Automated Era Recognition of Arabic Poetry: A Comparative Study of Deep Learning Models

Shaden Alsuhayan
Dept. of Computer Engineering
Imam Abdulrahman Bin Faisal University
Dammam, Saudi Arabia
2220007144@iau.edu.sa

Haneen Alhabib
Dept. of Computer Engineering
Imam Abdulrahman Bin Faisal University
Dammam, Saudi Arabia
22220001259@iau.edu.sa

Haneen Alomri
Dept. of Computer Engineering
Imam Abdulrahman Bin Faisal University
Dammam, Saudi Arabia
2210003424@iau.edu.sa

Fatima Alawami
Dept. of Computer Engineering
Imam Abdulrahman Bin Faisal University
Dammam, Saudi Arabia
2220005142@iau.edu.sa

***Abstract*— Arabic poetry has undergone rich historical, linguistic, and stylistic development over the centuries, making automated poetic era recognition a challenging natural language processing (NLP) task. This study expresses Arabic poetry era recognition as a multi-class text classification problem where each poem in which a poem is classified to one of twelve historical eras. A unified experimental pipeline is retained to guarantee a fair comparison between traditional deep learning models and transformer-based architectures. Traditional models are trained from scratch using word-level representations, whereas transformer models are fine-tuned from pretrained Arabic language encoders. All approaches are evaluated in term of accuracy, macro F1-score, and weighted F1-score on a held-out validation set. Experimental results have shown that transformer-based models significantly outperform traditional models, with CAMeLBERT-CA achieving the overall best performance. The analysis of confusion matrices also indicates that most misclassifications arise between historically adjacent eras, as there is an overlapping of historical and stylistic eras in Arabic poetry. These findings demonstrate the effectiveness and benefice of implementing transform-based NLP techniques for Arabic poetry analysis and focus on the remaining challenges related to stylistic ambiguity and class imbalance.**

Keywords—Natural Language Processing, Text classification, Arabic language, Arabic Poetry, Convolutional neural networks, Deep learning, Transformer-based models.

I. Introduction

Natural Language Processing (NLP) has achieved remarkable level of success in a broad range of language understanding tasks, allowing machines to comprehend and analyze textual data. Applications that have been developed in Arabic NLP include sentiment analysis, named entity recognition, and machine translation. Nevertheless, the linguistic, structural, and cultural complication of Arabic poems makes the automated analysis of Arabic poetry somewhat underexplored.

The Arabic poetry is a very essential part of the Arabic literary heritage with its rich syntactic, morphological, and stylistic diversity. In addition to usual grammatical and lexical structures, poetry depends strongly on poetic forms (e.g. meter, rhyme, figurative language) that are challenges for Automated Natural Language Processing (ANLP) systems. Identifying the historical era of a poem is a useful task for literary analysis applications. However, automatic Arabic poetry era classification remains difficult due to the complexity of poetic language nature, lexical overlap across eras, and the limited availability of balanced annotated datasets. Another source of difficulty lies in the nature of era recognition task itself, as increasing the number of target eras substantially raises classification complexity.

Most current research in Arabic NLP has primarily concentrated on Modern Standard Arabic (MSA) or Arabic Standard Modern Written (ASMW) text, prioritizing functional and informational language rather than artistic literature. Moreover, the models that are trained mostly on contemporary prose have a tendency to struggle to generalize to poetic text, especially when the task involves separating historical and stylistic variation. This gap highlights the need for specialized methodology to better deal with the linguistic richness and depth of Arabic poetry.

The recent developments in deep learning have contributed to the improvement of text classification across many NLP tasks. Traditional deep learning models, such as convolutional neural networks and recurrent architectures, depend on fixed word-level representations and local or sequential patterns. These architectures may limit the ability to capture long-range dependencies and stylistic variation that occurs in poetry, even though being successful in many other applications.

More recently, transformer-based models have established a paradigm shift by leveraging both self-attention mechanisms and pretraining on diverse and large corpora. Several pretrained transformer models have been proposed for Arabic NLP tasks, each one is different in pretraining data and linguistic focus. Despite the fact that they perform well in general NLP tasks, a systematic comparison of traditional and transformer-based models for Arabic poetry era recognition remains limited.

Inspired by these developments, this study focuses on conducting a full comparative evaluation and analysis of traditional deep learning and transformer-based architectures under a unified pipeline.

II. Objectives and Motivation

The primary objectives of this study are:

- To formulate Arabic poetry era recognition as a multi-class text classification problem that covers twelve historical eras.

- To implement and evaluate traditional deep learning models (TextCNN and BiLSTM) and transformer-based models (AraBERT, AraPoemBERT, QARiB, and CAMelBERT-CA) within a uniformed framework.
- To compare models performance based on general evaluation metrics, including accuracy and weighted F1-score.
- To examine how different pretraining strategies effect performance of classification in Arabic poetry.

The motivation for this work comes from the growing interest in applying modern NLP techniques to Arabic literature, which has special challenges due to its richness and depth. Current studies usually focus on individual models or lack controlled experimental conditions, which make fair comparison hard. In addition to that adopting a twelve class era formulation introduces more difficulty to the task. Furthermore, this study aims to explain the strengths and limitations of traditional and transformer-based models to determine which are better suited for historical Arabic poetry analysis by comparing them under same conditions.

