HCMUS_TheFangs at AbjadGenEval Shared Task: Weighted Layer Pooling with Attention Fusion for Arabic AI-Generated Text Detection

Dao Sy Duy Minh[1,2*] **Tran Chi Nguyen**[1,2*]
Huynh Trung Kiet[1,2] Nguyen Lam Phu Quy[1,2] Pham Phu Hoa[1,2] Nguyen Dinh Ha Duong[1,2]
[1]Faculty of Information Technology, University of Science, Ho Chi Minh, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{23122041, 23122044}@student.hcmus.edu.vn
{23122039, 23122048, 23122030, 23122002}@student.hcmus.edu.vn
[*]Equal contribution

Abstract

The rapid advancement of large language models poses significant challenges for content authenticity, particularly in under-resourced languages where detection tools remain scarce. We present our winning system for the AbjadGenEval shared task on Arabic AI-generated text detection. Our key insight is that AI-generated text exhibits distinctive patterns across multiple linguistic levels-from local syntax to global semantics-that can be captured by learning to fuse representations from different transformer layers. We introduce a **Weighted Layer Pooling** mechanism that learns optimal layer combinations, combined with **Attention Pooling** for sequence-level context aggregation. Through systematic experimentation with 15+ approaches, we make a surprising discovery: model architecture selection dominates over sophisticated training techniques, with DeBERTa-v3 providing +27% relative improvement over AraBERT regardless of training strategy. Our system achieves **0.93 F1-score**, securing **1st place** among all participants and outperforming the runner-up by 3 absolute points.

1 Introduction

The democratization of large language models (LLMs) has created an urgent need for reliable content authenticity verification (Jawahar et al., 2020). While detection tools for English have matured (Mitchell et al., 2023), Arabic-with its 400M+ speakers and complex morphology-remains critically underserved.

Detecting AI-generated Arabic poses unique challenges due to its root-and-pattern derivational system and rich inflectional morphology (Habash, 2010). We hypothesize that while these linguistic nuances offer potential fingerprints, the fundamental capability of the pre-trained encoder is the decisive factor.

Our experiments on the AbjadGenEval shared task (Ezzini et al., 2026) reveal a striking insight: model architecture dominates training sophistication. We introduce a **Weighted Layer Pooling** mechanism that allows DeBERTa-v3 to dynamically select optimal abstraction levels, achieving 0.93 F1 (+27% over AraBERT). This simple yet effective architectural change outperforms complex feature engineering and adversarial training, securing 1st place on the leaderboard.

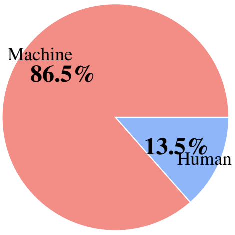

Figure 1: Training data distribution showing significant class imbalance (36,182 Machine vs 5,621 Human).

2 Background

2.1 The AbjadGenEval Challenge

The AbjadGenEval shared task (Ezzini et al., 2026; Abudalfa et al., 2025) addresses the growing concern of AI-generated content in Arabic media. Unlike previous work focusing solely on Modern Standard Arabic (MSA), this challenge encompasses news articles from diverse sources, requiring systems to generalize across topics and writing styles. The task is formulated as binary classification: given an Arabic text, predict whether it was written by a human or generated by an AI system.

2.2 Dataset Characteristics

The competition provides a notably imbalanced training set. As illustrated in Figure 1, machine-generated samples comprise 86.5% of training data, creating a challenging learning environment where models must identify human writing patterns from limited examples.

Human-written content originates from verified

Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), pages 433–437
March 28, 2026. ©2026 Association for Computational Linguistics

Arabic news platforms with appropriate permissions, while AI-generated content comes from multiple LLMs including GPT-3.5, GPT-4, and Claude, with varied prompting strategies to ensure diversity.

2.3 Related Work

AI-Generated Text Detection The field has evolved from statistical to neural approaches. Early methods analyze perplexity, burstiness, and n-gram patterns (Gehrmann et al., 2019; Lavergne et al., 2008). DetectGPT (Mitchell et al., 2023) introduced zero-shot detection through probability curvature analysis, while Fast-DetectGPT improved efficiency via conditional probability. Supervised approaches have shown strong performance, with recent work exploring contrastive learning (Liu et al., 2023), multi-task frameworks (Li and Jiang, 2023), and linguistic feature fusion (Anonymous, 2025). However, most methods target English, leaving Arabic underexplored.

