College of Computer and Information Sciences

King Saud University
College of Computer and Information Sciences
Computer Science Department

Arabic Poetry Generation using Large Language Models

CSC 496 – Final Report

Prepared by:

Msaad Alameel	443102435
Mohammed Alageel	443101692
Mohammed Aldawsari	443100998
Hatim Alhomid	443102109

Supervised by:
Prof. Mohamed Maher Ben Ismail

Research project for the degree of Bachelor's in Computer Science

Autumn 2024

Acknowledgements

We would like to express our sincere thanks to King Saud University faculty members for all their efforts to provide us with education and skills throughout our academic journey, especially Prof. Mohamed Maher Ben Ismail.

English Abstract

Arabic poetry represents an important art of the Arab culture. It is appreciated for its beauty, emotion, and unique styles. On the other hand, the recent advances in Artificial Intelligence, Machine Learning (ML) and Natural Language Processing (NLP) promoted researchers' efforts to exploit cutting edge capabilities to support Arabia poetry community. In particular, Generating Arabic poetry using Natural Language Processing (NLP) has attracted researchers' interest to address relevant challenges. Specifically, generating realistic Arabic poetry that exhibits the right rhythm, correct rhyme that conveys a deep meaning remains a challenging task for the involved community. One should note that complex language, large vocabulary, and special poetic forms make the problem even more acute for machines to generate pseudo-authentic poetry. Besides, Large Language Models (LLMs) have emerged as a powerful tool with promising capabilities to overcome such challenges. By leveraging vast amounts of linguistic data and advanced architectures, LLMs have made big steps in managing the difficult specificities of Arabic, such as capturing the flow, meaning, and style of the poems. While these models are improving, creating high-quality Arabic poetry with technology remains a tough challenge.

In this project, we plan to investigate cutting-edge LLMs ability to generate realistic Arabic poetry. Namely, recently released models such as Llama, Jais and Mistral will be explored. Specifically, different architectures that are designed to embed the considered models into the proposed solution. Moreover, we intend to collect the required data and pre-process it to validate and assess the performance of the models. Furthermore, suitable performance measures will be used to assess the results obtained using the implemented models.

Arabic Abstract

يمثل الشعر العربي فنًا مهمًا في الثقافة العربية، حيث يُقدر لجماله وعاطفته وأساليبه الفريدة. من ناحية أخرى، أدت التطورات الأخيرة في مجالات الذكاء الاصطناعي وتعلم الآلة ومعالجة اللغات الطبيعية إلى تعزيز جهود الباحثين لاستغلال القدرات المتطورة لدعم مجتمع الشعر العربي. وعلى وجه الخصوص، جذب توليد الشعر العربي باستخدام معالجة اللغات الطبيعية اهتمام الباحثين لمواجهة التحديات ذات الصلة. وبشكل محدد، لا يزال توليد الشعر العربي الواقعي الذي يتميز بإيقاع صحيح وقافية سليمة ونقل المعنى العميق مهمة صعبة للمجتمع العربي. وتجدر الإشارة إلى أن تعقيد اللغة والمفردات الواسعة والأشكال الشعرية الخاصة تجعل المشكلة أكثر حدة بالنسبة للآلات توليد الشعر الشبه الأصيل. إلى جانب ذلك، ظهرت نماذج اللغة الكبيرة كأداة قوية ذات قدرات واعدة للتغلب على مثل هذه التحديات. من خلال الاستفادة من كميات هائلة من البيانات اللغوية والنماذج المتقدمة، حققت نماذج اللغة الكبيرة خطوات كبيرة في إدارة الخصائص الصعبة للغة العربية، مثل التقاط أبحر وقوافي القصائد ومعانيها وأساليبها. وبينما تتحسن هذه النماذج، لا يزال إنشاء الشعر العربي عالي الجودة باستخدام التكنولوجيا تحديًا صعبًا.

في هذا المشروع، نخطط لاستكشاف قدرة نماذج اللغة الكبيرة المتطورة على توليد الشعر العربي الواقعي. وتحديدًا سيتم استكشاف النماذج التي تم إطلاقها مؤخرًا مثل Llama ، Jaisو Mistral على وجه التحديد، سيتم دراسة هندسات مختلفة مصممة لدمج النماذج المعتبرة في الحل المقترح. علاوة على ذلك، نعتزم جمع البيانات المطلوبة ومعالجتها مسبقًا للتحقق من أداء النماذج وتقييمها. وفضلاً عن ذلك، سيتم استخدام مقاييس أداء مناسبة لتقييم النتائج التي تم الحصول عليها باستخدام النماذج المنفذة.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Arabic poetry has been deeply intertwined with the linguistic and cultural identity of the Arab world for centuries. The most recognized form of traditional Arabic poetry, "Buhur", was established by Al-Farahidi in the 8th century [1]. He created a system of 15 poetic meters, and later expanded it to 16, which became the foundation of classical Arabic poetry. These meters are governed by a set of rules known as "Arud". They structure the rhythmic patterns of the verses. Specifically, each verse adheres to a repetitive pattern called "Taf'eelat", and the rhyme, or "Qafiyah", is crucial in shaping the final structure of the poem [2].

In Saudi Arabia, poetry holds immense cultural significance, particularly in the oral tradition of "Nabati" poetry, which connects modern Saudis to their Bedouin ancestors. This tradition is not only a form of artistic expression but also a tool for preserving historical narratives and social values. Poetry's role as a cultural cornerstone is reflected in how it is still practiced and revered today [3] [4].

Moreover, the Saudi government has recognized the importance of poetry in preserving the nation's cultural heritage. Through initiatives such as poetry competitions and festivals organized by the Ministry of Culture, and its inclusion in the Saudi Vision 2030, poetry continues to thrive as a means of both cultural preservation and modern expression [5]. Despite this promotion Arabic poetry, replicating the structured nature of Arabic poetry using machine learning techniques remains a challenging task due to its complex rules governing rhythm, rhyme, and diacritics [6].

1.1 Problem Statement

The intricate structure of Arabic poetry, including its reliance on Buhur, Arud, and Qafiyah, presents a unique challenge for Artificial Intelligence (AI). While human poets intuitively follow these rules, state-of-the-art Large Language Models (LLMs) struggle to replicate this process. In fact, the difficulty lies in teaching AI to adhere to these traditional poetic forms while also ensuring the generated poetry maintains the linguistic beauty and cultural authenticity of classical Arabic poetry [7].

1.2 Goals and Objectives

This project aims to fine-tune LLMs such as Llama-3.1, Jais [8], and Allam 13B [9] using relevant datasets like Ashaar and Aldiwan [10] which contain extensive examples of classical Arabic poetry. Specifically, this research objective consists in developing models that can generate poetry in accordance with traditional rules of Buhur, Arud, and Qafiyah. Moreover, the intended models would maintain the linguistic and cultural depth of the generated poetry.

Accordingly, the ultimate goal of this research consists in bridging the gap between traditional poetry and modern AI, contributing to the preservation of Arabic poetry by promoting its accessibility through modern technologies.

1.3 Research Scope

The scope of this research involves applying AI and LLMs to generate structured Arabic poetry. In particular, the considered process will include:

- Data preprocessing: Preparing the Ashaar and Aldiwan datasets for training the AI models.
- Fine-tuning models: Training Llama-3.1, Jais, and Allam 13B on these datasets to generate poetry that adheres to traditional poetic forms.
- Model evaluation: Assessing the accuracy of the models in following the rules of Arabic poetry while maintaining linguistic and cultural authenticity.

Chapter 2: Background

This chapter is divided into two main parts: First, we provide a background about Arabic Poetry. In particular, we highlight how important it is in Saudi Arabia and the Arabic world. Then, the second part outlines the technical background relevant to this project. Specifically, it depicts the concept of machine learning as the intelligent module of the proposed solution. More specifically, supervised and unsupervised machine learning paradigms are introduced in this chapter.

2.1 Arabic Poetry

Arabic poetry has many definitions that have changed over time. In the past, it was defined as “structured speech, honored by rhyme and meter, even though all knowledge could be considered poetry” (Ibn Manzur: Lisan al-Arab). It was also described as *"a composition that must have a coherent structure, with rhyme and meter, and must be intended to be such. If it lacks any of these, it is not called poetry, and its author is not called a poet."* Therefore, verses from the Quran or Hadith that are metrically structured are not considered poetry, as they were not intended to follow rhyme and meter. Likewise, what people say unintentionally is not poetry, since the term poetry comes from the word sha'rt meaning to perceive or understand. Thus, if it is not intended, it is as if the person did not perceive it (Al-Fayoumi). According to this view, poetry requires four essential elements: meaning, meter, rhyme, and intention [2].

وَمَنْ لَمْ يَذُقْ مُرَّ التَّعَلُّمِ سَاعَةً تَجَرَّعَ ذُلَّ الْجَهْلِ طُولَ حَيَاتِهِ

وَمَنْ فَاتَهُ التَّعْلِيمُ وَقْتَ شَبَابِهِ فَكَبِّرْ عَلَيْهِ أَرْبَعًا لِوَفَاتِهِ

Figure 1: Sample Arabic poetry [11]

2.1.1 Importance of poetry in Saudi Arabia

Arabic poetry holds significant cultural, historical, and social importance in Saudi Arabia. It plays a central role in the country's identity, as well as in the broader Arab world. In

recognition of the deep-rooted tradition of poetry in the Arab world, Saudi Arabia declared 2023 as the "Year of Arabic Poetry." This initiative aimed to celebrate, promote, and preserve Arabic poetry by holding events, festivals, and workshops that highlight both classical and modern poetic forms. It was also intended to encourage new generations to engage with this art form and appreciate its value in Arab culture [12].

Poetry competitions have long been a tradition in the Arab world, fostering creativity and cultural pride. In the United Arab Emirates, renowned competitions like Prince of Poets [13] and Million's Poet [14] provide platforms for poets from across the Arab world to showcase their skills in front of a wide audience. These events celebrate not only poetic talent but also the preservation of Arabic heritage, encouraging poets to explore themes of identity, culture, and social issues. While these competitions are held in the UAE, their influence and importance resonate throughout the entire Arab region, including Saudi Arabia, where poetry remains a vital part of cultural expression.

2.1.2 Vertical Poetry and Free Verse Poetry

Vertical poetry (Al-shi'r al-'amudi) is the traditional form of Arabic poetry. It follows strict rules of meter and rhyme, with each poem made up of verses that have two parts: the “sadr” (first half) and the “ajz” (second half). This form of poetry is based on a system created by Al-Khalil bin Ahmad Al-Farahidi, called Ilm al-'arud. This system ensures that the poetry is rhythmic and balanced, making it pleasant to listen to. Vertical poetry is deeply connected to Arabic heritage and depends on key elements such as language, rhyme, meaning, and rhythm [15].

Example of Vertical Poetry:

قِفَا نَبْكِ مِنْ ذِكْرَى حَبِيبٍ وَمَنْزِلِ بِسِقْطِ اللِّوَى بَيْنَ الدَّخُولِ فَحَوْمَلِ

فَتُوْضِحَ فَالمِقْرَاةِ لم يَعْفُ رَسْمُهَا لِمَا نَسَجَتْهَا مِنْ جَنُوبٍ وَشَمْأَلِ

Figure 2: An example of Vertical Poetry [16]

In contrast, free verse poetry (Al-shi'r al-hurr) is a more modern form of poetry. It became popular around the world in the late 19th century and in the Arab world during the 1930s. Free verse does not follow the strict rules of traditional poetry, giving poets more freedom

with how they structure their poems. While it still maintains rhythm, it doesn't stick to a fixed pattern of meter. Poets like Nazik Al-Malaika argue that free verse has some rhythmic patterns, while others, like Jabra Ibrahim Jabra, believe it is completely free from traditional forms [17].

يا تونس الخضراء .. جئتك عاشقاً
وعلى جبيني وردة وكتاب

إني الدمشقي الذي احترف الهوى
فاخضوضرت بغنائه الأعشاب

Figure 3: An example of Free Verse Poetry [17]

2.1.3 Meter

In Arabic poetry, the meter (Al-Bahr) refers to the poetic meter or rhythm that governs the structure of a poem. It is the system that organizes verses into rhythmic patterns. Al-Khalil bin Ahmad Al-Farahidi developed 15 poetic meters, known as "Buhoor" (plural of Bahr), while his student added a 16th meter. Each Bahr is divided into two halves, and it determines the rhythm and flow of the poem [18].

2.1.4 Taf'eela

Al-Taf'eela refers to the individual rhythmic units or "feet" that make up the meter of a poem. There are ten key patterns used to measure the syllables in a verse, such as Fa'oolun and Mustaf'ilun. These patterns help maintain balance between long and short syllables in the poem, creating a structured rhythm [19].

فَعُولُنْ مَفَاعِيلُنْ فَعُولُنْ مَفَاعِيلُنْ فَعُولُنْ مَفَاعِيلُنْ فَعُولُنْ مَفَاعِيلُنْ
ستبدي لك الأيام ما كنت جاهلاً ويأتيك بالأخبار من لم تزود

Figure 4: Taf'eela of the Taweel Bahr [20]

2.1.5 Wazn

Al-Wazn, or weight, is the overall measure of the poem's meter. It defines how the syllables are arranged and is closely linked to the Taf'eela. Essentially, it is the framework that ensures the poem adheres to a specific rhythm, making the poem pleasing to the ear and keeping its structure intact [21].

2.1.6 Rhyme

In Arabic poetry, the rhyme (Al-Qafiyah) refers to the ending part of a verse. It starts from the last vowel sound in the line and includes the last syllable or group of sounds. The rhyme is a crucial element, providing the poem with a musical flow and helping to unify the poem's structure.

There are two main types of rhyme:

- Muqayyad: Restricted rhyme, where the rhyme ends with a consonant.
- Mutlaq: Unrestricted rhyme, where the rhyme ends with a vowel.

Rhyme plays a key role in giving rhythm and harmony to Arabic poetry, working hand-in-hand with the poem's meter (Al-Wazn). Because of its importance, ancient poets gave as much attention to rhyme as they did to meter, both forming the musical backbone of the poem. Each poem ends with the same rhyming sound, making the rhyme a defining feature of the verse and often used to represent the poem itself [22].

قسما برفعة ذي العلا عباس حلى سمى الفضل خير الناس

لم يحل لي مدح بغير صفاته فبغيرها مدحي بغير اساس

الله انسان شريف أصله وخصاله للناس كالنبراس

مولى تفرد بالسماحة والحجا وبأعظم الاخلاق والاحساس

وافيته بالعيد اهدى ماصفا من خير تهنئة صفاء الماس

Figure 5: A sample Qafiyah with the (Seen) letter [11]

2.2 Artificial Intelligence

Let's dive deeper into the details of how Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP) works, and how these techniques can be applied to generate Arabic poetry, considering aspects like neural networks, transformer architecture, and training methods.