This piece of work contributes to the field of NLP by enhancing automated understanding and classification of this unique genre of literature.

III. RELATED WORKS

The emergence of Artificial Intelligence has greatly transformed the way Arabic Poetry is classified, chronologically, through automated classification. Traditional classification methods have relied heavily on human feature extraction and analysis; however, the use of artificial intelligence has allowed researchers to develop deep learning models that allow for automatic feature extraction from large datasets collected from historical periods of Arabic Poetry. By these advanced techniques, researchers are now able to detect highly complex relationships among multiple Arabic writing styles distinguish one literary period from another.

In the study conducted by [1], a strong deep learning approach was adopted, and a Convolutional Neural Network (CNN) System was constructed that can classify five distinct literary eras of Arabic poetry. The CNN creates a dataset of 45,696 labelled poems. Each poem was converted to vector space using FastText Word Embedding techniques. Subsequently, CNN layers were created to find local semantic features within the generated vector space, achieving an accuracy rate of 80.1%. This indicates that by using deep learning methods, it is possible to track the evolution of the stylistic features of Arabic poetry without rigid, handcrafted rules.

Using this technique and focusing mostly on the differentiating characteristics of modern and classical Arabic structure, [2] created a specialized system to identify Modern Arabic Poetry. This machine learning algorithm recognizes the unique "visual" and "structural" features found in modern free verse, such as alignment of the text and the absence of any rigid rhyme structure (as in classical poetry). Prior to feeding the data into the machine learning algorithm, the text underwent preprocessing to remove diacritical marks and normalize Arabic characters. The researchers concluded that while a classical Arabic poem can be identified by its strict metrical structure, the identification of modern Arabic poems requires the identification of features that characterize their dynamic flow structure.

As deep learning architectures evolved, a deep learning framework using Bidirectional Gated Recurrent Units (Bi-GRU) was introduced in [3]. This tool processes the poem's verses at a character level from both forward and backward directions, allowing the Bi-GRU model to gain insight into the rhythmic context of each verse, allowing the model to determine which literary era the poem most likely originated from through their bidirectional techniques, surpassing the difficulties of the previous systems wherein complex "Arud" transformations had to be performed to classify meters in poetry, achieving state-of-the-art results in classifying poetic meters.

To handle the issue of authorship attribution that is associated with era recognition, [4] created a Stacked Long Short-Term Memory (LSTM) network. The initial step of the process involved embedding the characters of the poem, as mentioned previously, into a high-dimensional vector space. The embeddings are then passed through multiple LSTM layers to learn long-term memory of writing style. The final output layer predicts the poet, and consequently the era, by recognizing unique stylistic fingerprints, confirming that deep learning models could distinguish between poets of the same era with high precision.

In an effort to combine different models in order to achieve the best possible performance, [5] has proposed a combination of convolutional neural networks (CNN) and long short-term memory networks (LSTM). The model operates in the following two steps; the first step is when the CNNs function as a feature extractor of local patterns of vocabulary usage, such as those found during the Jahili or Abbasid periods. The second step is when the LSTM layers take those features and analyze how they are sequenced, which allows them to capture the rhythmic quality of the prosodic feature in both cases. This hybrid model outperformed the same task when done with either just the CNN or the LSTM networks alone, because it was able to equally balance both local semantic detection and temporal sequence modelling.

Though deep learning has been so successful, [6] provided a comparative analysis of the three major architecture types of Machine Learning, Template Matching and Deep Learning (VGG16). Their results showed that while deep learning models performed much better with larger training sets, many of the traditional machine learning models performed better with smaller data sets that had specific rules for Arabic prosody. Thus, they recommend that in resource constrained environments, a simpler model may still provide better performance than a more complex neural network.

The introduction of the 'Ashaar' framework in [7] marked a significant technological transition, with a dataset made up of 2.7 million verses, and utilizing BERT Fine-tuning for Poetry Tasks. Tokens are created for each verse through models and the Transformer Architecture's self-attention has been leveraged to weigh the contextual importance of individual keywords when modelling word placement, allowing for a much more accurate distinction of pre-Islamic, Umayyad, Abbasid, and Modern periods of Arabic poetry than was ever possible by traditional RNN based techniques.

Addressing the problem of class imbalance in historical datasets, [8] investigated the use of Data Augmentation (DA) using Generative Pre-trained Transformers (GPT-2). This tool works by generating synthetic verses that mimic the style of underrepresented eras, such as the pre-Islamic period. These synthetic samples are then combined with the original real data to train the classification model. The study showed that this augmentation strategy significantly improves the robustness of classifiers, preventing them from being biased toward the more abundant Abbasid or Modern era data.

Pushing the boundaries of domain-specific modelling, [9] developed 'AraPoemBERT', a language model pre-trained exclusively on Arabic poetry. This tool works by masking random words in a poetic verse and training the model to predict them based on the context, effectively teaching it the archaic vocabulary and complex syntax of classical Arabic. When applied to era classification, AraPoemBERT outperformed general Arabic models like AraBERT, achieving unprecedented accuracy by leveraging its deep understanding of poetic nuances that general models often miss.

