Sebaweh at AraGenEval Shared Task: BERENSE - BERt based ENSEmbler for Arabic Authorship Identification

Muhammad Helmy*
mu.helmy@nu.edu.eg

Batool Balah*
batoolnajeh@gmail.com

Ahmed Mohamed Sallam
ahmedm.sallamibrahim@gmail.com

Ammar Sherif
ammarsherif90@gmail.com

Abstract

Authorship Identification for Arabic texts is challenging due to the language's dialectal diversity and the wide stylistic variation across genres, cultures, and historical periods. It has critical applications in copyright enforcement, forensic linguistics, and literary analysis. Recognizing its importance, we addressed this challenge using the *AraGenEval 2025* shared task dataset, which contains works by writers from diverse backgrounds and time periods. We conducted extensive experiments with multiple architectures and proposed an ensemble model that combines the strengths of four fine-tuned transformer-based models. We applied data augmentation to enrich the dataset and class weighting to handle class imbalance during training. Our system achieved a **Macro-F1 score of 90%**, representing a **15% improvement** over our baseline, and ranked **1st** in the competition.

1 Introduction

Transformer architectures have revolutionized the way we analyze and understand textual data, demonstrating a remarkable ability to capture deep contextual and stylistic patterns highly effective for tasks such as Authorship Identification. This task involves determining the author of a given text based on its stylistic and linguistic characteristics and has critical applications in plagiarism detection, forensic linguistics, and historical literature analysis. However, Arabic remains underrepresented in this line of research, despite its rich literary tradition (Alqurashi, 2024).

The task presents four core challenges: language-related complexities, feature selection, data availability, and preprocessing decisions. The structural challenges of Arabic, such as morphological richness, inflection, diglossia, and diacritics, complicate preprocessing and obscure stylistic cues. Additionally, the scarcity of large, balanced corpora and suitable modeling tools further hinders progress (Alqahtani and Dohler, 2023).

Our main contributions to the Arabic Authorship Identification task:

- Ranked 1st in AraGenEval's Subtask 2 on Arabic Authorship Identification (Abudalfa et al., 2025), a multiclass classification task predicting the author of an Arabic paragraph.
- Performed data augmentation to enrich the samples of underrepresented authors and applied class weighting during training.
- Extensively experimented with multiple Arabic transformer models (Alqurashi, 2024; Alqahtani and Dohler, 2023) and combined them into an ensemble, which reduced variance and improved robustness.
- Achieved a +15% improvement in macro-averaged F1 over the baseline, reaching 90%.

2 Background

The dataset for AraGenEval's Subtask 2 includes 21 Arabic authors spanning novelists, philosophers, historians, social activists, and politicians, and covers diverse time periods. Each author is represented by one to ten books, segmented into semantically coherent paragraphs. The texts are exclusively in Arabic, encompassing Classical Arabic, Modern Standard Arabic (MSA), and Egyptian dialect. Class distributions vary widely, from fewer than 100 to over 3000 samples per author, reflecting real-world authorship identification challenges such as long-form input, class imbalance, genre variability, and subtle stylistic overlap.

Authorship identification in English has evolved from classical machine learning with handcrafted features to deep learning and transformer-based approaches. Huertas-Tato et al. (Huertas-Tato et al.,

Proceedings of The Third Arabic Natural Language Processing Conference, pages 59–64

