NLP_wizard at AraGenEval shared task: Embedding-Based Classification for AI Detection and Authorship Attribution

Mena Hany
King Fahd University of Petroleum and Minerals / Saudi Arabia, Dammam
g202411920@kfupm.edu.sa

Abstract

This paper presents a lightweight system for the *AraGenEval* shared task, addressing AI-generated text detection and authorship identification in Arabic. Using pretrained `xlm-roberta-large` embeddings with mean pooling and `[CLS]` token strategies, combined with classical classifiers (RidgeClassifierCV and LinearSVC), our approach achieved F1-scores of 0.7400 and 0.8130 on the *ARATECT* and authorship datasets, respectively. Mean pooling outperformed `[CLS]` by 3%, demonstrating efficiency and robustness for limited Arabic data while capturing stylistic nuances critical for both tasks.

1 Introduction

The rapid advancements in large language models (LLMs) have enabled the generation of fluent, human-like text at unprecedented scale (Vaswani et al., 2017; Brown et al., 2020). This has intensified the need for robust systems capable of both detecting AI-generated content and identifying the authorship of text (Jawahar et al., 2020; Uchendu et al., 2020). Such capabilities are critical for preserving content authenticity, combating misinformation, and supporting forensic linguistic analysis (Uchendu et al., 2020). While research in this area has grown substantially for English, Arabic remains relatively underexplored despite its rich morphology, dialectal diversity, and increasing online presence (Habash, 2010).

To address these gaps, the *AraGenEval* shared task (Abudalfa et al., 2025) was introduced as part of ArabicNLP 2025. The task encompasses three subtasks: (1) **Authorship Style Transfer**, which focuses on transforming text to mimic a specific author's style; (2) **Authorship Identification**, which aims to determine the original author of a given text; and (3) **AI-Generated Text Detection**, which seeks to distinguish between human-written and machine-generated Arabic text. The competition provided a unified benchmark for evaluating system performance on these interrelated challenges.

Our participation focused on the **Authorship Identification** and **AI-Generated Text Detection** subtasks. We employed the `xlm-roberta-large` multilingual model to extract contextual embeddings for Arabic text. Instead of using the conventional `[CLS]` token representation, we computed the average of all token embeddings to form document-level feature vectors. These embeddings were then fed into various traditional machine learning classifiers. For AI-generated text detection, the `RidgeClassifierCV` achieved the best performance with an F1-score of 0.74 on the blind test set, ranking **10th** among all submissions. For authorship identification, the `LinearSVC` classifier attained an F1-score of 0.81303 on the blind test set, also ranking **10th** in the respective leaderboard.

Our findings highlight that averaging contextual embeddings from `xlm-roberta-large` can serve as a strong baseline for Arabic authorship and AI detection tasks, even when combined with relatively lightweight classifiers. We also observed that the choice of classifier plays a substantial role in performance, with linear models showing competitive results.

2 Background

The *AraGenEval* shared task (**?**) was designed to benchmark system performance on three Arabic NLP challenges: **Authorship Style Transfer** (Task 1), **Authorship Identification** (Task 2), and **AI-Generated Text Detection** (Task 3). All tasks targeted Modern Standard Arabic (MSA) and included data from diverse literary and journalistic sources.

2.1 Task Setup

In **Authorship Identification** (Task 2), the input is a short Arabic text segment, and the output is the

Proceedings of The Third Arabic Natural Language Processing Conference, pages 37–41

predicted author identity from a set of 21 possible authors. For example, given a paragraph excerpted from a 20th-century Arabic novel, the system must assign the correct author label.

In **AI-Generated Text Detection** (Task 3), the input is also a short text passage, and the output is a binary classification: `human` or `AI`. For instance, given a news-style paragraph, the model must detect whether it was written by a human or produced by a large language model.

2.2 Dataset Details

Authorship Identification. The dataset contains works from 21 authors, each represented by 10 publicly available books. Texts were segmented into semantically coherent paragraphs, and for style transfer tasks, selected paragraphs were rephrased into a standardized formal style using GPT-4o mini2. The dataset is split into training, validation, and test sets per author. Table 1 summarizes the distribution of samples.