Finally, a comprehensive performance analysis comparing Transformer models against Bi-LSTMs was provided in [10]. This tool works by utilizing the "attention" maps of the Transformer to visually highlight which words (e.g., specific archaic terms) influenced the model's decision to classify a poem into a specific era. Their study demonstrated that while Bi-LSTMs are computationally efficient, Transformer-based models provide better interpretability and accuracy for complex classification tasks, offering valuable insights for literary scholars.

IV. Data Acquisition & Preprocessing

For this study, the data was obtained is the FannOrFlop dataset, which is a comprehensive collection of Arabic poetry and its corresponding explanations. The dataset contains more than 6,000 samples of Arabic poetry, covering many different poetic forms, periods, styles and meters, and is accessible on Hugging Face.

The proposed workflow includes cleaning and normalizing data, splitting the data into training and validation datasets, balancing class distribution, model implementation, and evaluating the model's performance using performance metrics. A diagram of the workflow for this study is shown in Fig. 1.

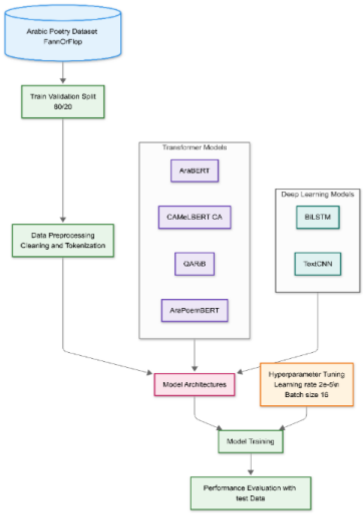


Fig. 1. Workflow diagram of the proposed Arabic poetry classification study

A. Data Acquisition

This study utilizes the FannOrFlop dataset introduced in [11], The data consists of a set of Arabic poems labelled with their corresponding historical eras. The dataset components: an ID, title and author for the poem's name and poet, source for the original reference, and tags providing metadata such as the historical era, genre, and meter. In order to classify poems based or historical eras , the "poem verses" field, which holds the poem's text was used as input, while "era" was set to be the target label.

B. Data Preprocessing

Arabic text is very rich in its morphological description and therefore may inject unnecessary noise into the Embedding Space. To handle this we created a multi-step Cleaning Pipeline as highlighted in Fig. 2 .

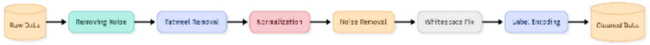


Fig. 2. Text cleaning pipeline applied during preprocessing.

Following text cleaning, the categorical labels (eras) were encoded into numerical format using a Label Encoder, mapping the 12 distinct historical eras to integers ranging from 0 to 11.

C. Class Balancing

Prior to data preprocessing, an exploratory analysis to examine the characteristics of the dataset and identify any potential sources of bias was performed. Fig. 3 shows that there is a strong class imbalance present in the dataset; since the Abbasid Era (العصر العباسي) dominates 2,342 samples, this means that the corpus has a very high representation of the Abbasid Era. On the other hand, the Islamic Era (العصر الإسلامي) and the Era Between the Two States both have a much lower amount of representation, with are only 294 samples of the Islamic Era and 321 samples of the Era Between the Two States. Due to this strong bias toward the Abbasid Era, a robust strategy to mitigate the risk of model overfitting is required.

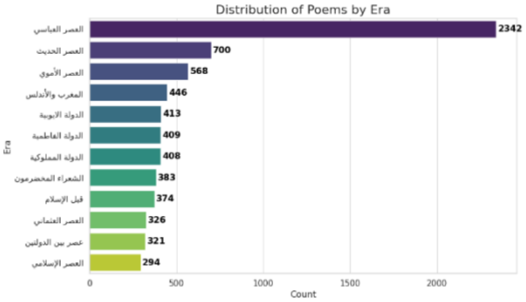


Fig. 3. Class distribution of historical eras before balancing

To address the imbalance identified in Fig. 3, an upsampling strategy was used that allowed to separate training samples by class which enabled resampling all the samples in the minority classes with replacement until they matched the number of samples in the majority class. By applying this, a perfectly balanced training data set that contained approximately 1,874 samples from each of the eras was created, which provided final balanced training size of 22,488. As a result of this step, the model treated all historical eras with equal importance during the optimization process.

D. Tokenization Strategy

To establish a tokenization approach, word count per poem was calculated to examine the formatting and structure of poems. Fig. 4 illustrated the strong right skewness of poem lengths, as most samples approximately 97% contain less than 500 words.

