Instruction-Guided Poetry Generation in Arabic and Its Dialects

Abdelrahman Sadallah[1] **Kareem Elozeiri**[1] **Mervat Abassy**[1] **Rania Elbadry**[1]
Mohamed Anwar[1] **Abed Alhakim Freihat**[1] **Preslav Nakov**[1] **Fajri Koto**[1]
[1]Mohamed bin Zayed University of Artificial Intelligence
{abdelrahman.sadallah, fajri.koto}@mbzuai.ac.ae

Abstract

Poetry has long been a central art form for Arabic speakers, serving as a powerful medium of expression and cultural identity. While modern Arabic speakers continue to value poetry, existing research on Arabic poetry within Large Language Models (LLMs) has primarily focused on analysis tasks such as interpretation or metadata prediction, e.g., rhyme schemes and titles. In contrast, our work addresses the practical aspect of poetry creation in Arabic by introducing controllable generation capabilities to assist users in writing poetry. Specifically, we present a large-scale, carefully curated instruction-based dataset in Modern Standard Arabic (MSA) and various Arabic dialects. This dataset enables tasks such as writing, revising, and continuing poems based on predefined criteria, including style and rhyme, as well as performing poetry analysis. Our experiments show that fine-tuning LLMs on this dataset yields models that can effectively generate poetry that is aligned with user requirements, based on both automated metrics and human evaluation with native Arabic speakers.[1]

1 Introduction

Poetry occupies a uniquely central position in the Arabic language and its culture (Al-Musawi, 2006). For centuries, Arabic poetry has served not only as an artistic medium, but also as a primary vehicle for preserving linguistic norms, expressing collective memory, and articulating social and emotional experiences (Jayyusi, 1977). Classical poetic traditions shaped the grammar, vocabulary, and rhetorical devices of Arabic (Zwettler, 1978; Orabi et al., 2020), while modern and dialectal poetry continue to reflect the lived realities of contemporary Arab societies (Badawi, 1975). As a result, poetry remains one of the richest and most demanding forms of written Arabic, encompassing complex structures

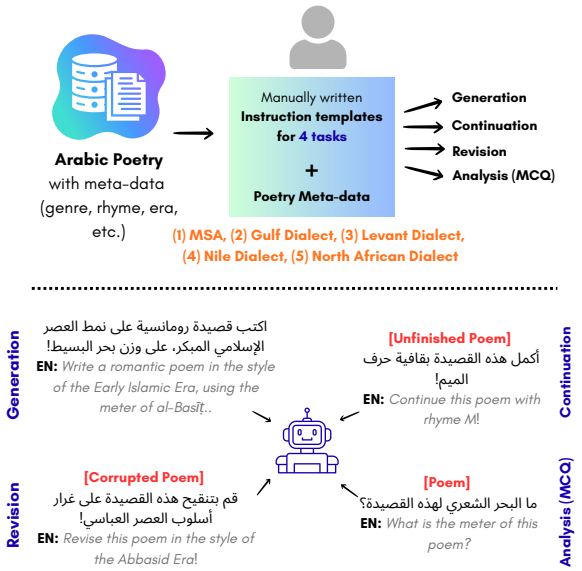

Figure 1: Instruction framework for Arabic poetry tasks using metadata and templates across five dialects, supporting generation, continuation, revision, and analysis.

such as meter, rhyme, imagery, and stylistic variation across historical eras and regional dialects.

Despite this significance, Arabic poetry remains underrepresented in current Large Language Model (LLM) research. Existing work has largely focused on analytical (Al Ghallabi et al., 2025) or classification-oriented tasks (Abbas et al., 2019; Shahriar et al., 2023; Alyafeai et al., 2023; Ahmed et al., 2025; Mutawa and Alrumaih, 2025), such as poet attribution, meter detection, or theme identification, often using relatively small or narrowly scoped datasets. In contrast, comparatively little attention has been paid to *poetry generation* in Arabic, particularly controllable generation that allows users to specify poetic constraints such as style, meter, rhyme, or dialect. This gap is especially pronounced when compared to the growing body of work on English poetry generation (Yi et al., 2018) and creative text instruction tuning (Chakrabarty et al., 2023). As a result, current LLMs often strug-

[1]The data and code available here:https://github.com/mbzuai-nlp/instructpoet-ar

Findings of the Association for Computational Linguistics: ACL 2026, pages 38761–38780

gle to produce poetry that is structurally sound, culturally appropriate, and responsive to explicit user instructions (Elkaref et al., 2022).

In this work, we aim to address this gap by framing Arabic poetry generation as an instruction-following problem and by providing the resources needed to support it at scale. We begin by aggregating a large and diverse Arabic poetry corpus from multiple publicly available sources spanning different historical eras, poetic genres, and linguistic varieties. We then unify these sources into a single, clean format with standardized verse structure and harmonized metadata, enabling consistent downstream use.

Building on this unified corpus, we construct a comprehensive instruction fine-tuning (IFT) dataset focused on Arabic poetry, as illustrated in Figure 1. We define four core tasks—*generation*, *continuation*, *revision*, and *analysis in MCQ*—covering both understanding and creative production. These tasks are further divided into 54 subtasks targeting specific poetic skills, such as identifying meter, completing missing verses, generating poetry under stylistic constraints, and reconstructing corrupted text. For each subtask, we manually design detailed instruction templates in Modern Standard Arabic (MSA) and four major dialects: Gulf, Levantine, North African, and Nile Valley. Each template includes multiple paraphrases to ensure robustness to linguistic variation and prompting style. In total, this process yields 3,220 high-quality instruction templates, forming one of the most comprehensive instruction-based resources for Arabic poetry to date.

Using this dataset, we fine-tune four large language models representing both Arabic-centric and general-purpose LLMs: Fanar (Abbas et al., 2025) and Allam (Bari et al., 2025) as Arabic-centric models, and Qwen3 (Yang et al., 2025) and LLaMA-3.1 (Grattafiori et al., 2024) as widely used multilingual models. We explore two training regimes: a standard joint training setup with randomly mixed tasks, and a curriculum-based approach in which the tasks are introduced in increasing order of difficulty. This allows us to examine whether structured exposure to poetic skills improves model performance and stability.

Our contributions are as follows: (i) We aggregate and standardize a large Arabic poetry corpus spanning multiple eras, genres, and dialects. (ii) We create a comprehensive instruction fine-tuning dataset for Arabic poetry with four core tasks and 54 subtasks, including 3,220 templates across Modern Standard Arabic and four major dialects, resulting in 1.35M training and 24.8K testing pairs. (iii) We fine-tune and evaluate four large language models under joint and curriculum-based regimes to assess their ability to handle poetic structure, stylistic constraints, and dialectal variation.

2 Related Work

Research on Arabic poetry has mainly focused on two directions: analysis and generation. However, existing approaches to generation are largely uncontrolled and limited to Modern Standard Arabic (MSA), overlooking dialectal diversity and user-specified constraints.

2.1 Poetry Analysis and Classification

The field has benefited from the release of large-scale and high-quality resources. Most notably, the **Diwan Corpus** (Al-Onazi et al., 2025) provides a massive repository of 14 million verses, offering the granular metadata required to train robust prosodic models. This shift toward data-rich approaches is evident in meter classification tasks; while early work relied on rigid rule-based systems, recent deep learning approaches utilize bidirectional RNNs to classify meter from undiacritized text with high accuracy (AlShaibani et al., 2020).

Beyond structural analysis, the focus is on expanding to semantic and cultural understanding. The release of the **Fann or Flop** benchmark (Al Ghallabi et al., 2025) represents an important shift, moving evaluation beyond simple metric accuracy to assess how well Large Language Models (LLMs) grasp metaphor and historical context. Similarly, **AraPoemBERT** (Qarah, 2024) demonstrates the value of domain-specific pretraining, setting new state-of-the-art results for nuanced tasks like poet gender and sub-meter classification.

2.2 Poetry Generation

The evolution of poetry generation from evolutionary algorithms (Manurung, 2004) to modern neural architectures (Talafha and Rekabdar, 2021) has been characterized by trade-offs between *fluency* and *controllability*.

From Sequence Modeling to Planning. Foundational neural approaches, such as Zhang and Lapata (2014) in Chinese poetry, demonstrated that RNNs could capture basic poetic forms. However, these models often suffered from thematic drift.

To counter this, Wang et al. (2016) introduced a "planning-based" architecture, separating the generation of global thematic sub-topics from line-by-line surface realization to ensure long-range coherence.

Controllability and Collaboration. Recent work emphasizes user control over the creative process. Approaches such as **PoeLM** (Ormazabal et al., 2022) and **CoPoet** (Chakrabarty et al., 2022) reduce dependence on rigid templates, allowing users to guide generation through natural language prompts or control codes. In the context of Arabic, this evolution is evident in the shift from basic LSTM-based synthesis (Hejazi et al., 2021) to rhythm-aware Transformer models. A notable example is **Tahḏīb** (Elzohbi and Zhao, 2025), which employs a byte-level transformer (Xue et al., 2022) (ByT5) to address the challenge of inserting phrases into classical verse without breaking the strict metrical structure. In contrast to Tahḏīb, which focuses on byte-level, rhythm-constrained insertion, our work centers on instruction-tuning large language models through natural language supervision, enabling a broader range of constrained co-creation tasks.

