Brightskies Inc.

Documents

Allam Challenge 2024 – Documentation

Prepared By: Marwan ElSafty, Ayman ElGhotni, Hossam ElKady & Islam Bassuni

Content

Challenge Overview

The Allam Challenge 2024 is an initiative organized by the Saudi Data and Artificial Intelligence Authority (SDAIA) to drive innovation and enhance Arabic language technology using Large Language Models (LLMs). The challenge seeks to identify and cultivate exceptional talent in artificial intelligence, particularly in the domain of Arabic language applications. Through this competition, SDAIA aims to inspire participants to develop creative solutions that improve and expand the capabilities of LLMs, with a specific focus on the Arabic language.

Goals of the Challenge:

- **Fostering Talent**: One of the primary objectives of the Allam Challenge is to discover and nurture AI talent, providing participants with the opportunity to develop their skills and turn their innovative ideas into practical, implementable projects.
- **Enhancing Arabic Language Models**: The challenge emphasizes improving the performance of LLMs, specifically designed to cater to the complexities of the Arabic language, including poetry generation, grammar parsing, and conversational capabilities with historical figures.
- **Promoting Collaboration**: Participants are encouraged to work collaboratively in teams, blending technical and creative expertise to deliver impactful solutions.

Key Features:

- **Workshops and Mentorship**: Participants benefit from workshops led by experts in AI, data science, and Arabic literature, providing valuable guidance and insights. The challenge also offers mentorship to help teams refine their projects and bring their ideas to fruition.
- **Support for Execution**: SDAIA provides a conducive environment for innovation by offering resources and guidance to help teams navigate the development process, including support for startup-level execution.

Competition Phases:

1. **Preliminary Phase (Remote)**: Teams work on their initial ideas remotely, developing their projects in various fields related to the Arabic language, such as poetry generation, grammar parsing, and AI-driven language learning tools.
2. **Final Phase (On-site in Riyadh)**: The final phase will take place between November 7-9, 2024, where participants will present their solutions to a panel of judges. During this period, they will also engage in collaborative activities and present their final submissions.

Judging Criteria:

The challenge is judged based on several key aspects:

- **Technical Implementation (30%)**: The depth and sophistication of the technical solution.
- **Idea-Solution Fit (20%)**: The alignment between the problem being addressed and the proposed solution.
- **User Experience (15%)**: How intuitive and user-friendly the solution is.
- **Evaluation and Accuracy (20%)**: The precision and effectiveness of the LLM-based solutions.
- **Presentation and Documentation (10%)**: The clarity and professionalism of the project presentation and supporting materials.
- **Ethics, Responsible AI, Teamwork & Collaboration (5%)**: Emphasis on ethical practices and collaboration among team members.

Awards and Prizes:

The competition offers significant rewards, with the top three teams receiving financial prizes to further support the development and implementation of their solutions. The total prize pool includes:

- **1st Place**: SAR 500,000
- **2nd Place**: SAR 300,000
- **3rd Place**: SAR 200,000

This challenge is not only a platform to showcase AI and language model innovation but also a significant step toward advancing the application of AI technologies in Arabic language services.

Proposed Idea

The solution we are presenting for the Allam Challenge is a comprehensive AI-powered platform dedicated to **Arabic poetry analysis, generation, and interactive poetic debates**. Our platform harnesses advanced language models to create a multifaceted tool that addresses various aspects of Arabic poetry, from critical analysis to creative generation and even engaging in poetic debates. The platform serves as a powerful tool for students, researchers, literature enthusiasts, and educators, elevating their understanding and engagement with Arabic poetry.

1. Poetry Analysis and Artistic Criticism

One of the core features of our platform is the **poetry analysis and artistic critique tool**, powered primarily by the **ALLAM LLM**—a state-of-the-art Arabic language model. This model is specifically designed for understanding and processing the nuances of the Arabic language. The platform can:

- **Interpret Poetic Nuances**: The ALLAM model, with its in-depth training on Arabic texts, can analyze the subtleties of Arabic poetry, including metaphors, similes, and cultural references.
- **Cultural and Historical References**: Leveraging the vast linguistic and cultural knowledge embedded in ALLAM, the model can recognize historical and literary allusions within Arabic poetry, providing a rich and contextual analysis.
- **Artistic Criticism**: Beyond literal interpretation, the model can deliver a comprehensive artistic critique, evaluating thematic depth, rhetorical techniques, and stylistic devices, which is particularly beneficial for students and scholars.
- **Supplementary NLP Models**: Small NLP models are also employed for tasks like classification for poetic meters and the rhyme (Qafya), ensuring that all aspects of the poetry are critically analyzed.

For example, if a user inputs a classical poem by Al-Mutanabbi, the model will not only interpret its literal meaning but also analyze the deeper themes, such as power and pride, while offering a critique of its stylistic and rhetorical elements.

2. Poetry Generation

Our platform also features an innovative **Poetry Generation** tool, driven by the ALLAM LLM:

- **Poet Selection**: Users can choose from a variety of well-known Arabic poets, both classical and modern. The ALLAM model will mimic the selected poet's style of writing as if the poet was writing the verse themselves, while adhering to the topic and the rhyme.
- **Topic Selection**: Users input a topic, and the model generates poetry that adheres to this theme while maintaining the poet's unique stylistic elements.
- **Maintaining Rhyme and Structure**: One of the key capabilities of few-shot prompting with ALLAM is its ability to generate poetry that follows traditional Arabic rhyme schemes and meters, ensuring that the output reflects the rich linguistic structure of Arabic poetry.

For example, if a user selects Nizar Qabbani and the topic of love, the model will generate verses that mirror Qabbani's emotional intensity and modern stylistic nuances, all while adhering to the chosen rhyme and thematic constraints.

3. Poetic Debate (Poetic Battles)

The **Poetic Debate** feature is the most innovative aspect of our solution. This feature simulates an interactive poetic debate between two well-known poets on a given topic. ALLAM serves as the primary engine for generating poetry during these debates:

- **How it Works**: Users can choose two poets from the poets list of classical and moderns poets and a topic of interest. The poets then engage in a turn-based exchange of verses, each responding in their own style, powered by ALLAM mimicking the poets' unique styles.
- **Turn-Based Verse Generation**: Each poet generates a verse in response to the other, creating a rich, dynamic exchange that reflects both poets' styles.
- **Judgement and Analysis**: Once the debate concludes, a virtual judge analyzes the generated poetry. The judge, leveraging supplementary NLP models for tasks like rhyme checking and rhetorical analysis, evaluates each poet's adherence to the topic, poetic quality, and rhetorical mastery.
- **Winner Selection**: The judge then selects a winner based on criteria such as thematic depth, creativity, and the quality of rhetorical devices used.

For instance, a debate between Al-Mutanabbi and Nizar Qabbani on the theme of "resistance" would showcase two distinct poetic approaches. Al-Mutanabbi's grandiose verses would be contrasted with Qabbani's modern, intimate reflections, with the judge ultimately deciding which poet delivered the most compelling performance.

4. Use Cases and Impact

- **Educational Tool**: This solution serves as a comprehensive educational tool for students and instructors alike, particularly in academic settings where the analysis and debate features can be used to demonstrate and study the intricacies of Arabic poetry.
- **Research**: Researchers can explore the evolution of Arabic poetic styles, using the platform to analyze different periods and themes in Arabic literature.
- **Creative Tool for Writers**: Poets and writers can use the platform's poetry generation feature as a source of inspiration or as a collaborative tool to explore new creative directions.
- **Entertainment**: The poetic debate feature offers an engaging, interactive experience for users who wish to see legendary poets "battle" on a creative and intellectual level.

Technology Stack

Our solution is primarily powered by **ALLAM**, a leading LLM designed specifically for the Arabic language. ALLAM provides the core capabilities for generating and analyzing Arabic poetry, ensuring that the platform is able to handle the unique challenges posed by the language's structure and poetic traditions. In addition, **small NLP models** are used for supplementary tasks such as:

- **Meter Classification**: To ensure that the generated poetry follows traditional Arabic meter rules, particularly important in classical poetry forms.
- **Rhyme Classification:** Categorizing the output based on its rhyme scheme and its adherence to poetic conventions.

By combining the power of Prompt Engineering with ALLAM in addition to supplementary NLP models, our platform offers an unparalleled experience in Arabic poetry creation, analysis, and interactive engagement.

Methodology

Our proposed solution for the Allam Challenge 2024 is an advanced AI-powered platform designed for analyzing, generating, and debating Arabic poetry. The architecture and implementation are based on several modern techniques in AI, natural language processing (NLP), microservice architecture, and containerization. This section outlines the methodology, including the workflows, techniques, trained models, and system components used to develop the platform.

1. Agentic Workflow and Multi-Agent System

Some of our platform's supported features rely heavily on **agentic workflows**, where multiple agents collaborate under a shared environment. Each agent has a defined role and operates independently but communicates with other agents to achieve the platform's overall objectives. The key agents include:

- **Poetry Analysis Agent**: Specializes in analyzing poetry by breaking down its structure, identifying rhetorical elements, and providing in-depth artistic critique. This agent utilizes small NLP models for tasks like **meter classification** and **qafya detection** (rhyme). In addition to detecting and analyzing rhetorical devices such as metaphors, similes, and allusions within poems.
- **Poetry Generation Agent**: Responsible for generating poetry in the style of a specified poet based on a given topic. This agent leverages the **ALLAM LLM** for language generation and performs **similarity searches** to retrieve relevant poems, which are used in **few-shot learning** to better emulate the poet's style.
- **Poetry Judge Agent**: Evaluates the poetic quality of generated verses, particularly in **poetic debates**. This agent checks adherence to rhyme, thematic relevance, and rhetorical quality to declare a winner.

Each agent follows a **shared interface**, allowing modularity and ease of extension. The system is designed to scale, where new agents can be introduced to support additional features without disrupting the overall architecture.