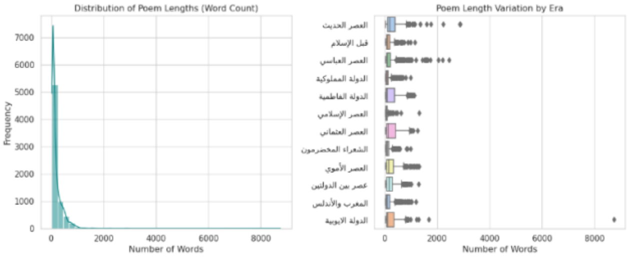


Fig. 4. Poem length distribution

Using the distribution of data and considering the computing limitations of each architecture, the maximum sequence lengths for each architecture are determined to fall between 32 and 512 tokens. This approach enables striking the best balance between the model's ability to represent information and its computational efficiency. All sequences longer than this maximum length were truncated while all sequences shorter than the length of the maximum length were padded to create a uniform dimension for each tensor used in the forward pass.

V. Model Development & Training

A. Overall Experimental Pipeline

This study formulates Arabic poetry era recognition as a multi-class text classification task, where each input poem is assigned to one of the twelve historical eras. All models in this study, ranging from traditional deep learning architectures to modern transformer-based ones, were trained and evaluated on the same dataset, using a unified experimental pipeline to ensure a fair and controlled comparison as shown in Fig. 5.

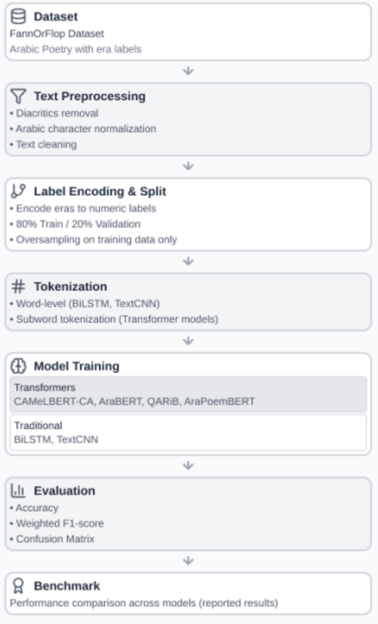

Dataset
FannOrFlop Dataset
Arabic Poetry with era labels
Text Preprocessing
• Diacritics removal
• Arabic character normalization
• Text cleaning
Label Encoding & Split
• Encode eras to numeric labels
• 80% Train / 20% Validation
• Oversampling on training data only
Tokenization
• Word-level (BiLSTM, TextCNN)
• Subword tokenization (Transformer models)
Model Training
Transformers
CAMeLBERT-CA, AraBERT, QARiB, AraPoemBERT
Traditional
BiLSTM, TextCNN
Evaluation
• Accuracy
• Weighted F1-score
• Confusion Matrix
Benchmark
Performance comparison across models (reported results)

Fig. 5. Overall experimental pipeline for Arabic poetry era classification

The overall pipeline follows a consistent sequence of steps shared across all different approaches. First, the dataset was cleaned and preprocessed (as described in the previous section). Poetic era labels were then encoded into numeric class identifiers using a label encoding scheme with mappings between label IDs and era names. The dataset was split into training (80%) and validation (20%) subsets using stratified sampling, preserving the original dataset distribution across eras. Due to the relatively small size of some historical classes, a separate test set was not introduced, instead, the validation set was treated as a held-out evaluation split across all experiments.

To address class imbalance, resampling strategies were applied only to the training set, while keeping the validation set in its original imbalanced form. All models were trained in a supervised manner using the cross-entropy loss function for multi-class classification. Training was conducted for a predefined number of epochs, with validation performed at the end of each epoch. For transformer-based models, fine-tuning was done end-to-end using pretrained encoders with task-specific classification heads, whereas traditional models were trained from scratch. For each approach, the best-performing model checkpoint was selected based on validation performance and saved, guaranteeing reproducibility and allowing future experimentation without retraining. This unified pipeline enables fair comparison in this study between traditional and transformer-based deep learning models.

B. Traditional Deep Learning Models

Two widely adopted and well-known traditional deep learning architectures were implemented: TextCNN and Bidirectional LSTM (BiLSTM). These models rely on word-level representations and were trained from scratch.

For both models, poem texts were processed using word-level tokenization following Arabic normalization. A vocabulary was constructed from training data to prevent data leakage, with unseen words mapped to <UNK> token. Padding (<PAD>) was used to ensure uniform sequence lengths. Poems were encoded as fixed-length sequences, with longer texts truncated and shorter ones padded.

1) TextCNN

The TextCNN model detects discriminative local patterns through convolutional filters applied over word embeddings. Its architecture consists of:

- A trainable embedding layer that maps word indices to dense vector representations.
- Multiple one-dimensional convolutional layers with varying kernel sizes (3, 4, and 5), to capture n-gram (e.g. 5-gram) features of different lengths.
- ReLU activation function.
- Global max pooling, which extracts the most prominent (max) feature from each filter across sequences.
- A dropout layer for regularization to prevent overfitting.
- A fully connected classification layer producing logits over the twelve poetic eras (classes).

2) BiLSTM

The BiLSTM model is designed to capture long-range sequential dependencies within poems. Its architecture consists of:

- A trainable word embedding layer shared across all tokens.
- Stacked bidirectional LSTM layers, enabling the model to process the poetic sequences in both forward and backward directions.
- Concatenation of the final hidden states from both directions to form a global sequence representation.
- A dropout layer for regularization to prevent overfitting.
- A fully connected classification layer producing logits over the twelve poetic eras.

