# Arabic Poet Classifier: Poet Detection from Poetry

Reem Almutairi[1], Rawan Alghamdi[2], Shahad Balkhashir[3], Asmaa Alwaal[4],
Roaa Altunsi[5], Ryam Alsahafi[6], Nuha Makki[7], Rahaf Koshak[8]

Department of Computer Science, King Abdulaziz University, Saudi Arabia
[1]rawadhalmutairi@stu.kau.edu.sa, [2]rsaeedalghamdi0007@stu.kau.edu.sa, [3]skbaalkhashir@stu.kau.edu.sa, [4]aameralwaal@stu.kau.edu.sa,
[5]rmustafaaltunsi@stu.kau.edu.sa, [6]rhameadalsahafi@stu.kau.edu.sa, [7]nalimakki@stu.kau.edu.sa, [8]rabdulkarimkoshak@stu.kau.edu.sa

**This paper presents an innovative approach to classifying Arabic poetry by poet, addressing the growing need to authenticate poetic works in the face of advanced deepfake technologies. Leveraging a comprehensive dataset of 254,630 poems from 348 poets, we implemented various machine learning models, including FastText, LSVC, Random Forest, and Logistic Regression, to identify the authorship of poems. Our methodology encompassed extensive preprocessing steps such as normalization, tokenization, stop word removal, stemming, and transforming text into numerical vectors. Feature extraction techniques, including character n-grams and Term Frequency-Inverse Document Frequency (TF-IDF), were employed to capture the unique stylistic elements of each poet. Among the models tested, the FastText model demonstrated superior performance, achieving a precision, recall, and F1-score of over 90%. The results underscore the effectiveness of machine learning in literary analysis and provide a foundation for further research to integrate additional poetic features such as meter, theme, and era, and to scale up the dataset for enhanced classification accuracy and reliability. This study not only highlights the potential of machine learning in literary analysis, but also sets the stage for future research to incorporate additional poetic features and expand dataset size, thereby enhancing the robustness and reliability of poet classification systems.**

***Index Terms*—Natural Language Processing, Poetry, Arabic Language, Arabic Poetry, Poet Detection, Text Classification, Text Mining**

## I. Introduction

Arabic poetry, an integral part of Arabic literature, has a rich legacy spanning over a millennium, with roots extending back to the pre-6th century era. Historically, Arabic poets have employed poetry as a versatile medium to convey a diverse range of emotions, chronicle events, impart knowledge, and engage in both praise and satire. The discipline of Arabic Prosody, initiated in the 8th century, has played a crucial role in deciphering and classifying the poetic forms and rhythms that characterize classical Arabic poetry. This profound tradition encompasses 16 primary rhythms, each governed by its own set of rules and patterns, formulated by the pioneering scholar al-Farahidi [1].

Arabic poetry, with its extensive historical background, is replete with metaphors and symbolism. This study aims to explore the categorization of Arabic poems, with a particular emphasis on determining and verifying authorship. It will also address the distinct challenges posed by the structural elements of Arabic poetry, including form, rhythm, rhyme, and weight. In recent times, the advent of deepfake technology has enabled the generation of synthetic poetry that closely mimics the style of real poets. This development raises concerns about the authenticity of poetic works – whether a poem is a genuine creation of the attributed poet or a deepfake imitation. To address this issue, this paper proposes a machine learning-based approach to classify Arabic poems based on the poet's name. The overarching goal is to distinguish between genuine poems and those generated by deepfake technology.

## II. Related Work

The classification of text using machine learning has been extensively researched in various languages, particularly in English. However, the unique complexities of the Arabic language, including its morphological richness and the necessity of learning classical Arabic for poetic analysis, have posed significant challenges. Despite these hurdles, recent studies have begun to explore the potential of machine learning in classifying Arabic texts, including Arabic poetry.

Ahmed et al. (2019) addressed this gap by proposing a method to classify modern Arabic poetry into four categories: love, Islamic, social, and political poems. They utilized machine learning algorithms, including Naïve Bayes (NB), Support Vector Machines (SVM), and Linear Support Vector Classification (LSVC), to enhance classification accuracy through data preprocessing steps such as tokenization, removal of non-Arabic terms and stop words, and stemming. Their approach involved building a dataset of modern Arabic poems and segmenting it into words, then applying the machine learning algorithms mentioned above. The results showed that the LSVC algorithm achieved the highest average precision, recall, and F-measure, outperforming NB and SVM. Specifically, LSVC demonstrated maximum precision for social poems, maximum recall, and F-measure for love poems. This study is significant as it navigates the linguistic complexities of Arabic poetry, implementing a structured approach that has proven successful in text classification tasks in other languages [2].