Beyond Arabic, prior work on poetry generation has explored alternative modeling and decoding strategies. (Belouadi and Eger, 2023) proposes a token-free decoding model for poetry generation in English and German, focusing on improving generation quality through modeling and decoding innovations. (Yu et al., 2024) presents a system for classical Chinese poetry generation using character-by-character generation, a strategy well-suited to the linguistic properties of Chinese. (Zhang and Eger, 2024) explores diverse poetry generation through a combination of prompting-based and training-based agents to enhance stylistic diversity. However, these approaches remain largely centered on generation or decoding strategies and on a limited set of languages. In contrast, our work shifts the focus toward instruction-tuned LLMs that support controllable, multi-task creative interaction within a unified framework for Arabic poetry, covering generation, continuation, revision, and analysis across both Modern Standard Arabic and multiple dialect groups.

3 Dataset Construction

We followed a multi-step process to prepare the dataset for instruction tuning, as illustrated in Figure 2. The pipeline begins with raw sources (PoetsGate, Adab, Ashaar), followed by data unification, metadata enrichment, and deduplication. Finally, we create instruction fine-tuning (IFT) templates for MSA and four dialects, supporting tasks such as generation, continuation, revision, and analysis.

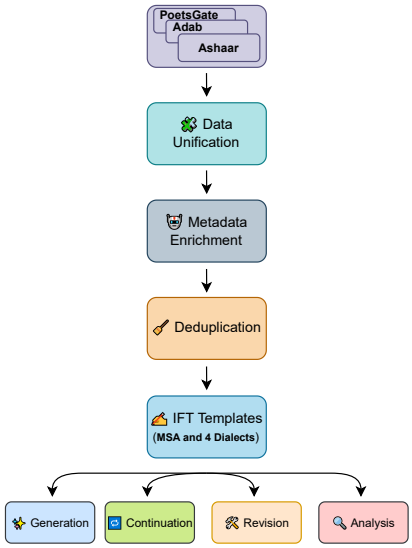

Figure 2: Overview of the dataset construction pipeline. Raw data are unified and normalized, enriched with metadata, deduplicated, and transformed into human-curated instruction-following templates. The final dataset is organized into four task-specific partitions: generation, continuation, analysis, and revision.

3.1 Raw Data

Most of the raw data comes from well-known Arabic poetry websites, such as Mawsooaa,[2] Adab,[3] Diwany,[4] Al-Diwan,[5] and PoetsGate.[6] The majority of the poems in these sources are written in Modern Standard Arabic, with some entries appearing in regional dialects. The collection spans many historical periods, including classical, medieval, and modern eras, and covers a wide range of poetic styles and voices.

3.2 Data Unification

After collecting the raw data, we unified all sources into a single clean and consistent format to prepare them for model training. Because the datasets originated from different websites and public collections with varying structures and labels, unification was essential. We first standardized the poem text by representing each verse in a fixed format, with

[2] https://poetry.dctabudhabi.ae/
[3] https://www.adab.com
[4] http://www.diwany.org/
[5] https://www.aldiwan.net/
[6] https://poetsgate.com/

one verse per line. This ensured structural consistency across all poems. We then removed poems containing only a single verse, as they provide limited learning value and introduce noise. Next, we extracted and separated metadata fields that were combined in some sources, such as era, genre, and meter. We also normalized the spelling and naming of all metadata categories to a single canonical form across datasets. For poems missing rhyme information, we applied automatic rhyme detection by analyzing verse-ending letters and assigning a rhyme label when at least 70% of verses shared the same ending

3.3 Metadata Enrichment

In addition to the metadata available from the original sources, we enriched the dataset with two new forms of semantic and syntactic metadata: keywords and keyphrases. The keywords capture the main themes or intentions of the poem (such as love, war, or pride), while the keyphrases are short spans taken from the poem that provide a concise syntactic summary of its meaning. These additional metadata fields were automatically generated using Gemini 2.5 Pro, allowing us to expand the descriptive layer of the dataset without manual annotation.

To assess the quality of the extracted metadata, we conducted a manual evaluation. We randomly sampled 100 poems and examined whether the generated keywords were relevant and representative of the poem content. Our analysis showed that 96% of the extracted keywords were of good quality, indicating that the metadata generation process is both reliable and effective for our purposes.

3.4 De-duplication

Once enrichment and unification were complete, we performed deduplication at several levels. First, we removed intra-source duplicates (i.e., identical poems within the same split). Next, we ensured there was no data leakage between training and testing splits by removing any poem from the training set that appears in the FannOrFlop benchmark (Al Ghallabi et al., 2025), which we use as our test set for evaluation. Deduplication was performed using string matching after normalization, including removing elongation and diacritics and standardizing orthographic variants. Table 1 summarizes the statistics of our clean and deduplicated Arabic poetry dataset.

Source	# Samples	Avg. Verses
Train Split		
Ashaar	123,581	19.81
PoetsGate*	112,482	15.58
Adab*	70,277	35.33
AraPoems[7]	62,963	22.01
Diwan*	38,005	22.65
Mawsooaa*	18,002	10.25
Arapoet*	1,303	9.25
Arabic Poetry Dataset	662	19.41
Arabic-Poetry-Melody	48	21.44
Adab World*	6	93.33
Other	8	24.88
TOTAL	427,337	21.39
Test Split		
FannOrFlop (Al Ghallabi et al., 2025)	6,984	17.97

Table 1: Arabic poetry data used for IFT. (*) indicates scraped sources.

3.5 Poetry IFT Data

To prepare the dataset for instruction fine-tuning, we designed four task families: generation (composing new poetry following specific constraints), continuation (completing verses), revision (fixing missing or corrupted lines), and analysis (identifying metadata such as themes or meter), each leveraging unified poem text and enriched metadata such as era, genre, meter, rhyme, keywords, and keyphrases. By grounding subtasks in metadata, the model learns both surface-level patterns and deeper poetic and linguistic features.

As shown in Table 2, generation, continuation, and analysis tasks dominate the dataset with over 427K training samples each, while revision is smaller with 68K samples, reflecting its specialized nature. The number of subtasks varies across families, ranging from 8 for revision to 19 for generation, where each subtask corresponds to a different target attribute (e.g., conditioned on rhyme, meter, or genre), ensuring diversity in instruction formats.

3.5.1 IFT Templates

For each of the main IFT tasks, we created several subtasks, and for every subtask, we designed a set of instruction templates. Each template expresses the request in a slightly different way so the model learns to understand the same instruction across multiple phrasings. This variation helps the model generalize better and handle a wider range of real-world prompts.

In Appendix A, we show the distribution of each task and its subtasks. In total, we have 246, 176,

Task	Split	Total Samples	# Subtasks
Generation	Train	427,337	19
	Test	6,984	19
Continuation	Train	427,276	11
	Test	6,984	11
Revision	Train	68,947	8
	Test	3,863	8
Analysis	Train	427,337	16
	Test	6,984	14

Table 2: Overall statistics for Arabic poetry IFT dataset across tasks and data splits.

214, and 8 templates for generation, continuation, analysis, and revision tasks, respectively. These templates form the backbone of the instruction layer used to build the IFT dataset [8].

To further increase coverage and mirror actual user behavior, we expanded the templates beyond Modern Standard Arabic (MSA). In addition to the MSA versions, we created dialectical templates representing **Gulf**, **Levant**, **Nile Valley**, and **North African** dialects. This allows the model to interact naturally with users. The dialect templates were written and revised by native speakers from each corresponding region to ensure accuracy, natural phrasing, and authentic usage. We illustrate the structure of our instruction templates in Appendix A.

3.5.2 Tasks

Generation Generation tasks prompt the model to compose new Arabic poems conditioned on metadata such as era, genre, meter, rhyme, or keywords. These tasks aim to produce authentic poetry that adheres to stylistic and structural constraints.

Continuation Continuation tasks prompt the model to extend partial poems by generating the remaining verses. To create continuation examples, poems are split at random ratios (10%–90%), exposing the model to diverse completion scenarios. This teaches the model to maintain coherence in meter, style, and meaning.

Revision Revision tasks prompt the model to fix corrupted poems, aiming to assist real users in refining poetry during the writing process. For training, we use Gemini 2.5 Pro to automatically corrupt poems and pair each corrupted version with its original as the target. Corruptions include altered wording, disrupted meter, missing verses, or syntactic errors. By learning to map corrupted text back to its clean form, the model improves its robustness to noisy inputs and internalizes a deeper understanding of the expected poetic structure. This task also complements the continuation and generation tasks, as successful revision requires strong modeling of poetic rhythm, semantics, and stylistic norms.

Analysis Analysis tasks are framed as multiple-choice questions where the model predicts a target metadata attribute (e.g., poet, era, genre, or meter) based on the poem text and optional contextual metadata. Each MCQ includes one correct answer and four randomly sampled distractors to avoid class bias.

4 Experimental Setup

4.1 Finetuning

To evaluate how our Arabic poetry dataset improves alignment with user requirements, we apply parameter-efficient LoRA fine-tuning (Hu et al., 2022) to four instruction-tuned base models: two multilingual (LLaMA-3-8B (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025)) and two Arabic-centric (ALLaM-7B–instruct (Bari et al., 2025), Fanar-1-9B (Abbas et al., 2025)). This design isolates (i) how far multilingual models can be specialized to Arabic poetry with our data, and (ii) the added benefit of starting from Arabic-optimized models.

Fine-tuning is performed on dataset using a standard causal language modeling objective applied to the concatenated instruction-output pairs for all tasks in the corpus. We train LoRA adapters for two epochs with rank $r = 64$ and scaling factor $\alpha = 32$, while keeping the base model parameters frozen. We consider two training regimes: (i) *joint training*, where all tasks are randomly shuffled and optimized together, and (ii) *curriculum learning*, where tasks are presented in a fixed order (analysis $\rightarrow$ continuation $\rightarrow$ generation $\rightarrow$ revision). This design allows us to systematically assess how curriculum structure influences the models' downstream performance on Arabic poetry understanding and generation.