At its core, AI involves the simulation of human-like intelligence in machines, allowing them to perform tasks that typically require human cognition, such as learning, reasoning, problem-solving, and even creativity. AI systems can be rule-based, data-driven, or both [23].

Rule-based AI: Early AI systems relied on hard-coded rules to make decisions. For example, you could develop a simple AI system to generate poetry by explicitly coding the rules of meter, rhyme, and diacritical placement. However, this approach becomes rigid and inefficient, especially when dealing with complex tasks like natural language generation.

Data-driven AI: Modern AI systems rely heavily on large amounts of data and statistical techniques to learn patterns from the data. AI is learning from vast datasets to generate new verses that adhere to traditional rules [23].

AI works through a combination of algorithms (step-by-step procedures to solve a problem) and models (mathematical representations of the problem). The more data the AI system has access to, the better it can generalize and improve over time [23].

2.2.1 Machine Learning

Machine Learning (ML) is a subset of AI focused on building systems that learn from data. Instead of manually coding every rule, ML models are trained on data from past examples, allowing them to recognize patterns and make predictions or decisions based on new data. In the following we will showcase the key steps involved in how ML works [24].

- Data Collection: The first step is to gather a large dataset. In our case, the dataset consists of Arabic poems (such as those in the Ashaar and Aldiwan datasets), annotated with details like meter, rhyme, and diacritical patterns. These structured forms provide the foundation for learning.

-Feature Extraction: Features are the important pieces of information that the model will learn from. In Arabic poetry generation, features might include word sequences (important for rhyme and meter), diacritics (important for rhythm and pronunciation) and patterns of line breaks and dominant meter, and the rhyme scheme.

-Model Training: During training, the ML model learns the relationship between input features (e.g., sequences of words in Arabic) and the output (e.g., new lines of poetry that conform to meter and rhyme). This is done by optimizing a loss function, which measures how far the model's predictions deviate from the desired output.

-Supervised Learning: A supervised learning approach is used where the model is fed labeled data (I.e., lines of poetry annotated with rhyme, meter, and diacritics). The model learns the patterns that adhere to Arabic poetry rules.

-Optimization: The model continuously adjusts its internal parameters using algorithms like gradient descent, which minimizes the error between the predicted output and the actual output.

-Testing and Validation: After training, the model is tested on unseen data to check its accuracy. In poetry generation, this would involve assessing whether the AI-generated verses adhere to traditional Arabic poetic forms in terms of both structure and meaning. [23]

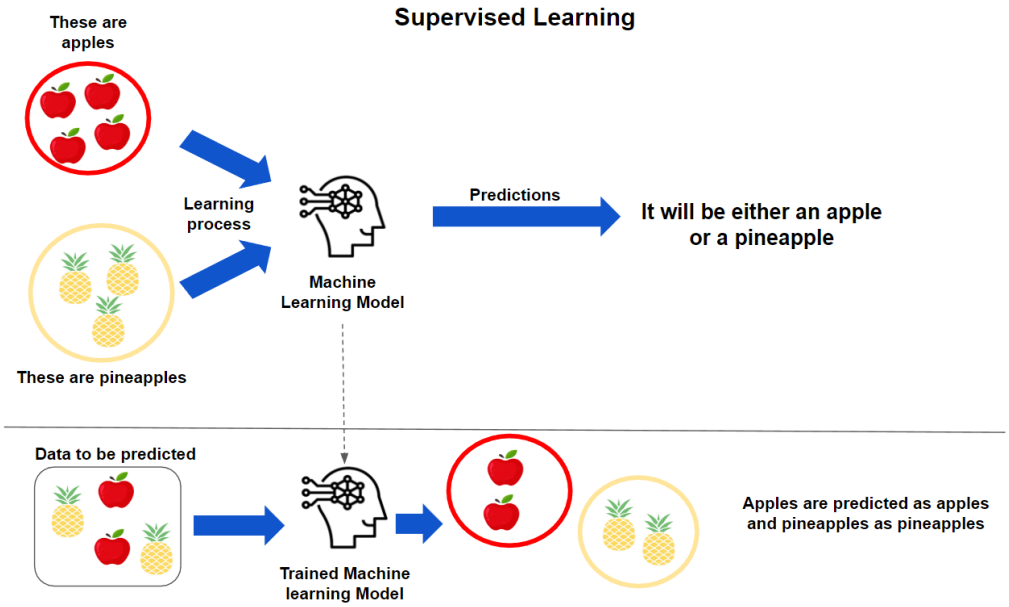

Figure 6: A typical supervised learning [24]

2.2.2 Deep Learning

Deep Learning (DL) is a branch of ML that uses neural networks with many layers (hence the term “deep”) to learn from data. It’s particularly useful for complex tasks like natural language generation because it can model intricate patterns and dependencies in data.

a. Neural Networks

A neural network consists of interconnected layers of nodes (neurons), which transform the input data into an output. Here's how it works [24].

Input Layer: Receives the input data, in our case it is sequences of words or characters in Arabic poetry.

Hidden Layers: These layers perform the complex transformations needed to make sense of the input data. Each hidden layer learns abstract features from the data. The first few layers might learn simple features, such as individual letters and diacritical marks. Deeper layers might learn higher-order patterns, such as rhyme schemes and meter structures.

Activation Functions: At each layer, activation functions decide which neurons should be activated, allowing the model to introduce non-linearity. This is important because poetic structures are highly non-linear (e.g., maintaining rhyme and rhythm across lines).

Output Layer: This layer produces the final prediction, such as the next word in a poetic verse. The output could also include suggestions for diacritical marks, ensuring that the generated poetry is metrically correct.

b. Backpropagation

The model is trained using a process called backpropagation, where the error (difference between the predicted output and the actual output) is propagated backward through the network. The model's parameters (weights) are adjusted at each layer to minimize this error. Over time, the model becomes better at predicting the output as shown in Figure 7.

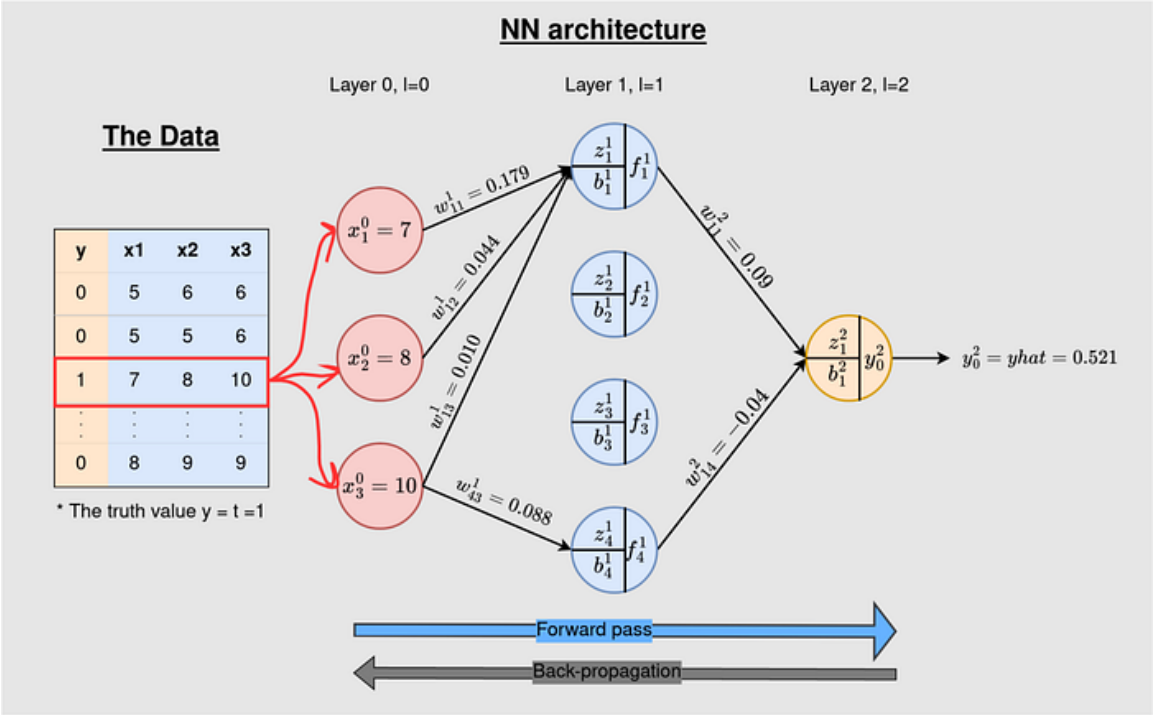

Figure 7: Typical neural network architecture [25]

2.2.3 Recurrent Neural Networks

In poetry generation, understanding the sequence of words and how they relate to each other is essential. Recurrent Neural Networks (RNNs) and their variants, like LSTMs (Long Short-Term Memory) and GRU's (Gated Recurrent Units), are well-suited for this task because they maintain a "memory" of previous inputs, allowing the model to generate coherent, sequential poetry. In fact, Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to effectively handle long-range dependencies in sequential data. They feature three gates (input, forget, and output) that manage the flow of information, allowing LSTMs to retain important contextual information while mitigating issues like the vanishing gradient problem.

Gated Recurrent Units (GRU) are a simpler alternative to LSTMs, combining the input and forget gates into a single update gate and eliminating the separate cell state. GRU's consist of two main components: the update gate, which controls how past and new information are combined, and the reset gate, which determines how much past information to forget.

Both LSTMs and GRU's are highly effective for tasks involving sequential data in Figure 8, such as natural language processing and speech recognition, with LSTMs being more complex and GRU's providing a more computationally efficient option [24].

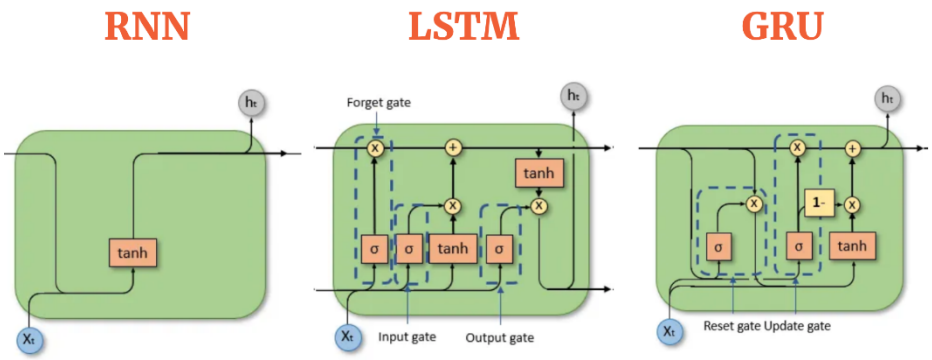

Figure 8: Main difference between RNN, LSTM, GRU [26]

2.2.4 Natural Language Processing

Natural Language Processing (NLP) can be introduced as a branch of AI that enables machines to understand, interpret, and generate human language. It sits at the intersection of linguistics, computer science, and AI, and is fundamental for tasks such as text generation, language translation, and sentiment analysis. In our case, NLP allows models to process and generate text that adheres to linguistic and poetic rules. In the following, we outline the key components of NLP [24].

-Tokenization: Splitting text into individual words, phrases, or even characters. In Arabic poetry generation, tokenization involves handling both the surface form of words and the associated diacritics, which are critical for maintaining the meter.

-Part-of-Speech (POS) Tagging: Assigning grammatical categories (like noun, verb, adjective) to each token. This helps ensure the correct syntactical structure is maintained.

-Named Entity Recognition (NER): Identifying proper nouns, such as names of places or people. And it can help the model insert culturally relevant terms or themes in a structured way.

-Dependency Parsing: Understanding the grammatical structure of a sentence by identifying relationships between words. This helps ensure that the generated poetry is grammatically correct and maintains the necessary syntactical relationships.

-Semantic Analysis: Capturing the meaning of words and phrases to ensure that the generated text not only follows poetic rules but also conveys meaningful and contextually appropriate content.

a. Large Language Models

A Large Language Model (LLM) is a type of artificial intelligence (AI) model designed to understand and generate human language. These models are trained on vast amounts of text data, allowing them to grasp the complexities of language such as grammar, syntax, and meaning while being capable of tasks like text generation, translation, summarization, and question answering. LLMs like Llama-3.1 [27] [28], Jais and Allam 13B are trained to generate Arabic poetry by learning from large datasets, such as Ashaar and Aldiwan, that represent the structure of traditional poetry. These models are incredibly powerful because they have billions of parameters (numerical values that represent learned information), which allow them to generate coherent and contextually appropriate language

LLMs are typically pre-trained using massive datasets, usually containing billions of words, sentences, and documents. This allows them to learn general language patterns, such as grammar, syntax, and even subtle cultural nuances. During pre-training, the model learns to predict the next word in a sentence by optimizing a loss function that measures the difference between its predictions and the actual next word. After pre-training, the LLM can be fine-tuned on a more specific dataset. In our case, the model is fine-tuned on the Ashaar and Aldiwan datasets, which consist of traditional Arabic poetry. The model learns the specific poetic rules, such as:

- Meter: Ensuring that the generated lines of poetry follow established rhythmic patterns.
- Rhyme: Learning to generate lines that end with consistent rhyming sounds (Qafiyah).
- Diacritics: Understanding and applying the correct Diacritics (Tashkeel), which is essential for both meaning and rhythm.

b. Transformer Architecture in LLMs

Transformers are a type of neural network architecture that has revolutionized the field of natural language processing (NLP) and other sequential data tasks. Introduced in the paper "Attention is All You Need" by Vaswani in 2017 [28], Transformers use a mechanism called self-attention to process input sequences in parallel, allowing for more efficient training and better handling of long-range dependencies compared to previous models like RNNs. Self-attention enables the model to weigh the importance of different words in a sequence relative to each other. For each word, it computes attention scores for all other words, allowing the model to focus on relevant context. This capability helps the model understand relationships and dependencies within the input data [28].

Besides, instead of computing a single set of attention scores, Transformers use multiple attention heads to capture different types of relationships in the data. Each head learns to focus on different parts of the input, enriching the model's understanding of the context [28].

Unlike RNNs, Transformers do not have an inherent sense of order since they process input tokens simultaneously. To retain sequential information, Transformers use positional encodings, which are added to the input embeddings to provide information about the position of each word in the sequence [28]. After the attention layers, the output is passed through feedforward neural networks that apply non-linear transformations. This helps the model learn complex representations of the input data [28].