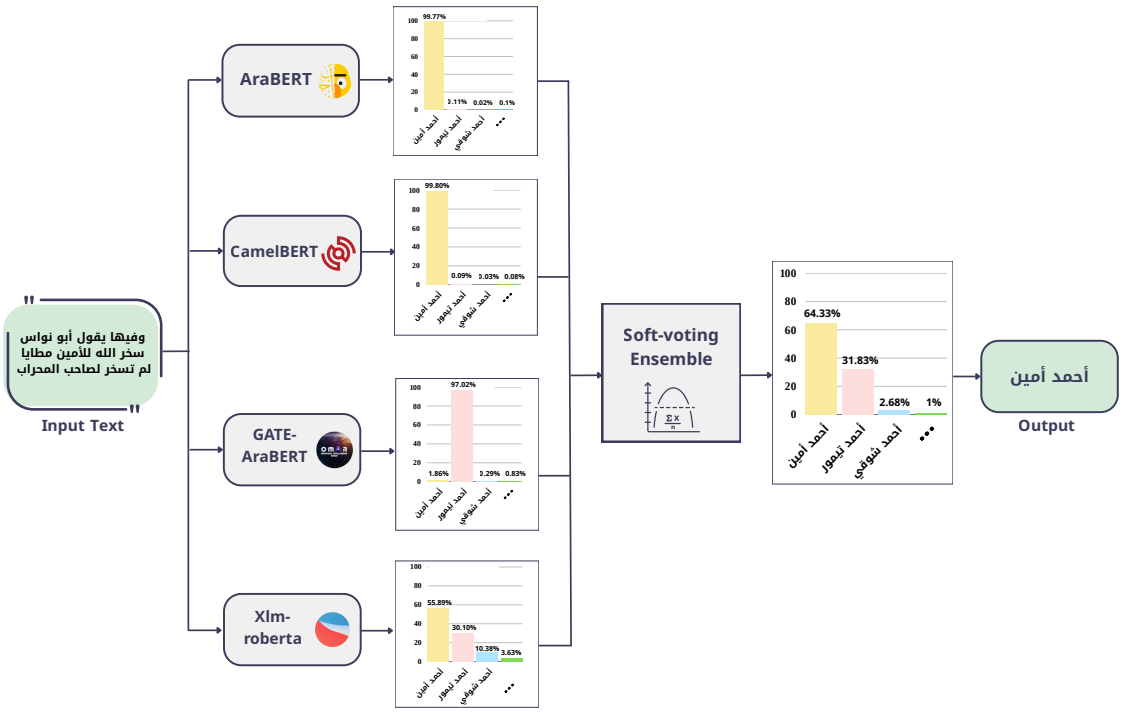


Figure 1: **System overview**. Our system ensemble is composed of 4 models: AraBERT, CAMeLBERT, XLM-RoBERTa-Arabic, and GATE-AraBERT-v1. The final output is then computed via soft-voting of all the outputs.

2022) introduced PART, a pre-trained transformer using contrastive learning to capture author-specific styles. Silva et al. (Silva et al., 2023) applied GAN-BERT to attribute late 19th-century novels and later extended it to detect AI-generated forgeries (Silva et al., 2024). While highly effective across genres and large author sets, comparable work in Arabic remains scarce due to its morphological richness and dialectal variation, which both complicate modeling and offer unique stylistic cues.

A related task, Author Profiling, predicts attributes such as gender, dialect, or age. Zhang and Abdul-Mageed (Zhang and Abdul-Mageed, 2022) developed a transformer-based system for profiling Arabic social media users. However, such work focuses on trait prediction for short, informal texts, not full-text identity attribution, highlighting the need for dedicated Arabic authorship identification methods across domains.

Arabic authorship studies have often been small-scale (fewer than 15 authors) and domain-specific, such as classical literature, Islamic legal texts, or poetry. These works aimed to identify authors using statistical and machine learning methods adapted to the domain. Al-Sarem et al. (Al-Sarem et al., 2020) used an artificial neural network for fatwa texts, while Sayoud (Hadjadj and Sayoud, 2021) applied PCA and SMOTE to address feature dimensionality and class imbalance. Earlier works (Altheneyan and Menai, 2014; Ahmed et al., 2019) employed Naïve Bayes, SVM, or LDA with lexical, syntactic, and structural features. While effective in restricted settings, these approaches relied heavily on manual feature engineering and often failed to capture semantic or stylistic depth across genres.