Our motivation for the model's choice is encompassed in three points, which are:

1. **Multilingual vs. Arabic-centric pretraining**: LLaMA-3-8B and Qwen3-8B are strong multilingual instruction-tuned models with Arabic support, serving as high-capacity

[8] Templates are available here: https://huggingface.co/datasets/MBZUAI/instructpoet-ar

generalist baselines. ALLaM-7B–instruct and Fanar-1-9B are Arabic-focused models (morphology and script nuances), allowing us to test whether an Arabic linguistic prior improves poetic control (meter, rhyme, and style) after fine-tuning.

2. **Model size**: The selected models span a moderate parameter range (7–12B), which is large enough to capture complex poetic patterns, such as long-range rhyme schemes and stylistic consistency across multiple verses. Prior work has shown that instruction-tuning even mid-sized models (e.g., T5-11B) improves their ability to follow structural constraints such as rhyme and lexical requirements in poetry generation (Chakrabarty et al., 2022).

3. **Instruction following**: All models are instruction-tuned, matching our task format. This lets fine-tuning focus on poetry-specific skills (form constraints, register, and thematic fidelity) rather than learning generic instruction adherence from scratch.

4.2 Evaluation

To systematically assess the performance of our system across all subtasks, we design a comprehensive evaluation framework that combines LLM-as-a-judge and lm-eval-harness, automated assessment, and human evaluation.

Evaluation with LLM-as-a-Judge We adopt *Gemini 2.5 Flash* as an external evaluation model in order to avoid bias toward any of the four model families evaluated and to ensure more reliable and consistent judgments. We formulate a dedicated evaluation prompt that outlines the expected criteria and incorporates relevant task meta-data when necessary for each Generation, Continuation and Revision subtask, developing 38 task-specific evaluation prompts, each tailored to the unique requirements and constraints of its corresponding subtask. We evaluate both baseline and fine-tuned versions of all models.

Given an input instance and the corresponding system-generated output, the evaluator model produces aspect-wise assessments across several dimensions. These include **compliance**, which measures the adherence to explicit task conditions and constraints; **fluency**, which captures grammaticality, readability, and linguistic naturalness; **coherence**, which evaluates structural consistency, topical alignment, and logical flow; and **poetic quality**, a dimension particularly relevant to creative-generation tasks such as poetry, assessing the use of imagery, metaphor, stylistic devices, and overall poetic expression.

Each aspect is rated on a 1–5 Likert scale, and we compute an overall score for each sample by averaging the four aspect-level ratings. For stylistic or poetry-generation subtasks, we additionally provide the judge model with meta-information (e.g., meter, theme, and required rhetorical devices) to enable more targeted and accurate evaluation. All 38 subtasks are evaluated using our task-specific LLM-judge prompts. The automated evaluation covers every subtask, ensuring consistent, model-agnostic assessment across diverse task types.

LM-Eval-Harness As we mentioned in 3.5.2, we designed the analysis task as an MCQ task. To evaluate the models on this task, we created a task in the lm-eval-harness (Gao et al., 2023) framework formatted as a completion task. We append each answer choice to the instruction, and take the completion with the highest likelihood.

Human Evaluation To assess the quality of generated Arabic poetry, we conducted a comprehensive human evaluation study on the generation task. We randomly sampled 100 input prompts from our test set and generated poetry outputs using four models: (1) ALLaM-7B-Instruct-preview (base), (2) ALLaM-7B-Instruct-preview fine-tuned, (3) Qwen3-8B (base), and (4) Qwen3-8B fine-tuned. This resulted in 400 generation samples (100 inputs × 4 models).

We recruited two Arabic-speaking annotators with experience in Arabic poetry and literary arts to evaluate the generated poems. To ensure an unbiased assessment, we conducted a blind evaluation in which the model identities were anonymized. The evaluation samples were shuffled to avoid ordering effects.

Each annotator independently rated all 400 samples on a 5-point Likert scale across the four criteria defined in 4.2:

5 Results

5.1 Automatic Metric Evaluation

Generation, Continuation, and Revision Table 3 reports the average Gemini-2.5-Flash scores (1–5) across generation, continuation, and revision, broken down by dialect variant. Overall, there is no

Model	MSA			Gulf			North Africa			Levant			Nile Valley		
	G	C	R	G	C	R	G	C	R	G	C	R	G	C	R
ALLaM-7B–instruct	1.74	1.46	1.27	2.17	1.36	1.27	2.04	1.36	1.29	1.96	1.32	1.31	1.86	1.41	1.25
ALLaM-7B–instruct (Curriculum)	**2.44**	2.00	1.56	**2.72**	**1.97**	**1.64**	**2.66**	1.96	**1.64**	2.57	1.88	**1.67**	2.42	1.93	**1.60**
ALLaM-7B–instruct (Random)	**2.44**	**2.03**	**1.57**	**2.72**	**1.97**	1.60	2.57	**2.03**	1.62	**2.59**	**1.98**	1.66	**2.44**	**1.98**	1.59
Fanar-1-9B	1.28	1.07	1.08	1.25	1.06	1.07	1.28	1.11	1.11	**2.06**	1.05	1.07	1.24	1.03	1.09
Fanar-1-9B (Curriculum)	**1.50**	**1.42**	1.29	**1.57**	**1.43**	1.36	**1.60**	**1.41**	1.29	1.57	**1.37**	1.36	**1.51**	**1.42**	1.28
Fanar-1-9B (Random)	1.46	1.40	**1.32**	1.50	1.38	**1.37**	1.54	1.39	**1.37**	1.54	1.36	**1.45**	1.46	1.37	**1.38**
LLaMA-3-8B	1.50	1.18	1.06	1.62	1.19	1.09	1.50	1.19	1.05	1.48	1.18	1.12	1.41	1.20	1.07
LLaMA-3-8B (Curriculum)	1.82	1.97	**1.59**	**1.90**	1.92	**1.70**	**1.88**	**1.97**	1.70	1.88	1.88	**1.78**	1.80	1.85	1.59
LLaMA-3-8B (Random)	**1.90**	**2.06**	**1.59**	1.89	**1.95**	1.67	1.87	**1.97**	**1.72**	**1.98**	**2.00**	1.74	**1.82**	**1.96**	**1.64**
Qwen-3-8B	**1.84**	1.41	1.38	**2.24**	1.38	1.34	**1.91**	1.35	1.49	2.02	1.39	1.35	1.75	1.35	1.38
Qwen-3-8B (Curriculum)	1.76	1.98	1.51	1.77	1.94	**1.57**	1.84	1.97	1.53	1.79	1.90	**1.64**	1.75	1.94	**1.58**
Qwen-3-8B (Random)	1.79	**2.05**	**1.52**	1.84	**2.01**	1.55	1.82	**1.99**	**1.58**	**1.80**	**1.98**	1.63	**1.76**	**1.96**	1.51

Table 3: LLM-as-a-Judge evaluation across models, training types, and Arabic dialects. G = Generation, C = Continuation, R = Revision.

single dominant dialectal trend across all models. Instead, different models exhibit different levels of proficiency across MSA and the dialect variants, with performance varying modestly depending on the model family and the evaluated task. This suggests that fine-tuning improves performance broadly across dialects, but without revealing a uniform pattern in which one dialect is consistently easiest or hardest for all models.

In contrast, Table 4 breaks down performance by task and evaluation aspect rather than by dialect, reporting scores for compliance, fluency, coherence, and poetic quality. A clear trend emerges across all model families: performance is highest for generation, lower for continuation, and lowest for revision, reflecting the increasing difficulty of the tasks. Generating poetry from scratch gives the model greater freedom to satisfy poetic and stylistic constraints, whereas continuation requires maintaining coherence, style, and often meter with an already provided poetic context. Revision is the most challenging setting, since the model must restore corrupted poetry while preserving meaning and recovering structural constraints. This pattern is also consistent with real-world writing scenarios, where continuing or repairing an existing poem is often more constrained than composing a new one. Prior work likewise highlights that maintaining coherence across poetic context is challenging and often requires explicit planning mechanisms, while phrase insertion under metrical constraints restricts word choice and sentence construction (Wang et al., 2016; Elzohbi and Zhao, 2025).

Analysis (MCQ) Table 5 reports the average performance across all analysis tasks [9]. Across all model families, both curriculum-based and random corruption strategies substantially outperform the base models, indicating that structured corruption is effective for improving analytical reasoning. The strongest gains are observed for LLaMA-3-8B and Qwen3-8B , where accuracy increases by more than 30 percentage points relative to their respective baselines. In most cases, random corruption achieves marginally higher performance than curriculum learning, though the differences are small.

5.2 Human Evaluation

Inter-Annotator Agreement We assessed inter-annotator reliability using three complementary metrics: Pearson correlation, Spearman correlation, and quadratic weighted kappa (Cohen, 1968). Quadratic weighted kappa is particularly appropriate for ordinal Likert-scale data, as it accounts for chance agreement and penalizes larger disagreements more heavily than smaller ones.