By modeling context from both past and future tokens, the BiLSTM is well-suited for capturing structural and syntactic dependencies that extend beyond local n-grams, which are common in poetic language.

3) Training Configuration for Traditional Models

Both models were trained in a supervised learning manner. All training was performed using GPU acceleration.

TABLE I. TRAINING CONFIGURATIONS USED FOR TRADITIONAL MODELS.

Loss function	Cross-entropy loss
Optimizer	Adam
Batch	Mini-batch training with fixed batch sizes
Epoch	10
Validation	Performed at the end of each epoch.

C. Transformer-based Models

In contrast to traditional models, transformer-based architectures leverage contextualized subword representations pretrained on large Arabic corpora. Four transformer models were fine-tuned in this study: AraBERT, AraPoemBERT, QARiB, and CAMelBERT-CA. All models follow a common high-level fine-tuning paradigm and are built upon the standard BERT encoder architecture [12], differing primarily in their pretraining corpora and linguistic focus.

1) Fine-tuning Strategy

Each transformer model was adapted for poetry era classifications by using a pretrained transformer encoder as the backbone and attaching a task-specific classification head. The contextual representation of the [CLS] token was used as a summary that captures the overall style and meaning of the whole poem and was fed this to the classifier to predict the poem era. Training and evaluation were implemented using Hugging Face Transformers framework.

2) Tokenization and Input Encoding

Transformer models employ subword tokenization, which is suited for Arabic morphology. Each model uses its original tokenizer based on its pretraining scheme. The tokenization process included converting text into subword token IDs, creating attention masks, truncating sequences to a predefined maximum length, and applying dynamic padding of batches.

3) Transformer-Based Models Description

Although the fine-tuning procedure was unified, each transformer model introduces distinct characteristics.

TABLE II. TRANSFORMER-BASED MODELS USED IN THIS STUDY AND THEIR PRETRAINING CHARACTERISTICS.

AraBERT	A general-purpose Arabic BERT model used as a baseline. It is used to evaluate how well standard Arabic transformers capture poetic era patterns [13].
AraPoemBERT	A domain-specific model pretrained exclusively on Arabic poetry text to learn poetic stylistic and lexical patterns [14].
QARiB	A BERT-based Arabic language model pretrained on a large and diverse collection of Arabic tweets and text, and it was intended to be fine-tuned on a downstream task [15].
CAMelBERT-CA	A model pretrained on CA (classical Arabic) data and was intended to be fine-tuned on NLP tasks [16].

These differences enable the study to assess the effect of pretraining domain and focus on downstream performance in poetic era classification.

4) Training Configuration for Transformer-Based Models

Transformer models were fine-tuned under a coherent optimization framework. All training was performed using GPU acceleration.

TABLE III. TRAINING CONFIGURATIONS USED FOR TRADITIONAL AND TRANSFORMER-BASED MODELS.

Optimizer	AdamW.
Learning rate	2×10^{-5}, selected from standard fine-tuning ranges of BERT-based models.
Learning rate scheduling	Cosine decay with warmup.
Warmup ratio	10%
Batch size	16 per device, with gradient accumulation applied when required to achieve an effective batch size of 32.
Weight Decay	0.01
Validation	Performed at the end of each epoch.

Training was conducted under predefined number of epochs. In particular, AraBERT, QARiB, and CAMelBERT-CA were trained on 10 epochs, while AraPoemBERT was trained for 8 epochs which reflects its faster convergence.

VI. EVALUATION AND ANALYSIS

Since no separate test set was introduced, all models were evaluated using the held-out validation set which served as the evaluation split across all experiments. The validation set was used exclusively for evaluation and was never involved in training or fine-tuning.

To provide a comprehensive assessment of performance across the twelve poetic eras, accuracy, macro averaged F1-

score, and weighted F1-score were used as the major evaluation metrics. Class-wise precision, recall, and F1-score were additionally reported in the source code [17], which is publicly available, to support detailed analysis.

In addition, a confusion matrix was generated for each model to enable error analysis and to better understand inter-era misclassification patterns.

TABLE IV. EVALUATION RESULTS OF ALL MODELS.

Model	*Accuracy*	*Macro F1*	*Weighted F1*
TextCNN	0.363	0.103	0.228
BiLSTM	0.336	0.099	0.219
AraBERT	0.652	0.592	0.650
AraPoemBERT	0.513	0.401	0.492
QARiB	0.578	0.524	0.577
CAMelBERT-CA	**0.706**	**0.656**	**0.703**

The TextCNN model achieved an overall accuracy of 36.3% with a weighted F1-score of 0.228, indicating limited classification ability across the twelve poetic eras. As reflected in the confusion matrix (Fig. 6(a)), the model is heavily biased toward the Abbasid era (العصر العباسي), achieving extremely high recall for this class, while failing to correctly identify many smaller eras. The extremely low macro F1-score 0.103 underlines severe imbalance in class-wise performance. There are several eras where the recall is close to zero, meaning that the model fails to generalize beyond dominant lexical patterns. This behavior is expected, as TextCNN is based primarily on local n-gram features and lacks methods for modeling long-range context dependencies.