More recent Arabic work with transformers remains narrow in scope. AlZahrani and Al-Yahya (AlZahrani and Al-Yahya, 2023) focused on Islamic legal texts with small author sets, while Alqurashi et al. (Alqurashi et al., 2025) used a CAMeLBERT-based ensemble for classical poetry, achieving F1 scores from 0.97 to 1.0. Despite strong results, their focus was limited to a single genre.

To address these gaps, our work presents a transformer-based model trained on Arabic texts spanning diverse dialects and genres, capable of learning stylistic patterns directly from raw text without manual feature engineering.

3 System Overview

We reached this system design after experimenting with several alternative architectures, includ-

ing BERT embeddings with RNN/LSTM heads, frozen BERT embeddings with SVM/RF classifiers, and BERT embeddings concatenated with extracted topic distributions followed by a fully connected softmax layer. However, the pure BERT embeddings followed by a fully connected softmax layer outperformed the other approaches (see Figure 1).

3.1 Model Architecture

Following the best-performing architecture, we fine-tuned four transformer-based models from Hugging Face: AraBERT v0.2 (136M), CAMeLBERT-Mix (110M), Arabic XLM-RoBERTa (270M), and GATE-AraBERT (135M), each leveraging the same fully connected softmax classification head. To ensure robust inference, we employed a soft-voting ensemble that averaged the predicted probability distributions of all four models, thus reducing variance and exploiting complementary stylistic features captured by each transformer (see Appendix B).

3.2 Handling Class Imbalance

The dataset exhibited a significant imbalance in the number of samples per author, which could bias the model toward overrepresented classes. To address this, we modified the standard cross-entropy loss to include class weights inversely proportional to class frequencies, thereby penalizing errors on underrepresented authors more heavily (see Appendix C for the formal definition).

3.3 Data Augmentation

To increase stylistic variation and expand data diversity, we collected additional works from the Hindawi Books dataset (Filali, 2022), targeting underrepresented authors: Tharwat Abaza, Kamel Kilani, Gobran Khalil Gobran, Ahmad Taymour Basha, Ahmad Shawqy. After using the validation set to select the hyper-parameters and do initial experiments, we appended it with the training set at the end to increase the training data before the final evaluation on the test set.

4 Experimental Setup

4.1 Data Splits

We followed the official Shared Task 2 data split provided by the organizers. The dataset was divided into *training*, *validation*, and *test* sets. The validation set was used for model selection and hyperparameter tuning, while the test set was reserved for final evaluation.

4.2 Preprocessing

To address statistical imbalances and reduce noise that could obscure stylistic cues, we applied three preprocessing steps to the dataset. First, we removed a total of 2,740 duplicates to avoid overrepresentation of specific expressions. Second, we performed length capping by splitting 1,381 texts exceeding 3,000 characters into chunks of approximately 2,000 characters, corresponding to the mean text length across authors and remaining within the tokenizer's maximum sequence length. (see Appendix D for illustrative examples).

This step was intended to reduce overfitting risks, improve gradient updates for underrepresented authors, and encourage reliance on stylistic rather than length cues. Finally, we removed diacritics, as they are often inconsistently applied or auto-inserted in digital-born text, which can introduce noise into the stylistic signal.

4.3 Parameter Settings

We fine-tuned four transformer-based models with carefully selected hyperparameters, including learning rate, optimizer, training epochs, warmup ratio, and weight decay. The best configurations for AraBERT, CAMeLBERT, and XLM-RoBERTa-Arabic are the same: learning rate of $8e10^{-5}$, Adam as optimizer, cosine scheduler, 10% warmup ratio, 4 epochs, and 0.1 of weight decay. GATE-AraBERT-v1 is the same with the only difference in learning rate: $2e10^{-5}$

4.4 External Tools and Libraries

The implementation was carried out in Python 3.10 using Google Colab and Kaggle environments. We used **pandas** and **numpy** for data handling, **matplotlib** and **seaborn** for visualization (e.g., histograms and bar charts), **langdetect** for language identification, and **langchain** for text splitting.