Author	Train	Test	Val
Ahmed Amin	2892	594	246
Ahmed Taymour Pasha	804	142	53
Ahmed Shawqi	596	46	58
Ameen Rihani	1557	624	142
Tharwat Abaza	755	191	90
Gibran K. Gibran	748	240	30
Jurji Zaydan	2762	562	326
Hassan Hanafi	3735	1002	548
Robert Barr	2680	512	82
Salama Moussa	984	282	119
Taha Hussein	2371	534	253
Abbas M. Al-Aqqad	1820	499	267
Abdel G. Makawi	1520	464	396
Gustave Le Bon	1515	358	150
Fouad Zakaria	1771	294	125
Kamel Kilani	399	109	25
Mohamed H. Heikal	2627	492	260
Naguib Mahfouz	1630	343	327
Nawal El Saadawi	1415	382	295
William Shakespeare	1236	358	238
Yusuf Idris	1140	349	120

Table 1: Authorship identification dataset statistics.

AI-Generated Text Detection. The *ARATECT* dataset contains human-written and AI-generated Arabic texts. Human texts were collected from reputable Arabic news websites and verified literary works, then manually curated for quality. AI-generated texts were produced using several Arabic-capable LLMs, including Mistral, GPT-4, and LLaMA, prompted under diverse strategies. Each text is annotated with a binary label (`human` vs. `AI`) and covers two main domains: news and literature.

2.3 Related Work

Authorship attribution has been extensively studied across languages, with foundational surveys such as (Stamatatos, 2009) and transitions from stylometric to deep learning methods highlighted by (Kestemont, 2014). AI-generated text detection research has grown recently with large language models, with multilingual studies focusing on cross-lingual generalization (Uchendu et al., 2020) and detection surveys (Jawahar et al., 2020). Our approach applies multilingual transformer embeddings (`xlm-roberta-large`) averaging token vectors for Arabic authorship identification and AI-detection within the competitive *AraGenEval* shared task.

3 System Overview

Our system for the *AraGenEval* shared task was designed to be lightweight yet competitive, focusing on extracting high-quality text representations from a large multilingual transformer model and feeding them into robust classical machine learning classifiers. Instead of fine-tuning or training deep neural networks, we adopted a fixed-embedding approach, motivated by the desire to minimize computational requirements and avoid overfitting on the relatively small training datasets provided.

3.1 Key Algorithms and Design Decisions

We selected the `xlm-roberta-large` model due to its proven effectiveness in multilingual contexts and its strong coverage of Arabic. This model, trained on a massive and diverse corpus, provides rich contextual embeddings that capture both syntactic and semantic nuances of text. Given that the shared task focuses on style-related distinctions (authorship identification and AI-generated text detection), we hypothesized that `xlm-roberta-large`'s high-capacity representations could encode stylistic patterns without task-specific fine-tuning.

Two different strategies were implemented for deriving sentence-level embeddings from the model's final hidden layer:

1. **Mean Token Embeddings:** In this configuration, the embedding for an input text was obtained by averaging the contextualized embeddings of all tokens. This approach is expressed as:

$$h_{\text{mean}} = \frac{1}{n} \sum_{i=1}^{n} e_i$$

where $e_i \in \mathbb{R}^d$ represents the embedding of token i and n is the total number of tokens in the input sequence. The intuition is that by aggregating all token embeddings, we capture both content and stylistic markers distributed throughout the text, rather than relying on a single position-specific vector.

2. **[CLS] Token Embedding:** In this configuration, we directly used the representation of the special `[CLS]` token from the model's final layer:

$$h_{\text{[CLS]}} = e_{\text{[CLS]}}$$

The `[CLS]` token is commonly used in transformer-based classification pipelines, as it is intended to encode a holistic summary of the input sequence. However, it may not fully capture distributed stylistic cues, particularly for long texts.

Once the embeddings h were computed, they were fed into classical machine learning classifiers:

- **AI-Generated Text Detection:** RidgeClassifierCV was chosen for its efficiency, robustness to multicollinearity, and ability to handle high-dimensional input spaces without explicit feature selection.
- **Authorship Identification:** LinearSVC was selected for its scalability to large feature sets, strong generalization properties, and suitability for high-dimensional sparse representations.

3.2 Resources Beyond Provided Data

The system used no additional annotated datasets beyond those provided in the shared task. The only external component was the publicly available `xlm-roberta-large` model from the HuggingFace Transformers library. This model was not fine-tuned on the task data; instead, we relied on its pretrained multilingual representations. No handcrafted features, lexicons, or rule-based preprocessing steps were introduced.

3.3 Addressing Task Challenges

Two main challenges guided our design decisions:

1. **Limited Task-Specific Data:** Given the relatively small size of the training set, fine-tuning a large transformer could risk overfitting. Using fixed embeddings allowed us to leverage the model's pretrained linguistic knowledge while avoiding costly gradient-based updates.
2. **Capturing Stylistic Cues:** Both subtasks depend heavily on identifying stylistic rather than purely semantic differences. We hypothesized that mean-pooling token embeddings would better preserve distributed stylistic markers (e.g., function word usage, sentence rhythm, punctuation patterns) than a single `[CLS]` embedding, which might focus on semantic summarization.