Overall agreement across all evaluation criteria was substantial, with Pearson ($r = 0.58$), Spearman ($\rho = 0.58$), and quadratic weighted kappa ($\kappa = 0.57$) yielding nearly identical values, indicating robust and consistent measurement across metrics. Agreement was highest for *Fluency* ($r = 0.65$, $\rho = 0.65$, $\kappa = 0.65$), reflecting the relatively objective nature of assessing linguistic correctness and rhythmic patterns. *Poetic Quality* also demonstrated strong agreement ($\kappa = 0.59$), followed by *Coherence* ($\kappa = 0.54$). The lowest agreement was observed for *Compliance* ($\kappa = 0.51$), suggesting that judgments regarding adherence to constraints involve a higher degree of subjectivity.

These agreement levels fall within the expected range (0.50–0.65) for subjective evaluation of creative text (Artstein and Poesio, 2008) and corre-

[9] A detailed breakdown of performance for each individual analysis task is provided in Table 13.

Model	Generation					Continuation					Revision				
	Comp	Flue	Cohe	Poet	Over	Comp	Flue	Cohe	Poet	Over	Comp	Flue	Cohe	Poet	Over
ALLaM-7B-Instruct	1.96	2.09	2.14	1.70	1.97	1.48	1.46	1.33	1.27	1.38	1.14	1.41	1.31	1.25	1.28
ALLaM-7B-Instruct (Cur)	2.45	**2.79**	**2.78**	**2.29**	**2.58**	2.14	2.06	1.83	1.76	1.95	**1.43**	**1.84**	**1.64**	**1.58**	**1.62**
ALLaM-7B-Instruct (Rand)	**2.46**	2.77	2.75	2.28	2.56	**2.20**	**2.10**	**1.89**	**1.80**	**2.00**	**1.43**	1.81	1.63	1.57	1.61
Fanar-1-9B	1.04	**1.91**	1.34	1.02	1.40	1.07	1.08	1.07	1.04	1.06	1.04	1.11	1.11	1.05	1.08
Fanar-1-9B (Cur)	1.73	1.61	**1.36**	**1.51**	1.55	**1.63**	**1.44**	**1.26**	**1.30**	**1.41**	1.25	1.42	1.28	1.27	1.30
Fanar-1-9B (Rand)	**1.66**	1.55	1.32	1.46	1.50	1.61	1.41	1.24	1.27	1.38	**1.30**	**1.50**	**1.35**	**1.36**	**1.38**
Llama-3-8B	1.48	1.45	1.80	1.30	1.51	1.23	1.19	1.26	1.06	1.19	1.05	1.10	1.10	1.05	1.08
Llama-3-8B (Cur)	2.04	1.99	1.70	1.71	1.86	2.23	2.03	1.71	1.71	1.92	1.51	1.87	**1.68**	**1.62**	**1.67**
Llama-3-8B (Rand)	**2.08**	**2.01**	**1.72**	**1.74**	**1.89**	**2.32**	**2.09**	**1.78**	**1.77**	**1.99**	**1.53**	**1.88**	**1.68**	1.60	1.67
Qwen-3-8B	**2.01**	**1.94**	**2.23**	**1.68**	**1.96**	1.44	1.45	1.40	1.22	1.38	1.28	1.45	1.50	1.34	1.39
Qwen-3-8B (Cur)	1.96	1.89	1.62	1.65	1.78	2.21	2.06	1.76	1.75	1.94	**1.43**	**1.76**	**1.54**	**1.51**	**1.56**
Qwen-3-8B (Rand)	1.96	1.93	1.67	1.66	1.80	**2.24**	**2.14**	**1.83**	**1.79**	**2.00**	**1.43**	**1.76**	**1.54**	**1.51**	**1.56**

Table 4: Results across different models and tasks. Comp = Compliance, Flue = Fluency, Cohe = Coherence, Poet = Poetic Quality, Over = Overall.

Model	Analysis Accuracy
ALLaM-7B–instruct	66.2
ALLaM-7B–instruct (Curriculum)	77.8
ALLaM-7B–instruct (Random)	**78.1**
Fanar-1-9B	51.2
Fanar-1-9B (Curriculum)	**60.5**
Fanar-1-9B (Random)	60.1
LLaMA-3-8B	47.0
LLaMA-3-8B (Curriculum)	78.7
LLaMA-3-8B (Random)	**79.0**
Qwen3-8B	44.6
Qwen3-8B (Curriculum)	**77.4**
Qwen3-8B (Random)	77.2

Table 5: Analysis task accuracy (%). **Bold** indicates the best result within each model family. The results are averaged over all subtasks.

spond to moderate to substantial agreement according to Landis and Koch (Landis and Koch, 1977), supporting the reliability of the human annotations used in this study.

Results and Analysis Table 6 presents the averaged scores across both annotators for each model. All observed differences between models were statistically significant (ANOVA, $p < 0.0001$ for all criteria).

Overall, fine-tuning substantially improved the performance of both base models. ALLaM-7B increased by 0.97 points (+32%), rising from 3.02 to 3.99, while Qwen3-8B exhibited a larger relative gain of 1.42 points (+63%), improving from 2.24 to 3.66. This confirms the effectiveness of domain-specific fine-tuning for Arabic poetry generation, with Qwen3-8B benefiting more strongly due to its weaker baseline performance on Arabic poetic tasks.

Model	Comp	Flue	Cohe	Poet	Over
ALLaM-7B (Base)	3.09	3.10	3.02	2.87	3.02
ALLaM-7B (FT)	**3.93**	**4.20**	**4.00**	**3.82**	**3.99**
Qwen3-8B (Base)	2.48	2.13	2.46	1.91	2.24
Qwen3-8B (FT)	**3.66**	**3.86**	**3.54**	**3.58**	**3.66**

Table 6: Human evaluation results averaged across two annotators. Scores are on a 1–5 scale (higher is better). All pairwise differences between models are statistically significant ($p < 0.0001$). Comp = Compliance, Flue = Fluency, Cohe = Coherence, Poet = Poetic Quality, Over = Overall.

In terms of model comparison, the fine-tuned ALLaM-7B achieved the highest overall score (3.99/5.0) and ranked first across all four evaluation criteria—compliance, fluency, coherence, and poetic quality—demonstrating its superior capability in Arabic poetry generation. The fine-tuned Qwen3-8B followed with an overall score of 3.66/5.0, trailing ALLaM-7B by 0.33 points. Among the base models, ALLaM-7B (3.02) significantly outperformed Qwen3-8B (2.24) by 0.78 points, indicating that ALLaM has a stronger foundational understanding of Arabic language and poetic structure.

A criterion-specific analysis reveals that Fluency consistently received the highest scores across all models, suggesting that generating linguistically correct and rhythmically sound Arabic text is a relative strength of current large language models. Conversely, Poetic Quality was the lowest-scoring criterion across models, highlighting the difficulty of capturing the artistic depth and aesthetic nuances of Arabic poetry. The fine-tuned ALLaM-7B performed particularly well in Fluency (4.20/5.0) and Coherence (4.00/5.0), approaching human-level

quality in these dimensions. However, even the best-performing model achieved only 3.82/5.0 in Poetic Quality, indicating substantial room for improvement in modeling poetic creativity and artistic expression.

Finally, ANOVA tests confirmed that all performance differences between models were statistically significant ($p < 0.0001$) across every evaluation criterion, including the overall score. This demonstrates that the observed improvements are systematic rather than due to random variation, validating both the effectiveness of the fine-tuning approach and the superiority of the ALLaM-7B model for Arabic poetry generation.

5.3 Analysis

Further **subtask-level analysis** reveals that random fine-tuning consistently improves performance across generation, continuation, and revision subtasks. Fine-tuned models show the strongest improvements in tasks requiring precise control and structure, particularly compositional generation with multiple constraints, rhyme-focused revision demanding prosodic awareness, and continuation tasks that require maintaining long-range poetic consistency. Analysis across dialects revealed models achieve highest performance on MSA and major regional varieties (Gulf, Levantine, North African), with fine-tuning enhancing robustness to dialectal variation in all settings. Detailed results and figures are provided in the Appendix G.

We observe that LLM-based evaluation yields consistently lower scores than human judgments. This discrepancy is in line with prior work showing that LLM evaluators tend to apply stricter and more consistent criteria, particularly penalizing deviations in structure, form, or constraint satisfaction, whereas human evaluators may allow for greater stylistic variation and interpretive flexibility (Zhu et al., 2025). This suggests that automatic evaluation may underestimate perceived quality in creative tasks such as poetry, especially when outputs intentionally deviate from rigid conventions.

6 Conclusion and Future Work

We introduced an instruction-following framework for Arabic poetry that treats poetic creation and understanding as controllable, user-driven tasks. Our dataset spans Modern Standard Arabic and four major dialect groups, covering four core tasks: generation, continuation, revision, and analysis. By grounding instructions in rich metadata and curated templates, we enable models to handle structural, stylistic, and dialectal complexities such as meter, rhyme, genre, and era. Our experiments across automatic evaluation measures, LLM-as-a-judge, and human evaluation show that instruction fine-tuning significantly improves analytical accuracy and overall output quality.

Our work lays the foundation for research at the intersection of Arabic NLP and literary scholarship, moving toward practical collaboration between language models and Arabic poetic expression.

In future work, we plan to explore deeper literary engagement, such as critique, interpretation, and stylistic analysis, while expanding coverage to contemporary poetry, free verse, and dialectical compositions. Another direction is to enrich the dataset with contemporary poetry, free verse, and natively dialectal compositions, which are currently underrepresented due to data availability constraints but are essential for modeling modern poetic practice. We also plan to investigate full fine-tuning, larger model scales, and prosody-aware objectives to further enhance performance.