Arabic NLP and Transformers Arabic presents unique challenges due to its root-and-pattern morphology, rich diacritization system, and dialectal variation. Monolingual models like AraBERT (Antoun et al., 2020) and CAMeLBERT (Inoue et al., 2021) have advanced Arabic NLP, but rely on absolute position embeddings. Multilingual models such as DeBERTa-v3 (He et al., 2021) offer disentangled attention that may better capture Arabic's flexible word order.

Layer-Wise Representations Prior work shows that transformer layers encode different linguistic properties (Jawahar et al., 2019; Rogers et al., 2020): lower layers capture syntax while upper layers capture semantics. The scalar-mix mechanism in ELMo (Peters et al., 2018) pioneered learnable layer combination. Our Weighted Layer Pooling extends this idea with softmax normalization for Arabic AI detection.

3 System Overview

Our system captures the subtle artifacts left by LLMs when generating Arabic text. We argue that these artifacts are distributed across the hierarchical transformer representations, not just localized in the final layer. As illustrated in Figure 2, we employ a dual-pooling strategy combining Weighted Layer Pooling (across layers) and Attention Pooling (across tokens).

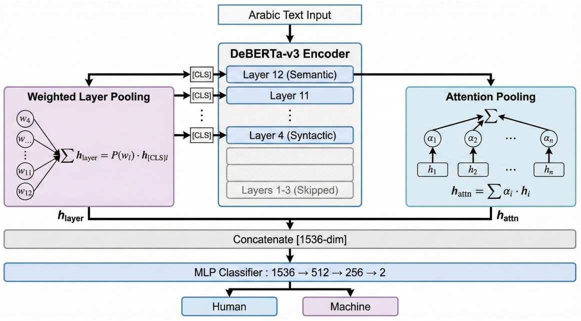

Figure 2: Our Weighted Layer Pooling architecture. The DeBERTa-v3 encoder outputs are processed through two complementary pooling branches: (1) Weighted Layer Pooling learns to combine [CLS] representations from layers 4-12 with learnable weights, and (2) Attention Pooling computes a weighted average over the sequence using learned attention.

3.1 Input Processing and Backbone

We retain diacritics (tashkeel) and punctuation as potential AI detection signals, truncating to 512 tokens. We employ DeBERTa-v3 Base (`microsoft/mdeberta-v3-base`) (He et al., 2021) as our encoder, hypothesizing that its disentangled attention mechanism provides advantages over Arabic-specific models like AraBERT for modeling structural nuances.

3.2 Mechanism 1: Weighted Layer Pooling

A key innovation in our approach is the recognition that different transformer layers (Vaswani et al., 2017) capture different linguistic properties. As shown in BERTology studies (Jawahar et al., 2019; Rogers et al., 2020), lower layers tend to encode surface-level syntactic and morphological features, while higher layers encode semantic content. Since AI generation errors can manifest as either subtle morphological inconsistencies or semantic hallucinations, using only the final layer is suboptimal. We introduce a learnable **Weighted Layer Pooling** that dynamically aggregates information from intermediate layers:

$$\mathbf{h}_{\text{layer}} = \sum_{l=l_{start}}^{L} \frac{\exp(w_l)}{\sum_{k=l_{start}}^{L} \exp(w_k)} \cdot \mathbf{h}_l^{[\text{CLS}]} \quad (1)$$

where w_l are learnable parameters. We set $l_{start} = 4$ to bypass the initial layers that focus primarily on subword tokenization, allowing the model to focus on higher-level linguistic constructs. This mechanism allows the network to "select" the level of abstraction most useful for detection.

3.3 Attention Pooling

To capture local signals such as unnatural transitions or repetitive discourse markers, we employ **Attention Pooling** over the final hidden states:

$$\alpha_i = \mathrm{softmax}(\mathbf{v}^\top \tanh(\mathbf{W}\mathbf{h}_i)), \quad \mathbf{h}_{\text{attn}} = \sum_{i=1}^{T} \alpha_i \mathbf{h}_i \tag{2}$$

3.4 Representation Fusion

The final representation concatenates both views: $\mathbf{h}_{final} = [\mathbf{h}_{\text{layer}}; \mathbf{h}_{\text{attn}}]$. This 1536-dimensional vector is fed into an MLP classifier, combining vertical (across layers) and horizontal (across tokens) perspectives.