2. Retrieval-Augmented Generation (RAG) and Few-Shot Prompting

To enhance the model's performance in generating accurate and contextually relevant outputs, we implemented a **Retrieval-Augmented Generation (RAG)** approach:

- **RAG**: This approach allows agents to retrieve supplementary data from a **datastore** (powered by the Ashaar dataset) to provide context to the **ALLAM LLM**. This improves the accuracy of poetry generation, analysis, and critique by feeding relevant historical and cultural data into the model.

- **Few-Shot Prompting**: To mimic the style of specific poets, the Poetry Generation Agent performs a **similarity search** on the retrieved poems. By feeding a few relevant examples into the LLM, the model learns the poet's distinctive voice and style, generating new verses that align closely with the selected topic.

3. Ashaar Dataset Integration

The **Ashaar dataset**, a vast collection of Arabic poems available on Hugging Face, serves as the foundation for our model's context:

- **Dataset Statistics**:
  - Number of poems: 254,630
  - Number of baits (verses): 3,857,429
  - Number of poets: 7,167 This dataset enables the model to access a wealth of poetic works, enriching both the generation and analysis features by providing examples, structure, and stylistic references.

4. Trained Models

As part of our platform, we trained several models to handle specific tasks related to poetry analysis and generation, one of the most crucial being the **Arabic Poetic Meter Classification Model**:

- **Arabic Poetic Meter Classification Model**: A transformer-based model that classifies Arabic poetic meters (Bahr). It predicts which of the 17 traditional Arabic meters a given verse belongs to. The architecture includes:
  - **Embedding Layer**: Combining token and positional embeddings to capture the sequence of text.
  - **Transformer Block**: Multi-head self-attention with 3 attention heads, bidirectional GRU layers, and dense layers for processing the input.
  - **Final Layers**: Outputting the predicted meter using a softmax layer for classification.

  The meter classification model enhances the **Poetry Analysis Agent** by providing structured analysis of poetic meters, which is critical for evaluating and generating poetry that conforms to traditional Arabic forms.

- **Qafya Detection with Qawafi** :قوافي Our platform leverages Qawafi's ready-to-use **Qafya detection model** to identify and analyze rhyme schemes in Arabic poetry. This model specializes in detecting rhyme patterns (Qafya) within verses, ensuring that generated and analyzed poetry aligns with traditional Arabic poetic conventions.

By integrating Qawafi's Qafya detection, the platform provides precise rhyme classification, which is essential for maintaining the structural and aesthetic integrity of Arabic verse. This feature significantly enhances our poetry generation and analysis capabilities by ensuring adherence to classical Arabic rhyme patterns, adding an authentic layer to the poetic experience.

5. LangChain and ALLAM Integration

To streamline the interaction between agents and the ALLAM LLM, we utilized the **LangChain platform**:

- **LangChain Platform**: By using LangChain's components, particularly l**angchain_ibm** and **ibm_watsonx_ai**, we can seamlessly access the ALLAM LLM on IBM Watson. LangChain also simplifies the implementation of **RAGs**, allowing agents to efficiently retrieve data from the Ashaar dataset and feed it into the model during poetry generation and analysis.
- **Agent-Oriented Framework**: Each agent communicates with LangChain components for task-specific NLP processes, such as meter classification, rhyme detection, and rhetorical analysis.

6. Microservice Architecture

Our application follows a **microservice architecture**, splitting the three main functionalities into separate services:

- **Poetic Debate Service**: Handles all requests related to the interactive poetic debate feature, where users select poets and a topic, and the system generates a turn-based poetic exchange.
- **Poetry Analysis Service**: Manages poetry analysis tasks, including structural breakdown, rhetorical device detection, and meter classification.
- **Poetry Generation Service**: Processes requests for poetry generation, leveraging ALLAM LLM to create original poems in the style of a chosen poet.

Each microservice is hosted on a separate server using **FastAPI**, providing fast response times and clear separation of concerns. This setup allows for easier scaling, extensibility, and portability across different environments.

7. Containerization and Orchestration

To ensure that the platform can run efficiently in different environments without complex setup, we employed **containerization**:

- **Docker**: Each microservice is containerized using Docker, isolating dependencies and ensuring consistency across development, testing, and production environments. This also simplifies deployment and scaling of individual components.
- **Docker Compose**: The entire system is managed using Docker Compose, which integrates all the microservices and agents into a single, cohesive environment. This allows for streamlined orchestration of services and agents, enabling the platform to function as a unified application.

8. Frontend Development

The **frontend** of the application is designed to offer a seamless and interactive user experience. Built using **React**, the frontend allows users to:

- Select poets and topics for poetry generation.
- Initiate poetic debates and view real-time verse exchanges.
- Request detailed poetry analysis and receive feedback in an easily understandable format.

Special attention was given to **UI/UX design**, with a focus on color coordination, easily recognizable buttons, and intuitive navigation. This ensures that users of all levels can easily interact with the platform's powerful features.

9. Integration and Final Packaging

To integrate the backend and frontend, we connected the microservices to the React frontend using **API endpoints** for each server. This integration allows users to:

- Post requests to different endpoints (e.g., poetic debate, poetry generation) and receive results in real time.
- Experience a seamless transition between features, from generation to analysis to interactive debates.

The final application is wrapped using **Docker Compose**, bringing together all the Docker containers into a single environment. This ensures that the entire platform can be deployed effortlessly across different systems, maintaining consistency and reducing deployment complexities.

Conclusion

Our methodology encapsulates the latest advancements in AI and NLP, focusing on agentic workflows, RAG, few-shot prompting, microservices, and containerization. The system is designed to be scalable, extensible, and portable, ensuring that it can evolve with future needs. By integrating both backend efficiency and a user-friendly frontend, we deliver a powerful and interactive platform for Arabic poetry enthusiasts, scholars, and students alike.

*(Screenshots and system architecture diagrams will be included in the appendix under **UI/UX** and **Architecture** sections respectively.)*

Performance Evaluation of ALLAM vs. Other Arabic LLMs

In developing our platform, we evaluated various state-of-the-art Arabic Large Language Models (LLMs) to determine which would best serve our needs for poetry analysis, generation, and debates.

Among these models, **ALLAM** consistently outperformed other notable models such as **JAIS** when prompted correctly and provided with the necessary context and examples.

1. ALLAM's Strengths in Arabic Language Processing

ALLAM demonstrated superior capabilities across several key areas:

- **Accuracy in Generation and Analysis**: When given well-structured prompts and valid examples through few-shot learning, ALLAM excelled in generating poetry that was not only contextually relevant but also stylistically faithful to the selected poets. This level of accuracy far exceeded our initial expectations and proved invaluable in ensuring that the content generated was coherent and culturally appropriate.
- **Factual Reliability**: One of the standout features of ALLAM is its ability to minimize **hallucination**—the phenomenon where models generate inaccurate or irrelevant information. Unlike other models that occasionally diverge from factual data, ALLAM almost never hallucinated during factual tasks, such as retrieving historical context about poets or interpreting specific poetic forms. This made it a highly reliable tool for both creative generation and analytical tasks.
- **Cultural Nuances and Contextual Understanding**: ALLAM's training on a wide variety of Arabic texts, including classical and modern poetry, enabled it to capture intricate cultural references and historical allusions, which are critical in poetry analysis and generation. Other models like JAIS struggled to maintain this level of cultural fidelity in their responses, especially in more nuanced poetic forms.

2. Comparison with JAIS and Other LLMs

While **JAIS** and other Arabic-focused LLMs are recognized for their capabilities in general Arabic language tasks, they fell short in key areas when compared to ALLAM:

- **Stylistic Mimicry**: In the context of poetry generation, JAIS was able to generate grammatically correct verses, but it often failed to replicate the distinct styles of classical poets with the same level of precision as ALLAM. ALLAM's few-shot learning approach, combined with its robust training data, enabled it to better capture the essence of each poet's style.
- **Factual Consistency**: JAIS exhibited a higher frequency of factual inaccuracies, particularly when handling historical or biographical queries about poets. In contrast, ALLAM's factual responses were consistently accurate, making it the superior choice for our platform's use cases, where reliability is critical.

3. Optimizing ALLAM through Prompting

One of the key factors that contributed to ALLAM's outstanding performance was the **proper construction of prompts**:

- **Few-Shot Prompting**: By providing ALLAM with carefully selected examples from the **Ashaar dataset**, we were able to fine-tune its responses, ensuring that the generated poetry adhered to the expected style, rhyme, and meter.
- **Contextual Prompts**: When tasked with analyzing or generating poetry, ALLAM responded more accurately when given a rich context, such as background information on the poet or the topic. This allowed the model to produce more coherent and relevant outputs, further distinguishing it from other models.

Conclusion

Based on our evaluations, **ALLAM** clearly stands out as the most effective Arabic LLM for our platform's needs. Its ability to minimize hallucinations, maintain factual accuracy, and capture cultural nuances makes it an indispensable tool for tasks related to Arabic poetry. By leveraging ALLAM with optimized prompting techniques and supplementary data, we were able to build a platform that consistently delivers high-quality analysis and generation, far surpassing the capabilities of other models like JAIS.

Challenges Faced

During the development of our AI-powered platform for the Allam Challenge, we encountered several technical and logistical challenges. Overcoming these hurdles required careful problem-solving, team collaboration, and leveraging available resources. Below is an outline of the key challenges faced, and the solutions implemented to address them.

1. Initial Difficulties with IBM Watson Integration

At the start of the project, we experienced some difficulties integrating with **IBM Watson**, particularly in accessing and utilizing the **ALLAM LLM** on the platform. The IBM Watson environment was new to our team, and we initially struggled with setting up the necessary components for our workflow. However, after dedicating time to studying tutorials and documentation, as well as seeking guidance from supervisors and coordinators involved in the challenge, we were able to successfully integrate IBM Watson into our platform. This allowed us to fully utilize ALLAM for our poetry-related tasks.