The transformers incorporate layer normalization and residual connections to stabilize training and improve convergence. Residual connections allow gradients to flow more easily through the network, enhancing performance [28].

Accordingly, the original transformer architecture consists of an encoder and a decoder. The encoder processes the input sequence and generates representations, while the decoder uses these representations to produce the output sequence, making it suitable for tasks like translation. However, many modern implementations, such as BERT [29] and GPT [30], focus on just the encoder or decoder.

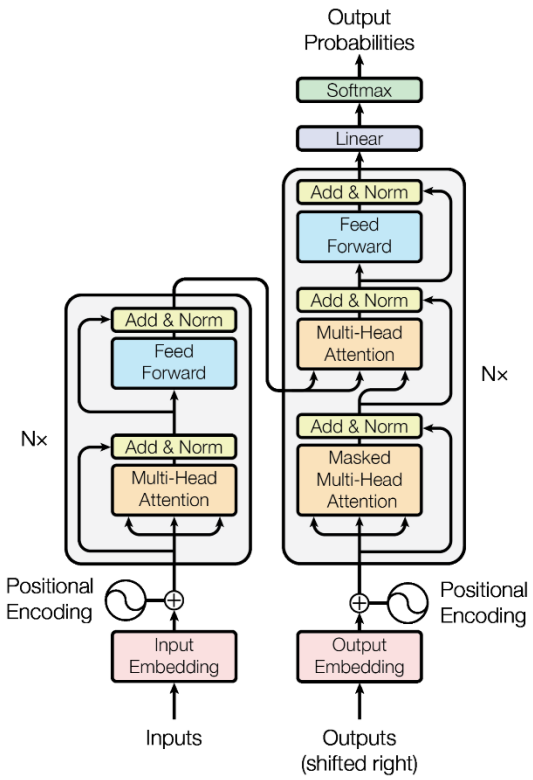

Figure 9: Typical transformer Architecture [28]

In summary, the transformers represent a significant advancement in deep learning architecture, enabling efficient and effective processing of sequential data and setting new benchmarks in various AI applications like generating poetry [28].

c. Data Preprocessing

For NLP models to effectively generate structured Arabic poetry, proper data preprocessing is essential. Arabic presents unique challenges due to its rich morphology, diacritical marks, and complex poetic rules. Preprocessing steps include:

Tokenization: Breaking down the text into manageable units (words or sub-words). Tokenization in Arabic is challenging because of the intricate morphology, but modern tokenization techniques (e.g., sub-word tokenization) allow for a more granular understanding of language structure.

Normalization: Handling diacritical marks (tashkeel), which are essential for maintaining the meter of Arabic poetry.

Phonetic Embeddings: For generating poetry with a focus on rhyme and meter, it's useful to represent words based on their phonetic properties. This helps the model generate lines that sound poetic while maintaining structural integrity.

By combining the power of AI, ML, DL, and advanced models like transformers and LLMs, we can build a system capable of generating Arabic poetry that respects the intricate rules of meter, rhyme, and diacritics, blending computational techniques with centuries-old literary tradition.

Chapter 3: Literature Review

This chapter delves into the exciting advancements in using artificial intelligence (AI) for creating Arabic poetry. It highlights key systems, which leverage deep learning to craft verses that respect traditional poetic forms while capturing the essence of Arabic culture. The chapter also discusses how AI-generated poetry is evaluated based on criteria such as rhythm, rhyme, and emotional depth. Furthermore, it examines the role of transformer models, such as AraGPT2, in enhancing the fluidity and richness of poetic expression.

Arabic poetry generation using artificial intelligence has advanced rapidly in recent years, with several prominent systems emerging to address the complexities of classical Arabic verse. One of the most significant efforts is Ashaar (2023) [10], which integrates deep learning techniques to analyze and generate Arabic poetry. Ashaar system focuses on multiple dimensions, including meter classification, theme analysis, and era recognition, while also offering automatic diacritization—a feature crucial for ensuring the generated poetry is metrically accurate. By leveraging a GPT-based architecture, Ashaar can generate poetry conditioned on specific themes like love (ghazal) or satire (heja), and it can adhere to traditional meters such as Buhur, a critical aspect of classical Arabic poetry [10] [31].

In addition, Ashaar goes beyond merely generating poetry; it offers a comprehensive framework for analyzing and generating poetry across several dimensions. It includes automatic diacritization and can extract the Arudi style, a vital component of classical Arabic meter. The system uses pre-trained models and datasets specifically designed for Arabic poetry, which include datasets for poetry generation, diacritization, and Arudi-style prediction. This system significantly enhances the quality of generated Arabic poetry by incorporating cultural and linguistic aspects that are crucial for its authenticity. Moreover, Ashaar's ability to generate poetry in multiple classical meters and themes has been a key development in improving the fluency and depth of AI-generated Arabic poetry [10] [31].

Ashaar also introduces an innovative method for recognizing and preserving the historical and cultural contexts of classical Arabic poetry. By analyzing the era in which a poem is set, Ashaar ensures that the generated verses reflect the stylistic nuances typical of different historical periods in Arabic literary history. This capacity for era-based generation makes Ashaar a powerful tool for researchers, poets, and educators looking to study or recreate

the rich literary traditions of the Arab world. Furthermore, Ashaar's advanced diacritization mechanism, which assigns diacritical marks to words, is instrumental in improving the pronunciation and rhythm of generated poetry, aligning it with traditional metrical patterns and enhancing the overall coherence of the poems [31] [10][14].

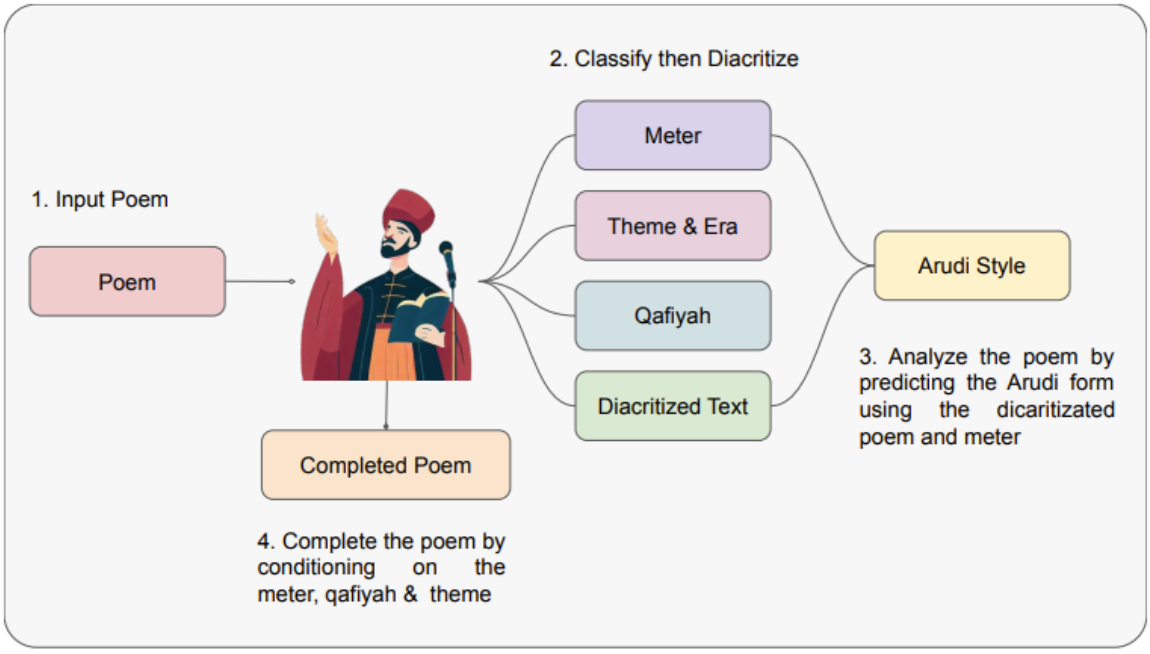

Figure 10: Flowchart of Ashaar poetry generation [10]

In terms of performance, Ashaar has demonstrated significant improvements over previous models. Its use of large-scale, specialized datasets allows it to generate fluent and culturally accurate poetry. The system excels at maintaining metrical accuracy and rhyme consistency, outperforming traditional models in these areas. The pre-trained models also allow Ashaar to handle the generation of poetry with specific themes and eras, making it highly adaptable to various poetic contexts [10].

In 2022, AraGPT2 [32] was introduced as a fine-tuned version of GPT-2 [10]specifically for Arabic poetry generation. Trained on a large corpus of Arabic poetry, this model improves upon earlier techniques like RNNs and GRU's by generating verses that capture the rich linguistic and metrical structure of classical Arabic poetry. Both quantitative (e.g., BLEU scores) and qualitative human evaluations have shown that AraGPT2 is able to produce fluent and metrically accurate poetry that preserves the cultural and emotional depth typical of Arabic poetry [10]. Moreover, GPT-J [33] and BERTShared [29], two

other transformer-based models, have also been used for Arabic poetry generation. They were tested using the Ashaar dataset and showed that GPT-J is particularly effective in generating consistent rhyme schemes, while BERTShared excels in creating more coherent and fluent verses. These advancements highlight how the combination of transformer models and classical Arabic poetic forms can lead to more natural and expressive poetry generation [10] [31].

Model/Metric	Diacritics	Training Corpus size	Prediction Time / 1024	Accuracy
(Abandah et al., 2020)	✓	1,493,086	388 ms	96.18 %
Transformer Model	✓	806,062	84 ms	95.51 %
Transformer Model	✓	1,460,255	84 ms	**96.24** %
(Abandah et al., 2020)		1,493,086	388 ms	94.18 %
Transformer Model		806,062	84 ms	93.90 %
Transformer Model		1,460,255	84 ms	**95.22** %

Figure 11: Ashar models Comparison [10]

Authors in [34] proposed three approaches for generating Arabic praise poetry: A character-based model made up of three layers that predicts the next character from an input sequence. A Markov-LSTM approach where verses are first generated using a Markovify function where the models guess the next word based on its probability. The output verses of this model are then passed as input to an LSTM model which then generates new verses from the set of input verses. The final approach involves fine-tuning a pre-trained GPT-2 model with 124 million parameters. The models were evaluated using BLEU scores as well as expert opinion. Overall, the Markov-LSTM model scored highest in both metrics with the GPT-2 model performing not far behind.

Marjan Ghazvininejad and Xing Shi et al in [35]. introduce Hafez, an interactive web interface to an automated poetry generation system built up from an RNN and a Finite State Acceptor (FSA). The website allows users to request poems from the system by providing the topic of the poem as well as by giving a style configuration through the user interface to encourage things such as alliteration, sentiment, word length among other things. The backend server then generates the poem and displays it on the user's screen. If the user dislikes the generated poem, they can reconfigure the style and ask the system to regenerate

the poem without changing the rhyming words. Users give the poem a rating from 1-5 stars which helps guide the model when generating new poems in the future.

This study [36], focuses on applying generative language models for poetry creation using a sequence-to-sequence (Seq2Seq) architecture. The method employs an encoder-decoder framework, where the encoder processes the input, and the decoder generates the output poetry. Key enhancements include attention mechanisms, which help manage longer dependencies in poetic structures. Additionally, the introduction of rhyme and topical constraints helps improve the model's creative output. For rhyme, the output distribution is fine-tuned based on phonetic similarities, ensuring the generated poetry aligns with traditional rhyme patterns. Non-Negative Matrix Factorization (NMF) is employed to manage topical consistency across the poem. This approach focuses on improving both the rhyme quality and the thematic coherence without altering the model's core structure. Results show effective constraint management in producing more aesthetically pleasing poetry with high fluency and adherence to stylistic norms

In [37] , the authors discuss the techniques for controlling the balance between creativity and structure in language generation models. The approach involves the use of reinforcement learning to adjust model parameters, such as temperature and nucleus sampling, which influence the creativity and coherence of the generated text. By fine-tuning these parameters, the model can produce outputs that range from highly creative and free-form text to more structured, predictable language. This flexibility allows for greater control over the generation process, enabling the model to generate diverse text while maintaining structural consistency. The techniques presented are particularly relevant for tasks like poetry generation, where balancing creativity and form is essential

The work in [38] outlines a neural network approach to poetry generation that integrates syllable-level constraints into a recurrent neural network (RNN). The model predicts the next word in a poem while ensuring the total syllables per line meet predefined requirements. It incorporates syllable counting as part of the generation process to maintain rhythmic consistency in the generated verses. The model successfully handles structural demands, demonstrating effective syllable-based control for generating poetry

The work in [39], introduced a model that integrates Phonetic CNN subword embedding to better handle the phonetic nuances of Arabic poetry. Unlike conventional embeddings,

which focus on semantic and syntactic features, their model captures phonological aspects essential for rhythm and rhyme, both vital to classical Arabic poetry. This innovation addresses a key gap in prior models that failed to consider prosodic features.

كَفّى فُرَاقِ الْمُحّبِ أَنَمَابَقَى لِهَوَاكَ مسْعَى أَرْتَجِيّه واتقي
Enough of the the separation, For I hope and fear your passion
يَاعُبْيلُ الْشوْقُ اضّنَا رُوَح مْتعبْ بِهَوَاكَ لكْنَ جْفَاكُمُ اعْيَانَيْ وجع
O Obail, longing aches my tired soul, your aversion has made me ill
اْنْ امتْلَأَ القَلْبُ بْحُبِكْ مولعّوا فَلِمَا يُصغِيَ قَلْبُكُمُ لْمن يَخْدعُ
If the heart is full of your love, Why would it listen to the deceitful

Figure 12: Example of a poem composed by the proposed model [39]

The proposed framework involves three key stages: Word Embedding, Keywords Extraction and Expansion, and Poem Generation shown in *Figure 12*. The model generates the first verse using a Backward and Forward Language Model (B/F-LM) with GRU units, ensuring the initial verse aligns with the input theme. For subsequent verses, it employs a Hierarchical Attention Sequence-to-Sequence (HAS2S) model, using both word-level and verse-level attention mechanisms to maintain coherence across the poem. This approach ensures that new verses build meaningfully on previous lines, preserving both semantic relevance and poetic structure.

The research in [40] explored the use of pre-trained models, BERTShared and GPT-J, to generate classical Arabic poetry, emphasizing meter and rhyme adherence. Their dataset included 15 traditional meters, and the models were fine-tuned to generate structured verses. The evaluation of both machine-based and human, showed that GPT-J excelled in maintaining rhyme while BERTShared was better at fluency shown in *Figure 13*. However, both models struggled with less common meters.