Orabi et al. (2020) explored the classification of Arabic poetry based on its era using a Convolutional Neural Network (CNN)-based deep learning model. They constructed an updated Arabic Poetry Dataset (2020) and used FastText word embeddings for their model. The preprocessing steps included cleaning the dataset by removing punctuation, symbols, numbers, and diacritics, and filtering out stop words and frequent terms. The study involved training two classifiers: a supervised deep-learning classifier using CNN and a FastText-based classifier. The research undertook three classification tasks. First, they performed binary classification of poems into modern and non-modern eras, achieving the highest accuracy and F1-score of 0.913 and 0.914, respectively. Second, they categorized poems into three eras: pre-Islamic & Islamic, Umayyad & Andalusian, and modern, with an accuracy and F1-score of 0.875 each. Third, they classified poems into five distinct eras: pre-Islamic, Islamic & Umayyad, Andalusian, Abbasid, and modern, achieving an accuracy and F1-score of 0.801 and 0.796, respectively. The CNN classifier exceeded the FastText-based classifier, particularly in the multiclass classification task. Assigning class weights helped address the dataset imbalance, improving the classifier's performance. This study significantly contributes to the field by incorporating deep learning techniques and addressing the classification of Arabic poetry across multiple historical periods, providing a comprehensive and refined dataset for future research [3].

Alaa M. El-Halees (2022) employed a transfer learning approach to attribute and verify the authorship of Arabic poetry using Arabic-BERT models. The research tackled two primary tasks: identifying the poet of an anonymous poem and verifying whether a given poem was written by a specific poet. They focused on automatic feature extraction, contrasting with traditional machine learning methods that relied on manual extraction. The dataset consisted of approximately 11,840 poems from 10 poets, with preprocessing steps including the removal of punctuation, diacritics, and non-Arabic characters. The study utilized four Arabic-BERT models (Mini, Medium, Base, and Large), with the Large model achieving the highest performance: an F1-score of 85% for authorship attribution and 96% for authorship verification. These results significantly outperformed traditional methods like k-Nearest Neighbours, Naive Bayes, and Support Vector Machines, , which were also tested for comparison. The research demonstrated the effectiveness of transfer learning in handling the structural intricacies of Arabic poetry, highlighting its potential for future advancements in Arabic text classification [4].

Saad Alanazi (2018) employed a random forest approach for authorship attribution of Classical Arabic poetry. The dataset comprised 100 poems from four poets who lived between the 8th and 12th centuries AD, with each contributing 25 poems in Classical Arabic. After cleansing the dataset by removing punctuation, twelve features were meticulously extracted and applied. These features included metrics such as metre, rhyme, the frequency of letters, part of speech tags of the first word, and the grammatical state of the last word in each verse. The 'Random Forest' method, which averages results of multiple decision trees, was employed, leading to a precision of 76.4%, recall of 66.7%, and F-measure of 64.8%. However, precision and recall varied across poets, indicating potential data insufficiency. The study sheds light on machine learning's promise in analyzing Classical Arabic poetry, showcasing the variety of features that can be extracted and utilized, and underscoring the need for further research [5].

Abandah et al. (2022) delved into the task of classifying Arabic text into the 16 poetry meters using a deep recurrent neural network (RNN) architecture. Their model incorporated bidirectional long short-term memory (LSTM) cells, enabling it to capture dependencies in both forward and backward directions within input sequences. By analyzing rhythmic structures and linguistic features, the model achieved an accuracy of 97.27% in distinguishing between different poetry meters, which is significantly higher than previous work. Additionally, the same model was adept at diacritizing Arabic poetry verses, predicting diacritics based on context. They also proposed a solution that approaches 100% accuracy when multiple verses of the same poem are available by predicting the class from the aggregate probabilities of the multiple verses. This work underscores the challenges of diacritizing poetry due to the poet's meticulous selection of phrases and relaxation of some diacritization rules.

Despite these advancements, the classification of Arabic text, particularly poetry, presents unique challenges. The rich morphological structure, extensive vocabulary, and diverse dialects of the Arabic language complicate the feature extraction and model training processes. Additionally, the limited availability of annotated datasets for Arabic poetry hampers the development of robust machine learning models. Researchers like Farghaly and Shaalan (2009) have emphasized the need for more comprehensive linguistic resources and annotated corpora to advance the field of Arabic NLP [6].

This research focuses on categorizing Arabic poems into several eras based on linguistic changes. It evaluates several categorization algorithms to judge their efficacy in automating this procedure. The technique examines the development of word usage in Arabic poetry to determine the era of a poem by analyzing the structure of hemistichs (parts of poem lines) without taking rhyme patterns into account. This study used several classification techniques to identify poetry from the Abbasid and Andalusian eras. SVM outperformed the other classifiers examined, including decision trees, random forest, and logistic regression. The dataset poems from the Abbasid and Andalusian eras were pre-processed by removing prepositions, repeating letters, and common terms before being separated into training and testing sets. The training data was used to train various classifiers, including logistic regression, random forest, decision trees, and support vector machines. The models, stripped of rhyme and word order, were tested for accuracy in predicting the appropriate age of poetry hemistichs. This study's evaluation metric was model accuracy, the Support Vector Machine (SVM) classifier has the highest accuracy, around 70%. Logistic regression produced the second-highest accuracy [7].