Limitations

LoRa vs. Full Fine-tuning Our fine-tuning approach relies on LoRA-based parameter-efficient adaptation rather than full model fine-tuning. While LoRA offers practical advantages in terms of computational efficiency and accessibility, it may limit the model's capacity to fully internalize complex poetic structures and stylistic nuances, particularly given the large scale of our instruction dataset. Full fine-tuning may therefore be more effective for this task, and exploring this direction is an important avenue for future work.

Smaller Models Our experiments focus on a limited set of mid-sized language models. We do not evaluate smaller model variants, which prevents a systematic study of how model size influences performance on Arabic poetry tasks. Examining a broader range of model sizes would help clarify the trade-offs between model capacity, efficiency, and poetic competence.

Modern Poetry Although the dataset covers multiple eras and includes dialectal instructions, the poetic content itself is largely historical and written in Modern Standard Arabic, reflecting the nature of available public sources. A more diverse dataset

containing contemporary poetry and native dialectal poems could better capture modern linguistic usage and stylistic variation, and may lead to further improvements in model performance.

Ethics and Broader Impact

The dataset used in this work consists of publicly available Arabic poetry collected from open literary sources. No private, personal, or sensitive information was created, introduced, or used, and all data processing was conducted solely for research purposes in line with standard academic practice.

While our models are designed to assist with poetic generation and analysis, there is a risk that generated content could be misinterpreted as authentic human-authored poetry or incorrectly attributed to specific authors. We therefore emphasize that outputs should be used with appropriate disclosure and not presented as verified historical or literary artifacts.

Another limitation stems from data coverage: publicly available poetry is skewed toward well-documented, canonical forms and may underrepresent contemporary, dialectal, or community-sourced poetic traditions. This could impact model behavior on underrepresented genres. We encourage future work to broaden dataset diversity and continue evaluating models on socially and culturally inclusive benchmarks.

We expect the primary impact of this work to be positive, supporting research, education, and creative exploration in Arabic language and literature.

References

Mourad Abbas, Mohamed Lichouri, and Ahmed Zeggada. 2019. Classification of arabic poems: from the 5th to the 15th century. In *New Trends in Image Analysis and Processing – ICIAP 2019: ICIAP International Workshops, BioFor, PatReCH, e-BADLE, DeepRetail, and Industrial Session, Trento, Italy, September 9–10, 2019, Revised Selected Papers*, page 179–186, Berlin, Heidelberg. Springer-Verlag.

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsaneddin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur A. Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed K. Elmagarmid, Mohamed Y. Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, and 22 others. 2025. Fanar: An Arabic-centric multimodal generative AI platform. *ArXiv preprint*, abs/2501.13944.

Munef Abdullah Ahmed, Raed Abdulkareem Hasan, Mostafa Abdulghafoor Mohammed, Peter Mwangi, and Tirus Muya. 2025. Classification arabic language (classical Arabic poetry, al-hur Arabic poetry and prose) using machine learning. *EDRAAK*, pages 94–104.

Wafa Al Ghallabi, Ritesh Thawkar, Sara Ghaboura, Ketan Pravin More, Omkar Thawakar, Hisham Cholakkal, Salman Khan, and Rao Muhammad Anwer. 2025. Fann or flop: A multigenre, multiera benchmark for Arabic poetry understanding in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20224–20244, Suzhou, China. Association for Computational Linguistics.

Muhsin J Al-Musawi. 2006. *Arabic poetry: Trajectories of modernity and tradition*. Routledge, London, UK.

Badriyya B. Al-Onazi, Wadee A. Nashir, and Asma A. Al-Shargabi. 2025. Diwan: constructing the largest annotated corpus for arabic poetry. *IEEE Access*, 13:58927–58941.

Maged Saeed AlShaibani, Zaid Alyafeai, and Irfan Ahmad. 2020. Meter classification of arabic poems using deep bidirectional recurrent neural networks. *Pattern Recognit. Lett.*, 136:1–7.

Zaid Alyafeai, Maged Saeed AlShaibani, and Moataz Ahmad. 2023. Ashaar: Automatic analysis and generation of arabic poetry using deep learning approaches. *ArXiv preprint*, abs/2307.06218.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Muhammad Mustafa Badawi. 1975. *A critical introduction to modern Arabic poetry*. Cambridge University Press, Cambridge, UK.

M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan Alrashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Majed Alrubaian, Ali Alammari, Zaki Alawami, and 2 others. 2025. ALLaM: Large Language Models for Arabic and English. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

Jonas Belouadi and Steffen Eger. 2023. ByGPT5: End-to-end style-conditioned poetry generation with token-free language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7364–7381, Toronto, Canada. Association for Computational Linguistics.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. *Help me write a poem*: Instruction tuning as a

vehicle for collaborative poetry writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6848–6863, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tuhin Chakrabarty, Vishakh Padmakumar, He He, and Nanyun Peng. 2023. Creative natural language generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 34–40, Singapore. Association for Computational Linguistics.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Nehal Elkaref, Mervat Abu-Elkheir, Maryam ElOraby, and Mohamed Abdelgaber. 2022. Generating classical Arabic poetry using pre-trained models. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 53–62, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mohamad Elzohbi and Richard Zhao. 2025. Tahḏīb: A rhythm-aware phrase insertion for classical Arabic poetry composition. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 194–202, Suzhou, China. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. A framework for few-shot language model evaluation.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models.

Hani D. Hejazi, Ahmed A. Khamees, Muhammad Turki Alshurideh, and Said A. Salloum. 2021. Arabic text generation: Deep learning for poetry synthesis. In *Advances in Machine Learning and Data Analytics*. Springer.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Salma Jayyusi. 1977. *Trends and Movements in Modern Arabic Poetry*. Brill, Leiden, The Netherlands.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Hisar Maruli Manurung. 2004. *An Evolutionary Algorithm Approach to Poetry Generation*. Ph.D. thesis, University of Edinburgh.

A. M. Mutawa and Ayshah Alrumaih. 2025. Determining the meter of classical arabic poetry using deep learning: a performance analysis. *Frontiers in Artificial Intelligence*, 8.

Mariam Orabi, Hozayfa El Rifai, and Ashraf Elnagar. 2020. Classical Arabic poetry: classification based on era. In *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–6, Antalya, Turkey.

Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. 2022. PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3655–3670, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Faisal Qarah. 2024. AraPoemBERT: a pretrained language model for arabic poetry analysis. *arXiv preprint arXiv:2403.12392*.

Sakib Shahriar, Noora Al Roken, and Imran Zualkernan. 2023. Classification of arabic poetry emotions using deep learning. *Computers*, 12(5).

Sameerah Talafha and Banafsheh Rekabdar. 2021. Poetry generation model via deep learning incorporating extended phonetic and semantic embeddings. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 48–55, Laguna Hills, CA, USA.

Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1051–1060, Osaka, Japan. The COLING 2016 Organizing Committee.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report.

Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153, Brussels, Belgium. Association for Computational Linguistics.

Chengyue Yu, Lei Zang, Jiaotuan Wang, Chenyi Zhuang, and Jinjie Gu. 2024. CharPoet: A Chinese classical poetry generation system based on token-free LLM. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 315–325, Bangkok, Thailand. Association for Computational Linguistics.

Ran Zhang and Steffen Eger. 2024. LLM-based multi-agent poetry generation in non-cooperative environments. *arXiv preprint arXiv:2409.03659*.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, Doha, Qatar. Association for Computational Linguistics.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. Judgelm: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Michael Zwettler. 1978. *The oral tradition of classical Arabic poetry: its character and implications*. Ohio State University Press, Columbus.

A Additional Data Statistics

We provide additional statistics of the dataset, covering both corpus-level properties and instruction-level distributions. At the corpus level, we analyze the distribution of key attributes such as poetic meter, poet era, and genre. Table 7 reports the top 10 most frequent values for each category.

At the instruction level, we summarize the composition of the IFT dataset across tasks and subtasks. Table 2 presents the overall distribution of samples across the main task families. Tables 8, 9, 10, and 11 provide a detailed breakdown of the individual subtasks for analysis, generation, continuation, and revision, respectively.

B IFT Examples

In Table 12 we show one template from each different task. In Figure 3, we show different samples for our tasks, highlighting the input and output.

C Human Annotations

For the template generation task, we relied on four human annotators, each representing a different regional Arabic dialect. All annotators are native speakers of the respective dialects they contributed to.

For the human evaluation study, we employed two native Arabic speakers with demonstrated familiarity with Arabic poetry and literary texts.

All annotators were provided with detailed task-specific guidelines prior to annotation. They were informed about the scope and expected workload of their tasks in advance. Compensation was determined based on the amount of work completed.

D Detailed Analysis Task Results

Table 13 presents the detailed performance across individual analysis subtasks. Fine-tuning yields substantial improvements across all model families, with near-perfect performance on structurally grounded tasks such as meter and rhyme prediction. In contrast, tasks requiring deeper semantic understanding, such as genre and poet identification from text, remain more challenging.

E Inference Parameters

In tables 14 & 15, we show the metadata generation prompts and the generation hyperparameters, respectively, used during inference.

F Automatic Evaluation

In addition to human and LLM-based evaluation, we report automatic metrics to assess models' adherence to structural constraints and content fidelity. Table 16 summarizes automatic evaluation metrics for each model family and its fine-tuned variants. It reports multiple-choice accuracy for the analysis task, and semantic similarity (BERTScore), lexical overlap (ROUGE-L), and rhyme adherence for the corruption, generation, and continuation tasks. For generation, it additionally includes a key-phrase inclusion score to measure content anchoring.