Model	Prompt	Generated Poem
BERTShared	Meter: الوافر *al-Wāfir*	ألا من للموت أهون من حياة ومن لي أن أعيش بغير عيش عجبت لمن يموت وليس يدري وما في الموت موت بعد موت لعمرك ما الحياة وإن تناهت على ما قد مضى من الدنيا وعيش أرى الموت الشقاء لكل حي ولا خير الحياة لمن لحي
	Topic: (Romantic) رومنسيه	ألا أيها القلب الذي هو واجد على كل حال من سرورك موقد وإني لأرجو أن يدوم وصاله وما أنا إلا مغرم متزيد فيا ليت شعري ما الذي بي من الجوى إذا لم يكن لي في هواك مزيد أيا كبد حرى عليك تقطعت ويا لك من قلب عليك شديد فديتك لو كان الوصال بناظري إلى الله أشكو بعد ذاك الصدود ولو كنت أدري كيف أصغي لقولكم ولكن قلبي لا يزال يصيد
GPT-J	Meter: الطويل *al-Ṭawīl*	ومن جاءني بالخمس منه بحقه وكيف يحامي حاجب بالأجارب لقد حط في بعض اللقاء مكانه وما حط في بعض اللقاء مناسبي وقد جاءني بالخمس والجهل مذنب فما هو لي في بيت مجد وكاذب ولكن إذا ما أعطيت كف مجدها وليس لها غير الملاقاة ثاقب فلا برحت أيدي الليالي مخافة تقر لها عيني وتلوى الحواجب
	Topic: (Sad) حزينه	وعين على الأشجان بات ينيرها ومن دونه عود اللمى فيعيرها وحالف صبري في التجلد إنها إذا قام منها الوجد في الدهر حورها إذا نازحتها الدار بات يجودها بما جاد في الأيام منها أسيرها أتطلب دار لم ينل من بلادها ويعذب إلا في التراب بدورها إذا ما ارتدى بالماء لم يحو مقلة سوى ذلك الصافي ولا ذلك النورها

Figure 13: Examples of generated poetry [40]

Overall, the study demonstrated that pre-trained transformers have significant potential for classical poetry generation, but challenges remain in handling complex poetic structures and imbalanced data. Future work could focus on incorporating human feedback to improve model performance.

In [41], the authors explore the use of deep learning models, specifically LSTM and GPT-2, for generating Arabic poetry. Leveraging the Arabic Poem Comprehensive Dataset (APCD), which contains over 1.7 million verses from various historical periods of Arabic poetry. The study compares the performance of these models when trained on both the full dataset and a subset focusing on a single meter, "Taweel," one of the most used meters in classical Arabic poetry shown in Figure 14.

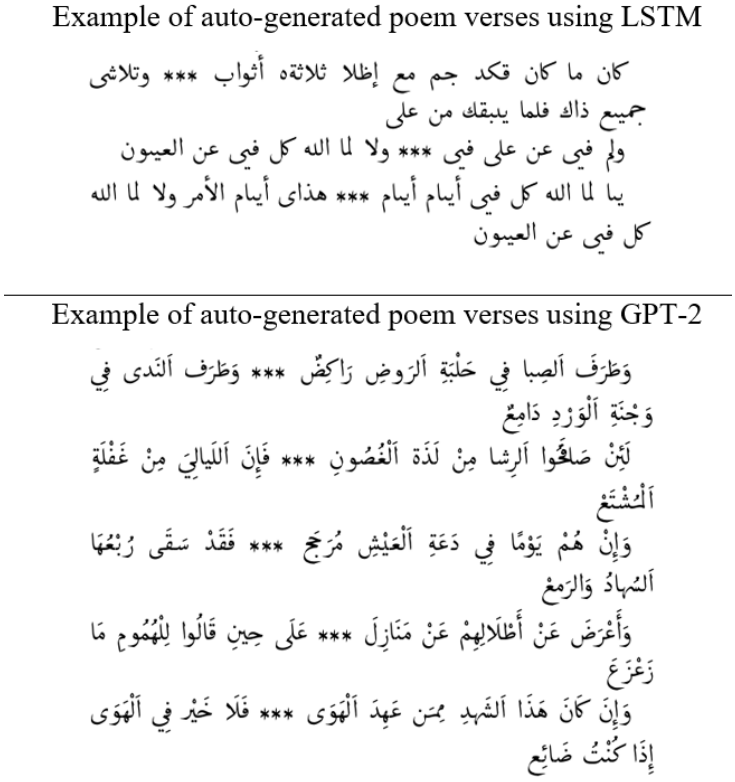

Figure 14: Examples of auto-generated poem verses using GPT-2 and LSTM [41]

The results indicate that GPT-2 outperforms LSTM across all metrics fluency, coherence, meaningfulness, and poetic quality, achieving an average human evaluation score of 2.94 out of 5, though both models lagged behind human poets.

The work in [42] depicts a deep learning model designed for Arabic poetry, using a two-stage process, Keyword Extraction and Verse Generation. In the Keyword Extraction stage, the model extracts a set of key terms representing the poem's themes, where each keyword corresponds to a verse in the poem. And in the Verse Generation stage each verse is generated using a keyword along with all previously generated verses, ensuring semantic relevance and thematic consistency throughout the poem. In the Verse Generation stage, it uses two primary approaches. The first approach is called First Verse Generation; it's a Bi-directional Gated Recurrent Unit (Bi-GRU) model used to generate the first line of the poem. The approach uses both forward and backward GRU structures, allowing the model to better predict and capture the structure of the first verse by incorporating both past and

future word contexts. The second approach is called Subsequent Verses Generation, it's a modified Bi-GRU encoder-decoder with a hierarchical neural attention mechanism that is used to generate the remaining verses sequentially. This hierarchical attention model is designed to focus on two levels of information: "Word-level attention" This mechanism captures associations between individual words within a verse, helping the model focus on the most informative words. And "Verse-level attention" This level captures relationships between entire verses, ensuring coherence and continuity across the poem.

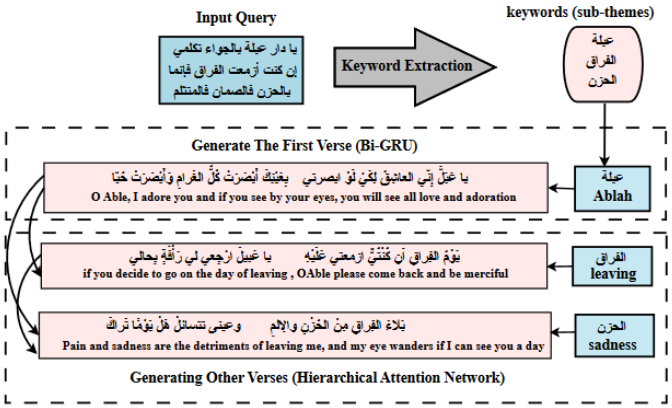

Figure 15: Sample keyword extraction and verse generation [42]

The paper also employs FastText embeddings, which are suitable for morphologically rich languages like Arabic. FastText breaks words into character n-grams, enabling the model to capture both syntactic and semantic relationships more effectively, especially for rare or complex words. The performance of the model is evaluated using both human judgment (on metrics such as meaning, coherence, and poetic-ness) and quantitative measures (like BLEU scores). The results indicate that the proposed model outperforms baseline models in generating high-quality Arabic poetry.

The research in [43] explores the generation of Arabic poetry using GPT-2, the authors utilize GPT-2, a decoder-only transformer, to predict the next word in a sentence based on a large corpus of text. Due to computational constraints, they employ the small GPT-2 model (117 million parameters). The model processes Arabic text using BPE (Byte-Pair Encoding) for efficient tokenization. In Data Processing The model is pre-trained using two Modern Standard Arabic corpora, Khaleej-2004 and Watan-2004, containing millions of words from news articles. Fine-tuning is done on a large Arabic poetry dataset, scraped

from Aldiwan [11]. After that in training they performed on an NVIDIA Tesla T4 GPU, took 32 hours over 25 epochs, while fine-tuning for Arabic poetry generation took 12 hours over 6 epochs. The authors used a custom BPE tokenizer for Arabic and pre-trained the model with Hugging Face's GPT-2 in Arabic. In Poem Generation, they employed top-k and top-p sampling strategies to ensure a balance between creativity and coherence. These techniques optimize the generation by filtering probable next words, enhancing diversity while maintaining relevance. The results show that GPT-2 can successfully generate high-quality Arabic poetry, outperforming existing models, especially after fine-tuning on poetry-specific datasets.

The field of natural language processing has seen a dramatic transformation with the development of large language models, which are capable of performing a wide range of tasks. However, their effectiveness often depends on their ability to adapt to specific domains and applications. In this project, we aim to explore multiple advanced approaches to harness the potential of these models and determine the most effective methodology for our objectives.

Our exploration will include leveraging innovative architectures like RWKV (Receptance Weighted Key Value), which combines the computational efficiency of recurrent neural networks with the scalability of transformers, and RAG (Retrieval-Augmented Generation), which integrates external knowledge bases to enhance factual accuracy and contextual depth. Additionally, we will focus on fine-tuning large language models, including LLaMA, Qwen, and ALLaM, by adapting them to specific tasks and datasets to achieve greater precision and alignment.

By implementing and evaluating these approaches, we aim to measure their performance across various metrics such as accuracy, computational efficiency, and relevance to the domain at hand. This systematic evaluation will allow us to identify the strengths and limitations of each method, ultimately guiding us to the approach that delivers the best results. In the following section, we delve into the details of fine-tuning large language models for specialized applications.

Chapter 4: Proposed works

The field of natural language processing has seen a dramatic transformation with the development of large language models, which are capable of performing a wide range of tasks. However, their effectiveness often depends on their ability to adapt to specific domains and applications. In this project, we aim to explore multiple advanced approaches to harness the potential of these models and determine the most effective methodology for our objectives. Specifically, this research explores leveraging innovative architectures such as the Receptance Weighted Key Value (RWKV) [44] which combines the computational efficiency of recurrent neural networks with the scalability of transformers, and the Retrieval-Augmented Generation (RAG) [45] which integrates external knowledge bases to enhance factual accuracy and contextual depth. Additionally, we intend to focus on fine-tuning large language models, including LLaMA [46], Qwen [47] and ALLaM [9] by adapting them to specific tasks and datasets to achieve greater precision and alignment.

By implementing and evaluating these approaches, we aim to measure their performance across various metrics such as accuracy, computational efficiency, and relevance to the domain at hand. This systematic evaluation will allow us to identify the strengths and limitations of each method, ultimately guiding us to the approach that delivers the best results. In the following section, we delve into the details of fine-tuning large language models for specialized applications.

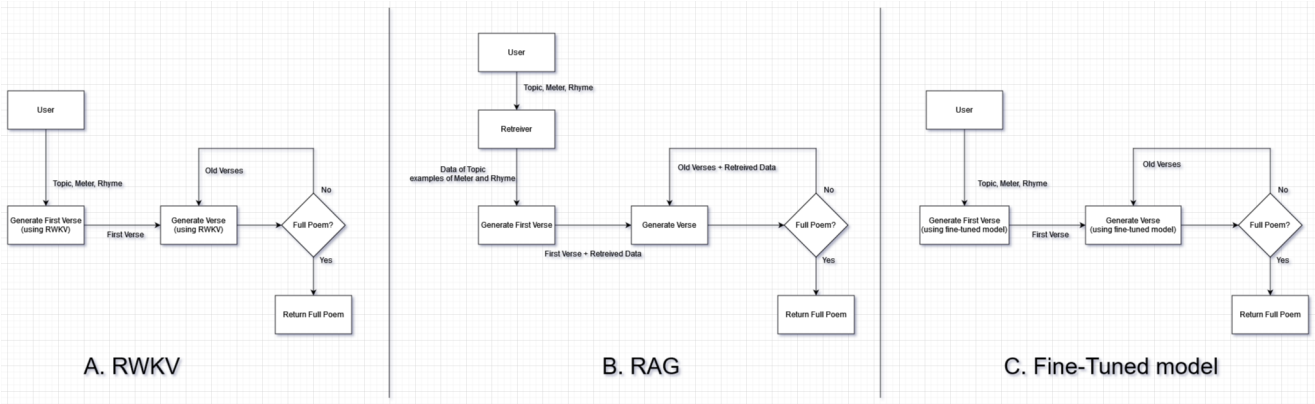

Figure 16: Proposed Models' Flowcharts

4.1 Receptance Weighted Key Value for Arabic poetry Generation

Receptance Weighted Key Value (RWKV) is a novel architecture that combines the advantages of Recurrent Neural Networks (RNNs) and transformers. It was designed to handle the limitations of traditional RNNs, which struggle with long-term dependencies, while also benefiting from the efficiency and scalability of transformers. By merging these two powerful techniques, RWKV offers a new approach for processing sequential data, such as language, in a way that maintains context over long sequences [44].

In the context of Arabic poetry generation, RWKV's unique strengths are especially valuable. Arabic poetry follows strict rules of rhyme (al-qafiya) and meter (al-bahr), and its structure requires a model capable of maintaining long-term dependencies across multiple lines and stanzas. The ability to maintain coherence, rhyme, and rhythm over long sequences of text is a challenging task for conventional models. RWKV, with its time-decay mechanism and weighted key-value processing, addresses these challenges effectively, making it an ideal enhancement for large language models (LLMs) like LLaMA [46], Qwen [47] and ALLaM [9] when generating Arabic poetry.

The Arabic poetry generation process begins by tokenizing the input text, breaking it down into smaller units (tokens). These tokens are then converted into vector representations, which capture both the syntactic and semantic relationships inherent in the Arabic language. This allows the model to understand the meaning behind each word or phrase.

Once the input is tokenized and embedded, it is passed through the RWKV core, where the magic happens. The RWKV architecture uses a Receptance Mechanism that preserves long-term dependencies by weighing the relevance of context throughout the generation. This mechanism is particularly critical in poetry, where earlier lines influence the structure and meaning of the subsequent ones.

One of the most important features of RWKV in poetry generation is its time-decay mechanism. As new lines of the poem are generated, RWKV places greater importance on the most recent context while gradually "decaying" the influence of earlier lines. This helps the model maintain the current flow and focus on the evolving themes, without the generation being disrupted by distant context. In Arabic poetry, where the poem must feel fluid and connected, this feature is key in maintaining both coherence and creativity.

In addition to handling dependencies, RWKV is highly effective at stylistic fine-tuning. Arabic poetry adheres to specific rules of rhyme (al-qafiya) and meter (al-bahr), and RWKV fine-tunes these elements as the poem is generated. The architecture ensures that each line conforms to the correct rhyme and rhythm, preserving the traditional forms while allowing for artistic expression.