2. Data Cleaning and Preprocessing in the Ashaar Dataset

The **Ashaar dataset**, which forms the backbone of our platform's poetry generation and analysis functionalities, contained a significant number of missing values. This posed a challenge for performing efficient **similarity searches** and feeding the data into the model for **few-shot learning**. To address this, we performed extensive **data cleaning** and **Arabic text normalization**. This included:

- Handling missing values by removing incomplete entries.
- Normalizing the Arabic text to ensure consistency in tokenization, which improved the performance of the **similarity search** and the overall effectiveness of the model.

These preprocessing steps were crucial in enhancing the quality of the data fed into ALLAM, ultimately leading to better results in poetry generation and analysis.

3. Challenges with Datasets for Meters, Qafya, and Rhetorical Elements

One of the more complex challenges we faced was the **lack of readily available datasets** for specific poetry-related tasks, such as **meter classification**, **qafya detection** (rhyme), and identifying **rhetorical elements** in Arabic verses. While we managed to train a **meter classification model** using a custom-built transformer-based architecture, there were still gaps in datasets for qafya and rhetorical analysis.

To overcome this:

- We leveraged **ALLAM's capabilities** to handle **qafya detection** directly through its language understanding, without needing a separate dataset. ALLAM was able to accurately detect rhymes through careful prompting.

- For rhetorical elements, we relied on **example-based prompting**, which allowed ALLAM to extract and identify rhetorical devices in the verses.

This creative use of available tools and resources enabled us to overcome the absence of specialized datasets and deliver effective poetry analysis.

4. Cloud Hosting Constraints

Another significant challenge was the **restriction on hosting or deploying models** and applications directly on **IBM Watson**. As we were unable to host the code for our platform on IBM Watson, we needed to find an alternative hosting solution for the duration of the challenge. After exploring various options, we decided to use **cloud hosting** to deploy the application. This ensured that the platform was fully operational and accessible to users despite the limitations on IBM Watson.

While this added extra complexity to our deployment process, the cloud hosting solution proved to be reliable and allowed us to maintain the platform's functionality throughout the challenge.

5. *ALLaM's Challenges*

In implementing Arabic poetry generation features—such as stylistic emulation of specific poets, topic-based rhyming, and poetic debates—we encountered several technical challenges:

- **Inconsistent Structured Output**: Allam often struggled to consistently produce structured outputs like JSON format and did not support tool-calling features compatible with frameworks like Langchain. Structured output libraries, such as DSPY and Outlines, were also incompatible with Allam, and at times, it did not fully adhere to system prompts.
- **Reliable Parsing of Allam's Output**: Parsing Allam's responses posed reliability issues, as outputs were occasionally inconsistent or unstructured.

Solutions Implemented

To address these challenges, we applied advanced prompt engineering techniques, including LLAMA2-specific strategies, to improve Allam's response structure. Few-shot prompting was used to help the model adhere to the required formats and style consistently. We implemented regex and string processing methods with exception handling and fallback mechanisms to enhance output parsing reliability. For the poetic debate functionality, we focused on optimizing system stability and responsiveness to ensure a smooth user experience.

Conclusion

Throughout the development process, we faced several technical and logistical challenges, from integrating new technologies like IBM Watson, to managing incomplete datasets, and dealing with

hosting restrictions. However, by leveraging team collaboration, problem-solving, and external resources such as tutorials and cloud hosting, we successfully overcame these obstacles. Each challenge provided valuable learning experiences and helped us refine our approach to building a robust and effective poetry analysis and generation platform.

Evaluation and Accuracy

1. GPT-Like Metric for Evaluating Allam Model's Judging and Summarization

Objective: The goal is to evaluate the judging and summarization capabilities of the Allam model in comparison with state-of-the-art models, specifically the gpt-4o-mini model. This comparison model was chosen for its similar parameter count, making it a closer match for evaluating Allam's effectiveness. This evaluation focuses specifically on prompt engineering within the judge agent, assessing its performance in providing comments and summaries.

Methodology: To measure the similarity between the Allam model's responses (candidate responses) and those of GPT (reference responses), we employ contextual embedding comparisons:

Metric: We calculate precision, recall, and F1 scores to determine how closely Allam's responses align with the reference responses from GPT.

Embeddings: Contextual embeddings are computed for both candidate and reference responses using BERT embeddings, providing a high-resolution view of similarity in terms of content and context.

Implementation:

- **Algorithm:** We referenced the methodology from "BERTScore: Evaluating Text Generation with BERT" and implemented this using the open-source BERTScore library (available on GitHub here).
- **Model:** To compute the contextual embeddings, we used the DeBERTa-large-mnli model from Hugging Face. This BERT-based model is optimized for identifying nuanced semantic similarities, making it well-suited for our evaluation of judging and summarization. In addition, this model works well with Arabic language.

Automation and Results Logging:

- **Judge Agent Enhancements:** An evaluation mode was added to the judge agent, enabling automatic calculation of these metrics during backend server operations.
- **Results Storage:** All computed metrics (precision, recall, and F1) are logged in a CSV file, allowing for easy review and further analysis of the Allam model's performance.

2. Theme Classification Metric for Arabic Poetry Generation

Objective: This metric aims to assess the Allam model's capability to generate Arabic poetry verses that align with the theme requested by the user. By embedding generated verses and comparing them to pre-existing theme-specific embeddings, this evaluation verifies if the generated content accurately reflects the desired theme.

Methodology:

To evaluate whether the generated verse aligns with the intended theme:

- **Embedding Space Creation:** We first embed theme-specific verses from a dataset to establish an embedding space. This space serves as a reference for each theme.
- **Generated Verse Embedding:** The verse generated by Allam is embedded and then tested for proximity within the correct thematic embedding space. A high similarity score indicates successful theme alignment.

Implementation:

- **Dataset:** We use the Ashaar dataset from Hugging Face for both poetry generation and the evaluation process, as it offers a broad selection of Arabic poetic themes and styles.
- **Embedding Model:** To compute embeddings, we use the BERT-base-multilingual-cased model, chosen for its strong performance in understanding Arabic text. This model enables robust embedding comparisons for theme accuracy

Poetry Generation Evaluation Table Examples

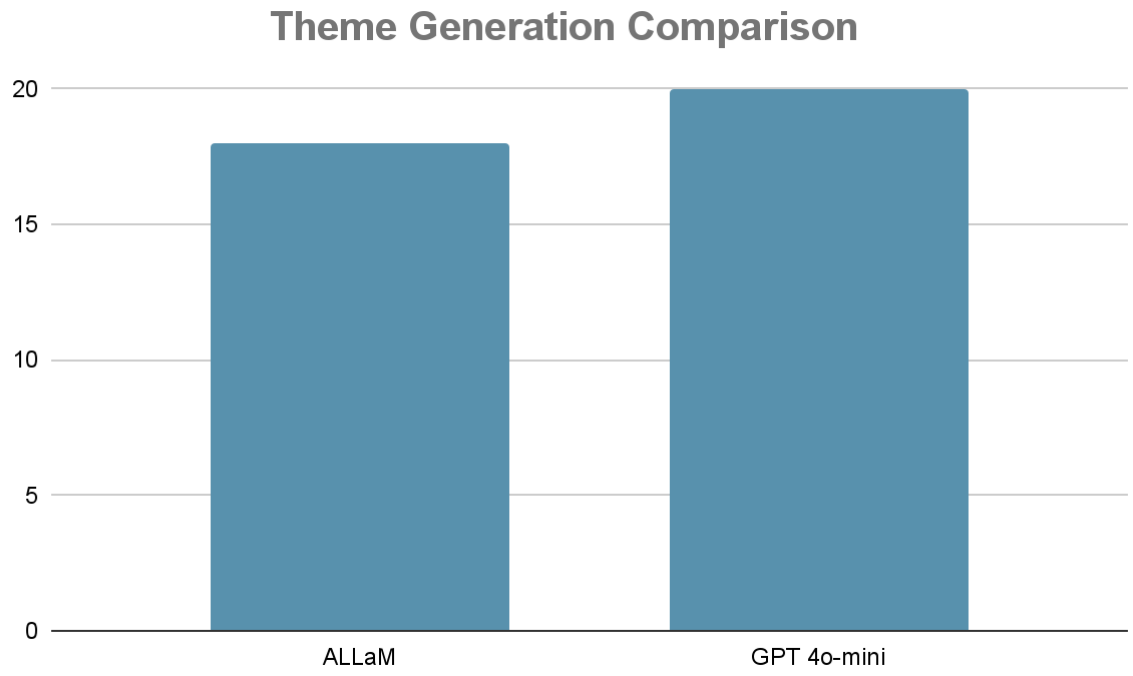
Here are a few sample cases illustrating how we generated data using both ALLAM and GPT-4o-mini and compared their outputs.