In this table, we focus on rhyme adherence and BERTScore to evaluate whether models generate poems that both follow the required structural constraints and remain semantically aligned with reference outputs. Across all model families, fine-tuning leads to substantial improvements in rhyme

Value	Count	%
Meter		
بحر الطويل (Al-Ṭawīl)	15164	20.31
بحر الكامل (Al-Kāmil)	12017	16.10
بحر البسيط (Al-Basīṭ)	9162	12.27
بحر الوافر (Al-Wāfir)	6376	8.54
بحر الخفيف (Al-Khafīf)	5553	7.44
بحر السريع (As-Sarīᶜ)	3544	4.75
بحر الرجز (Ar-Rajaz)	3298	4.42
بحر المتقارب (Al-Mutaqārib)	3002	4.02
بحر الرمل (Ar-Raml)	2905	3.89
بحر المجتث (Al-Mujtath)	1897	2.54
Poet Era		
العصر الحديث (Modern)	82487	37.17
العصر العباسي (Abbasid)	50125	22.59
العصر المملوكي (Mamluk)	22298	10.05
العصر العثماني (Ottoman)	17011	7.66
العصر الأندلسي (Andalusian)	13507	6.09
العصر الأموي (Umayyad)	9207	4.15
العصر الفاطمي (Fatimid)	6893	3.11
العصر الأيوبي (Ayyubid)	5736	2.58
المخضرمون (Mukhaḍramūn)	5477	2.47
العصر الجاهلي (Pre-Islamic)	4906	2.21
Genre		
عامه (General)	14139	22.89
قصيره (Short)	7171	11.61
مدح (Praise)	4947	8.01
رومانسيه (Romantic)	4442	7.19
هجاء (Satire)	2923	4.73
حزينه (Sad)	2163	3.50
عتاب (Reproach)	1575	2.55
دينيه (Religious)	1244	2.01
رثاء (Elegy)	1239	2.01
غزل (Ghazal)	1189	1.93

Table 7: Top 10 most frequent values for poetic meter, poet era, and genre.

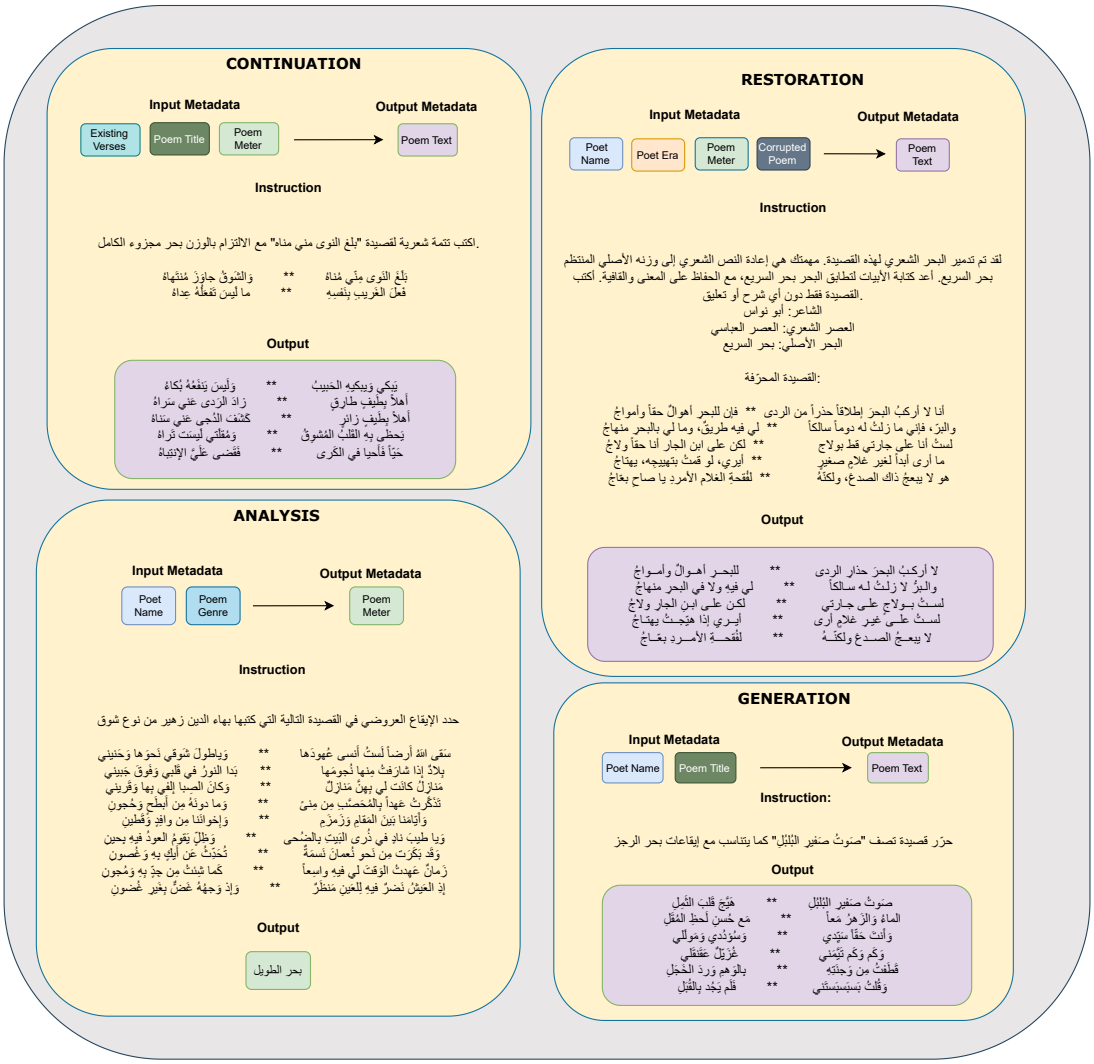

Figure 3: Example instances from the IFT dataset for each of the four main poetry tasks. Each example illustrates the instruction format along with the relevant input fields and expected output.

Subtask (Input → Output)	Samples Count
Train Split	
poem_text → poet_name	142,583
poem_text → poem_title	88,464
poem_text → keywords	58,700
poem_text → poet_era	32,892
poet_name → poet_era	26,142
poet_name, poem_text → poet_era	18,055
poet_name, poem_text → rhyme	15,389
poem_text → meter	11,168
poem_text → genre	6,331
poet_name, → meter	6,306
poet_name, poem_text → meter	5,132
poet_name → genre	4,843
poet_name, poem_text → genre	3,889
poet_name, poem_text, genre → meter	1,241
Test Split	
poem_text → poet_name	1,042
poem_text → poem_title	876
poem_text → meter	745
poem_text → poet_era	638
poem_text → keywords	621
poem_text → genre	550
poet_name → poet_era	476
poet_name → genre	416
poet_name → meter	364
poet_name, poem_text → genre	321
poet_name, poem_text → meter	284
poet_name, poem_text → rhyme	226
poet_name, poem_text → poet_era	225
poet_name, poem_text, genre → meter	200

Table 8: Detailed statistics per subtask for the *Analysis* task in the Arabic poetry IFT dataset.

Subtask (Input → Output)	Samples Count
Train Split	
poem_title, → poem_text	99,852
poet_name, → poem_text	85,765
poem_title, poet_name → poem_text	52,501
keywords, → poem_text	45,128
key_phrases, → poem_text	34,584
poet_name, poet_era → poem_text	21,509
rhyme, → poem_text	18,398
poet_era, poem_title → poem_text	16,781
meter, → poem_text	11,761
poet_name, rhyme → poem_text	10,290
poem_title, rhyme → poem_text	7,594
poet_name, meter → poem_text	6,879
genre, → poem_text	6,067
poet_name, genre → poem_text	3,142
poem_title, genre → poem_text	2,378
rhyme, meter → poem_text	1,348
genre, poet_era → poem_text	1,337
poem_title, meter → poem_text	1,226
genre, meter → poem_text	797
Test Split	
poem_title → poem_text	849
poet_name → poem_text	728
meter → poem_text	630
genre → poem_text	549
keywords → poem_text	513
key_phrases → poem_text	443
rhyme → poem_text	430
poem_title, poet_name → poem_text	428
poet_era, poem_title → poem_text	376
poet_name, poet_era → poem_text	333
poet_name, meter → poem_text	295
poet_name, genre → poem_text	262
poem_title, meter → poem_text	210
poet_name, rhyme → poem_text	208
poem_title, genre → poem_text	187
poem_title, rhyme → poem_text	148
genre, poet_era → poem_text	139
rhyme, meter → poem_text	131
genre, meter → poem_text	125

Table 9: Detailed statistics per subtask for the *Generation* task in the Arabic poetry IFT dataset.

Subtask (Input → Output)	Samples Count
Train Split	
existing_verses, poem_title → poem_continuation	138,055
existing_verses, poem_title, poet_name → poem_continuation	77,174
existing_verses, keywords → poem_continuation	64,862
existing_verses, poet_era → poem_continuation	49,948
existing_verses, meter → poem_continuation	25,602
existing_verses, rhyme → poem_continuation	24,826
existing_verses, poem_title, poet_era → poem_continuation	19,147
existing_verses, poem_title, rhyme → poem_continuation	12,453
existing_verses, genre → poem_continuation	8,082
existing_verses, poem_title, genre → poem_continuation	4,164
existing_verses, poem_title, meter → poem_continuation	2,963
Test Split	
existing_verses, poem_title → poem_continuation	1,218
existing_verses, meter → poem_continuation	999
existing_verses, poet_era → poem_continuation	833
existing_verses, keywords → poem_continuation	730
existing_verses, genre → poem_continuation	702
existing_verses, rhyme → poem_continuation	536
existing_verses, poem_title, meter → poem_continuation	521
existing_verses, poem_title, poet_name → poem_continuation	449
existing_verses, poem_title, poet_era → poem_continuation	391
existing_verses, poem_title, genre → poem_continuation	340
existing_verses, poem_title, rhyme → poem_continuation	265

Table 10: Detailed statistics per subtask for the *Continuation* task in the Arabic poetry IFT dataset.