Finally, RWKV employs weighted key and value processing, which dynamically adjusts the importance of different tokens in the sequence. This allows the model to focus on the most relevant information at each step, helping it generate poetry that is both thematically consistent and stylistically accurate.

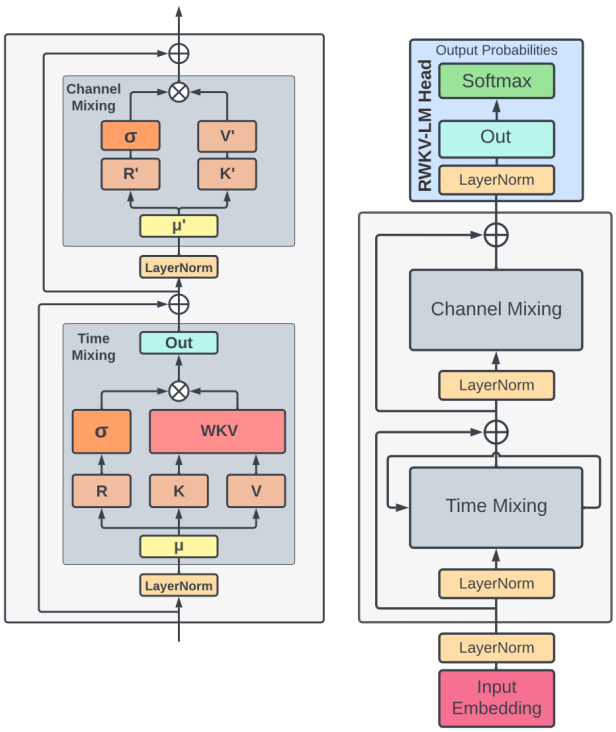

Figure 17: Illustration of the RWKV architecture [44]

Figure 17 provides a visual overview of how the RWKV core processes sequences. Key components such as the receptance mechanism and time-decay mechanism are crucial for maintaining long-term dependencies, enabling contextually accurate and coherent outputs over extended sequences.

4.1.1 RWKV based Arabic Poetry Generation

Building on RWKV's core strengths, the process for generating Arabic poetry adapts these mechanisms to adhere to the strict rules of Arabic poetic form. One of the most important features of RWKV in poetry generation is its time-decay mechanism. As new lines of the poem are generated, RWKV places greater importance on the most recent context while gradually "decaying" the influence of earlier lines. This helps the model maintain the current flow and focus on the evolving themes without the generation being disrupted by distant context.

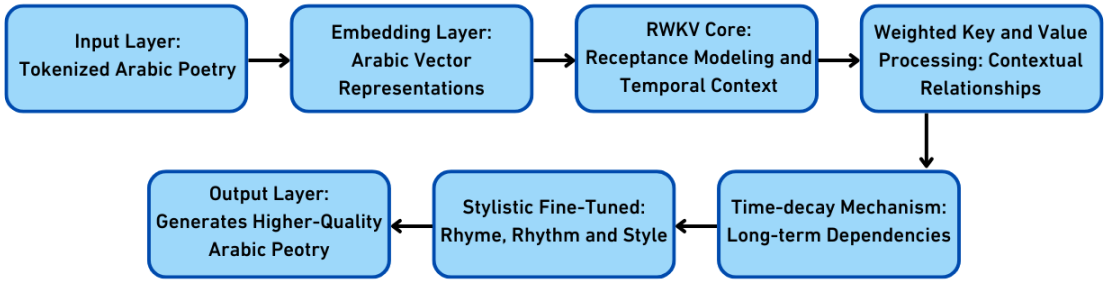

Figure 18: RWKV Pipeline for Arabic Poetry Generation.

Figure 18 visualizes the entire process of generating Arabic poetry using RWKV. It shows how the input is processed through each layer, and where RWKV's mechanisms like time-decay and key-value weighting come into play to ensure high-quality output. As it can be seen, in the input layer, the Arabic poetry input is tokenized into smaller units (tokens). This step is necessary for converting the raw text into a format that the model can process. Tokenization breaks the text into manageable chunks, allowing the model to understand individual words or phrases. Then, in the embedding Layer, the tokenized input is then transformed into vector representations that capture the relationships between words. This step is crucial for understanding the deep meaning behind each token in the context of Arabic grammar and poetry. Next, in RWKV Core, the embedded input passes through the RWKV core, which utilizes its receptance mechanism to preserve long-term dependencies. The model adjusts the relevance of different parts of the input, ensuring that the poem maintains its thematic coherence over time. Besides, as new lines are generated, RWKV applies a time-decay mechanism, reducing the influence of older lines and ensuring that the model stays focused on the most relevant context. This helps prevent the poem from becoming disjointed or losing its thematic focus.

Moreover, RWKV fine-tunes the stylistic elements of the poem, adjusting the rhyme and meter to ensure that the poem adheres to traditional Arabic poetic forms. This is a crucial step in Arabic poetry generation, as the aesthetic quality of the rhyme and rhythm is a central part of the art form. Finally, the model generates the output he refined Arabic poetry. This output respects the required rhyme, meter, and coherence, ensuring that the poem flows naturally and aligns with the traditional structure of Arabic poetry.

4.2 Retrieval-Augmented Generation based Arabic poetry Generation

RAG is an advanced technique designed to enhance the performance of LLMs by integrating external knowledge sources. This approach addresses several limitations of LLMs, such as knowledge gaps, factual inaccuracies, and hallucinations, which are especially problematic in knowledge-intensive or domain-specific tasks like Arabic poetry. By dynamically retrieving relevant information, RAG enables LLMs to generate outputs that are accurate, contextually relevant, and up to date without requiring retraining [48].
As it can be seen in Figure 19, RAG operates by integrating two primary components: a retriever and a generator. This approach enables large language models (LLMs) to access external knowledge sources. Here's how the retriever and generator components work together in RAG [45] :

1. User Query: A user provides a query or prompt.
2. Retrieval Step: The retriever fetches relevant information from the external knowledge base based on the query.
3. Generation Step: The generator combines the retrieved content with the user's query to produce a response.
4. Final Output: The system returns a response that is informed by both the model's internal knowledge and the retrieved external context.

RAG offers unique advantages for generating Arabic poetry by addressing several challenges intrinsic to the task:

1. Maintaining Poetic Structures: Arabic poetry demands adherence to specific meters (Bahr) and rhymes (Qafiyah). RAG can retrieve examples of similar poetic forms or structural templates, guiding the LLM to generate a poem that conforms to these rules.
2. Contextual Enrichment: Arabic poetry often incorporates historical, cultural, or religious allusions. RAG enables the retrieval of relevant background information, like historic events from books, ensuring the generated poetry resonates with its intended audience.

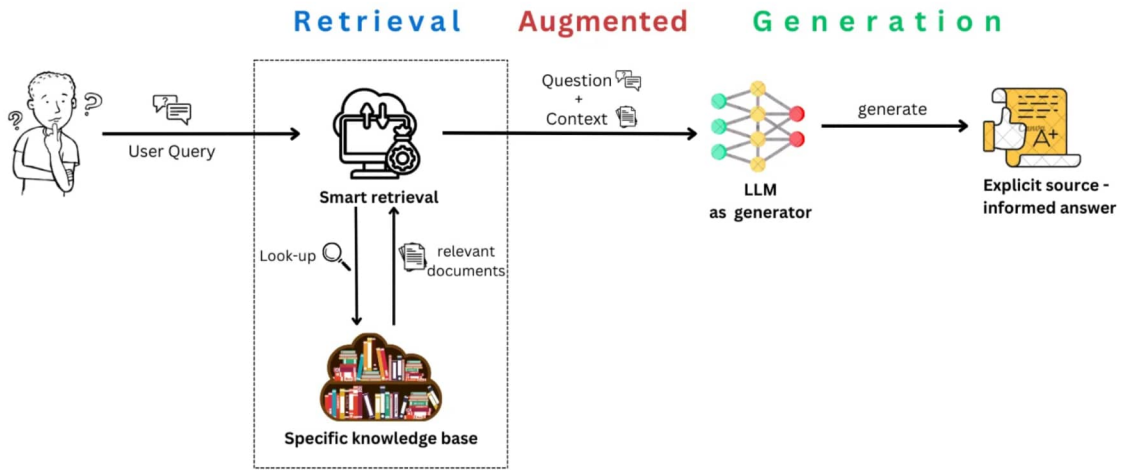

Figure 19: Retrieval Augmented Generation [49]

The retriever in RAG is responsible for finding the most relevant information from external sources to support the LLM's response. It uses methods like embeddings and semantic search to match the user's query with the right content, ensuring the retrieved data is accurate and useful [45]. In our case, the retriever can access collections of poems or historical texts to provide examples that fit the query's style or theme. To better handle poetic tasks, meter (Bahr) and rhyme (Qafiyah) can also be encoded into the embeddings, allowing the retriever to prioritize texts with similar meters or rhymes. This ensures the generated poetry stays true to the poetry rules.

4.3 Fine-Tuning Large Language Models (LLMs)

Fine-tuning large language models (LLMs) [50] is essential for adapting them to specific tasks or domains. While pretrained models excel at generalization, fine-tuning allows them

to specialize using smaller, task-specific datasets, improving their accuracy and contextual relevance. A key innovation in this process is the use of appended neurons, which add task-specific layers to the model without altering its core structure. This approach preserves the original model's general capabilities while efficiently integrating new knowledge. By focusing adjustments on appended neurons, fine-tuning becomes more resource-efficient, enabling models to adapt quickly and effectively to specialized applications.

In particular, fine-tuning involves modifying a pretrained model by adding new layers or adjusting existing ones to optimize performance for specific tasks. There are a lot of approaches but the most common one is low-rank adaptation (LoRA) [50] that we will use it, which appends task-specific layers to the model while freezing the original pretrained parameters. This ensures efficient resource usage and avoids overwriting the base model's general capabilities.

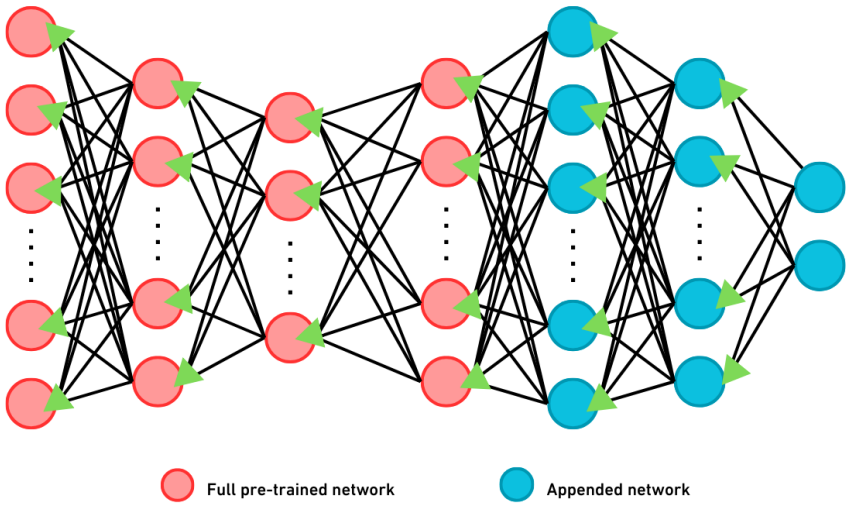

Figure 20: Fine-Tuning process.

As depicted in Figure 20, the structure of fine-tuning consists of several key components*:*

Appended Layers: New layers are added to the pretrained model, specifically designed to capture task-specific patterns. These layers interact with the output of the frozen pretrained layers but focus solely on learning the features relevant to the new task.

Input and Preprocessing: Task-specific datasets are cleaned, tokenized, and aligned with the model's input requirements. For example, in fine-tuning for Arabic poetry, the dataset must include structured examples of poetic forms, rhythms, and rhymes.

Gradient Flow: During training, gradients are calculated and applied only to the appended layers. This ensures that the pretrained model's parameters remain fixed, preserving its general knowledge while the appended layers specialize in the task.

Optimization: Fine-tuning uses optimizers like Adam or its variants to adjust the weights in the appended layers. Loss functions such as cross-entropy are used to evaluate and improve the model's outputs.

Evaluation: The fine-tuned model is tested on validation datasets that represent the target domain. Metrics like BLEU for text coherence, perplexity for fluency, and domain-specific metrics (e.g., rhyme and rhythm accuracy in poetry) are used to measure performance.

This structured approach ensures that fine-tuning is both efficient and effective, leveraging the pretrained model's strengths while specializing it for new challenges.

4.3.1 Fine-Tuning for Arabic Poetry Generation

Fine-tuning large language models such as LLaMA [46], Qwen [51], and ALLaM [9] for Arabic poetry generation involves systematically adapting the pretrained architecture to the intricate linguistic and structural characteristics of classical and modern Arabic poetry. Arabic poetry is governed by strict rules, including Buhur (meter systems), Arud (rhythmic frameworks), and Qafiyah (rhyme schemes), requiring the model to generate text that adheres to these constraints.

The fine-tuning process begins with the preparation of a carefully curated and annotated dataset. The dataset must include examples of structured poetic forms, with explicit annotations for rhythm patterns, rhyme structures, and stylistic features. Preprocessing ensures tokenization aligns with the model's architecture, with attention to preserving the morphological and syntactic richness of Arabic.

To adapt the model, Low-Rank Adaptation (LoRA) is employed. LoRA inserts task-specific rank-restricted matrices into the pretrained model's architecture without modifying its original parameters. These additional layers are designed to capture task-specific features by learning low-dimensional updates, significantly reducing memory and

computational requirements. The pretrained weights are frozen, and gradients are computed only for the LoRA layers, isolating task-specific learning while maintaining the model's general capabilities.

The fine-tuning optimization is driven by a cross-entropy loss function, tailored for the constraints of Arabic poetry. The training process involves feeding prompts and evaluating outputs for their compliance with annotated rhythmic and rhyme patterns. The LoRA layers learn to map the poetic rules encoded in the dataset to model predictions while maintaining high efficiency in training.

The evaluation metrics for the fine-tuned model include perplexity for fluency, rhyme coherence scores, rhythmic accuracy based on Taf'eelat (metrical units), and human assessments for artistic and cultural fidelity. Iterative tuning and hyperparameter adjustments further refine performance, ensuring that the model generates outputs that respect both the formal rules and the creative essence of Arabic poetry.