Another research investigates the meter classification of Arabic poems using Deep Bidirectional Recurrent Neural Networks. It investigates a method for automatically classifying the meters of Arabic poetry, which is difficult due to the complexity and diversity of Arabic poetic structures. The authors used deep bidirectional recurrent neural networks (Bi-RNNs) to perform classification tasks. This technique considers context from both directions, which is very useful for sequential data such as poetry. The algorithm can detect complicated patterns and connections in the text, which are critical for correctly detecting the poem meter. The model was trained on a specifically built dataset of 55,440 verses across 14 poem meters, and it achieved 94.32% accuracy on an independent test set [8].

Abbas et al. (2019) significantly advanced Arabic text classification by exploring various machine-learning algorithms. Their study emphasized the importance of tailored preprocessing techniques for Arabic text, including diacritic handling, tokenization, and normalization. Evaluating classifiers such as Support Vector Machines (SVM), Naïve Bayes (NB), and Decision Trees (DT) on Arabic datasets, they found that preprocessing steps such as removing stop words and normalizing text enhance classifier performance. Notably, the Multinomial Naïve Bayes (MNB) model achieved an impressive 70.21% accuracy with word tokenization and no stop words. Abbas et al. also highlighted the effectiveness of ensemble methods like Bagging and Random Forest (RF). Moving forward, they plan to explore lemmatization and Named Entity Recognition (NER) to further enhance classification results [9].

Oweis Alsharif et al. (2013) conducted a study on emotion classification in Arabic poetry using machine learning techniques, aiming to classify poetry into four main emotional categories: elegy, love, pride, and satire. An annotated dataset of 1231 poems of varying lengths was built, and the impact of different preprocessing settings, feature vector dimensions, and machine learning algorithms on classification accuracy was tested. The study relied on two methods for building feature vectors: frequency of unigrams and mutual information. Results showed that complex preprocessing methods, such as stemming and morphological analysis, did not improve classification accuracy, whereas the mutual information method outperformed the others in feature selection. Four algorithms were tested: Naïve Bayes, LibLinear (SVM), LibSVM, and Hyperpipes, with Hyperpipes achieving the highest accuracy of 79% using 2000 features. The findings highlight the challenges posed by the complexity of poetic language and the limitations of available linguistic analysis tools, suggesting the need for improving these tools and exploring new methods for classifying emotions in Arabic texts. [10].

Sakib Shahriar et al.(2023) addressed the automatic classification of emotions in Arabic poetry using deep learning techniques. They developed a novel dataset of 9452 Arabic poems labeled with emotions such as joy, sadness, and love. Various deep learning models were trained on this dataset, including one-dimensional Convolutional Neural Networks (1DCNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM) networks, and the transformer-based AraBERT model. The AraBERT model achieved the highest performance with an accuracy of 76.5% and an F1-score of 0.77, surpassing previous benchmarks. This research underscores the superior performance of transformer-based models in the emotional classification of Arabic poetry and suggests further enhancements in text processing and feature selection to improve model accuracy and efficiency. Key contributions include the introduction of an Arabic poetry dataset with labeled emotions, exploration of various deep learning models, and a performance comparison showing the superiority of transformer-based models. [11].

The paper "Naive Bayes for Classical Arabic Poetry Classification" by Iqbal AbdulBaki Mohammad, published in the Journal of Al-Nahrain University, explores the application of Naive Bayes classification techniques to Classical Arabic Poetry (CAP). The study focuses on the automatic categorization of poetry into various classes such as Ghazal, Medeh, Wasef, and Hijaa, by extracting word roots and using them as features for classification. The method involves preprocessing the text to extract roots without relying on predefined lists, instead assigning weights and order to letters based on their positions. This approach reduces the dimensionality of the feature space and is claimed to require only a small amount of training data to estimate the necessary parameters effectively. Experimental results indicate that while some preprocessing methods like stemming did not significantly improve classification performance, the Naive Bayes model, leveraging word root-based features, performed well across different poetic ages and layers. This suggests potential for broader application in text mining tasks within Arabic literary studies, though the paper notes the challenges posed by the linguistic complexity of Arabic and the subtleties of poetic expression. [12].