4.5 Evaluation Metrics

Following the AraGenEval guidelines, we evaluated our models using four primary metrics: Macro F1-score, Accuracy, Precision, and Recall on the test set. Macro F1-score was the main ranking

criterion in the shared task, defined as:

$$\text{Macro F1} = \frac{1}{N}\sum_{i=1}^{N}\text{F1}_i \quad (1)$$

where N is the number of classes, and F1_i is the F1-score computed for class i:

$$\text{F1}_i = 2\frac{\text{Precision}_i\text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (2)$$

$$\text{where Precision}_i = \frac{TP_i}{TP_i + FP_i}, \quad (3)$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (4)$$

Here, TP_i, FP_i, and FN_i denote the number of true positives, false positives, and false negatives for class i. Accuracy is computed as the proportion of correctly predicted instances over the total number of instances.

5 Results

We gradually enhanced performance over our initial BERT + RNN baseline. Table 1 compares alternative architectures we tested. The best single-model result came from BERT embeddings with a softmax layer, reaching **0.85**. This suggests that while BERT embeddings capture valuable stylistic information, and their effectiveness depends heavily on the classifier's capacity to exploit high-dimensional contextual features.

Table 1: Comparison of alternative architectures on the validation set.

Architecture	F1 Score
BERT + RNN (baseline)	0.75
Frozen BERT + SVM (bagging)	0.66
Frozen BERT + Random Forest	0.35
BERT + Fully Connected Layer	**0.85**
Our Ensemble[1]	**0.90**

Building on these findings, we adopted the BERT embeddings + fully connected softmax layer architecture as our main design and explored further enhancements. We evaluated various embedding models, including AraBERT v0.2, CAMeLBERT-Mix, Arabic XLM-RoBERTa, GATE-AraBERT, Arabic-labse-Matryoshka, and Arabic distilbert-base. We excluded the last two from the final ensemble as their validation F1 scores fell below **0.80**.

[1]Result on test set.

We incorporated external stylistic cues by performing topic modeling and concatenated the top topic keywords with the embedding representation, following the approach of Alqurashi et al. (Alqurashi et al., 2025). However, experiments with CAMeLBERT-Mix showed no measurable performance gain (F1 = **0.85** both with and without topic features), suggesting that topic distributions did not contribute additional discriminative power beyond the contextual embeddings.

Subsequently, augmenting training data with the Hindawy dataset yielded consistent validation improvements across most models. Table 2 reports macro-F1 scores with and without augmentation on the validation set.

Table 2: Macro-F1 with and without augmentation (validation set).

Model	Aug	No Aug
AraBERT v0.2	0.90 (↑ **2%**)	0.88
CAMeLBERT-Mix	0.90 (↑ **6%**)	0.84
Arabic XLM-RoBERTa	0.83 (**0**)	0.83
GATE-AraBERT	0.89 (↑ **5%**)	0.84

Although applying class-weighted loss improved performance in the frozen GATE-AraBERT + bagging SVM setup, increasing validation F1 from **0.56** to **0.66**, it did not show such an enhancement for the fully connected architecture. The effect was minimal overall, though we observed a slight gain from **0.82** to **0.83** validation F1 for XLM-RoBERTa. We retained this procedure as it did not degrade performance for other models and XLM-RoBERTa had not shown improvements from data augmentation.

To better understand model errors, we inspected the confusion matrix of the predicted authors. Misclassifications were often concentrated among authors with overlapping genres or historical contexts, reflecting the stylistic and thematic proximity between them. A detailed analysis of the most frequent confusions is provided in Appendix A.

Finally, our ensemble system achieved a macro-averaged F1 of **0.9046**, accuracy of **0.9327**, precision of **0.9012**, and recall of **0.9143**, ranking **1st** on the official test set of the AraGenEval 2025 Subtask 2, outperforming each single model.

