NileUn at AbjadGenEval Shared Task: Contrastive Learning with Stacking Ensemble for Efficient Arabic AI-Generated Text Detection

Mohamed Hussein
CIS Center, Nile University
m.hussein2558@nu.edu.eg

Shrouk Shalaby
CIS Center, Nile University
S.Shalaby@nu.edu.eg

Nesreen Mohamed
CIS Center, Nile University
N.Mohamed2442@nu.edu.eg

Abstract

We present a computationally efficient approach for detecting AI-generated Arabic text as part of the AbjadGenEval shared task. Our method combines Supervised Contrastive Learning with a Stacking Ensemble of AraBERT and XLM-RoBERTa models. Our training pipeline progresses through three stages: (1) standard fine-tuning without contrastive loss, (2) adding supervised contrastive loss for better embeddings, and (3) further fine-tuning on diverse generation styles. On our held-out test split, the stacking ensemble achieves F1=0.983 before fine-tuning. On the official workshop test data, our system achieved 4th place with F1=0.782, demonstrating strong generalization using only encoder-based transformers without requiring large language models. Our implementation is publicly available.[1]

1 Introduction

Large language models capable of generating human-like text raise concerns about content authenticity and academic integrity (Wu et al., 2025). While AI-generated text detection has been studied extensively for English, Arabic remains underexplored (Guellil et al., 2021).

The M-DAIGT shared task (Lamsiyah et al., 2025), AraGenEval shared task (Abudalfa et al., 2025), and the AbjadGenEval shared task (Ezzini et al., 2026) address this gap through AI-generated text detection for Arabic-script languages. We participate in the binary classification subtask: distinguishing human-written from AI-generated Arabic text. We contribute:

1. **Supervised Contrastive Learning**: We add contrastive loss to cross-entropy, pulling same-class embeddings together while pushing different classes apart.
2. **Stacking Ensemble**: A meta-learner that optimally weighs model predictions based on confidence scores, outperforming simple majority voting.
3. **Three-stage Training**: We first train without contrastive loss, then add contrastive learning, and finally fine-tune on diverse AI generation styles.

Our approach achieves strong results on validation (F1=0.983) and secured 4th place on the official test set (F1=0.782), all without expensive LLM fine-tuning.

2 Background

2.1 Task Setup

The ARATECT dataset contains 5,298 balanced training samples (50% human, 50% machine-generated by Mistral, GPT-4, and LLaMA) with 200 unlabeled test samples. Table 1 shows the distribution.

Split	Train	Test
Samples	5,298	200
Human (%)	50	–
Machine (%)	50	–

Table 1: ARATECT dataset distribution.

For further fine-tuning, we use the Arabic Generated Abstracts dataset[2] containing 8,388 samples from four AI models (Allam, Jais, LLaMA, OpenAI) across three generation strategies: text polishing, generation from title, and generation from title with content.

2.2 Related Work

Alshammari et al. (2024) fine-tuned AraELECTRA and XLM-R achieving 83% accuracy. Alghamdi

[1] https://www.kaggle.com/code/se7s609/abjadgeneval-arabic-ai-detection-contrastive-ens

[2] https://huggingface.co/datasets/KFUPM-JRCAI/arabic-generated-abstracts

Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), pages 489–492
March 28, 2026. ©2026 Association for Computational Linguistics

and Alowibdi (2024) used traditional ML on Arabic tweets with Naive Bayes reaching 93% in that domain-specific setting.

Contrastive learning has proven effective for representation learning (Khosla et al., 2020), and ensemble methods improve text classification (Dong et al., 2020). We combine both for Arabic AI detection.

3 System Overview

3.1 Model Architecture

We use AraBERT (Antoun et al., 2020), pre-trained on 1.5B Arabic words, and XLM-RoBERTa (Conneau et al., 2020), a multilingual model covering 100+ languages. Both have 136M parameters each.

Each model has two heads: (1) a classification head for binary prediction, and (2) a projection head mapping to 256-dimensional space for contrastive learning.

3.2 Training Pipeline

Our training follows three stages:

Stage 1: Standard Fine-tuning. We first train with cross-entropy loss only to establish baseline performance.

Stage 2: Adding Contrastive Loss. We then train with combined loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \cdot \mathcal{L}_{SCL} \tag{1}$$

where $\lambda = 0.3$. The supervised contrastive loss (Khosla et al., 2020):

$$\mathcal{L}_{SCL} = -\sum_i \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{e^{z_i \cdot z_p / \tau}}{\sum_{a \neq i} e^{z_i \cdot z_a / \tau}} \tag{2}$$

pulls same-class embeddings together while pushing different classes apart. Here $P(i)$ is the set of positive samples, z are L2-normalized embeddings, and $\tau = 0.1$.

Stage 3: Further Fine-tuning. Finally, we fine-tune on the Arabic Generated Abstracts dataset with reduced learning rate (1×10^{-5}) to prevent catastrophic forgetting while learning diverse AI patterns.

Figure 1 shows the embedding improvement after contrastive training.

3.3 Stacking Ensemble

Instead of majority voting, we use logistic regression to combine model outputs:

$$\hat{y} = \sigma(w_1 \cdot p_1 + w_2 \cdot p_2 + b) \tag{3}$$

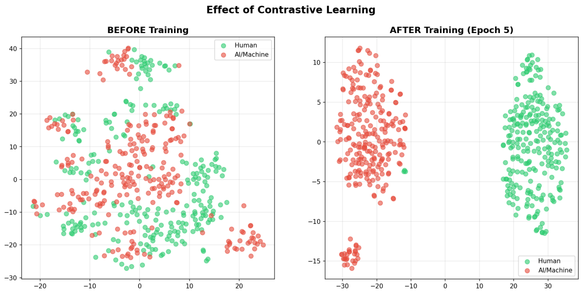

Figure 1: t-SNE of AraBERT embeddings before (left) and after (right) contrastive training, showing clearer class separation.