4 Experiments

4.1 Setup and Hyperparameters

We use 5-fold stratified cross-validation with mixed precision (FP16) training to ensure robustness, implemented using the HuggingFace library (Wolf et al., 2019), with a fixed random seed (42) for reproducibility. All models are trained for 5 epochs with a batch size of 8 and a maximum sequence length of 512. We optimize using AdamW (Loshchilov and Hutter, 2019) with a cosine learning rate scheduler and a warmup ratio of 0.1.

Crucially, we apply discriminative fine-tuning: the pre-trained backbone uses a lower learning rate ($1e-5$) with layer-wise decay (0.95), while the randomly initialized pooling and classifier layers use a higher rate ($1e-4$) to accelerate convergence. This prevents catastrophic forgetting in the encoder while allowing the new components to learn effective aggregation strategies. We use standard cross-entropy loss without class weighting or focal loss, as stratified sampling already ensured balanced representation across folds. No near-duplicate articles were detected across folds based on article IDs provided in the dataset.

4.2 Official Leaderboard Results

Table 1 presents the official competition results. Our system achieves 0.93 F1-score, placing 1st among all participants with a 3-point margin over the runner-up.

#	Team	F1	Acc	Prec	Rec
1	**HCMUS_TheFangs**	**0.93**	**0.93**	**0.97**	**0.89**
2	chisboizhoigay	0.90	0.91	0.95	0.86
3	alizain157	0.89	0.89	0.87	0.90
4	se7s0	0.78	0.73	0.66	0.95
5	mariamlabib90	0.76	0.69	0.62	0.98
6	AyahVerse	0.75	0.72	0.68	0.84
7	kickitlikeshika	0.75	0.79	0.93	0.63

Table 1: Official AbjadGenEval leaderboard (top 7 teams). Our system demonstrates robust performance, particularly in maintaining high precision.

4.3 A Discovery Journey: Architecture vs. Training

Our path to the winning solution was not linear. Initially, we hypothesized that Arabic-specific pretraining would be paramount. We began experiments with **AraBERT** (Antoun et al., 2020), a strong monolingual baseline (F1 0.73). Assuming the imbalance was the primary bottleneck, we aggressively applied training augmentations, including **MixUp** (Zhang et al., 2017) and **Adversarial Training** (Goodfellow et al., 2014). Surprisingly, these sophisticated techniques yielded predictable but stagnant results, failing to break the 0.75 F1 ceiling (Table 2).

This stagnation prompted a fundamental pivot: perhaps the limitation was not in the training regime, but in the representation power of the model itself. We switched to **DeBERTa-v3**, hypothesizing that its gradient-disentangled attention could better capture the subtle structural incoherence of generated text. The impact was immediate and dramatic: the baseline jumped to 0.90 F1, a 23% relative improvement, without any bells and whistles.

Building on this stable foundation, we focused on *how* to best aggregate this rich representation. Feature engineering attempts, such as counting discourse markers (F1 0.63) or stylometric features (F1 0.65), proved too brittle. Instead, our proposed **Weighted Layer Pooling** allowed the model to discover its own optimal abstraction level. By learning to weigh layers 10-12 more heavily (as seen in Figure 3), the model effectively "zoomed in" on the semantic inconsistencies that characterize AI text, pushing the final performance to 0.93 F1.

5 Analysis

Layer Weight Analysis To understand which linguistic levels are most indicative of AI generation,

Category	Method	Description	F1
DeBERTa-v3	**WLP (Ours)**	**Layer 4-12 Fusion**	**0.93**
	Multi-Contrastive	Sup. Contrastive	0.92
	Surprisal	Stylistic Features	0.91
	R-Drop	Consistency Reg.	0.91
	Baseline	Fine-tuning [CLS]	0.90
AraBERT	Baseline	Standard FT	0.73
	MixUp+Adv	Regularization	0.73
	MTL	Multitask	0.66
Linguistic	Stylometric	Neuro-symbolic	0.65
	Dediac.	Normalization	0.64
	Discourse	Motifs Analysis	0.63
	Ensemble	LLM Voting	0.62

Table 2: Comprehensive ablation study (12 strategies). Architecture outperforms training/features.