The study titled "The Classification of Modern Arabic Poetry Using Machine Learning," authored by Munef Abdullah Ahmed and others, and published in TELKOMNIKA in 2019, explores the application of machine learning algorithms to classify modern Arabic poetry. This research is significant due to the complex nature of Arabic poetry which requires understanding of intricate linguistic rules and cultural contexts. The paper discusses the classification of poetry into four categories—love, Islamic, social, and political—each defined by unique thematic elements. The authors employed Naïve Bayes, Support Vector Machines (SVM), and Linear Support Vector (SVC) algorithms to analyze the texts, with a strong emphasis on data preprocessing to improve accuracy. The results indicate that while all algorithms performed well, Linear SVC showed superior performance across multiple metrics. This study highlights the potential of machine learning in enhancing our understanding of Arabic literary texts and suggests further research to refine these techniques for broader applications in Arabic text analysis. [13].

Al-Falahi et al.(2019) investigate authorship attribution in Arabic poetry, aiming to determine the authorship of specific texts by employing text mining classification techniques such as Naïve Bayes, Support Vector Machine, and Linear discriminant analysis. Utilizing various features including lexical, character, structural, poetry, syntactic, semantic, and specific word features, the authors develop the Arabic Poetry Authorship Attribution Model (APAAM). Through experiments conducted on a dataset of Arabic poetry comprising known poetic texts for training and anonymous poetic texts for testing, the study achieves a high performance accuracy of 99.12%, with attribute-level performance at 98.246%, and technique-level performance at 92.836%. The conclusion highlights the effectiveness of specific word, character, and lexical features in achieving superior performance, while noting the limitations of poetry features, which are constrained by meter, rhyme, and sentence length, making them less ideal for authorship attribution when used independently. [14].

Sahin et all (2018) address the challenge of classifying poetry according to poet amidst the growing volume of text data, focusing on English poetry. Constructing a dataset comprising poems from three different poets, the study applies text categorization techniques, including feature selection via the Chi-Square technique and classification using five algorithms: Sequential minimal optimization, Naive Bayes, C4.5 decision tree, Random Forest, and k-nearest neighbors. While each classifier yields varied results, achieving classification success rates above 70%, Sequential minimal optimization stands out with the highest success rate of 0.7067 F-score when employing 700 features. Notably, as the number of features increases, the success rates of all classifiers improve, except for k-nearest neighbors. The authors suggest that larger datasets and alternative text classification methods, such as semantic classification, may enhance the effectiveness of classifying poetry genres in future research endeavors. [15].

The paper presents a thorough AI-based study aimed at automatically detecting misogyny and sarcasm in Arabic text, both in binary and multiclass scenarios. The key to this AI approach is to differentiate between various misogyny and sarcasm topics found in Arabic tweets on social media networks. The study involves the use of seven state-of-the-art NLP classifiers: ARABERT, PAC, LRC, RFC, LSVC, DTC, and KNNC. To fine-tune, validate, and evaluate these techniques, two Arabic tweet datasets (misogyny and Abu Farah datasets) are utilized. Both datasets are split for training and validation with 70%, and testing with 30%. For the experimental study, two scenarios are proposed for each case study (misogyny or sarcasm): binary and multiclass problems. The best accuracy for misogyny detection is achieved using the AraBERT classifier, with 91.0% for the binary classification scenario and 89.0% for the multiclass scenario. For sarcasm detection, the AraBERT also yields the best accuracy, with 88% for the binary classification scenario and 77.0% for the multiclass scenario. [16].

The author discusses the classification of poetry based on the poet. The study aims to classify poetry according to three different poets: Adryan Rotica, Lamar Cole, and Richard Allen Beevor. The process involves generating a balanced corpus of poetry from these poets and applying text classification techniques to the corpus.In terms of methodology, the study highlights the need to convert text files into numerical data for structured data analysis. Various text classification techniques are employed, including the Chi-Square (CHI) technique for feature selection and classification algorithms such as Sequential Minimal Optimization (SMO), C4.5 decision tree, Random Forest (RF), and k-nearest neighbors (KNN). The best classification rate is 0,7067 F-score that occurs with SMO in 700 features and the worst classifier is KNN. [17].

## III. Dataset

The Arabic poetry collection dataset was organized by associating each poem's verse with its respective poet name. Each poem has other features like meter, era, and theme. The total number of poems was 254,630 for 348 different authors. The dataset was split into training and testing sets using an 80-20 ratio, where 80% of the data was used for training and 20% was reserved for testing. This split was implemented to ensure that the model could be effectively trained while also being evaluated on a substantial portion of the data to measure its performance accurately. Table I represents the full details of the dataset.

| – | N.Poets | N.Poems | N.Words |
|---|---|---|---|
| Training dataset | 126 | 15145 | 1673465 |
| Testing dataset | 126 | 6784 | 89456 |
| Total | 252 | 21929 | 1762921 |

TABLE I
Dataset Details

## IV. Methodology

The methodology of this study involves preprocessing the data, splitting it into training and testing sets, extracting features, and applying classification models. Preprocessing includes handling missing values, normalizing and tokenizing text, removing stop words, stemming, and transforming text into numerical vectors. The data is then split into training and testing sets to ensure fair representation and reliable evaluation. Feature extraction involves character features extraction and Term Frequency-Inverse Document Frequency (TF-IDF) to identify latent topics. Finally, four different models are used for classification, using their strengths to handle all aspects of the Arabic language. This comprehensive approach ensures accurate and efficient Arabic poetry classification.