By leveraging LoRA's efficiency and the structural constraints of Arabic poetry, this fine-tuning approach enables the model to specialize in generating high-quality, metrically accurate, and stylistically authentic poetic outputs, while preserving the general linguistic capabilities of the base architecture.

Like we said before Fine-tuning a model allows it to specialize in a specific task or domain by adapting a pretrained model to new data. The main advantage is that it significantly enhances the model's performance in the targeted area, improving accuracy, relevance, and the model's ability to handle specialized tasks. By focusing on a specific dataset, fine-tuning ensures that the model can learn task-specific patterns that the original model may not have encountered during its initial training. It also requires fewer resources and less training time compared to training a model from scratch, making it an efficient approach.

However, fine-tuning can also have some drawbacks. One of the main issues is that the model may experience "catastrophic forgetting" [50] where it loses some of the knowledge it gained from its original training. This is particularly problematic when the fine-tuning data is narrow or limited, as it can overwrite generalizable capabilities with overfitting to the new task. Additionally, fine-tuning can increase the risk of overfitting if the dataset is too small or not diverse enough, leading to poor generalization to new or unseen data.

Furthermore, fine-tuning may require significant computational resources for large models, especially when multiple fine-tuning tasks need to be handled, which can be inefficient for applications requiring real-time responses. Lastly, fine-tuned models may become too specialized, making it challenging to apply them to different tasks without further adjustments or retraining.

Fine-tuning LLMs such as LLaMA, Qwen, and ALLaM not only maximizes their potential but also opens avenues for addressing specialized challenges across languages and domains. This approach integrates modern AI advancements with tailored applications, ensuring relevance and excellence in real-world implementations.

Chapter 5: Experiment Settings

This chapter introduces the dataset used for Arabic poetry generation, emphasizing its importance in training models to produce poetry that aligns with the traditional conventions of Arabic verse. The dataset comprises classical Arabic poetry, encompassing a diverse range of themes, meters, and rhyme schemes. It is meticulously annotated with critical metadata such as meter (al-bahr) and rhyme scheme (al-qafiya), which are foundational for generating poetry that respects these structural norms. This chapter discusses the dataset's structure, preprocessing steps, and its essential role in creating linguistically accurate and culturally authentic Arabic poetry.

5.1 Ashaar Dataset

The Ashaar dataset is a carefully curated collection of classical Arabic poetry, serving as a cornerstone for training advanced models like RWKV, LLaMA, Qwen, and ALLaM. It spans multiple periods of classical Arabic literature, offering a vast repertoire of poetic forms, themes, and structural conventions. Each poem is enriched with essential metadata, such as its meter (al-bahr) and rhyme scheme (al-qafiya), which are crucial for enabling models to grasp and replicate the complex patterns inherent to Arabic poetry.

The dataset plays a pivotal role in preserving the artistic integrity of Arabic verse while facilitating an understanding of the cultural and linguistic nuances embedded within it. For instance, the inclusion of meter annotations allows models to capture the rhythmic essence of Arabic poetry, while the rhyme scheme metadata guides the generation of coherent and harmonious poetic lines. The dataset also categorizes poems by themes such as love (ghazal), praise (madh), or elegy (ritha'), ensuring that the generated poetry aligns contextually and stylistically with its intended purpose.

Beyond its focus on structure and coherence, the dataset also provides a comprehensive view of stylistic variations within Arabic poetry. This diversity enhances the model's ability to generate contextually rich poetry that reflects the varied traditions within Arabic literature. By combining structural rigor with creative freedom, the Ashaar dataset ensures that generated poetry is both authentic and innovative.

5.2 Dataset Annotation and Structure

The Ashaar dataset's annotations play a critical role in enhancing its utility for training. Each poem in the dataset is accompanied by detailed metadata, including annotations for its meter (al-bahr), rhyme scheme (al-qafiya), themes, period, and author. These structured annotations enable models to identify and reproduce the intricate structural elements that define Arabic poetry. Poems adhering to specific meters such as al-Tawil, al-Kamil, or al-Basir are labeled accordingly, providing the model with clear guidance on generating rhythmic compositions that align with classical patterns.

Rhyme scheme annotations are equally vital, ensuring that generated poetry maintains the repetitive harmony characteristic of Arabic verse. By embedding semantic information, such as thematic categorization, the dataset also allows models to generate poetry that is not only linguistically accurate but also contextually meaningful. For instance, poems categorized under specific themes, such as love or praise, allow models to produce poetry that is tailored to these themes, maintaining both poetic structure and thematic relevance.

The dataset's structure is designed to reflect the historical evolution of Arabic poetry, incorporating works from various periods and capturing shifts in poetic styles and cultural influences. This temporal diversity is essential for training models to generate poetry suited to specific historical or cultural contexts. By including a wide array of themes and poetic forms, the dataset provides a robust foundation for producing both classical and contemporary Arabic poetry, with the flexibility to adapt to various stylistic and contextual needs.

Figure 21 illustrates the structure of the Ashaar dataset, highlighting the various types of metadata annotations associated with each poem. These annotations include details on the poem's meter (al-bahr), rhyme scheme (al-qafiya), thematic classification, and additional information such as poet details and poem era. The dataset's structure allows for the generation of linguistically accurate and contextually relevant Arabic poetry by enabling models to understand the rhythmic and thematic nuances of traditional Arabic verse [52].

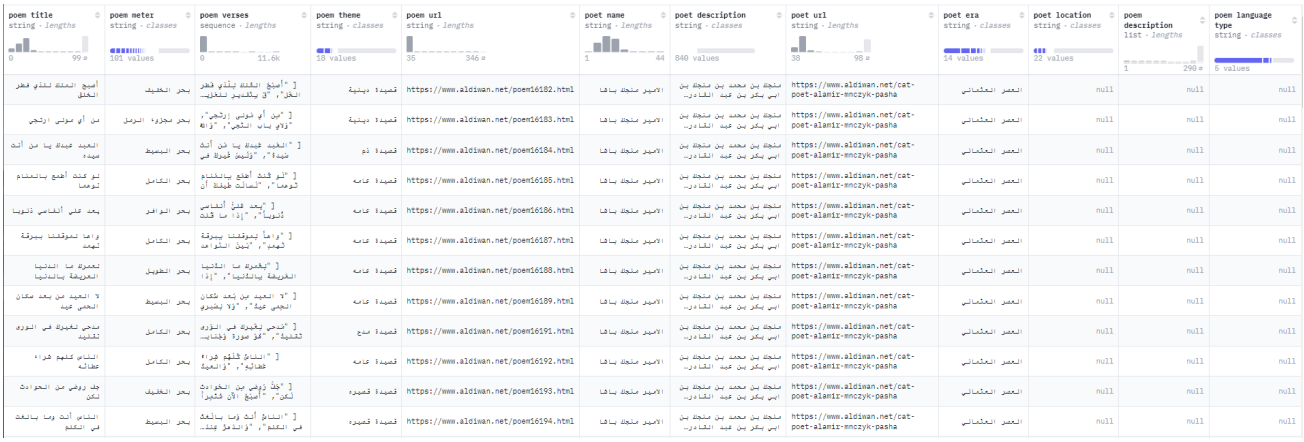

Figure 21: Structure and Annotations of the Ashaar Dataset.

5.3 Preprocessing Steps

The preprocessing of the dataset involved a series of steps to prepare it for effective model training. The process is illustrated in Figure 22. Text cleaning was carried out to eliminate inconsistencies and extraneous characters, ensuring uniformity. Normalization processes addressed variations in Arabic spelling and diacritics, while tokenization divided the text into manageable units, such as words or syllables, to facilitate model training.

Special tokens were added to mark the start and end of each line, helping the model understand the line-by-line progression of Arabic poetry. The dataset was further enhanced by balancing its representation of different meters and rhyme schemes to prevent bias toward any specific poetic form. Semantic embeddings were integrated to enrich the contextual understanding of words and phrases, which is crucial for generating coherent and culturally nuanced poetry.

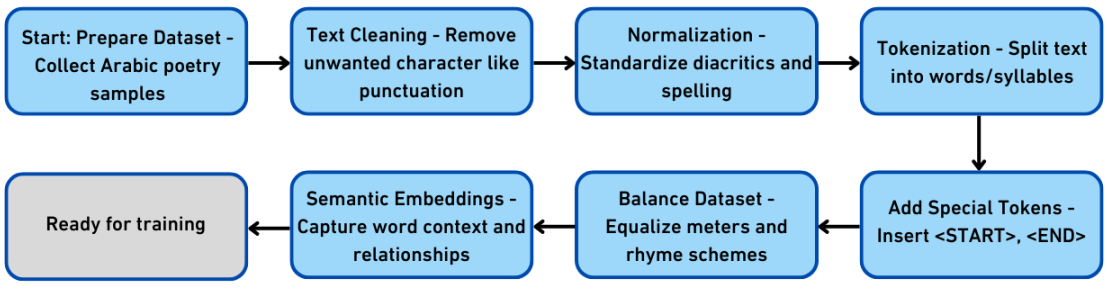

Figure 22: A flowchart detailing the preprocessing steps, such as text cleaning, normalization, and tokenization, applied to prepare the dataset for training

Chapter 6: Research Hypotheses

This research posits that the integration of advanced AI techniques, particularly fine-tuning, Retrieval-Augmented Generation (RAG), and Receptance Weighted Key Value (RWKV) architecture, would enable the generation of Arabic poetry that aligns with traditional poetic forms while preserving its linguistic beauty, rhythm, and cultural depth. The fusion of these methodologies aims to address the distinct challenges posed by Arabic poetry, such as maintaining rhythmic accuracy, thematic consistency, and adherence to its strict linguistic and stylistic conventions. The hypothesis is based on the belief that these techniques, when effectively combined, will enable the generation of Arabic poetry that is not only technically proficient but also culturally and artistically resonant with the rich literary tradition of the Arab world.

6.1 Receptance Weighted Key Value

The Receptance Weighted Key Value (RWKV) architecture is hypothesized to play a pivotal role in addressing the challenge of maintaining long-term dependencies and structural coherence in generated Arabic poetry. Arabic poetry is known for its intricate patterns, spanning multiple lines and stanzas with a fixed rhythm and rhyme. One of the key difficulties in generating such poetry is ensuring that the generated verses adhere to the rules governing meter and rhyme over long sequences, while also maintaining thematic cohesion.

RWKV's unique ability to process data sequentially while utilizing a time-decay mechanism to prioritize more recent context is expected to address this challenge. This feature will allow the model to generate verses that flow naturally, preserving rhythmic integrity while maintaining coherence across long sequences of text [44]. Additionally, the receptance mechanism inherent in RWKV is expected to improve the model's ability to generate poetry that is not only thematically cohesive but also stylistically consistent with traditional Arabic poetic forms. By leveraging RWKV, the model can respect the rhythmic patterns of Arabic verse, while ensuring that the overall structure remains fluid and coherent, regardless of the length of the generated text.

In the context of Arabic poetry, where adherence to meter and rhyme is critical, RWKV's ability to handle long-range dependencies and sequential data is expected to provide a significant advantage. Its unique approach will enable models to generate long-form Arabic poetry that adheres to traditional rules while sounding natural and artistically rich [53].

6.2 Retrieval-Augmented Generation (RAG)

The research further hypothesizes that the integration of Retrieval-Augmented Generation (RAG) will significantly enhance the contextual relevance and coherence of the generated poetry. RAG combines the strengths of retrieval-based methods with generative models, enabling the model to draw from a large pool of pre-existing examples during the generation process. By retrieving relevant examples or references from the Ashaar dataset or other relevant sources, the model ensures that the generated output is firmly rooted in authentic poetic traditions.

This approach is expected to address challenges such as ensuring thematic consistency across multiple lines and stanzas, enriching the generated poetry with culturally relevant references, and generating poetry that aligns with specific poetic themes or historical periods. RAG is anticipated to enhance the model's ability to generate contextually coherent poetry, minimizing the risks of factual inaccuracies or stylistic errors. Moreover, RAG's dynamic retrieval mechanism will ensure that the output remains grounded in the established traditions of Arabic poetry, while still being adaptable enough to incorporate the model's creative generative capabilities. Thus, the integration of RAG is hypothesized to enable the production of Arabic poetry that feels both authentic and relevant to contemporary contexts while respecting the formal conventions of the tradition [45].

6.3 Fine-Tuning

It is hypothesized that fine-tuning large language models (LLMs) on a well-curated and comprehensive dataset like Ashaar will empower the models with a specialized understanding of the formal and stylistic elements of Arabic poetry. Fine-tuning leverages the dataset's deep annotations, allowing models to grasp the intricate patterns of Arabic verse, such as meter (al-bahr), rhyme (al-qafiya), and syntactic structures that are unique to the Arabic language. By fine-tuning on this dataset, models are expected to learn not

only the rules that govern the rhythmic and phonetic aspects of Arabic poetry but also the cultural and historical context that shapes the thematic and stylistic elements of the verse [54].

The fine-tuning process is anticipated to enhance the models' ability to generate verse that adheres strictly to the formal elements of Arabic poetry while maintaining fluency and naturalness. Furthermore, it is expected that this process will enable the models to replicate the diversity of Arabic poetic forms across different historical periods, capturing the thematic richness of classical poetry—whether it be love (ghazal), praise (madh), or elegy (ritha'). The models, once fine-tuned, should exhibit an improved capacity to generate poetry that is not only linguistically precise but also rich in thematic content and stylistic nuances, reflecting the artistic creativity of classical Arabic poets.

6.4 Combined Effect

It is hypothesized that the combined application of fine-tuning, RAG, and RWKV will result in a synergistic effect that significantly enhances the quality of the generated Arabic poetry. Fine-tuning will provide the foundational knowledge of Arabic poetry's formal elements, ensuring that the generated verses adhere to established rules of meter and rhyme. RAG will enhance contextual relevance and thematic coherence by retrieving relevant examples, enabling the model to draw on a rich pool of poetic tradition. Meanwhile, RWKV's ability to maintain long-term dependencies and structural coherence will ensure that the generated poetry flows naturally across stanzas and maintains rhythm and rhyme consistency.

Together, these techniques are expected to complement each other, resulting in a model capable of producing Arabic poetry that is not only linguistically precise but also culturally rich and artistically expressive. The integration of fine-tuning will ensure that the model has a deep understanding of the structural and stylistic elements of Arabic poetry, while RAG will ensure that the output is contextually grounded and thematically coherent. RWKV's unique features will allow the model to handle long-form generation, producing poetry that remains consistent in style, rhythm, and thematic focus over extended passages.

