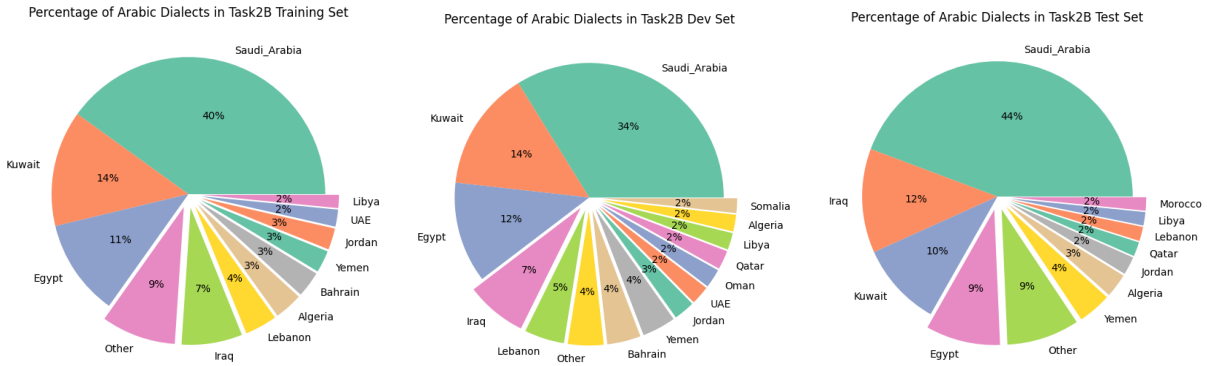
Table 8: Task 2B best hyperparameters for each model, determined post-hoc. Models marked with an asterisk (*) indicate the hyperparameters of the task submitted model.

the top three represent about 65% and 64% of the datasets for Task 2A, their percentage drops off particularly in task 2B Development set to 60% whereas the task 2B Training set is still at 65%. Thus, dialect-wise task 2B showed much more variation. The high concentrations of specific dialects imply that our models are significantly influenced by the linguistic features of Saudi Arabia, Kuwait, and Egypt. Upon reviewing the generalization error in Table 6, which compares the validation to testing set metrics, we hypothesize that this dialect variance may adversely affect model generalization. Such variance can introduce additional complexity and nuance to the classification task. When training a language model on a dataset largely influenced by three dialects and then tests it on a broader dialectal range, the model may find it challenging to generalize effectively.



(a) Training Set (b) Development Set (c) Test Set

Figure 1: Task 2A - Arabic Dialect Language Identification



(a) Training Set (b) Development Set (c) Test Set

Figure 2: Task 2B - Arabic Dialect Language Identification