Prompt	أريد بيت شعر من قصيدة دينية علي طريقة أبي بكر الصديق ينتهي بقافية النون
Ground Truth Theme	قصيدة دينية
ALLAM's Response	يا ربّ صلّ على النبيّ محمدٍ * ما ناح قمريّ على الأغصان
Detected Theme	قصيدة دينية
GPT 4o-mini Response	في القلب نور الإيمان يضيء، يهدينا درب الحق واليقين
Detected Theme	قصيدة دينية

Prompt	أريد بيت شعر مدح علي طريقة أحمد شوقي ينتهي بقافية الميم
Ground Truth Theme	قصيدة مدح
ALLAM's Response	يا عَليلَ الجَسمِ كَم عِيلَتِ الأسهُمُ بِنا مِن جَورِ عَينِكَ بِالسِهام
Detected Theme	قصيدة مدح
GPT 4o-mini Response	أنتَ الفخرُ في زمنٍ قد زادَ فيهِ الفُضلُ والمَجدُ عَظيما
Detected Theme	قصيدة مدح

Prompt	أريد بيت شعر هجاء علي طريقة حافظ ابراهيم ينتهي بقافية النون
Ground Truth Theme	قصيدة هجاء
ALLAM's Response	وَفَتىً لَو حَلَّ خاطِرُهُ في لَيالي الدَهرِ لَم تَخُن
Detected Theme	قصيدة هجاء
GPT 4o-mini Response	أَتَظُنُّ العَزَّ في كَذِبٍ يُسَطِّرُ فَتَسيرُ في مَوكِبِ الفَضائِحِ كَالأحلام
Detected Theme	قصيدة رومنسيه

Prompt	أريد بيت شعر عن الوطن علي طريقة امرؤ القيس ينتهي بقافية النون
Ground Truth Theme	قصيدة وطنية
ALLAM's Response	بلاد العرب أوطاني من الشام لبغدان ومن نجد إلى يمن إلى مصر فتطوان
Detected Theme	قصيدة وطنيه
GPT 4o-mini Response	يا وطني، فيك القلب ينمو ويزهر، كالنجم في ظلام الليل يلوح ويحنون
Detected Theme	قصيدة وطنيه



Out of 20 poems correctly generated by GPT 40-mini following a specific theme, ALLaM accurately generated 18, demonstrating impressive performance given the significant difference in parameter count between the two models. This highlights ALLaM's efficiency and capability despite its comparatively smaller size.

Poetic Debate Evaluation Table Example

The following is a randomly selected comparison from the poetic debates conducted between ALLAM and ChatGPT 4o.

ALLaM	ChatGPT
الجولة الأولى	
أحمد شوقي: • **النقاط**: 6 نقاط • **التعليق**: يعبر البيت عن فكرة مبتكرة ولكن يحتاج لتحسين التناسق اللغوي.	**أحمد شوقي:** • **البيت**: "وَلَم أَرَ مِثلَ القَلبِ لا وَجهَ لَهُ ما اِلتَمَعَ البَرقُ لَنا مِن غَيمِهِ اِبتَسَما" • **النقاط**: 6 • **التعليق**: يعبر عن عمق المشاعر الإنسانية، لكنه لم يكن كافيًا لإقناع الحكام.
حافظ إبراهيم: • **النقاط**: 8 نقاط • **التعليق**: يعبر البيت عن مشاعر تجاه شخص محبب بقوة تعبير عاطفي.	**حافظ إبراهيم:** • **البيت**: "بِأَبي وَأُمّي ذاكَ الوَجهُ ما غَدا لِطَيفُكَ يَزورُني وَاللَيلُ قَد جَلّا" • **النقاط**: 8 • **التعليق**: يحمل جمالية في التصوير الشعري، مما جعله يتفوق في هذه الجولة.
الجولة الثانية	
أحمد شوقي: • **النقاط**: 8 نقاط • **التعليق**: يعبر البيت عن حب الشاعر لوطنه ومشاعره بشكل جيد.	**أحمد شوقي:** • **البيت**: "يا مِصرُ أَنتِ كِناسَةٌ شَرَفَت وَوَطِئَت أَرضُكِ مَنزِلاً كَرُمُ" • **النقاط**: 8 • **التعليق**: يعكس الفخر الوطني، مما زاد من نقاطه.
حافظ إبراهيم: • **النقاط**: 6 نقاط • **التعليق**: يعبر البيت عن حب الشاعر لوطنه ومشاعره بشكل جيد.	**حافظ إبراهيم:** • **البيت**: "بِلادُكَ قَد سَمَت بِالأَمنِ حَتّى كَأَنَّ بِها لَم تُفجَع بِزَلزِلِ" • **النقاط**: 6 • **التعليق**: رغم جمال البيت، إلا أنه لم يكن بنفس القوة.

الجولة الثالثة	
أحمد شوقي: • **النقاط**: 6 نقاط • **التعليق**: يعبر البيت عن مشاعر تجاه شخص أهان الشاعر مع تعبير عاطفي قوي.	**أحمد شوقي:** • **البيت**: "أَوسَعتَنا يَومَ الوَداعِ إِهانَةً أَدَبٌ لَعَمرُكَ لا يُصيبُ مَثيلا" • **النقاط**: 6 • **التعليق**: يحمل معاني عميقة، لكنه لم يكن كافيًا لتحقيق نقاط أعلى.
حافظ إبراهيم: • **النقاط**: 10 نقاط • **التعليق**: يعبر البيت عن جمال المكان مع وصف قوي.	**حافظ إبراهيم:** • **البيت**: "أَرى المَصابيحَ فيها وَهيَ مُشرِقَةٌ كَأَنَّها النَورُ وَالوَسمِيُّ حَيّاهُ" • **النقاط**: 10 • **التعليق**: البيت كان قويًا جدًا، مما جعله يتصدر الجولة.
الجولة الرابعة	
أحمد شوقي: • **النقاط**: 6 نقاط • **التعليق**: يعبر البيت عن مشاعر تجاه شخص يعاني من السهر والألم.	**أحمد شوقي:** • **البيت**: "يا عَليلَ الجَفنِ كَم ذا تَسهَرُ أَنتَ لِلصَبرِ وَلِلرَزايا أَكبَر" • **النقاط**: 6 • **التعليق**: يحمل مشاعر عميقة، لكنه لم يكن كافيًا للتفوق.
حافظ إبراهيم: • **النقاط**: 10 نقاط • **التعليق**: يظهر الشاعر كراعٍ للمحتاجين مع تعبير عاطفي مؤثر.	**حافظ إبراهيم:** • **البيت**: "يا أَمينَ البَيتِ يا مَن بِالعَطايا يَمنَحُ الجائِعَ وَالخائِفَ وَالجاني" • **النقاط**: 10 • **التعليق**: البيت كان مؤثرًا جدًا، مما جعله يتفوق في هذه الجولة.
الجولة الخامسة	
أحمد شوقي: • **النقاط**: 6 نقاط	**أحمد شوقي:**

• **التعليق**: يعبر البيت عن فكرة مبتكرة مع مجال لتحسين التناسق اللغوي.	• **البيت**: "لَم يَزَل يَرمي بِقَولٍ سَهمَهُ ** وَيَقولُ الشِعرُ دَواءٌ لِمَن ** سَئِمَ العَيشَ وَعافَ المُنى ** وَهوَ لا يَدري بِأَنَّ الشِعرَ ** مِنهُ أَصلُ دائِهِ" • **النقاط**: 6 • **التعليق**: يحمل عمقًا فلسفيًا، لكنه لم يكن كافيًا لتحقيق نقاط أعلى.
حافظ إبراهيم: • **النقاط**: 7 نقاط • **التعليق**: يعبر البيت عن مشاعر تجاه شخص محبب مع قوة تعبير عاطفي.	**حافظ إبراهيم**: • **البيت**: "لَم يَزَل يَطوي هَواهُ عَنكُمُ زَمَناً حَتّى رَأَيتُكَ يا عَلِيُّ فَطَويتُهُ" • **النقاط**: 7 • **التعليق**: البيت كان جيدًا، لكنه لم يكن بنفس قوة الجولات السابقة.
النتائج النهائية	
احمد شوقي: مجموع النقاط: 32 **حافظ إبراهيم**: مجموع النقاط: 41	**أحمد شوقي**: مجموع النقاط: 34 **حافظ إبراهيم**: مجموع النقاط: 41
نظرة عامة	
بعد تحليل أداء كل شاعر، يظهر تفوق حافظ إبراهيم بمجموع 41 نقطة مقارنةً بـ 32 نقطة لأحمد شوقي، حيث أظهر حافظ إبراهيم إبداعًا وتناسقًا لغويًا وتعبيرًا عاطفيًا أعلى في أبياته.	تفوق حافظ إبراهيم في معظم الجولات، حيث أظهر قوة في التعبير الشعري وجمالية في التصوير. بينما قدم أحمد شوقي أبياتًا تحمل عمقًا شعريًا، إلا أنه لم يتمكن من تحقيق نفس مستوى النقاط. الأداء العام لكلا الشاعرين كان مميزًا، لكن حافظ إبراهيم كان الأبرز في هذه المنافسة.
الفائز	
حافظ إبراهيم	**حافظ إبراهيم**

Evaluation Scores

Poetry Generation

This evaluation assesses the similarity of generated poetic text from ALLaM as the candidate model against ChatGPT as the reference. Precision, recall, and F1 scores are computed to evaluate each generated text's alignment with the reference for each poet.

The mean values for each metric across all generated poetic texts are as follows:

- **Average Precision:** 0.7912
- **Average Recall:** 0.8005
- **Average F1 Score:** 0.7955

Interpretation of Results

- **Precision** (0.7912): ALLaM's generated text aligns closely with the reference, capturing much of the relevant wording.
- **Recall** (0.8005): High recall suggests comprehensive content coverage in the generated text compared to the reference.
- **F1 Score** (0.7955): Indicates a balanced performance in both capturing relevant information and closely matching the reference.

Juding and Poetry Analysis

This evaluation assesses the similarity of the judge's analysis and scoring text for each poet after completing a round, with ALLaM as the candidate and ChatGPT as the reference. Precision, recall, and F1 scores are computed to measure alignment between the generated and reference text.

The mean values for each metric across all judge analysis and scoring entries are as follows:

- **Average Precision:** 0.8891
- **Average Recall:** 0.8983
- **Average F1 Score:** 0.8930

Interpretation of Results

- **Precision** (0.8891): Shows that the judge's generated analysis text contains relevant elements closely aligned with the reference.
- **Recall** (0.8983): Indicates that the analysis text comprehensively captures the content and context of the reference text.
- **F1 Score** (0.8930): Reflects a balanced performance in both relevance and thoroughness, demonstrating high-quality alignment with the reference.

Scoring Conclusion

The BERTScore results reflect a strong alignment between ALLaM's generated poetic text and ChatGPT's reference text. The model demonstrates high recall, slightly outperforming precision, suggesting that it captures the essential content thoroughly, though there is room for minor adjustments in exact wording alignment. This analysis provides valuable insights into the quality and relevance of ALLaM's generated poetry, with potential areas for refinement to further enhance precision.