The combined effect of these methods is anticipated to result in poetry that is both formally accurate and creatively rich, adhering to the traditional conventions of Arabic verse while allowing for a degree of artistic innovation. This synergistic approach is expected to not only advance the field of Arabic poetry generation but also provide a robust foundation for future research and applications in AI-generated literature.

Chapter 7: Experiments

This chapter presents the findings from the experiments conducted to evaluate and refine Arabic poetry generation models. The primary aim of these experiments was to assess the effectiveness of different models in generating poetry that embodies the essence of traditional Arabic poetic forms. A series of distinct thematic prompts, including spiritual praise, romantic ghazals, nature, wisdom, and leadership, were used to guide the generation of poems. Expert reviewers were involved in assessing the generated poetry based on key attributes such as linguistic fluency, rhyme consistency, metrical accuracy, and emotional depth.

7.1 Fine-Tuning and Retrieval-Augmented Generation (RAG)

The proposed approach involves two key techniques: fine-tuning and Retrieval-Augmented Generation (RAG). Fine-tuning was employed to adapt the models specifically for the task of generating high-quality Arabic poetry, enabling them to capture the intricacies of rhyme, meter, and poetic style. A curated dataset of both classical and contemporary Arabic poetry was used to fine-tune the models, ensuring their outputs would align with the conventions of Arabic poetic tradition. In addition to fine-tuning, this research explored Retrieval-Augmented Generation (RAG), a method that enhances the model's generation capabilities by providing access to external contextual information during the generation process. The rationale was that RAG could potentially improve creativity and diversity in the poems by allowing the models to pull relevant poetic context during generation. However, the results were not as promising as expected. In fact, the integration of RAG failed to meaningfully improve the generated poetry and did not add value by introducing additional poetic sources. As a result, fine-tuning emerged as the only successful technique, with RAG contributing no measurable enhancement to the quality of the outputs.

Throughout the considered experimentation process, a number of models were evaluated for their ability to generate structured Arabic poetry. Namely, these models include LLaMa, Qwen, RWKV, and ALLaM. Unfortunately, LLaMa and Qwen showed

significant limitations when applied to Arabic language generation. Both models produced outputs with poor fluency, failed to maintain metrical structure, and struggled with coherent rhyme—making them unsuitable for Arabic poetry generation. Similarly, RWKV, while interesting due to its lightweight and recurrent-style architecture, was computationally impractical for our purposes. It required training resources beyond what was available and could not be effectively fine-tuned using the provided infrastructure. In contrast, ALLaM, when fine-tuned on the curated dataset, generated poetry that demonstrated strong alignment with the expected Arabic poetic form. It adhered well to the rules of *بحر الطويل*, maintained consistent rhyme with the required letter, and delivered semantically coherent content. Another model, Silma, also responded well to fine-tuning, showing high competence in producing metrically accurate and stylistically appropriate Arabic verse.

One should note that as the experimentation progressed, we expanded our evaluation to include non-open-source models known for their advanced generative capabilities—specifically, ChatGPT and Claude. Despite not being fine-tuned on our dataset, both models were able to generate Arabic poetry of surprisingly high quality. Their outputs demonstrated strong semantic fluency and a reasonable degree of conformity to rhyme and meter, making them suitable candidates for final comparison against our fine-tuned models.

These results reinforced the conclusion that fine-tuning domain-relevant models, such as ALLaM and Silma, is highly effective. At the same time, state-of-the-art proprietary models like ChatGPT and Claude can serve as competitive baselines or complementary tools in Arabic poetry generation, even without specialized fine-tuning.

7.2 Training Infrastructure

For the fine-tuning process, we relied on Google Colab with a T4 GPU, which provided the necessary computational resources to manage the extensive demands of training large language models. This infrastructure enabled efficient fine-tuning of the models over multiple iterations, ensuring high-quality outputs. The T4 GPU played a crucial role in speeding up the process, allowing us to train the models without long delays [55].

The combination of fine-tuning and the use of high-performance hardware was integral to the success of the experiment, as it enabled the models to learn complex patterns in Arabic poetry, thus improving their generative capabilities.

7.3 Evaluation Process

To assess the poetic quality of the generated outputs, we designed an evaluation process rooted in human expertise and deep familiarity with the Arabic literary tradition. Since Arabic poetry is a nuanced art form—where rhythm, emotion, and stylistic authenticity often transcend what automated metrics can capture—we chose to rely solely on human judgment. In the final evaluation phase, all selected models were given the same poetic task with well-defined constraints to ensure fairness and consistency. The models were instructed to generate a short Arabic poem that adhered to the following conditions:

- **Title:** *شكر لجامعة الملك سعود*
- **Meter (البحر):** *الطويل*
- **Rhyme (القافية):** The poem must end each line with the letter "د" (الدال)
- **Language Style:** Classical Arabic, with appropriate depth and formal structure

Each poem was evaluated across three key dimensions:

1. **Meaning (المعنى):** The expressiveness, coherence, and relevance of the poem's content
2. **Meter (الوزن):** Adherence to the metrical rules of *بحر الطويل*
3. **Rhyme (القافية):** Consistency and correctness in maintaining the required end rhyme

The evaluation was conducted using a 5-point scale, where 5 represents excellent performance and 1 indicates poor quality. In addition to numerical scores, reviewers could also provide optional narrative feedback that highlighted notable stylistic or technical points. This human-centered evaluation approach serves as a bridge between technical advancement and artistic fidelity, ensuring that our models are judged not only by what they generate but by how well they resonate with the rich tradition of Arabic poetry. Examples of generated poems are shown in **Error! Reference source not found.**.

Table 1: Sample poems generated by the four evaluated models.

Model	Generated Poem
ALLaM	شكراً لجامعة سعود درب المجد طريق التقدم يا سعود لك استمد من العلم قد أحييت أرضك بالندى ومن العلم قد أحييت أرضك بالجدد
Silma	جامعة سعود في نجد، فخرها يزداد في الشرق والغرب، تظل رمز الجداد ما من مكان في العالم إلا وهي حماة كالنجم في السماء، يضيء بالنجاد فصاحتها تعانق الأفق، في كل بلاد ومدائحها تتردد، بكل لسان وفؤاد منها العلم ينبع، ويمتد في كل مداد في نجد، جامعة سعود، فخر الأمجاد
ChatGPT	سعيت للعلا والمجد يا خير مقصد وسرت بعلم الناس تزْهين بالرشد رفعت منار الحق فوق ذرا العلا وغرست في الأرواح نورًا من السعد إذا ذكر التعليم كنت أولًا وصوتك بين الخلق أبهى من الغد لك الحمد يا دار العلوم ومهدها تفيضين علمًا في السهول وفي النجد
Claude	أيا صرح علم قد سمت بفضلكم وأشرقت نورًا في ربوع البوادي تنير عقولًا بالمعارف والنُهى وتبني رجالًا للغد المتجدد بك العلم يزهو والثقافة تزدهي وتسمو بك الآمال نحو السؤدد فكم أنجبت أجيال علم مضيئة وكم أطلقت أفكار فكر مسدد ستبقى منارًا شامخًا في بلادنا يُشِع بمجد راسخ متجدد فدُمت لأوطان العروبة مفخرًا ودام عطاؤك الزاكي على مر الأبد

7.4 Evaluation Methodology and Results

Evaluating the quality of generated poetry, especially in a language as rich and complex as Arabic with its stringent traditional forms, requires a multi-faceted approach. Relying solely on automated metrics designed for general text generation (like BLEU, ROUGE, which measure similarity to reference texts but often fail to capture creativity, nuance, meter, or rhyme) is insufficient. Therefore, our evaluation combined automated assessment of specific quantifiable linguistic features with a comprehensive human evaluation conducted by experts in Arabic poetry. This hybrid approach allowed us to assess both technical aspects and the subjective artistic and cultural fidelity of the generated verses.

7.3.1 Automated Evaluation: Type-Token Ratio (TTR)

Lexical diversity is an important characteristic of high-quality text, including poetry. A varied vocabulary can contribute to the richness, complexity, and aesthetic appeal of a

poem, avoiding repetition (unless intentional for effect). The Type-Token Ratio (TTR) is a simple yet useful metric for quantifying this aspect of language. In fact, Type-Token Ratio (TTR) is calculated as follows:

$$TTR = (Number\ of\ Unique\ Words\ (Types)\ /\ Total\ Number\ of\ Words\ (Tokens)) \times 100 \quad (1)$$

As it can be noticed, TTR is obtained dividing the number of unique words (types) in a text by the total number of words (tokens) and often multiplying by 100 to get a percentage A higher TTR generally indicates greater lexical diversity.

Relevance to Poetry: In Arabic poetry, while certain words or phrases might be repeated for emphasis or rhythm, a generally rich and varied vocabulary is desirable. Comparing the TTR of different models' generated poems provides an objective measure of how well they capture this aspect of linguistic richness relative to each other.

Methodology: We calculated the TTR for a sample of poems generated by each of the evaluated models: Silma, ALLaM, ChatGPT, and Claude. For consistency across poems of varying lengths, we used a sliding window approach (e.g., calculating TTR for every 100 tokens and averaging, or calculating per complete generated poem and averaging).

Table 2: Average Type-Token Ratio (TTR) for the Generated Poetry Samples.

Source	Average TTR (%)
Silma	81.63
ALLaM	78.26
ChatGPT	93.48
Claude	94.83

As shown in **Error! Reference source not found.**, Claude generated poetry with the highest average TTR at 94.83%, followed closely by ChatGPT at 93.48%. Silma exhibited a TTR of 81.63%, while ALLaM had the lowest TTR among the evaluated models at 78.26%. These results suggest that general-purpose models like ChatGPT and Claude generated text with significantly higher lexical diversity compared to models potentially more focused on specific Arabic poetic structures like Silma and ALLaM within the evaluated samples. The lower TTR for ALLaM and Silma might imply they tended to

repeat words more or use a more constrained vocabulary, potentially due to focusing on structural adherence (meter and rhyme), which can sometimes limit word choice compared to generating free-form text. Conversely, the high TTR for ChatGPT and Claude might reflect their training on vast, diverse general text corpora, allowing them to produce a wider range of vocabulary when attempting poetic output, even if this diversity doesn't necessarily translate to correct poetic structure.

It is important to note that TTR is a measure of diversity, not quality or appropriateness of vocabulary within the poetic context. A high TTR doesn't guarantee good poetry, just a wider range of words. It does not evaluate adherence to meter, rhyme, meaning, or cultural context, which are critical for Arabic poetry. Therefore, TTR serves as a supplementary metric to the more crucial human evaluation.

7.3.2 Expert Evaluation

Given the highly subjective and culturally embedded nature of Arabic poetry, expert human evaluation was the primary method for assessing the quality and authenticity of the poems generated. This approach allows for a nuanced judgment of aspects that automated metrics cannot reliably capture, such as artistic merit, emotional depth, cultural resonance, and the subtle correctness of meter and rhyme within the complex rules of Arabic prosody.

Our human evaluation was conducted by a panel of distinguished experts in Arabic poetry and literature. This panel consisted of two practicing Arabic poets and one professor of Arabic literature at King Saud University specializing in classical Arabic poetics and literary interpretation. The evaluation process involved presenting a set of poems generated by the evaluated LLMs (Silma, ALLaM, ChatGPT, and Claude) to each expert via a structured form. To ensure impartiality, the poems were anonymized, meaning the evaluators were not informed which model generated which poem. Each expert was asked to independently assess the poems based on a predefined set of criteria.

The Experts rated each poem based on the following criteria, designed to capture the essential elements of high-quality Arabic poetry:

- Poetic Meaning (المعنى): Assessed the depth, clarity, symbolism, emotional impact, and overall coherence of the ideas conveyed in the poem. (Rated on a scale from 1: Weak to 5: Excellent).

- Metrical Accuracy (الوزن): Evaluated the adherence of the verses to the rules of the specified Arabic meter (Bahr), focusing on the rhythmic patterns and syllabic balance (*Taf'eelat*). Given the focus in our experiments and examples (Figure 24, which likely represents output after fine-tuning), particular attention was paid to the *al-bahr al-taweel* if specified or attempted by the model. (Rated on a scale from 1: Weak to 5: Excellent).
- Rhyme Consistency (القافية): Checked if the end-rhyme (*Qafiyah*) pattern was consistent and correct throughout the poem, adhering to Arabic rhyme rules (Muqayyad or Mutlaq, depending on the poem's structure). (Rated on a scale from 1: Weak to 5: Excellent).
- Free Feedback: Experts were encouraged to provide open-ended comments on any aspect of the poem, including strengths, weaknesses, originality, grammatical errors, awkward phrasing, or specific observations regarding meter and rhyme, as well as overall poetic authenticity and artistic feel.

The scores and feedback from the expert panel provided crucial insights into the performance of the different models. The average scores for each model across the key criteria are presented in **Error! Reference source not found.** (on a scale of 1 to 5).

Table 3: Average Human Evaluation Scores per Criterion (Scale 1-5).

Criterion	Silma	ALLaM	ChatGPT	Claude
Poetic Meaning (المعنى)	2.0	2.5	3.25	4.0
Metrical Accuracy (الوزن)	1.0	2.0	2.75	2.75
Rhyme Consistency (القافية)	2.25	2.5	3.25	2.0

Overall Performance: Based on the averaged scores, ChatGPT shows the highest performance in Rhyme Consistency and strong performance in Meaning and Meter. Claude excels significantly in Poetic Meaning and performs comparably to ChatGPT in Meter. ALLaM and Silma struggled more with Metrical Accuracy and Poetic Meaning compared to

ChatGPT and Claude, although ALLaM performed comparably to Claude in Rhyme Consistency and better than Silma overall.

Poetic Meaning: Claude received the highest average score for Poetic Meaning (4.0), with experts frequently rating its verses as coherent, insightful, and emotionally impactful. ChatGPT also scored well in Meaning (3.25). ALLaM (2.5) and Silma (2.0) lagged behind, suggesting their potential focus on structural elements may have sometimes come at the expense of semantic depth or natural expression compared to the more general-purpose models.

Metrical Accuracy: ChatGPT and Claude achieved the highest scores for Metrical Accuracy (both 2.75). While still not perfect, these models demonstrated a better ability to produce verses that approximate traditional meter compared to ALLaM (2.0) and especially Silma (1.0), which received a consistently "ضعيف" rating from evaluators on this criterion. This indicates significant challenges remain in fully mastering Arabic prosody for all models, but ChatGPT and Claude showed a relative advantage in our evaluation of these specific generated poems.