6 Conclusion

We developed an ensemble-based system for Arabic Authorship Identification, achieving a macro-F1 of **0.9046** on the AraGenEval 2025 test set and

ranking **1st** in Subtask 2. Our analysis showed that while frozen embeddings with classical classifiers underperformed, a BERT + fully connected design, combined with data augmentation and ensembling, delivered strong gains. Class-weighted loss had mixed effects, benefiting some models but not others.

Limitations include the restriction to only 21 authors and the features are not guaranteed to be style-based rather than content-based, which might present a form of overfitting. Future work will investigate open-set authorship, experiment more with contrastive learning to enhance the features, assess potential data leakage, and apply interpretability techniques to better understand the model's decision-making process.

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)*, Suzhou, China. Association for Computational Linguistics.

A. Ahmed, R. Mohamed, and B. Mostafa. 2019. Arabic poetry authorship attribution using machine learning techniques. *Journal of Computer Science*, 15(7):1012–1021.

Mohammed Al-Sarem, Abdullah Alsaeedi, and Faisal Saeed. 2020. A deep learning-based artificial neural network method for instance-based arabic language authorship attribution. *International Journal of Advances in Soft Computing and its Applications*, 12:1.

Fatimah Alqahtani and Mischa Dohler. 2023. Survey of authorship identification tasks on arabic texts. *ACM Computing Surveys*.

Lama Alqurashi. 2024. *Investigating Authorship in Classical Arabic Poetry Using Large Language Models*. Ph.D. thesis, University of Leeds.

Lama Alqurashi, Serge Sharoff, Janet Watson, and Jacob Blakesley. 2025. Bert-based classical arabic poetry authorship attribution. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6105–6119, Abu Dhabi, UAE. Association for Computational Linguistics.

Alaa Saleh Altheneyan and Mohamed El Bachir Menai. 2014. Naïve bayes classifiers for authorship attribution of arabic texts. *Journal of King Saud University - Computer and Information Sciences*, 26(4):473–484. Special Issue on Arabic NLP.

Fetoun Mansour AlZahrani and Maha Al-Yahya. 2023. A transformer-based approach to authorship attribution in classical arabic texts. *Applied Sciences*, 13(12).

Ali El Filali. 2022. Hindawi books dataset. https://huggingface.co/datasets/alielfilali01/Hindawi-Books-dataset. Accessed: 2025-08-10.

Hassina Hadjadj and H. Sayoud. 2021. Arabic authorship attribution using synthetic minority oversampling technique and principal components analysis for imbalanced documents. *International Journal of Cognitive Informatics and Natural Intelligence*, 15(1):1–17.

Javier Huertas-Tato, Álvaro Huertas-Garcia, Alejandro Martin, and David Camacho. 2022. Part: Pre-trained authorship representation transformer. *arXiv preprint arXiv:2209.15373*. Preprint.

Kanishka Silva, Burcu Can, Frédéric Blain, Raheem Sarwar, Laura Ugolini, and Ruslan Mitkov. 2023. Authorship attribution of late 19th century novels using gan-bert. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Student Research Workshop), Volume 4*, pages 310–320. Association for Computational Linguistics.

Kanishka Silva, Ingo Frommholz, Burcu Can, Frédéric Blain, Raheem Sarwar, and Laura Ugolini. 2024. Forged-gan-bert: Authorship attribution for llm-generated forged novels. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 325–337. Association for Computational Linguistics.

Chiyu Zhang and Muhammad Abdul-Mageed. 2022. Bert-based arabic social media author profiling. *arXiv preprint arXiv:1909.04181*.

A Detailed Error Analysis

Inspection of the confusion matrix of the predicted authors revealed that **Tharwat Abaza** was often misclassified as **Ahmad Shawqi** and **Mohamed Hussein Heikal** due to narrative similarities. **Fouad Zakaria** and **Abd al-Ghaffar Mikkawi** occasionally confused, likely due to shared philosophical themes.