Similarly, the BERTScore results for the judge's analysis and scoring indicate a high level of alignment with the reference, also showing strong recall. This suggests that the generated text captures both the content and context of the reference accurately, achieving reliable quality in the judge's evaluations and scoring. Together, these results provide confidence in the alignment of ALLaM's generated texts with the reference, ensuring accurate, relevant poetic and scoring analysis.

Future Work and Marketing Strategy

As we look beyond the initial development of our AI-powered poetry platform, the potential for expansion and impact is vast. Our goal is to continue refining the platform's capabilities, making it more accessible to a diverse range of users, and ensuring it becomes a valuable tool for poetry enthusiasts, educators, researchers, and the broader Arabic-speaking community. In this section, we outline our future plans and marketing strategy to maximize the platform's reach and effectiveness.

1. Future Work and Expansion

Our current platform provides powerful tools for poetry analysis, generation, and debates. However, there are several enhancements we plan to implement in future iterations to improve functionality, personalization, and user experience.

- **User Accounts and Personalization**:
  In future versions of the platform, we will implement a **user account system** that allows users to create and log into personal accounts. This feature will enable users to **save their poetry sessions, chats, and analysis results**, allowing for a more personalized and seamless experience.
  - **Chat history** will enable users to revisit their previous poetry generation or debates, track their learning progress, and analyze recurring themes or styles in their work.
  - **Customization of outputs**: The platform can adapt its poetic generation by learning from each user's preferences, adjusting styles, topics, and complexity based on previous interactions. This creates a tailored experience that keeps users engaged over time.

  - **Social Media Integration**: Users will have the option to connect their social media accounts (e.g., Twitter, Facebook, Instagram) to the platform. By analyzing their shared content, interests, and writing style, the platform can refine its poetry generation and analysis to reflect the user's unique voice and preferences. This integration will provide a deeper level of personalization, making the experience more relevant to each user.

- **Voice Command and Text-to-Speech:**
  To make the platform more accessible and interactive, we plan to incorporate **voice command** and **text-to-speech** functionalities:
  - **Voice Command Input**: Users will be able to submit prompts, requests, or configuration preferences using voice commands. This feature will simplify the user experience and make interactions with the platform more natural and engaging.
  - **Text-to-Speech Output**: The platform will offer text-to-speech functionality, allowing it to read generated poetry aloud with accurate pronunciation and diacritical marks. This feature will enhance the experience of listening to Arabic poetry, with correct rhythm, rhyme, and meter, making it more enjoyable and educational.

  These features aim to create an immersive environment, enabling users to interact with poetry in both written and spoken forms.
- **Configurable Poetic Debates:**
  The poetic debate feature will be enhanced with new customization options, allowing users to tailor debates according to their preferences:
  - **Rounds and Qafya Selection**: Users can configure the number of rounds and select a specific qafya (rhyme scheme) for the debate, creating a more structured and competitive experience.
  - **Meter and Rhetorical Elements**: The platform will allow users to set the intended meter and choose specific rhetorical elements to be included in the poetic debate.
  - **Advanced Theme Control**: Users can define themes, tones, and even historical or stylistic references, giving them greater control over the debate's content and flow.

  These configurable features aim to make poetic debates more interactive, allowing users to challenge the AI in creative ways and explore diverse poetic forms.
- **Learning from User Inputs**:
  - We plan to integrate **adaptive learning** mechanisms, where the platform learns from the **user's prompts** and interactions. This includes analyzing user behavior, feedback, and corrections to refine future outputs.
  - The platform could even offer **suggestions or guidance** based on past activity, helping users explore new poets, styles, or rhetorical techniques they haven't encountered before.
- **Fine-tuning ALLAM on Larger Arabic Datasets**:
  - As part of our roadmap, we will work on **fine-tuning ALLAM** with **larger and more diverse Arabic poetry datasets**. This includes expanding the model's training data to encompass **regional dialects**, **contemporary poetry styles**, and more **historical works**. By doing so, we aim to make the model even more versatile in generating,

analyzing, and understanding the full breadth of Arabic poetry, from classical to modern times.
  - We also plan to build on the **Ashaar dataset** by incorporating more detailed datasets for **rhetorical elements** and **advanced poetic structures** to make the platform a more comprehensive tool for both education and creativity.
- **Interactive Educational Features**:
  - Expanding the platform's role as an **educational tool**, we envision the inclusion of **interactive lessons** on Arabic poetry forms, techniques, and history. The platform could offer **tutorials** on how to write different forms of Arabic poetry, such as Ghazals or Qasidas, and provide **real-time feedback** as users compose their own verses.

2. Marketing Strategy

To ensure the success of the platform and maximize its impact, we have devised a **multi-faceted marketing strategy** aimed at reaching a wide range of potential users, including poets, educators, students, researchers, and Arabic culture enthusiasts. Our strategy focuses on user engagement, brand visibility, and community building.

- **Target Audience Outreach**:
  - The platform is designed to appeal to a broad demographic, from **literature students** and **academic researchers** to **poetry enthusiasts** and **cultural institutions**. We will focus on key educational and cultural markets, including universities, schools, libraries, and art centers across the Arab world.
  - Special focus will be given to **educators**, where the platform can be used as a teaching tool in literature courses, helping students better understand classical Arabic poetry while engaging with modern AI technologies.
- **Content Creation and Partnerships**:
  - We will create **educational content** such as blog posts, video tutorials, and interactive workshops that demonstrate the platform's capabilities in analyzing and generating poetry. These materials can be shared on **social media platforms**, **YouTube**, and **poetry websites** to build awareness and engagement.
  - Forming **partnerships with cultural institutions**, universities, and literary festivals will also be key to expanding our reach. By collaborating with these organizations, we can introduce the platform to new audiences and integrate it into academic and cultural programs.
- **Building a Community**:
  - One of the most powerful ways to grow the platform's user base is by fostering an engaged **online community** of poets, educators, and AI enthusiasts. Through the development of **discussion forums**, **poetry challenges**, and **user-generated content competitions**, we will encourage users to share their creations and experiences with others, creating a viral effect that draws more users to the platform.

  - Hosting **poetry debates and competitions** in collaboration with cultural festivals and literary organizations could also generate significant interest and publicity for the platform.
- **Subscription Model and Monetization**:
  - While the base version of the platform will remain free to ensure wide accessibility, we plan to introduce a **premium subscription model** that offers enhanced features. Premium users can benefit from:
    - **Advanced poetic analysis**, including deeper critique and custom feedback.
    - **Personalized learning plans**, where the platform suggests poets, styles, and themes to explore based on past interactions.
    - **Access to exclusive poetry datasets** and historical archives for research purposes.

3. Vision for the Future

Our long-term vision for the platform is to become a central hub for Arabic poetry in the digital age—where technology and culture intersect to create new possibilities. We believe that with continuous refinement and user feedback, our platform will not only preserve the rich tradition of Arabic poetry but also introduce it to new generations in innovative ways.

- **Expanding Beyond Poetry**: As the platform evolves, we see the potential to expand beyond poetry into other areas of Arabic literature, such as prose, historical texts, and even philosophical works. The underlying technologies, especially the use of ALLAM and RAG, are versatile enough to support more complex forms of literary analysis.
- **Global Reach**: While our platform is currently focused on Arabic poetry, we aim to broaden its reach to international audiences. The integration of multi-language capabilities will allow users from diverse linguistic backgrounds to engage with the richness of Arabic literature and connect it to their own literary traditions.

By aligning technological innovation with cultural preservation, we believe this platform can serve as a **pioneering tool** in the world of digital Arabic literature. Our goal is not only to meet the current needs of users but to anticipate future trends, providing a product that grows and adapts with its audience.

Results

The results of our AI-powered platform for Arabic poetry generation, analysis, and poetic debate are a testament to the innovative methodologies we employed. Below, we present the key outcomes based on various evaluations:

1. Poetry Generation Quality

- **Poet Style Mimicry**: The platform's ability to generate poetry in the style of specific poets was evaluated.
- **Relevance to Topics**: The platform's capability to maintain thematic coherence was assessed.

N.B: Examples can be seen in Appendix D

2. Poetry Analysis Accuracy

- **Meter Classification**: Accuracy of the meter classification model was measured against a labeled dataset.
- **Rhetorical Elements Detection**: The platform's ability to detect rhetorical elements like metaphors and similes was evaluated using example-based prompting.

3. Poetic Debate Functionality

- **Debate Coherence**: The platform's ability to simulate coherent and stylistically consistent poetic debates was assessed.
- **Judgment Accuracy**: The Poetry Judge Agent's effectiveness in selecting a debate winner based on rhyme, rhetorical strength, and thematic relevance was evaluated.

N.B: Examples can be seen in Appendix D

4. User Experience

- **Sleek UI Design**: A user-friendly interface enables easy navigation, as shown in Appendix A (UI/UX).
- **Performance and Response Time**: Platform performance and response times for generating poetry and conducting analyses were taken into consideration.

N.B: Examples can be seen in Appendix A

These results demonstrate the effectiveness of our methodologies and the value of the platform in providing both creative and analytical insights into Arabic poetry. Further testing and user feedback will provide additional data to refine the results.

Conclusion

Our AI-powered platform for Arabic poetry generation, analysis, and debates represents a significant step forward in the application of modern technology to classical Arabic literature. By utilizing cutting-edge language models such as **ALLAM**, supported by agentic workflows and retrieval-augmented generation (RAG) techniques, we successfully developed a solution that caters to a wide range of users, from poetry enthusiasts to academic researchers.

We faced several challenges during the development process, including technical integration issues and data quality concerns, but were able to overcome these obstacles through careful problem-solving and collaboration. The use of advanced models for **poetic meter classification**, **qafya detection**, and **rhetorical element extraction** has greatly enhanced the platform's analytical capabilities, while the integration of **few-shot learning** and **custom prompting** ensured that our poetry generation outputs are both accurate and stylistically faithful to the chosen poets.