Rhyme Consistency: ChatGPT scored highest in Rhyme Consistency (3.25), suggesting it was most successful in generating verses with correct and consistent end-rhymes. ALLaM also performed reasonably well (2.5), comparable to Claude (2.0) and slightly better than Silma (2.25). This indicates that while rhyme is a challenge, some models are more adept at it than others, with ChatGPT leading in this specific evaluation.

Insights from Free Feedback: The free feedback provided additional context. Experts noted that while ChatGPT and Claude sometimes produced more fluent or meaningful lines, they often failed spectacularly on meter and rhyme, making the output inconsistent as poetry (reinforcing their lower structural scores despite higher TTR). Conversely, ALLaM and Silma, despite lower meaning scores, sometimes received comments about attempts at structure, though frequently marred by inaccuracies, particularly in meter for Silma. One evaluator noted for Claude (Poem 4) that "the style was consistent from beginning to end," suggesting a better overall structural flow compared to others, despite its average meter/rhyme scores. Another commented on Silma's (Poem 2) rhyme structure having

specific adherence (تأسيس وإعراب), which is a positive note despite the low overall structural scores.

Consistently, the human evaluation results clearly indicate that general-purpose models like ChatGPT and Claude can sometimes produce more fluent and meaningful Arabic text with higher lexical diversity (as shown by TTR), but they still struggle significantly with the strict structural requirements of traditional Arabic poetry (meter and rhyme). Models potentially adapted for Arabic poetry generation, such as ALLaM and Silma (depending on your experimental approach), showed varied performance, with ALLaM showing better structural scores than Silma, and ChatGPT unexpectedly performing best in Rhyme Consistency in this specific evaluation set. Achieving high quality across meaning, meter, and rhyme simultaneously remains a significant challenge for all evaluated models. While invaluable, human evaluation is subjective. The panel size was limited to three experts. However, the consistency in their feedback and scores across multiple poems strengthens the validity of these findings and highlights the specific areas where current LLMs fall short in generating culturally authentic and structurally sound Arabic poetry.

Besides, the success of this project depended heavily on the availability and efficiency of computing resources. In particular, fine-tuning large language models such as LLaMA, Allam, and Silma required high-end GPUs (e.g., A100, V100, or 3090+). For certain experiments, especially those involving instruction tuning or RAG pipelines, the VRAM usage exceeded 24 GB per model instance. Moreover, the training and inference were carried out using cloud-based environments (e.g., Google Colab Pro, Kaggle, and Hugging Face spaces) and, in some cases, access to local university HPC servers. However, hardware constraints limited the batch size, sequence length, and number of training epochs. Furthermore, in order to reduce latency and VRAM requirements during evaluation and deployment, quantization techniques (e.g., 4-bit/8-bit inference using bitsandbytes) were explored, especially for Silma and Allam models. Besides, running six distinct versions (including ChatGPT API, Claude, Silma, and hybrid enhanced outputs) for human evaluation required careful management of compute hours and API usage quotas.

These constraints highlight the need for optimized model selection, training strategies, and access to scalable infrastructure in future work. Consequently, this research can be extended to develop more powerful, efficient, and poetically authentic Arabic text generation systems.

Chapter 8: Conclusion and Future Work

This research focused on exploring the potential of advanced AI techniques for generating Arabic poetry that adheres to traditional poetic forms. By examining methodologies such as fine-tuning large language models, retrieval-augmented generation (RAG), and the RWKV architecture, the study aimed to address the unique challenges posed by Arabic poetry, including its complex structure, rhythmic patterns, and cultural significance.

The project proposed leveraging a carefully curated dataset, such as the Ashaar dataset, enriched with annotations for meter (al-bahr), rhyme (al-qafiya), and thematic classifications. These annotations were intended to support models in understanding and replicating the intricate rules and stylistic elements of Arabic poetry. By integrating fine-tuning to capture structural conventions, RAG for contextual relevance, and RWKV for long-term coherence, the study envisioned generating poetry that aligns with traditional standards while maintaining fluency and thematic consistency.

This research laid the groundwork for future efforts in combining AI and Arabic literature, emphasizing the potential of advanced models to contribute to the preservation and modern accessibility of Arabic poetry. The proposed approaches offer a promising direction for further development and experimentation to achieve high-quality, culturally authentic poetic generation.

To build upon the foundation established in this study, several directions for future research and improvement are proposed. Specifically, one can incorporate Arabic diacritization models during generation or as a post-processing step to enhance pronunciation accuracy and rhythmic adherence. In addition, LLMs can be combined with symbolic rules or meter-checking modules to enforce strict poetic structures (e.g., `Al-Khalil's meters`) during generation. Moreover, expert poet evaluations can be used to guide the training of models via RLHF, aligning outputs with subjective artistic judgment. Besides, The considered dataset can be extended to include Arabic dialects and translated poetic works, allowing for broader stylistic coverage and cultural nuance. One should also note that scaling future experiments will require dedicated access to multi-GPU clusters or inference servers with

distributed generation capabilities. Efficient training paradigms like LoRA or QLoRA should be further explored to reduce memory and compute footprint.

References

[1] "Al-Khalil_ibn_Ahmad_al-Farahidi," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Al-Khalil_ibn_Ahmad_al-Farahidi.

[2] "Arabic_prosody," [Online]. Available: https://en.wikipedia.org/wiki/Arabic_prosody.

[3] "al-Khalil-ibn-Ahmad," [Online]. Available: https://www.britannica.com/biography/al-Khalil-ibn-Ahmad.

[4] "alchetron," [Online]. Available: https://alchetron.com/Al-Khalil-ibn-Ahmad-al-Farahidi.

[5] "2030 Vision," [Online]. Available: https://www.vision2030.gov.sa/.

[6] "Challenges," [Online]. Available: https://www.researchgate.net/publication/341560819_Challenges_in_AI_Generated_Arabic_Poetry.

[7] "Al-Khalil_ibn_Ahmad_al-Farahidi," [Online]. Available: https://en.wikipedia.org/wiki/Al-Khalil_ibn_Ahmad_al-Farahidi.

[8] "JAISS," [Online]. Available: https://arxiv.org/abs/2308.16149.

[9] "ALLAM," [Online]. Available: https://arxiv.org/html/2407.15390v1.

[10] Z. A. . M. S. A.-S. and M. , "Ashaar: Automatic Analysis and Generation of Arabic Poetry Using Deep," 2023.

[11] "aldiwan," [Online]. Available: https://www.aldiwan.net.

[12] "Year-Of-Arabic-Poetry," [Online]. Available: https://www.moc.gov.sa/en/Modules/Pages/Cultural-Years/Year-Of-Arabic-Poetry.

[13] "Prince_of_Poets," [Online]. Available: https://en.wikipedia.org/wiki/Prince_of_Poets.

[14] "Million_Poet," [Online]. Available: https://en.wikipedia.org/wiki/Million%27s_Poet.

[15] "شعر عمودي," [Online]. Available: https://ar.wikipedia.org/wiki/%D8%B4%D8%B9%D8%B1_(%D8%A3%D8%AF%D8%A8).

[16] a. poem. [Online]. Available: https://www.aldiwan.net/poem50.html.

[17] "شعر حر," [Online]. Available: https://ar.wikipedia.org/wiki/%D8%B4%D8%B9%D8%B1_%D8%AD%D8%B1.

[18] "Meter," [Online]. Available: https://ar.wikipedia.org/wiki/%D8%A8%D8%AD%D8%B1_(%D8%B4%D8%B9%D8%B1).

[19] "Tafe'lah," [Online]. Available: https://ar.wikipedia.org/wiki/%D8%AA%D9%81%D8%B9%D9%8A%D9%84%D8%A9.

[20] "Taweel Bahr," [Online]. Available: https://ar.wikipedia.org/wiki/%D8%A8%D8%AD%D8%B1_%D8%A7%D9%84%D8%B7%D9%88%D9%8A%D9%84.

[21] "Wazn," [Online]. Available: https://ar.wikipedia.org/wiki/%D9%88%D8%B2%D9%86_(%D8%B4%D8%B9%D8%B1).

[22] "Rhyme," [Online]. Available: https://ar.wikipedia.org/wiki/%D9%82%D8%A7%D9%81%D9%8A%D8%A9.

[23] G. L. G. a. I. —. S. For Loy, "Risk and benedits of AI," in *Artificial Intelligence modern approach*, Stuart Russel Peter Norving, 2010, pp. 5-28.

[24] A. Géron, "PartI," in *Hands-On_Machine_Learning_with_Scikit-Learn-Keras-and-TensorFlow-2nd-Edition-Aurelien-Geron*, O.REALLY, 2019, pp. 15-22.

[25] "DL image," [Online]. Available: https://miro.medium.com/v2/resize:fit:1400/1*ZXAOUqmlyECgfVa81Sr6Ew.png.

[26] "RNN image," [Online]. Available: https://python.plainenglish.io/introducing-gru-rnn-and-lstm-a-beginners-guide-to-understanding-these-revolutionary-deep-35b509a34a5a.

[27] R. Xu-Darme, G. Q. and . Z. C. , "Sanity checks and improvements for patch visualisation in prototype-based image classification," 2023.

[28] A. V. . N. S. and N. P. a. , "Attention Is All You need," 2017.

[29] J. D. M.-W. C. K. L. and K. , "BERT paper," arxiv, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805.

[30] "GPT," [Online]. Available: https://arxiv.org/abs/2305.10435.

[31] "Toward Fluent Arabic Poem Generation Based on Fine-tuning AraGPT2 Transformer | Arabian Journal for Science and Engineering.," [Online]. Available: https://link.springer.com/article/10.1007/s13369-023-07692-1.

[32] W. A. . F. B. and H. H. , "AraGPT2: Pre-Trained Transformer for Arabic Language Generation," [Online]. Available: https://arxiv.org/abs/2012.15520.

[33] J.-S. Lee, "https://arxiv.org/abs/2306.05431," [Online]. Available: https://arxiv.org/abs/2306.05431.

[34] A. H. R. A. and M. A. , "Arabic Poems Generation using LSTM, Markov-LSTM and Pre-Trained GPT-2 Models," 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244231535.

[35] M. Ghazvininejad, X. Shi, J. Priyadarshi and K. Knight, "Hafez: an Interactive Poetry Generation System," in *Proceedings of ACL 2017, System Demonstrations*, M. Bansal and H. Ji, Eds., Vancouver, Association for Computational Linguistics, 2017, pp. 43-48.

[36] V. d. C. and T. , "Automatic Poetry Generation from Prosaic Text," [Online]. Available: https://aclanthology.org/2020.acl-main.223.

[37] "Introducing Aspects of Creativity in Automatic Poetry Generation," 2020. [Online]. Available: https://arxiv.org/abs/2002.02511.

[38] Y. X. . S. M. L. R. and . L. W. , "Automatic Poetry Generation with Mutual Reinforcement Learning," [Online]. Available: https://aclanthology.org/D18-1353.

[39] T. S. and R. B. , "Poetry Generation Model via Deep learning incorporating Extended Phonetic and Semantic Embeddings," 2021. [Online]. Available: 10.1109/ICSC50631.2021.00013.

[40] E. N. . A.-E. M. E. M. and A. M. , "Generating Classical Arabic Poetry using Pre-trained Models," 2022. [Online]. Available: https://aclanthology.org/2022.wanlp-1.6.

[41] E. Ali Refaee, "OKAZ: A Deep-learning-Based System for Automatic Arabic Poem Generation," 2023. [Online]. Available: 10.1109/ICCIT58132.2023.10273934.

[42] S. Talafha and B. Rekabdar, "Arabic Poem Generation with Hierarchical Recurrent Attentional Network".

[43] M. Beheitt and . M. Hmida, "Automatic Arabic Poem Generation with GPT-2".

[44] B. P. a. E. A. a. Q. A. a. A. A. a. S. A. a. S. B. a. H. C. a. X. C. a. M. C. a. M. G. a. K. K. G. a. X. H. a. H. H. a. J. L. a. P. Kazienk, "RWKV: Reinventing RNNs for the Transformer Era," [Online]. Available: https://arxiv.org/abs/2305.13048.

[45] "RAG Redis," [Online]. Available: https://redis.io/glossary/retrieval-augmented-generation/.

[46] "The Llama 3 Herd of Models," [Online]. Available: https://arxiv.org/abs/2407.21783.

[47] S. B. Y. C. Z. C. K. D. X. D. Y. F. W. G. Y. H. F. H. B. H. L. J. M. L. J. L. R. L. D. L. G. L. C. L. K. L. J. M. R. M. X. R. X. Jinze Bai, "Qwen Technical Report," [Online]. Available: https://arxiv.org/pdf/2309.16609.

[48] "RAG prompting guide," [Online]. Available: https://www.promptingguide.ai/research/rag.

[49] "Intro to RAG," [Online]. Available: https://ingestai.io/blog/introduction-to-rag.

[50] V. B. P. . A. Z. . A. K. and A. S. , "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities," [Online]. Available: The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities.

[51] J. B. a. S. B. a. Y. C. a. Z. C. a. K. D. a. X. D. a. Y. F. a. W. G. a. Y. H. a. F. H. a. B. H. a. L. J. a. M. L. a. J. L. a. R. L. a. D. L. a. G. L. a. C. L. a. K, "Qwen Technical Report," 2023. [Online]. Available: https://arxiv.org/abs/2309.16609.

[52] "Hugging face," Ashaar dataset, 2022. [Online]. Available: https://huggingface.co/datasets/arbml/ashaar.

[53] Z. Yang, "Restore-RWKV: Efficient and Effective Medical Image Restoration with RWKV," 7 2024. [Online]. Available: https://arxiv.org/html/2404.09516v1.

[54] "Generating Classical Arabic Poetry using Pre-trained Models - ACL Anthology.," [Online]. Available: https://aclanthology.org/2022.wanlp-1.6/.

[55] google, "csturn cloud," [Online]. Available: https://saturncloud.io/blog/whats-the-hardware-spec-for-google-colaboratory/.

[56] "supervised Learning," [Online]. Available: https://medium.datadriveninvestor.com/exploring-the-basics-of-supervised-learning-bfc18fe5ce72.

[57] :q.

[58] A. Hakami, R. Alqarni, M. Almutairi and A. Alhothali, "Arabic Poems Generation using LSTM, Markov-LSTM and Pre-Trained GPT-2 Models," in *3rd International Conference on Machinen Learning & Applications*, 2021.

[59] J. H. a. S. Ruder, "Universal Language Model Fine-tuning for Text Classification," 2018. [Online]. Available: https://arxiv.org/abs/1801.06146.

[60] C. R. a. N. S. a. A. R. a. K. L. a. S. N. a. M. M. a. Y. Z. a. W. L. a. P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683.