### *A. The prepossessing stage*

Data preprocessing is a critical step in any text classification task, ensuring that the dataset is clean, consistent, and in a suitable format for machine learning algorithms. This phase involves a series of methodical steps, each designed to enhance the quality and usability of the textual data, thereby improving the overall performance and accuracy of the classification model. The following approaches detail each of these preprocessing steps.

*1) Handling Missing Values:*

The first step in preprocessing involved checking for and addressing missing values within the dataset. Ensuring that no data points were missing was critical to maintaining the integrity and completeness of our analysis.

*2) Normalization:*

Normalization techniques were applied to standardize textual features. This included normalizing specific Arabic characters such as Hamza, Alef, and Yah, which helped enhance consistency and remove variations that could affect classification accuracy.

*3) Tokenization:*

The normalized text was then broken down into individual tokens through a process called tokenization. This step was essential for laying the groundwork for further text analysis.

*4) Stop Word Removal:*

Commonly occurring but contextually insignificant words, known as stop words, were filtered out of the tokenized data. This refinement helped streamline subsequent processing steps by focusing on more meaningful data.

*5) Stemming:*

Stemming reduced tokens to their root forms, which captured the underlying semantic content while minimizing dimensionality. This process consolidated semantically related tokens, enhancing the efficiency of our classifier.

*6) Text Representation:*

Finally, text representation techniques transformed the preprocessed textual data into numerical vectors. This transformation was crucial for facilitating the training of machine learning models, as it converted textual features into a format compatible with various classification algorithms.

### *B. Feature Extraction*

Feature extraction is a fundamental stage in text classification, facilitating the transformation of raw text into numerical representations understood by machine learning algorithms. This process captures key attributes and patterns, laying the foundation for accurate classification models. The following techniques detail various approaches employed to extract crucial information from text data.

*1) Character Features Extraction*

The character features extraction process entails several key steps. Initially, character 3-grams are extracted from the text data to capture sequences of three consecutive characters. Subsequently, feature selection techniques are applied to identify the 500 most informative character n-grams. Chi-Squared and Information Gain methods are employed to select the top features based on their relevance to the target variable. These selected features undergo Principal Component Analysis (PCA) to reduce dimensionality while retaining essential information. The PCA results are integrated into the original dataset, providing a compact representation of the character-based features. This comprehensive approach ensures the effective extraction and representation of character-level information.

*2) Term Frequency-Inverse Document Frequency (TF-IDF)*

To extract the features on word-level we used Term Frequency-Inverse Document Frequency (TF-IDF) a statistical measures to evaluate the importance of a word in a document relative to a collection of documents (corpus). And the measure combines two key metrics: term frequency and inverse document frequency.

*Term Frequency (TF)*

This measures how frequently a term (word) appears in a single poem. The idea is that the more a word appears in a poem, the more important it is to that poem. Term frequency can be calculated as:

$$\text{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

*Inverse Document Frequency (IDF)*

This measures how important a term is across the entire corpus. Some terms, like common stop words (e.g., "is", "and", "the"), appear in many documents and are therefore less informative. The inverse document frequency diminishes the weight of such common terms and is calculated as:

$$\text{IDF}(t,D) = \log\left(\frac{\text{Total number of documents } D}{\text{Number of documents containing term } t}\right)$$

where $D$ is the total number of documents in the corpus.

*TF-IDF*

The TF-IDF score is the product of TF and IDF and gives a measure that balances the frequency of a term in a poem with how unique the term is across the entire corpus. It is calculated as:

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D)$$

### C. The proposed method

For the classification task, different models were employed: FastText, Linear Support Vector Classification, Random Forest, and Logistic Regression.

The FastText model was chosen due to its efficiency in handling text data and its ability to generate high-quality word embeddings. FastText, developed by Facebook's AI Research (FAIR) lab, extends the Word2Vec model by representing each word as a bag of character n-grams, making it particularly effective for morphologically rich languages like Arabic. To prepare the data for FastText, we started by formatting the training and testing datasets according to FastText's requirements. The labels were prefixed with "label" to ensure compatibility. For training, we used the following parameters: a learning rate (lr) of 1.0, 25 epochs, and bi-grams (wordNgrams=2). These parameters balance training speed and model accuracy. The FastText model was then trained using the prepared training data file. This approach leverages FastText's strengths in dealing with the subtleties of the Arabic language, including its complex morphology and rich character structure, providing a powerful method for Arabic poetry classification.