Looking ahead, we have a clear roadmap for the platform's future development. Our plans to introduce **user accounts**, **learning from user inputs**, and **fine-tuning ALLAM** on larger datasets will ensure that the platform evolves with the needs of its users. Additionally, the marketing strategy we've outlined positions us well to reach a broad and engaged audience, with plans to expand the platform's reach and functionality in the coming years.

In conclusion, our platform not only preserves the rich tradition of Arabic poetry but also offers users a powerful tool for exploring, creating, and analyzing poetry in innovative ways. With ongoing refinement and expansion, we believe this platform has the potential to become a central hub for Arabic literary studies, creativity, and education.

Work Distribution

Task	Team Members
1. **IBM Watson Setup**: Configure and integrate IBM Watson with ALLAM for model access.	Marwan ElSafty, Ayman ElGhotni
2. Langchain Integration: Utilize Langchain to streamline agent interactions with ALLAM.	Ayman ElGhotni, Islam Bassuni
3. Ashaar Dataset Cleaning: Clean and preprocess the Ashaar dataset, removing incomplete entries.	Marwan ElSafty, Ayman ElGhotni
4. Embedding Space Creation: Create theme-specific embedding spaces for poetry generation.	Marwan ElSafty, Islam Bassuni
5. Arabic Poetic Meter Classification Model: Develop a transformer model for Arabic meter classification.	Islam Bassuni, Hossam ElKady

6. Qafya Detection with Qawafi: Integrate Qawafi for rhyme (Qafya) detection in poetry.	Islam Bassuni, Hossam ElKady
7. Rhetorical Elements Detection Model: Detect metaphors, similes, and other rhetorical devices.	Islam Bassuni, Ayman ElGhotni
8. Poetry Analysis Agent: Design an agent for structural breakdown and critique of poetry.	Ayman ElGhotni, Marwan ElSafty
9. Poetry Generation Agent: Create an agent for generating poetry in specified styles.	Ayman ElGhotni, Marwan ElSafty
10. Poetry Judge Agent: Develop an agent to evaluate and score poetic debates.	Ayman ElGhotni, Marwan ElSafty
11. Poetry Generation Microservice: Set up a service for generating poetry based on user selections.	Hossam ElKady, Ayman ElGhotni
12. Poetry Analysis Microservice: Establish a service for structural analysis and critique.	Hossam ElKady, Ayman ElGhotni
13. Poetic Debate Microservice: Build a service to manage interactive poetic debates.	
14. Contextual Embedding for Evaluation: Use BERT embeddings for poetry similarity comparison.	Hossam ElKady, Islam Bassuni
15. Theme Classification via Embeddings: Verify generated verses align with requested themes.	Hossam ElKady, Islam Bassuni
16. Precision, Recall, F1 Metric Logging: Calculate and log accuracy metrics for model evaluation.	Hossam ElKady, Islam Bassuni
17. Retrieval-Augmented Generation (RAG): Implement RAG to provide ALLAM with contextual data.	Marwan ElSafty, Hossam ElKady
18. Frontend Development (React): Develop a React-based user interface for user interactions.	Hossam ElKady, Islam Bassuni
19. API Integration: Connect frontend and backend services via API endpoints.	Ayman ElGhotni, Marwan ElSafty
20. Docker Containerization: Containerize microservices with Docker for consistency.	Hossam ElKady, Islam Bassuni
21. Docker Compose Orchestration: Orchestrate the system using Docker Compose.	Marwan ElSafty, Ayman ElGhotni
22. Comparative Evaluation with GPT-4o-mini: Benchmark ALLAM's performance against GPT-4o-mini.	Hossam ElKady, Islam Bassuni
23. BERTScore Metric Evaluation: Calculate precision, recall, and F1 scores for text alignment.	Islam Bassuni, Hossam ElKady

About Brightskies

Brightskies is a leading digital transformation enabler, providing cutting-edge solutions in **High-Performance Computing (HPC)**, **AI and Machine Learning**, **Automotive Services**, and **Enterprise Solutions**. Founded in 2010, Brightskies has grown into a market leader with over 400 talented professionals dedicated to delivering innovative solutions to clients worldwide.

Our expertise in **AI and Machine Learning** has been a cornerstone of our success. We have developed and deployed a range of **AI-driven solutions**, including Natural Language Processing (NLP) systems, language models, and AI-powered tools for various industries. This experience directly contributed to our approach for the **Allam Challenge 2024**, where we utilized our knowledge of AI to develop an interactive poetry analysis and generation platform powered by cutting-edge technologies such as **LLMs**, **agentic workflows**, and **Retrieval-Augmented Generation (RAG)** systems.

Some of our previous projects that resonate with the goals of the Allam Challenge include:

- **Advanced NLP-based Chatbots** for industries like customer service and healthcare, which demonstrated our ability to develop complex, context-aware AI systems.
- **Arabic NLP models**, which focused on improving and fine-tuning language models for the Arabic-speaking world, closely aligned with the goals of the Allam Challenge.
- **LLM Training and Deployment** at scale, allowing us to leverage state-of-the-art models such as ALLAM for creative and analytical tasks like poetry generation and analysis.

Our experience and ongoing dedication to pushing the boundaries of **AI in Arabic language processing** uniquely positioned us to participate and excel in this challenge.

Thank You

We would like to extend our deepest gratitude to **SDAIA** and **Tuwaiq Academy** for organizing the **Allam Challenge 2024** and creating such a fantastic platform for innovation and creativity. This competition provided an incredible opportunity for participants to come together from across countries, fostering cross-border collaboration and communication.

The in-person format of the challenge added an invaluable element, allowing for meaningful **knowledge exchange**, **networking**, and creating a vibrant, cheerful atmosphere where ideas and expertise could be shared freely. The inclusive and inspiring environment nurtured by SDAIA and Tuwaiq Academy was instrumental in the success of all participants, encouraging innovation while promoting a sense of community.

We are grateful for the opportunity to be part of such a remarkable initiative and look forward to future collaborations and challenges that continue to drive the advancement of **Arabic language technology** and **AI solutions**. Thank you for this unforgettable experience.