Subtask (Corruption Type)	Samples Count
Train Split	
rhyme_structure	11,196
full_style	9,896
rhyme_substitution	9,834
rhyme_content	9,822
era_corruption	8,915
meter_transformation	7,294
meter_destruction	7,292
meter_inconsistency	4,698
Test Split	
meter_destruction	483
rhyme_structure	483
rhyme_content	483
meter_inconsistency	483
rhyme_substitution	483
era_corruption	483
meter_transformation	483
full_style	482

Table 11: Detailed statistics per subtask for the *Corruption (Restoration)* task in the Arabic poetry IFT dataset.

adherence, indicating better compliance with structural constraints. Improvements are particularly pronounced for generation tasks, where baseline models struggle to follow rhyme but fine-tuned variants achieve large gains. In addition, most models show consistent improvements in BERTScore, suggesting that gains in structure do not come at the expense of semantic quality.

It is important to note that many poetic attributes, such as meter and poet era, are difficult to evaluate using rule-based methods. As a result, our evaluation framework primarily relies on human judgments and LLM-based evaluators for comprehensive assessment. Nevertheless, the automatic results reported here provide complementary evidence that fine-tuning improves both structural fidelity and content quality.

G Analysis

Figures 4–6 present the results of an LLM-as-a-judge evaluation across subtask variation on three Arabic poetry tasks: **generation**, **revision**, and **continuation**. Each figure compares a base model against its best-performing fine-tuned variant. Across all tasks, fine-tuning consistently improves performance, with the magnitude of gains varying by task complexity and training strategy.

Generation Subtasks Figure 4 reports results for **generation sub-tasks** on generations by ALLAM-7B-INSTRUCT, comparing the base model with its random fine-tuned variant. The fine-tuned model

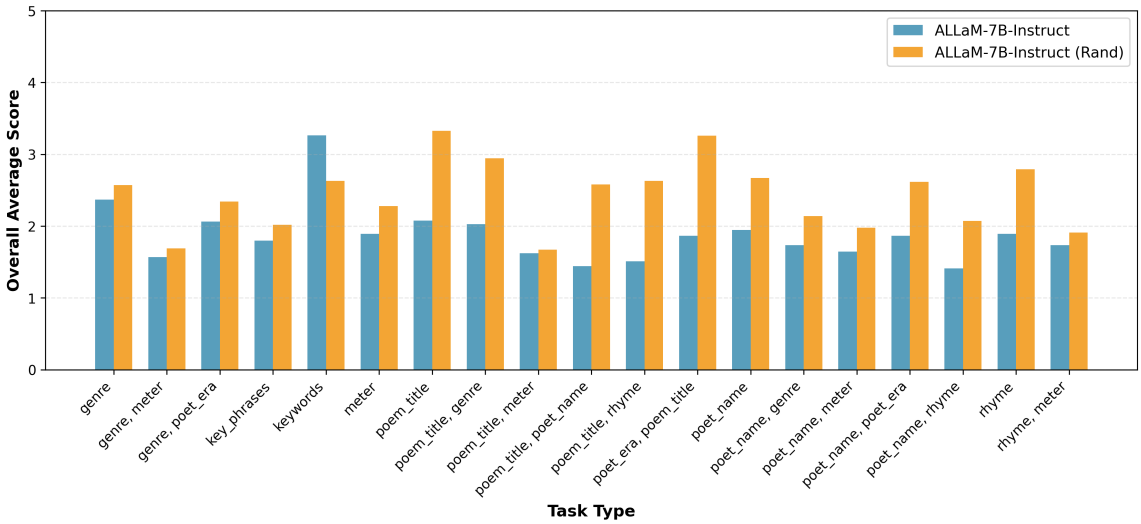

Figure 4: Generation Sub-tasks Results on ALLaM-7B–instruct .

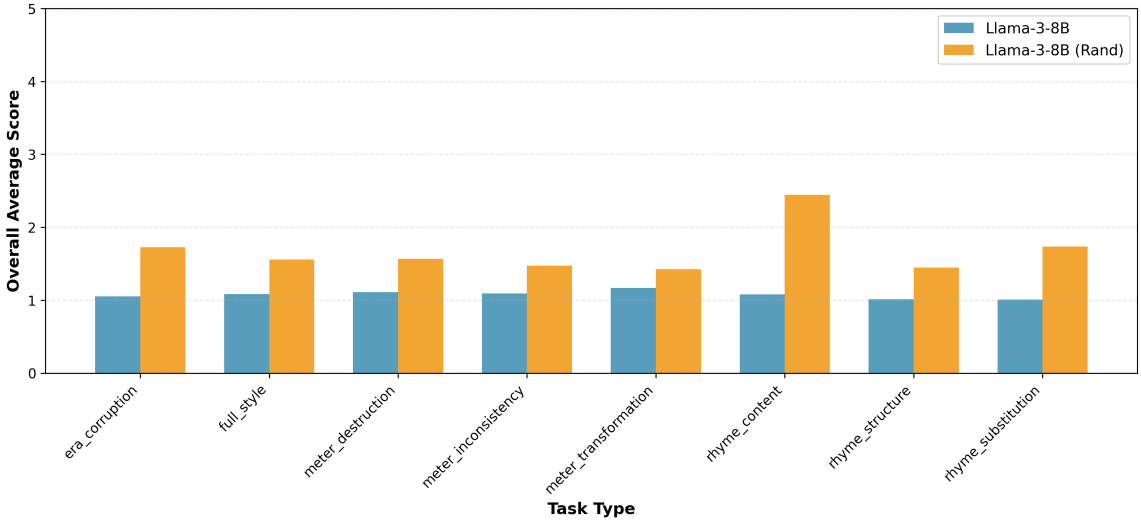

Figure 5: Revision Sub-tasks Results on LLaMA-3-8B .

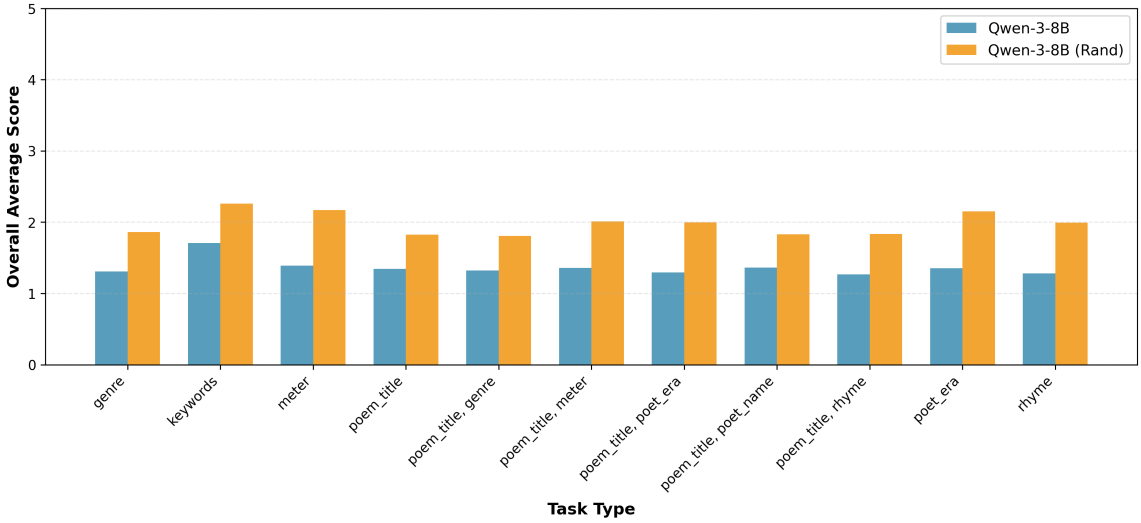

Figure 6: Continuation Sub-tasks Results on Qwen3-8B .