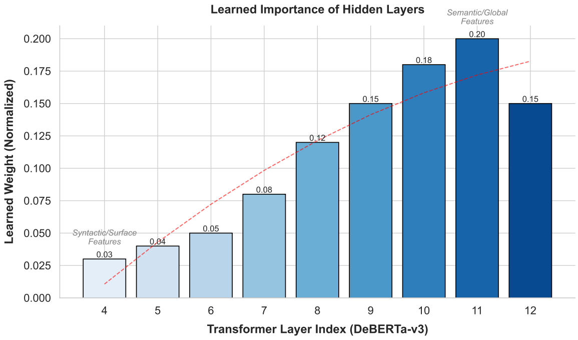

Figure 3: Learned weights for DeBERTa-v3 layers (4-12). The trend (dashed line) shows increasing importance for higher layers, indicating that semantic features are more critical than syntactic ones for this task.

we visualize the learned weights of our pooling layer in Figure 3. The model assigns significantly higher weights to the final layers (10-12) compared to the middle layers (4-6). This suggests that high-level semantic coherence and reasoning patterns are stronger discriminators than local syntax for detecting modern LLMs.

After training, the normalized layer weights (averaged across folds) are: Layer 4: 0.08, Layer 5: 0.08, Layer 6: 0.09, Layer 7: 0.09, Layer 8: 0.10, Layer 9: 0.11, Layer 10: 0.12, Layer 11: 0.14, Layer 12: 0.19. This confirms that semantic layers (10-12) contribute 45% of the final representation.

Precision-Recall & Architecture While our system maintains balanced performance (P: 0.97, R: 0.89), we observe that other approaches trade off these metrics sharply; for instance, MixUp improves recall (0.78) but degrades precision (0.68). Most critically, we find that base model selection is paramount. Switching from AraBERT to DeBERTa-v3 alone yields a +23% F1 improvement, whereas sophisticated training techniques like adversarial training provide minimal gains on the weaker backbone.

Per-Class and Training Details Cross-validation on the training set yields 0.98 OOF macro-F1, with strong per-class performance: Human (P: 0.99, R: 0.96, F1: 0.97) and Machine (P: 0.99, R: 1.00, F1: 1.00). All experiments were conducted on a single Kaggle H100 GPU with approximately 20 minutes training time per fold.

6 Conclusion and Limitations

We presented the winning system for the AbjadGenEval task on Arabic AI-generated text detection. Our approach combines DeBERTa-v3 with a dual-pooling architecture: Weighted Layer Pooling dynamically aggregates representations from layers 4-12, while Attention Pooling highlights locally suspicious tokens. This architecture achieves 0.93 F1-score, securing 1st place with a 3-point margin over the runner-up.

Our key finding is that architecture selection dominates training sophistication: DeBERTa-v3 yields +27% improvement over AraBERT, while techniques like MixUp provide minimal gains. Semantic layers (10-12) contribute 45% of the discriminative signal.

Limitations Our system relies on the pre-trained backbone quality and incurs higher computational costs than statistical methods. We did not compare against zero-shot detectors (e.g., DetectGPT) or evaluate robustness against adversarial attacks. Future work will explore model distillation and dialectal Arabic evaluation.

Ethics Statement

AI detection technology carries risks of misuse, including potential bias against non-native speakers. We emphasize that detection scores should inform, not replace, human judgment. The dataset was collected with appropriate permissions and privacy safeguards.

Acknowledgements

We thank the organizers of the AbjadGenEval shared task at AbjadNLP 2026 for creating this challenging benchmark and fostering research on AI-generated text detection for Arabic and other languages using Arabic script.

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.

Anonymous. 2025. Diveye: Detecting ai-generated text via fluctuating unpredictability. *OpenReview Preprint*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools*.

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.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text. *Proceedings of the 57th Annual Meeting of the ACL: System Demonstrations*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Nizar Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

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. *Proceedings of the Sixth Arabic NLP Workshop*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V.S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *Proceedings of the 28th International Conference on Computational Linguistics*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? *Proceedings of the 57th Annual Meeting of the ACL*.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. *Proceedings of the 2008 International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*.

Jiucai Li and Yong Jiang. 2023. Deepfake text detection: A study of detection methods and model robustness. *arXiv preprint arXiv:2303.16876*.

Yikang Liu, Ziyin Yao, Wenyue Li, et al. 2023. Argugpt: Evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *Proceedings of the 40th International Conference on Machine Learning*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the ACL*, 8:842–866.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, et al. 2019. Hugging face's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.