Appendix A – UI/UX

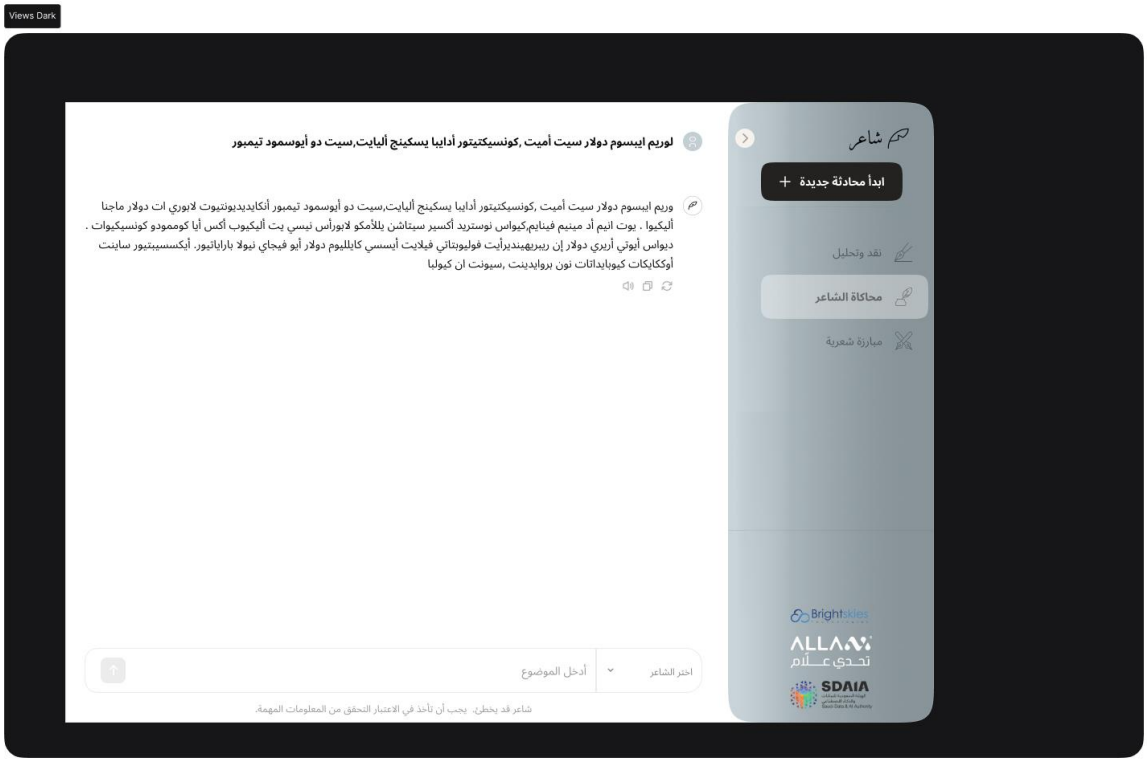


FIG1 – LIGHT FULL VIEW UI/UX

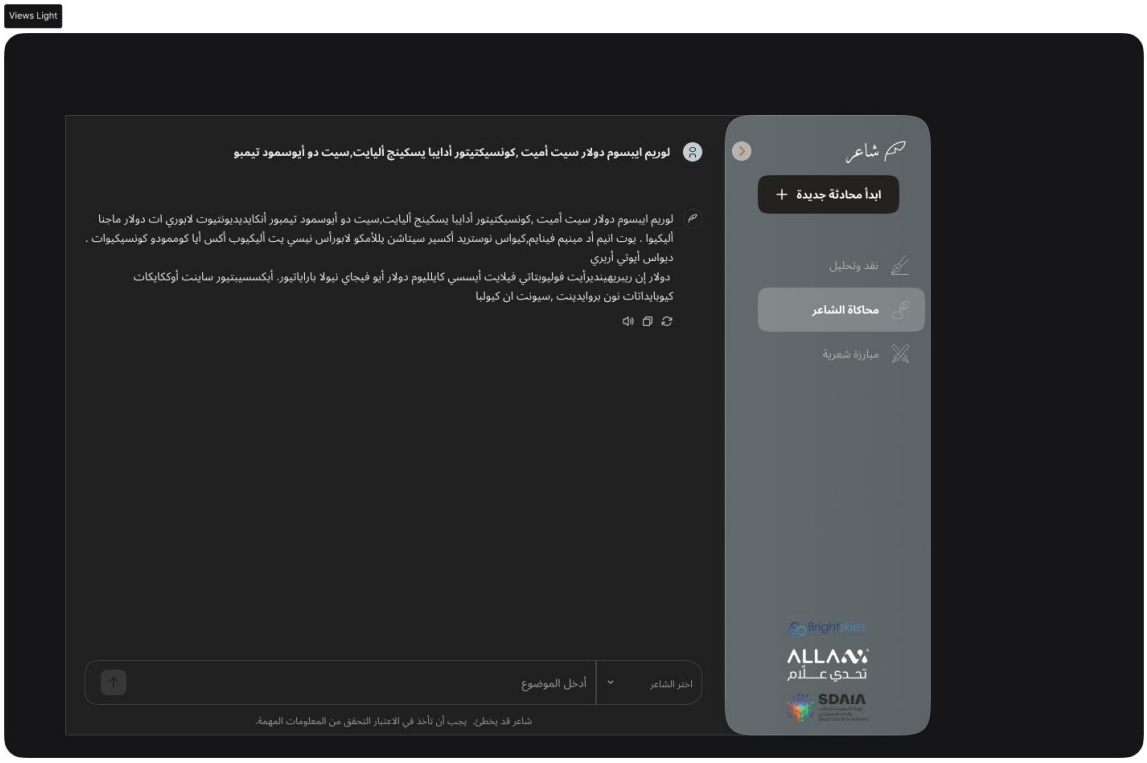


FIG2 – DARK FULL VIEW UI/UX

Appendix B – System Diagram

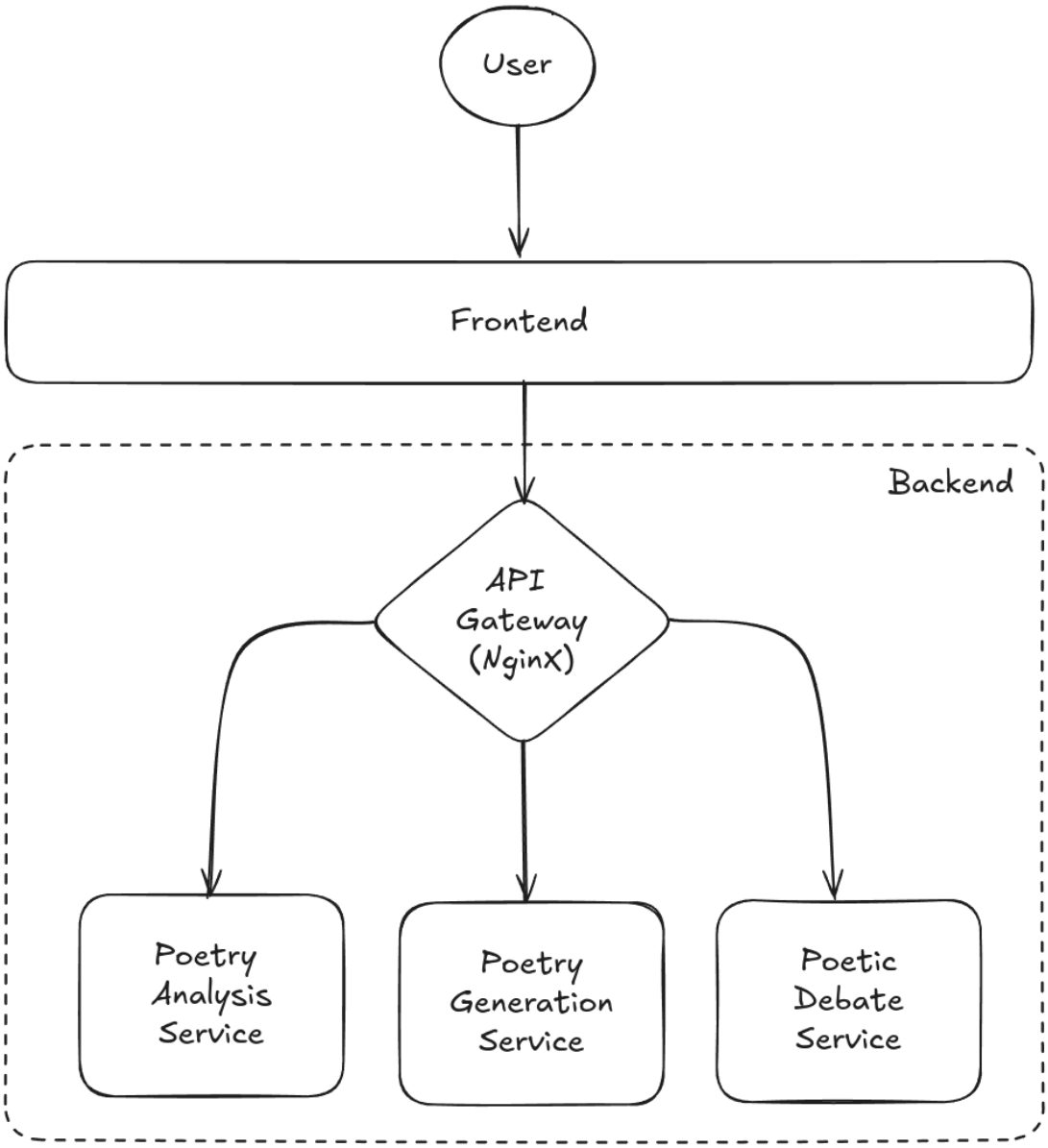


FIG3 – HIGH LEVEL SYSTEM DIAGRAM

Appendix C – Sequence Diagram

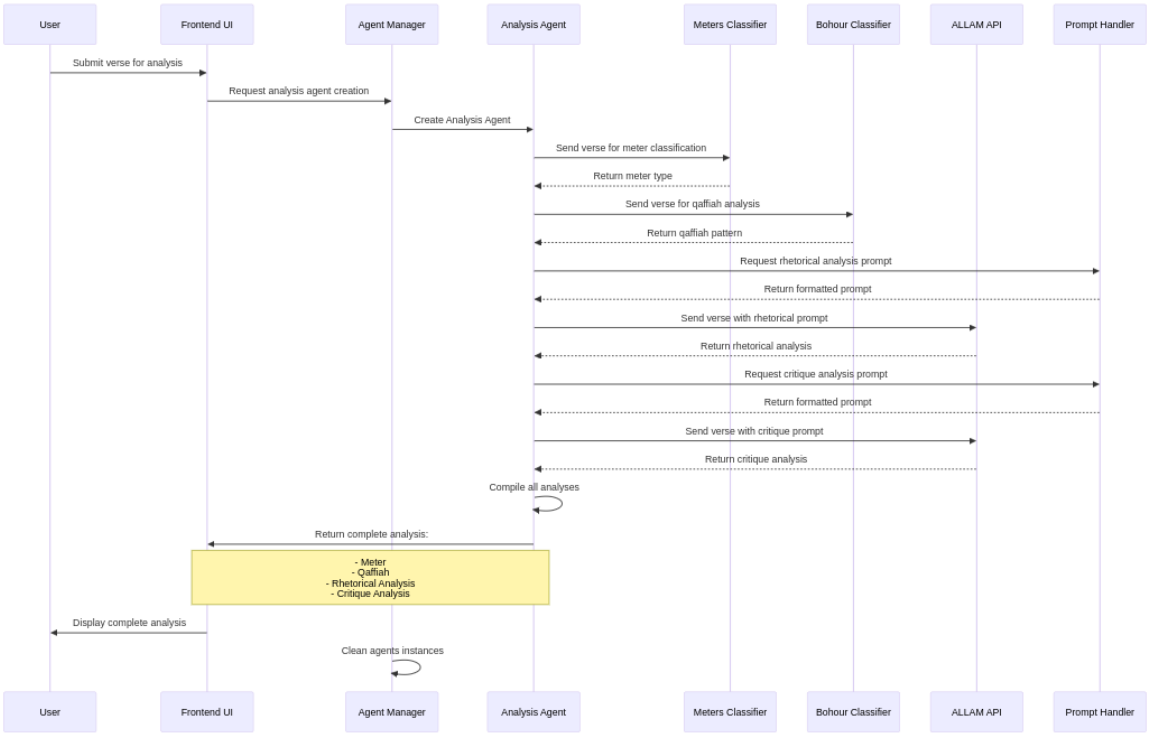


FIG4 - POETRY ANALYSIS & CRITICISM SEQUENCE DIAGRAM

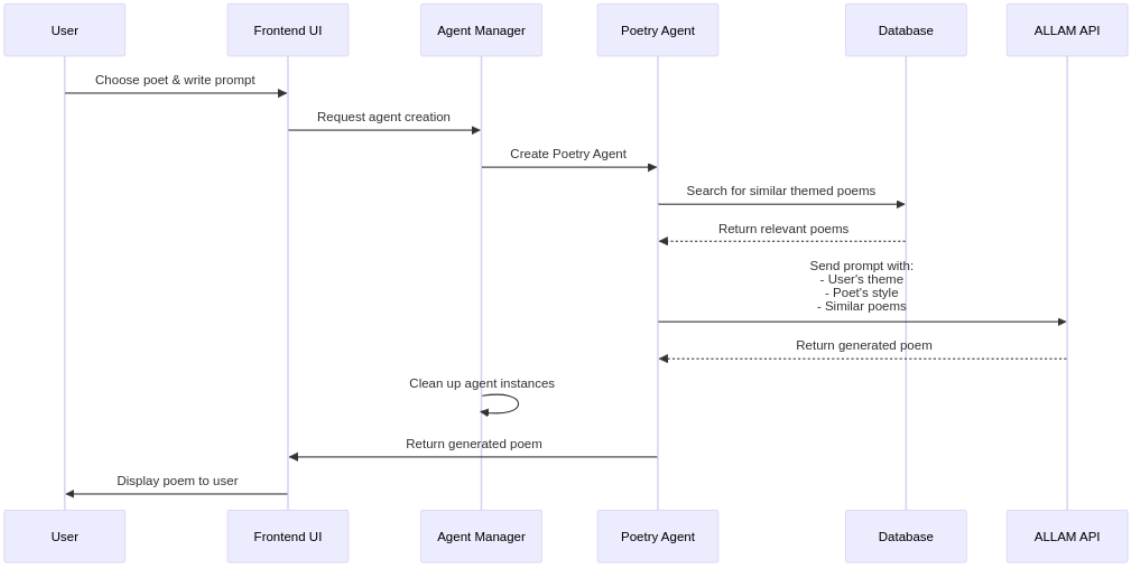