3.4 Configuration Comparison

We experimented with both configurations — mean token embeddings and `[CLS]` token embeddings — under otherwise identical conditions. While both approaches successfully leveraged the pretrained model's capacity, qualitative inspection during development suggested that mean token embeddings were more effective at preserving fine-grained stylistic patterns. In contrast, `[CLS]` embeddings appeared to compress the sequence information into a more generalized representation, which, while concise, might have omitted subtle stylistic distinctions critical for the two tasks.

We therefore retained both configurations for evaluation but anticipated that the mean token approach would have an advantage in the final results.

4 Experimental Setup

4.1 Data Splits

The *AraGenEval* shared task provided labeled data for both subtasks: (1) AI-generated text detection and (2) authorship identification. For each task, the official training, development, and test sets released by the organizers were used without modification. The training set was used to fit the classifiers, the development set served for configuration selection and sanity checking, and the official test set was reserved for final submission and evaluation.

4.2 Embedding Extraction

Embeddings were extracted using the `xlm-roberta-large` model from HuggingFace:

- Maximum sequence length: 512 tokens (truncation applied to longer texts)
- Pooling strategies: (1) mean pooling across all token embeddings; (2) using the final layer `[CLS]` token embedding

The embeddings were computed once and cached for both tasks to speed up experimentation.

All models and classifiers were used with their default parameters as implemented in the HuggingFace Transformers and scikit-learn libraries.

4.3 Computational Resources

All experiments were run on a single NVIDIA RTX 4060 GPU with 8GB VRAM, paired with a standard workstation environment.

4.4 Evaluation Metrics

The shared task organizers specified official metrics for each subtask:

- **AI-generated text detection:** Macro-averaged F1-score across classes.
- **Authorship identification:** Macro-averaged F1-score across authors.

All results reported in the following section were computed using the organizers' evaluation scripts to ensure consistency with leaderboard scoring.

5 Experimental Results

Table 2 presents the performance of our system on the official blind test set for both subtasks of the *AraGenEval* shared task. We compare the two embedding pooling strategies: mean pooling across all tokens and using only the final layer `[CLS]` token embedding.

Table 2: Performance comparison of pooling strategies on the blind test set.

Subtask	Pooling	F1	Rank
AI-generated text detection	Mean	0.7400	10
	CLS	0.7100	–
Authorship identification	Mean	0.8130	10
	CLS	0.7830	–

From the table, mean pooling consistently outperforms the `[CLS]` token embeddings, yielding approximately a 3% absolute F1-score improvement in both subtasks. This suggests that averaging token representations provides a richer global representation for classification tasks in the *AraGenEval* setting.

6 Conclusion

Our system for the *AraGenEval* shared task delivered competitive performance in both AI-generated text detection and authorship identification by leveraging pretrained `xlm-roberta-large` embeddings paired with efficient classical machine learning classifiers. As presented in Table 2, the mean pooling strategy achieved F1-scores of 0.7400 for AI-generated text detection and 0.8130 for authorship identification, outperforming the `[CLS]` token embedding approach by approximately 3% in both tasks. This improvement suggests that mean pooling better captures distributed stylistic patterns, which are critical for distinguishing AI-generated from human-written texts and identifying unique author signatures. The lightweight design, which avoided resource-intensive fine-tuning, proved well-suited for the limited training data provided in the *ARATECT* dataset and the authorship identification dataset, which spans 21 authors with diverse writing styles. The system's ability to handle varied text domains, including news and literature, underscores its robustness and potential for broader Arabic text analysis applications.

7 Future Work

To further enhance the system, several avenues can be explored. First, experimenting with hybrid pooling methods that combine mean pooling and `[CLS]` embeddings could produce more comprehensive text representations, balancing stylistic and semantic information. Second, applying targeted fine-tuning on the `xlm-roberta-large` model with task-specific Arabic data could improve its sensitivity to the language's unique morphological and stylistic features. Third, incorporating additional features, such as lexical patterns or syntactic structures, might strengthen the system's ability to detect subtle stylistic differences. Fourth, developing methods to process texts longer than 512 tokens, such as hierarchical embedding aggregation, could improve performance on extended literary works. Finally, testing the system on diverse real-world Arabic datasets, including social media or news articles, would help validate its effectiveness in practical settings and enhance its applicability to emerging challenges in text authenticity and authorship attribution.

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 (ArabicNLP 2025), Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Mike Kestemont. 2014. Function words in authorship attribution. from black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 59–66.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.