We also employed the Linear Support Vector Classifier (LSVC) model, a variant of the widely used Support Vector Machine (SVM) algorithm. The LSVC model was chosen due to its ability to effectively handle high-dimensional data and its robustness to overfitting, making it well-suited for text classification problems. The architecture of the LSVC model consists of an input layer, a feature transformation layer, and a decision function layer. The input layer receives the preprocessed text data, typically in the form of numerical feature vectors. The feature transformation layer applies the linear kernel function to project the input data into a higher-dimensional space, where the classes become more linearly separable. Finally, the decision function layer computes the hyperplane(s) that best separates the classes, based on the transformed features. In this training process, the text data is first transformed into TF-IDF feature vectors using the TfidfVectorizer. This process converts the textual data into numerical representations based on the importance of each word within the documents. Specifically, the training data is vectorized by fitting and transforming it, while the test data is first preprocessed using a custom function to normalize the text before applying the same transformation. The C parameter, which controls the regularization strength of the Linear Support Vector Classifier (LSVC), is then explored over a predefined range of values: [0.01, 0.1, 1, 10, 100]. For each C value, an LSVC model is initialized and trained on the training TF-IDF vectors, and predictions are made on the test set.

Furthermore, we utilized the Random Forest classifier, an ensemble model that combines multiple decision trees, for the Arabic poem classification task. This approach is particularly effective for managing high-dimensional text data and capturing intricate patterns, making it a robust choice for this type of analysis. The ensemble nature and resistance to class imbalance further enhance its suitability. While the performance of the Random Forest model was not optimal for this specific task, it still outperformed Logistic Regression and showed competitive results compared to the LSVC model. Random Forest's proficiency in handling textual data, along with its potential for refinement through tuning and enhanced ensemble strategies, warrants additional exploration for Arabic poem classification. This version maintains the essence of the original text but omits specific performance metrics, focusing instead on a qualitative evaluation of the model's performance.

In addition, we utilized Logistic Regression, a commonly used statistical model for classification tasks. This model is particularly adept at modeling the probability that an input belongs to various classes, which makes it well-suited for multi-class challenges such as Arabic poem classification. We transformed the preprocessed Arabic poem texts into numerical TF-IDF feature vectors using the TfidfVectorizer from scikit-learn. Subsequently, a Logistic Regression model was initialized and trained using these TF-IDF vectors from the training data, and then employed to generate predictions on the test set. While the performance of Logistic Regression was not as high as the LSVC model, it still serves as a valuable baseline, underscoring the need to explore various models for Arabic poem classification. Its simplicity and interpretability make Logistic Regression an excellent choice for initial testing and comparison purposes.

## V. Result and Discussion

In this section, the performance of four different models—FastText, Linear Support Vector Classification (LSVC), Logistic Regression, and Random forest —are evaluated and discussed. These models were chosen to leverage their unique strengths in handling text classification tasks, particularly for the complex structure of Arabic poetry. Each model was trained and tested on the same dataset to ensure a fair comparison of their capabilities. The results are analyzed based on key metrics such as accuracy, precision, recall, and F1-score, providing insights into the effectiveness of each approach in classifying Arabic poetry.

The performance of the FastText model was evaluated on the test dataset consisting of 10,937 samples. The model demonstrated impressive results, achieving a precision, recall, and F1-score of 0.926, indicating a high level of accuracy and consistency in its predictions. The training process, depicted in Figure 1, shows a rapid increase in accuracy within the first few epochs, stabilizing above 90% accuracy after approximately five epochs. This indicates that the model quickly learns the patterns in the data and maintains strong performance throughout the training process.

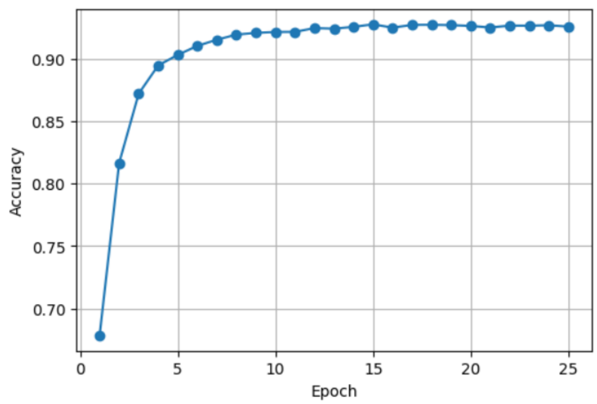


Fig. 1. Accuracy of FastText Model during Training

The performance of the Linear Support Vector Classification (LSVC) model was evaluated on the test dataset. The model achieved a precision of 0.576, recall of 0.576, and an F1-score of 0.549, indicating moderate effectiveness in its predictions. The training process, illustrated in Figure 2, shows the relationship between the regularization parameter $C$ and the accuracy of the model. The graph reveals that as the $C$ value increases, the accuracy initially improves, reaching 0.57 at $C = 10^0$ and maintaining this level of performance for higher $C$ values (up to $10^2$). This suggests that the model benefits up to a point, after which the accuracy stabilizes. Despite not achieving the highest performance metrics, the LSVC demonstrates consistent learning and reliable accuracy across different values of the regularization parameter.