FIG5 – POETRY GENERATION SEQUENCE DIAGRAM

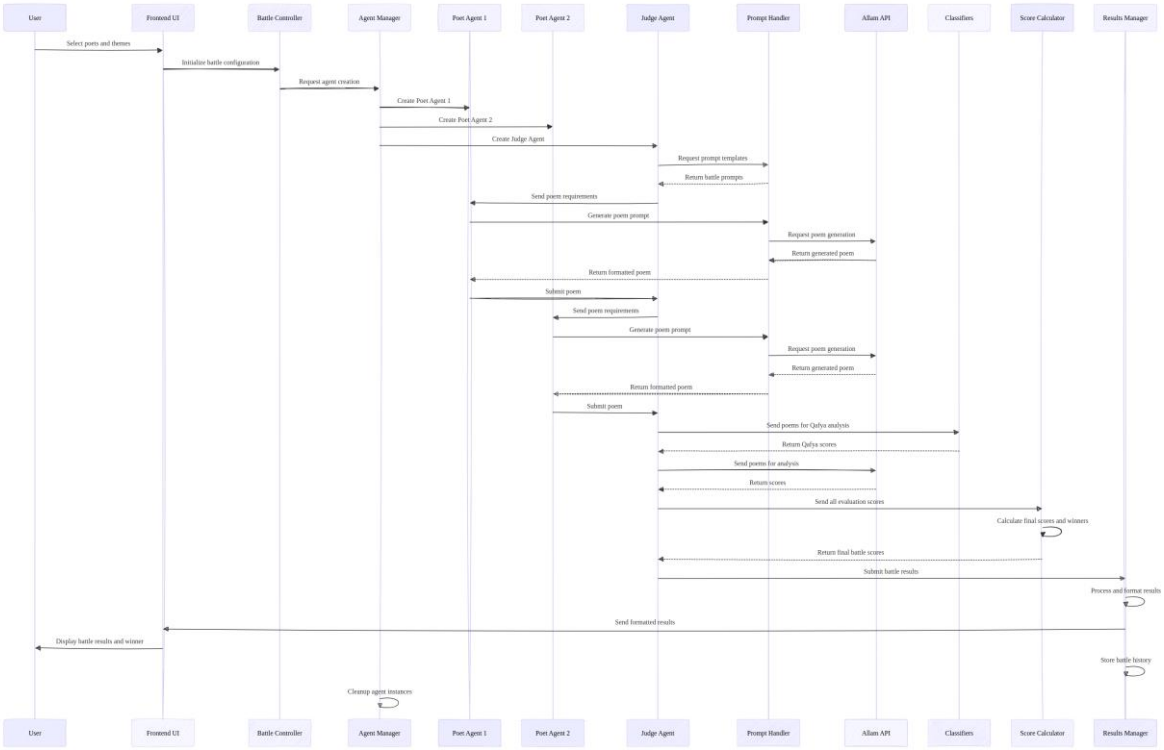


FIG6 – POETIC DEBATE SEQUENCE DIAGRAM

Appendix D – Examples

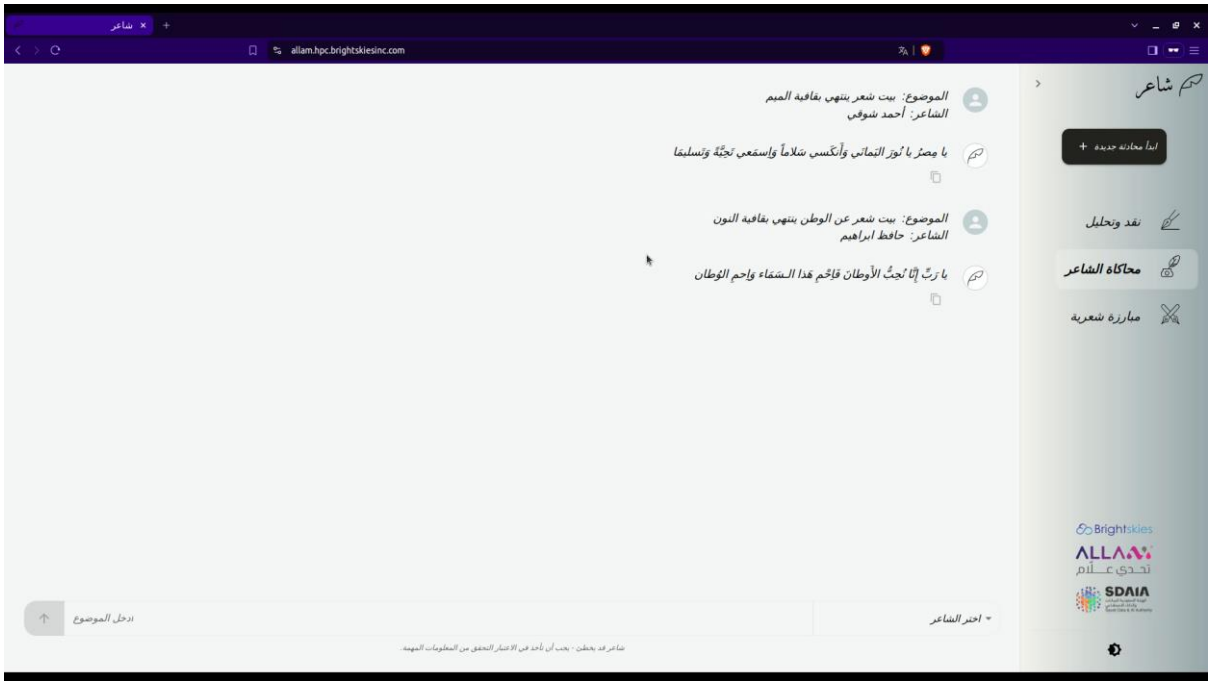


FIG7 – POETRY GENERATION EXAMPLE

FIG8 – POETIC DEBATE EXAMPLE 1-A

FIG9 – POETIC DEBATE EXAMPLE 1-B

FIG10 – POETIC DEBATE EXAMPLE 1-C

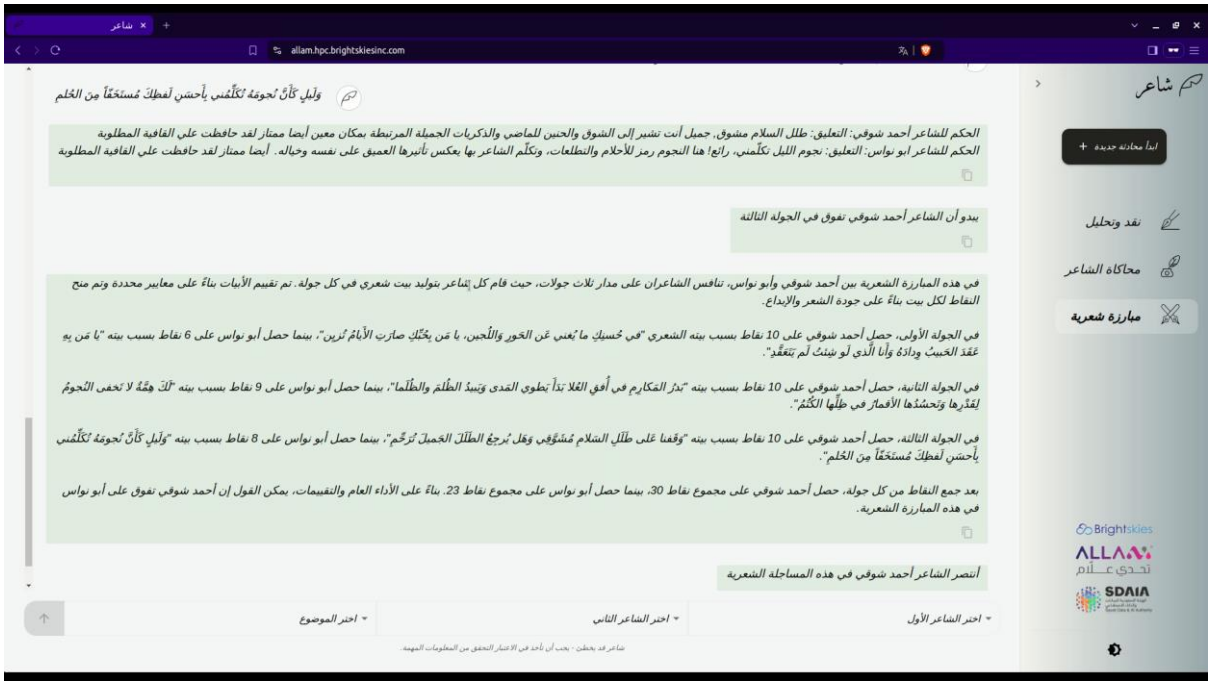


FIG11 – POETIC DEBATE EXAMPLE 1-D