The BiLSTM model reached an accuracy of 33.6% and a weighted F1-score of 0.219, slightly lower than that of TextCNN with regards to the overall accuracy. Despite its ability to model sequential dependencies, the macro F1-score (0.099) shows that the model performs poorly across a number of eras. As shown in its confusion matrix (Fig. 6(b)), BiLSTM show heavy preference towards the Abbasid era (العصر العباسي), achieving very high recall for this class while misclassifying most samples from other eras. This implies that, despite sequential modeling provides some contextual knowledge, word-level representations alone are insufficient for separating historical and stylistic overlaps in Arabic poetry.

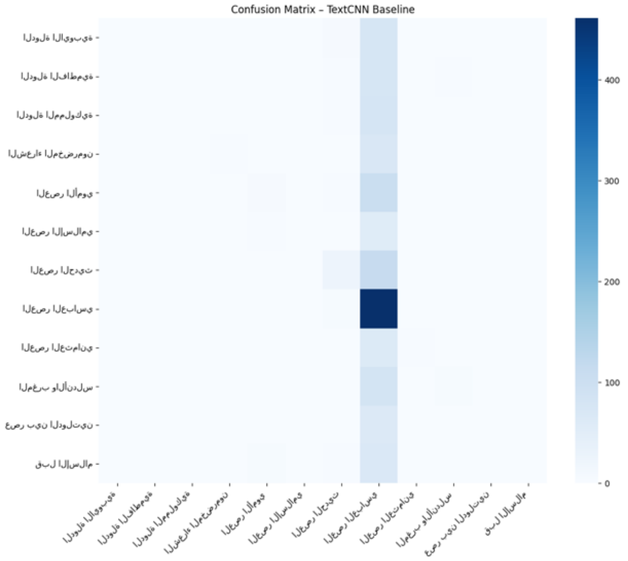
Transformer-based models noticeably outperform traditional deep learning approaches across all reported evaluation metrics. This improvement is attributed by the fact that contextualized subword representations and self-attention mechanisms, which helped modelling of largescale dependencies in poetic text.

AraBERT got the accuracy of 65.2%, the macro F1-score of 0.592 and the weighted F1-score of 0.650. These results show that a general-purpose Arabic transformer is able to effectively extract stylistic and lexical patterns. The confusion matrix (Fig. 6(c)) illustrates that most of the errors occur between historically adjacent eras, such as transitional periods, reflecting actual linguistic overlap and not random misclassification.

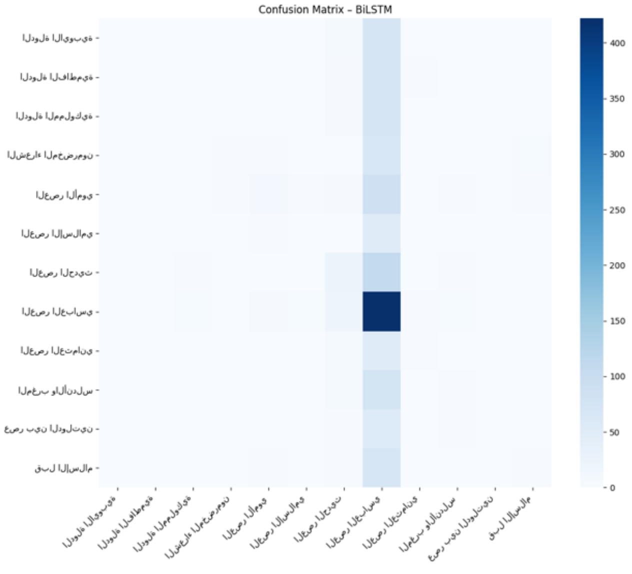
AraPoemBERT achieved an accuracy of 51.3%, with a macro F1-score of 0.401 and a weighted F1-score of 0.492. Although its overall performance is poorer than AraBERT, the model can recognize certain poetic eras better than traditional baselines. The confusion matrix in Fig. 6(d) demonstrates that misclassifications for several stylistically distinct eras are reduces, showing that poetry-specific pretraining is useful in capturing literature patterns and trends, even though the model was trained for fewer epochs.

QARiB had an accuracy of 57.8%, macro F1-score of 0.524 and weighted F1-score of 0.577, placing it between AraPoemBERT and AraBERT in terms of overall performance. The model shows balanced performance across multiple eras, benefiting from its diverse and large pretraining corpus. As shown in the confusion matrix in Fig. 6(e), misclassifications are mainly among historically and stylistically overlapping periods of time.