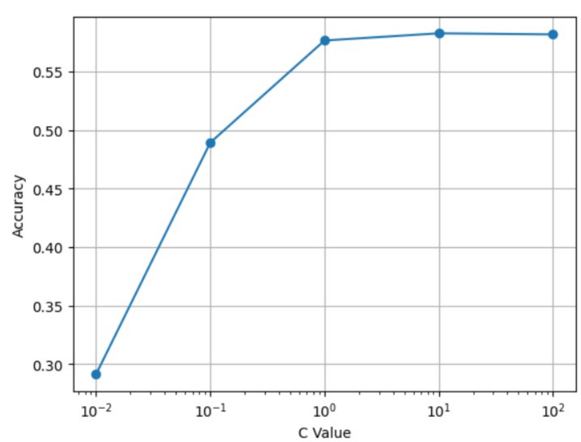


Fig. 2. Acuuracy of the LSVC model with different C values

The performance of the Logistics Regression Model (LR) model was evaluated on the test dataset. LR model achieved a precision of 0.494, a recall of 0.458, and an F1-score of 0.407, indicating low accurate classification.

Lastly, the Random Forest model. After testing the performance of the model in the testing set he model achieved a precision of 0.34, a recall of 0.31, and an F1-score of 0.26, which represents the worst performance in classification among all the models.

Among the models evaluated for Arabic poem classification, the Fast Text Model exhibited superior performance across all key metrics: precision, recall, and F1-score, each at an impressive 0.92566. This model demonstrated rapid convergence to optimal performance levels, as indicated by the accuracy graph that peaks sharply at around 0.90 within the initial few epochs and remains stable thereafter. In contrast, the Random Forest Model showed notably lower efficacy, with a precision of 0.34, recall of 0.31, and an F1-score of 0.26, reflecting its considerable struggles with the complexity of Arabic poetry classification. The Logistics Regression Model and the LSVC Model also showed limited effectiveness, with precision scores of 0.4941 and 0.57625, respectively. The LSVC's performance saw some improvement as the regularization strength was adjusted, yet it plateaued around 0.55, which suggests diminishing returns on accuracy with higher complexity in the model settings. This analysis starkly highlights the Fast Text Model's robustness and efficiency in handling high-dimensional text data, making it the standout choice for this specific classification task.

The Fast Text Model excelled in the Arabic poem classification task, outperforming other models due to its superior handling of the unique challenges presented by the Arabic language. It achieved impressively high scores across precision, recall, and F1-score, all at 0.92566, indicating strong accuracy and comprehensive coverage of the dataset without overfitting. Its effectiveness is further underscored by its rapid convergence to peak performance, which stabilized shortly after the training began, demonstrating its efficiency in reaching and maintaining an optimal solution. Fast Text's adeptness at processing text through word and subword embeddings allows it to capture nuanced semantic relationships and contextual usages that are essential in Arabic, a language characterized by rich morphology. This capability, combined with its robust generalization across varying inputs, makes the Fast Text Model particularly suited for managing the linguistic complexities of Arabic poetry, ensuring it delivers consistent and reliable classification results. Taable II represents a summary of all models' results.

| Model | Fast text | LSVC | Random Forest | LR |
|---|---|---|---|---|
| Precision | 0.92566 | 0.57625 | 0.34 | 0.4941 |
| Recall | 0.92566 | 0.57636 | 0.31 | 0.4583 |
| F1-score | 0.92566 | 0.54944 | 0.26 | 0.4075 |

TABLE II
MODEL EVALUATION METRICS

## VI. Conclusion and Future work

In this paper, we provide an overview of research on the classification of Arabic poetry using machine learning algorithms. We discuss the various approaches researchers have taken to categorize poems based on themes, eras, and authors. Our analysis demonstrates the importance of preprocessing techniques, feature extraction methods, and model selection in achieving precise classification results. Additionally, we discuss the challenges posed by the linguistic complexity of Arabic poetry and the scarcity of annotated datasets.