Model	F1	Acc	Prec	Rec
Before Further Fine-tuning				
AraBERT	0.978	0.977	0.960	0.996
XLM-R	0.965	0.964	0.936	0.996
Stacking	**0.983**	**0.983**	**0.971**	**0.996**
After Further Fine-tuning				
AraBERT	0.912	0.904	0.839	1.000
XLM-R	0.788	0.730	0.650	1.000
Stacking	0.922	0.915	0.855	1.000

Table 2: Results on held-out test split with contrastive learning. The stacking ensemble achieves F1=0.983 before fine-tuning.

This leverages confidence information: when one model is uncertain but another is confident, stacking weighs their contributions appropriately.

4 Experimental Setup

We split the ARATECT data 80/10/10 (4,238/530/530 samples) for train/validation/test. The Arabic Generated Abstracts dataset (8,388 samples) uses 90/10 split for further fine-tuning.

Hyperparameters: Batch size 16, learning rate 2×10^{-5} (halved for fine-tuning), 5 epochs initial training, 3 epochs fine-tuning, max length 512, AdamW with weight decay 0.01.

Environment: Kaggle with dual T4 GPUs, PyTorch, Hugging Face Transformers, mixed-precision training.

5 Results

5.1 Results on Training Split

We evaluate on our held-out test split (530 samples). Table 2 shows results across training stages.

5.2 Official Workshop Results

On the official AbjadGenEval workshop test data (200 samples), our stacking ensemble with further fine-tuning achieved **4th place** with F1=0.782,

Model	F1	Acc	Prec	Rec
Without Contrastive Loss				
AraBERT	0.53	0.57	0.60	0.48
XLM-R	0.55	0.58	0.61	0.50
Stacking	0.56	0.59	0.62	0.51
With Contrastive Loss				
AraBERT	0.58	0.62	0.65	0.53
XLM-R	0.60	0.63	0.66	0.55
Stacking	0.61	0.64	0.67	0.56

Table 3: Ablation study: contrastive loss provides +0.05 F1 improvement.

Accuracy=0.74, Precision=0.66, and Recall=0.95. The performance gap between our held-out split and the official test data suggests distribution shift between training and test sets.

5.3 Ablation: Effect of Contrastive Loss

Table 3 compares training with and without contrastive loss. Adding contrastive learning improves stacking F1 from 0.56 to 0.61 (+9% relative).

5.4 Analysis

Training Split vs Official Test: On our held-out split, the stacking ensemble achieves F1=0.983, demonstrating strong learning. However, on the official test data, performance drops to F1=0.782, indicating distribution shift between training and test sets.

Effect of Further Fine-tuning: Interestingly, further fine-tuning on diverse AI styles slightly reduces performance on our held-out split (F1 from 0.983 to 0.922) but was designed to improve generalization to unseen AI patterns in the official test.

Precision-Recall Trade-off: On the official test, our system achieves high recall (0.95) but lower precision (0.66). This suits applications where missing AI content is costly, like academic integrity checking.

Computational Efficiency: Our encoder-only approach uses models with 136M parameters each, making it practical for resource-constrained deployment without requiring large language models.

6 Conclusion

We presented an efficient Arabic AI text detection system combining contrastive learning with stacking ensemble. Our system achieves F1=0.983 on our held-out test split and secured 4th place on the official AbjadGenEval workshop test data with F1=0.782.

Key findings: (1) Contrastive learning improves embedding quality, (2) stacking ensemble outperforms individual models and majority voting, (3) significant distribution shift exists between training and official test data.

Limitations: The performance gap between validation (F1=0.983) and official test (F1=0.782) indicates overfitting to training distribution. High recall but moderate precision on official test may not suit all applications.

Future Work: Investigating domain adaptation techniques to reduce distribution shift, extending to multi-class detection to identify specific AI models, and evaluating on Arabic dialects beyond MSA.

Acknowledgments

We thank the AbjadGenEval and AraGenEval shared task organizers for the dataset and evaluation framework.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval shared task on Arabic authorship style transfer and AI generated text detection. In *Proceedings of The Third Arabic Natural Language Processing Conference at EMNLP 2025*, Suzhou, China. Association for Computational Linguistics.

Noura Saad Alghamdi and Jalal Suliman Alowibdi. 2024. Distinguishing Arabic GenAI-generated tweets and human tweets utilizing machine learning. *Engineering, Technology & Applied Science Research*, 14(5):16720–16726.

Hamed Alshammari, Ahmed El-Sayed, and Khaled Elleithy. 2024. AI-generated text detector for Arabic language using encoder-based transformer architecture. *Big Data and Cognitive Computing*, 8(3).

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*, pages 9–15, Marseille, France. European Language Resource Association.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258.

Saad Ezzini, Irfan Ahmed, Salmane Chafik, Shadi Abudalfa, Mo El-Haj, Ahmed Abdelali, Mustafa Jarrar, Nadir Durrani, Hassan Sajjad, and Farah Adeeba. 2026. AbjadGenEval: Abjad AI generated text detection shared task for languages using Arabic script at AbjadNLP 2026. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), co-located with the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*, Rabat, Morocco.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University - Computer and Information Sciences*, 33(5):497–507.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673.

Salima Lamsiyah, Saad Ezzini, Abdelkader El Mahdaouy, Hamza Alami, Abdessamad Benlahbib, Samir El amrany, Salmane Chafik, and Hicham Hammouchi. 2025. M-DAIGT: A Shared Task on Multi-Domain Detection of AI-Generated Text. In *Proceedings of the Shared Task on Multi-Domain Detection of AI-Generated Text*, pages 1–9, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.