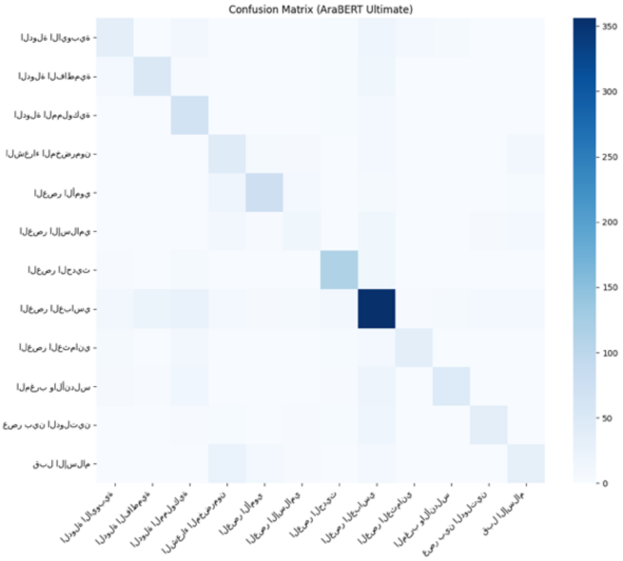
CAMeLBERT-CA achieved the highest performance among all models, with an accuracy of 70.6%, a macro F1-score of 0.656, and a weighted F1-score of 0.703. Its confusion matrix (Fig. 6(f)) presents strong diagonal dominance and weaker off-diagonal confusion, indicating enhanced separation between poetic eras. The model's high performance can be attributed to its linguistically informed pretraining on Classical Arabic, which aligns with the language characteristics of historical Arabic poetry.



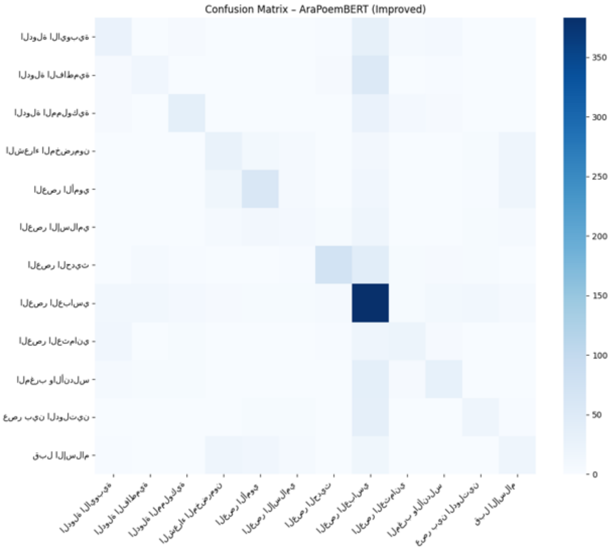
(a) TextCNN Confusion Matrix



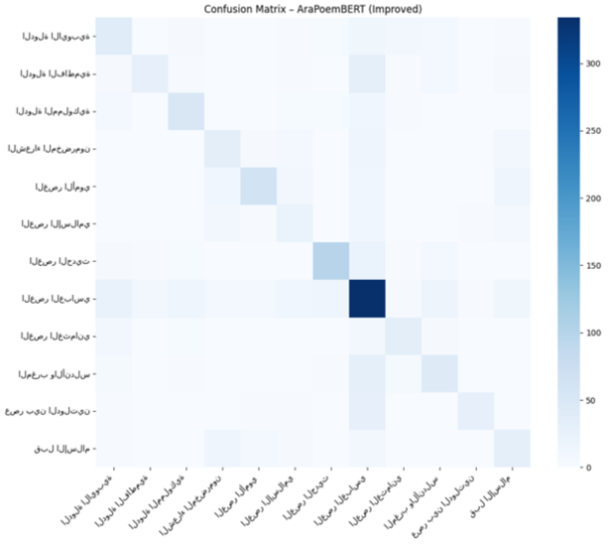
(b) BiLSTM Confusion Matrix



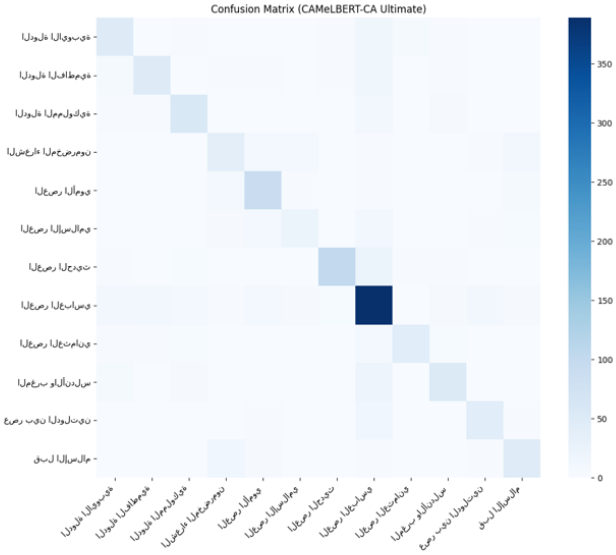
(c) AraBERT Confusion Matrix



(d) AraPoemBERT Confusion Matrix



(e) QARiB Confusion Matrix



(f) CAMelBERT-CA Confusion Matrix

Fig. 6. Confusion metrices for Arabic poetry era classifiation

APPENDIX

Name (Team Leader): Shaden Aksuhayan

Student ID: 2220007144

Group No.: 2

Name (Team Member 2): Haneen Alhabib

Student ID: 2220001259

Group No.: 2

Name (Team Member 3): Haneen Alomri

Student ID: 2210003424

Group No.: 2

Name (Team Member 4): Fatima Alawami

Student ID: 2220005142

Group No.: 2

ACKNOWLEDGMENT

We would like to express our sincere thanks to Dr. Mustafa Youldash for his support and guidance throughout this project. And we also would like to thank the researchers whose valuable work and contributions set the foundation for this project.