We highlight the potential of machine learning in enhancing the understanding of Arabic literary texts and emphasize the need for further research to refine classification techniques and expand dataset sizes. Our exploration sheds light on the advancements made in Arabic text analysis and sets the stage for future endeavors to improve the robustness and reliability of poet classification systems.

In future research, we believe that addressing the challenge of predicting the poet as a classification problem can benefit from incorporating additional poetry features beyond the poem verse itself. Specifically, categorical attributes such as poem meter, poem theme, plot era, and other stylistic elements could significantly aid in identifying the poet.

Additionally, we propose extending this investigation to process a larger dataset encompassing works from over 350 poets, or by integrating multiple datasets to enhance the robustness of our findings. Furthermore, we recommend expanding the scope of our study to include a wider variety of classification algorithms and conducting a comparative analysis of their performance.

## Paper Dataset and Code

you can access paper code and dataset by visiting the project page in GitHub here.

## References


[1] Cushman, S., Cavanagh, C., Ramazani, J., & Rouzer, P. (2012). The Princeton Encyclopedia of Poetry and Poetics. Princeton University Press.

[2] Ahmed, M. A., Hasan, R. A., Ali, A. H., & Mohammed, M. A. (2019). The classification of the modern arabic poetry using machine learning. TELKOMNIKA (Telecommunication Computing Electronics and Control), 17(5), 2667-2674.

[3] Orabi, M., El Rifai, H., & Elnagar, A. (2020, November). Classical Arabic Poetry: Classification based on Era. In 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-6). IEEE.

[4] El-Halees, A. M. (2022, January). Arabic Poetry Authorship Attribution and Verification Using Transfer Learning. Egyptian Computer Science Journal, 46(1), ISSN 1110-2586.

[5] Alanazi, S. (2018). Classical Arabic Authorship Attribution Using Simple Features. College of Computer and Information Sciences, Jouf University, Saudi Arabia.

[6] Abandah, G. A., Khedher, M. Z., Abdel-Majeed, M. R., Mansour, H. M., Hulliel, S. F., & Bisharat, L. M. (2022). Classifying and diacritizing Arabic poems using deep recurrent neural networks. *Journal of King Saud University. Computer and information sciences*, 34(6), 3775–3788. https://doi.org/10.1016/j.jksuci.2020.12.002

[7] Discovering the applicability of classification algorithms with Arabic poetry. (2019, April 1). IEEE Conference Publication — IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8717387

[8] AlShaibani, M. S., Alyafeai, Z., & Ahmad, I. (2020). Meter classification of Arabic poems using deep bidirectional recurrent neural networks. Pattern Recognition Letters, 136, 17. https://doi.org/10.1016/j.patrec.2020.05.028

[9] Abbas, M., Lichouri, M., & Zeggada, A. (2019). Classification of Arabic Poems: from the 5th to the 15th Century. In Lecture notes in computer science (pp. 179–186). https://doi.org/10.1007/978-3-030-30754-7_18

[10] Ghneim, N. (2013, March 15). Emotion Classification in Arabic Poetry using Machine Learning. Aiu-sy.

[11] Shahriar, S., Roken, N. A., Zualkernan, I. (2023, April 22). Classification of Arabic Poetry Emotions Using Deep Learning. Computers. https://doi.org/10.3390/computers12050089

[12] Iqbal AbdulBaki Mohammad, "Naive Bayes for Classical Arabic Poetry Classification," Journal of Al-Nahrain University, vol. 12, no. 4, pp. 217-225, December 2009.

[13] Munef Abdullah Ahmed, Raed Abdulkareem Hasan, Ahmed Hussein Ali, Mostafa Abdulghafoor Mohammed, "The Classification of Modern Arabic Poetry Using Machine Learning," TELKOMNIKA, vol.17, no.5, pp.2667-2674, October 2019. DOI: 10.12928/TELKOMNIKA.v17i5.12646

[14] Al-Falahi Ahmed, Ramdani Mohamed, and BellafkihMostafa, "Arabic Poetry Authorship Attribution using Machine Learning Techniques", Journal of Computer Science, vol. 15, no. 7, pp. 1012-1021, July 2019.

[15] Durmus Ozkan Sahin, Oguz Emre Kural, Erdal Kilic, and Armagan Karabina, "A Text Classification Application: Poet Detection from Poetry", Ondokuz Mayıs University, October 2018.

[16] Muaad, A. Y., Davanagere, H. J., Benifa, J. B., Alabrah, A., Saif, M. A. N., Pushpa, D., ... & Alfakih, T. M. (2022). Artificial intelligence-based approach for misogyny and sarcasm detection from Arabic texts. Computational Intelligence and Neuroscience, 2022.

[17] Sahin, D. O., Kural, O. E., Kilic, E., & Karabina, A. (2018). A text classification application: Poet detection from poetry. arXiv preprint arXiv:1810.11414.