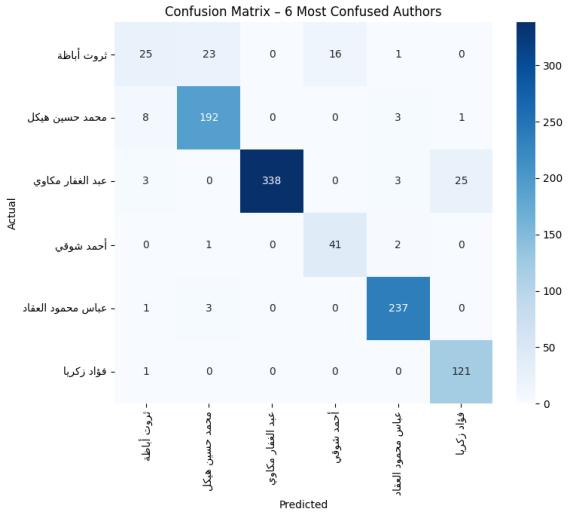


Figure 2: Confusion matrix showing frequent misclassifications between authors with overlapping styles.

B Soft-Voting Ensemble

In the soft-voting ensemble, the class probability distributions predicted by each model are averaged before selecting the final class label. Formally, let $p^{(m)} \in R^K$ denote the probability vector predicted by model m over K classes, and let M be the total number of models. The ensemble probability distribution $\hat{p}$ and the final predicted label $\hat{y}$ are defined as:

$$\hat{p} = \frac{1}{M} \sum_{m=1}^{M} p^{(m)}, \quad \hat{y} = \arg\max_k \hat{p}_k$$

where $\hat{p}$ represents the averaged probability distribution and $\hat{y}$ is the predicted class corresponding to the maximum probability.

C Weighted Loss Function

Formally, let $y_i \in \{1, \ldots, K\}$ denote the true class label of the i-th sample, $p_{i,c}$ the predicted probability for class c, and w_c the weight assigned to class c. The weighted cross-entropy loss is given by:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i} \log p_{i,y_i}$$

where N is the number of training samples and K the number of classes.

The weights w_c are set inversely proportional to the class frequencies, following the "balanced" option in `sklearn.compute_class_weight`:

$$w_c = \frac{N}{K \cdot n_c},$$

where n_c is the number of samples belonging to class c. This ensures that underrepresented classes receive higher weights during training.

D Preprocessing Examples

Duplicate Removal

The following excerpt, shown in Figure 3, appeared multiple times in the dataset and was reduced to a single occurrence during preprocessing:

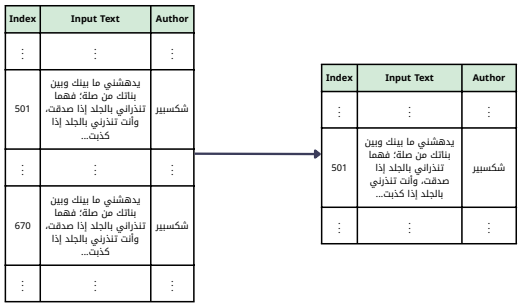
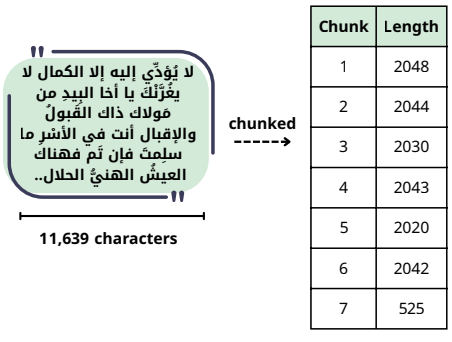


Figure 3: Example of a duplicate sample being reduced to one unique sample.

Splitting Large Texts

Figure 4 illustrates how a long text of 11,639 characters was split into seven smaller chunks of approximately 2,000 characters each, respecting the tokenizer's maximum input length.



Chunk	Length
1	2048
2	2044
3	2030
4	2043
5	2020
6	2042
7	525

Figure 4: Example of length splitting: a long text was divided into seven chunks with sizes [2048, 2044, 2030, 2043, 2020, 2042, 525].