VII. CONCLUSION & FUTURE WORK

In all models, confusion matrices show consistent trends that are typical of poetry era classification. In particular, high confusion is clearly observed between historically adjacent and transitional eras, whereas better performance is obtained for eras with different stylistic and lexical patterns. Thus, many misclassifications indicate ambiguity in the dataset rather than flaws in model design.

Traditional deep learning models struggle with this twelve-class poetry era classification task, whereas transformer-based models demonstrate noticeable performance improvements across all evaluation metrics. Moreover, linguistically informed domain-specific pretraining yields the best results, as confirmed by the high performance of CAMeLBERT-CA. The analysis of confusion matrix analysis upholds the quantitative findings and reveals historical connections between eras. Overall, the results point out the effective abilities of transformer-based architectures for classifying the Arabic poetry era, while emphasizing ongoing challenges related to overlapping of styles and the imbalance of classes.

Future work may address these challenges through integration of larger and more balanced dataset, and exploring poetry-specific features such as rhyme and meter. Additionally, explainable AI techniques could be used to achieve a greater understanding and deeper insights into how models analyze Arabic poetry.

References

[1] Orabi, M., El Rifai, H., & Elnagar, A. (2020). Classical Arabic Poetry: Classification Based on Era. *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, Antalya, Turkey.

[2] Zeyada, M., & Researchers. (2020). A Machine Learning Method for Recognizing Modern Arabic Poems. *Information Processing & Management*, 57(5), 102286.

[3] Al-Shaibani, M. S., Alyafeai, Z., & Ahmad, I. (2020). Meter Classification of Arabic Poems Using Deep Bidirectional Recurrent Neural Networks. *Pattern Recognition Letters*, 138, 176-182.

[4] Albaddawi, H. S., & Abandah, G. A. (2021). Arabic Poetry Authorship Attribution Using Deep Learning. *Jordan Journal of Electrical Engineering*, 7(4), 312-326.

[5] Al-Falahi, A., Al-Shargabi, B., & Al-Qurishi, M. (2022). A Hybrid Deep Learning Model for Arabic Poetry Classification. *Journal of King Saud University - Computer and Information Sciences*, 34(6), 3456-3465.

[6] Almomani, A., Nahar, K. M. O., & Abual-Rub, M. S. (2023). Automatic Recognition of Arabic Poetry Meter Using Machine Learning, Template Matching, and Deep Learning. *2023 3rd International Conference on Computing and Information Technology (ICCIT)*, Tabuk, Saudi Arabia.

[7] Alyafeai, Z., Al-shaibani, M. S., & Ahmad, M. (2023). Ashaar: Automatic Analysis and Generation of Arabic Poetry Using Deep Learning Approaches. *IEEE Access*, 11, 89234-89245.

[8] Refai, D., Abu-Soud, S., & Abdel-Rahman, M. J. (2023). Data Augmentation Using Transformers and Similarity Measures for Improving Arabic Text Classification. *IEEE Access*, 11, 12345-12356.

[9] Qasem, F., & Researchers. (2024). AraPoemBERT: A Pretrained Language Model for Arabic Poetry Analysis. *arXiv preprint arXiv:2403.12392*.

[10] Mutawa, A., & Sruthi, S. (2025). A Comparative Evaluation of Transformers and Deep Learning Models for Arabic Meter Classification. Applied Sciences, 15(9), 4941.

[11] Omkar Thawakar, "FannOrFlop," *Huggingface.co*, 2025. https://huggingface.co/datasets/omkarthawakar/FannOrFlop

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019. Available: https://arxiv.org/pdf/1810.04805

[13] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," *arXiv:2003.00104 [cs]*, Mar. 2020, Available: https://arxiv.org/abs/2003.00104

[14] F. Qarah, "AraPoemBERT: A Pretrained Language Model for Arabic Poetry Analysis," *arXiv (Cornell University)*, Mar. 2024, doi: https://doi.org/10.48550/arxiv.2403.12392.

[15] A. Abdelali, S. Hassan, H. Mubarak, K. Darwish, and Y. Samih, "Pre-Training BERT on Arabic Tweets: Practical Considerations," *arXiv.org*, Feb. 21, 2021. http://arxiv.org/abs/2102.10684

[16] "CAMeL-Lab/bert-base-arabic-camelbert-ca · Hugging Face," *Huggingface.co*, Nov. 07, 2023. https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-ca

[17] S. Al-Suhayan *et al*., *"NLP-Project-Automated-Era-Recognition-of-Arabic-Poetry,"* GitHub repository, 2025. [Online]. Available: https://github.com/shadenalsuhayan/NLP-Project-Automated-Era-Recognition-of-Arabic-Poetry.