Task	Input	Output	MSA	Nile Valley	North Africa	Gulf	Levant
Generation	Poet Name Era	Poem Text	اكتب قصيدة تحاكي أسلوب ((eman_teop)) من زمن ((are_teop)).	اكتب قصيدة تقلد أسلوب ((eman_teop)) من زمن ((are_teop)).	كتب قصيدة بالأسلوب ديال ((eman_teop)) من زمان ((are_teop)).	اكتب قصيدة على أسلوب ((eman_teop)) من زمن ((are_teop)).	اكتب قصيدة بنفس أسلوب ((eman_teop)) من زمن ((are_teop)).
Continuation	Existing Verses Meter	Poem Text	تابع هذه الأبيات بنفس الوزن الشعري ((retem)). ((sesrev_gnitsixe))	كمل الأبيات دي بنفس الوزن الشعري ((retem)): ((sesrev_gnitsixe))	كمّل هاد لْبيوت بنفس الوزن الشعري ((retem)). ((sesrev_gnitsixe))	كمل هالأبيات هذي بنفس الوزن الشعري ((retem)). ((sesrev_gnitsixe))	كمّل هالأبيات بنفس الوزن الشعري ((retem)). ((sesrev_gnitsixe))
Analysis	Poet Name Poem Text Genre	Meter	هذه القصيدة كتبها ((eman_teop)) وتنتمي إلى نوع ((erneg)): ((txet_meop)) ما هو البحر الشعري المستخدم؟	القصيدة دي كتبها ((eman_teop)) ودي من نوع ((erneg)): ((txet_meop)) إيه هو البحر الشعري المستخدم؟	هاد القصيدة كتبها ((eman_teop)) وكتنتمي لنوع ((erneg)): ((txet_meop)) شنو هو البحر الشعري اللي مستعمل؟	هذي القصيدة كتبها ((eman_teop)) ونوعها ((erneg)): ((txet_meop)) وشو البحر الشعري المستخدم؟	هي القصيدة كتبها ((eman_teop)) وبتنتمي لنوع ((erneg)): ((txet_meop)) شو هو البحر الشعري المستخدم؟
Restoration	Poem Text Era Poet Name Meter	Poem Text	لقد تم تدمير البحر الشعري لهذه القصيدة. مهمتك هي إعادة النص الشعري إلى وزنه الأصلي المنتظم ((retem)). أعد كتابة الأبيات لتطابق البحر ((retem)) مع الحفاظ على المعنى والقافية. الشاعر: ((eman_teop)) العصر: ((are_teop)) البحر: ((retem)) القصيدة المحرفة: ((txet_meop)) القصيدة المستعادة:	البحر الشعري للقصيدة دي اتدمر. مهمتك إنك ترجع النص الشعري لوزنه الأصلي ((retem)). اكتب الأبيات بما يطابق البحر ((retem)) مع الحفاظ على المعنى والقافية. الشاعر: ((eman_teop)) العصر: ((are_teop)) البحر: ((retem)) القصيدة المحرفة: ((txet_meop)) القصيدة المستعادة:	تخرب البحر ديال هاد القصيدة. خاصك ترجع النص الشعري للوزن الأصلي ((retem)). كتب الأبيات باش يطابقو البحر ((retem)) وحافظ على المعنى والقافية. الشاعر: ((eman_teop)) العصر: ((are_teop)) البحر: ((retem)) القصيدة المبدلة: ((txet_meop)) القصيدة المصلحة:	البحر الشعري لهالقصيدة تخرب. مهمتك إنك ترجع النص الشعري لوزنه الأصلي ((retem)). عيد كتابة الأبيات لتطابق البحر ((retem)) وحافظ على المعنى والقافية. الشاعر: ((eman_teop)) العصر: ((are_teop)) البحر: ((retem)) القصيدة المحرفة: ((txet_meop)) القصيدة المستعادة:	البحر الشعري لهي القصيدة تخرب. مهمتك ترجع النص لوزنه الأصلي ((retem)). كتوب الأبيات لتتطابق مع البحر ((retem)) مع الحفاظ على المعنى والقافية. الشاعر: ((eman_teop)) العصر: ((are_teop)) البحر: ((retem)) القصيدة المحرفة: ((txet_meop)) القصيدة المستعادة:

Table 12: Instruction template examples for IFT tasks across MSA and regional Arabic dialects.

Model	Joint	Poem-text						Poem+Poet				Poet-name		
		G	K	M	Title	Era	Poet	G	M	Era	Rhy	G	M	Era
ALLaM-7B–instruct	79.5	35.6	72.0	85.8	99.9	48.0	72.3	36.4	77.8	63.6	25.2	26.7	49.7	88.0
ALLaM-7B–instruct (Curriculum)	99.5	29.8	94.7	99.7	99.9	66.1	87.9	29.3	99.7	79.6	98.2	30.0	54.7	88.9
ALLaM-7B–instruct (Random)	99.5	32.9	95.3	99.6	99.9	61.9	87.8	33.6	99.7	76.9	99.6	31.0	58.2	89.3
Fanar-1-9B	40.5	28.9	65.5	38.4	99.9	38.4	60.3	24.6	41.5	46.7	21.2	26.0	49.2	53.8
Fanar-1-9B (Curriculum)	59.5	33.8	92.6	48.0	99.8	48.3	60.0	33.3	66.9	59.1	22.6	28.1	57.7	78.4
Fanar-1-9B (Random)	59.5	32.4	93.9	52.2	99.9	37.2	60.3	30.5	60.9	60.0	29.6	32.7	55.5	78.6
LLaMA-3-8B	38.5	34.7	61.7	35.4	99.9	28.5	52.3	29.0	35.9	40.4	28.8	22.4	35.2	40.6
LLaMA-3-8B (Curriculum)	99.0	39.3	92.4	98.3	100.0	64.3	84.6	40.2	98.2	88.0	99.6	37.0	53.9	89.9
LLaMA-3-8B (Random)	98.0	38.5	93.7	98.4	100.0	63.3	86.3	40.5	98.2	85.8	96.5	36.5	56.3	91.6
Qwen3-8B	45.5	22.7	49.3	43.6	99.8	17.1	54.1	28.4	46.5	29.8	26.6	23.6	31.0	33.8
Qwen3-8B (Curriculum)	98.0	31.6	93.6	98.9	99.9	62.2	84.9	30.8	100.0	83.6	100.0	35.8	54.1	87.4
Qwen3-8B (Random)	99.5	29.6	92.6	99.1	99.9	62.4	85.4	30.5	100.0	85.3	100.0	31.7	58.5	85.7

Table 13: Analysis task evaluation results in % (higher is better). **Joint** corresponds to `poet_name, poem_text, genre` $\rightarrow$ `meter`. Abbreviations: G=genre, K=keywords, M=meter, Rhy=rhyme. Models with (Curriculum)/(Random) are LoRA fine-tuned. Rows are grouped by backbone; horizontal rules separate groups.

consistently outperforms the base model across most attribute-controlled generation settings, including genre, meter, rhyme, poet identity, and their combinations.

Metadata generated	Model used in code	Prompt template
Keywords and key phrases	Gemini 2.5 Pro	You will be given an Arabic poem. Your task is to analyze its content and return: (i) 3 keywords that best represent the core themes or concepts of the poem, and (ii) 3 key phrases that are meaningful or characteristic expressions from the poem. All output must be in Arabic. Return the result strictly in JSON format with the following structure: `{"keywords": [], "key_phrases": []}`. Poem: `{poem_text}`

Table 14: Meta-data generation prompt used extracting poem-level metadata.

Hyperparameter	Value used
Prompt format	`chat`
Maximum new tokens	1024
Temperature	0.7
Top-p	0.9
Top-k	50
Repetition penalty	1.15

Table 15: Generation hyperparameters used for inference.

Performance gains are particularly pronounced when generation involves specific and identity- or structure-anchored constraints, such as `poem_title`, `poet_name`, or `rhyme`, either individually or in combination. In these settings, the fine-tuned variant shows substantially larger improvements over the base model, indicating enhanced control over semantically grounded and prosodic constraints. In contrast, gains are less visible for more general stylistic constraints such as `meter` or `genre`. This suggests that random fine-tuning is particularly effective at strengthening the model's ability to adhere to concrete, high-precision constraints, rather than broad stylistic cues alone.

Revision under Corruption Figure 5 focuses on **revision tasks** evaluated under different corruption types on LLAMA-3-8B results. These include era corruption, meter destruction, meter inconsistency, meter transformation, and several rhyme-based corruptions. The randomly fine-tuned model consistently outperforms the base model across all corruption categories.

The largest improvements are observed for *rhyme-related corruptions*, such as `rhyme_content`, and `rhyme_substitution`. This indicates that poetic revision places significant demands on prosodic awareness and controlled rewriting, capabilities that are not reliably recovered by base models without targeted fine-tuning.

Continuation Tasks Figure 6 presents results for **continuation tasks** on QWEN-3-8B's results, where models are prompted with existing verses and required to continue the poem while preserving specific attributes such as meter, rhyme, genre, poet era, or poet identity. The randomly fine-tuned variant consistently outperforms the base model across all continuation settings.

Performance gains are particularly notable when continuation requires maintaining *long-range structural consistency*, especially in meter-, era- and rhyme-constrained scenarios (e.g., `existing_verses + rhyme` and `existing_verses + poet_era`). These results suggest that fine-tuning enhances the model's ability to sustain global poetic structure beyond local fluency.

Dialectal Analysis Figure 7 presents a dialectal comparison of average performance across tasks for multiple base models and their fine-tuned (Random) variants. The evaluation spans Modern Standard Arabic (MSA), major regional dialects (Gulf, North African, Levantine, and Nile Valley). Across all model families, fine-tuning consistently improves performance for every dialect, although the magnitude of these gains varies.

Across model families, ALLAM-7B-INSTRUCT and QWEN-3-8B outperform FANAR and LLAMA in their base versions; after fine-tuning, ALLAM remains dominant while LLAMA surpasses QWEN, with all models benefiting from fine-tuning across both MSA and diverse Arabic dialects

Model	Resoration			Generation				Continuation		
	BERT-Score	Rouge-L	Rhyme	BERT-Score	Rouge-L	Rhyme	Key-Phrase	BERT-Score	Rouge-L	Rhyme
ALLaM-7B–instruct	83.3	21.9	21.5	80.6	5.9	14.4	**37.3**	84.8	4.4	52.0
ALLaM-7B–instruct (Curriculum)	**86.9**	30.1	55.2	80.8	**6.9**	54.5	31.7	**86.9**	**4.5**	78.8
ALLaM-7B–instruct (random)	**86.9**	**30.3**	**55.8**	**80.9**	**6.9**	**55.4**	33.1	**86.9**	**4.5**	**80.1**
Fanar-1-9B	76.2	7.2	14.6	74.8	3.1	7.1	21.6	77.4	4.4	24.6
Fanar-1-9B (Curriculum)	81.0	20.5	44.1	78.7	**6.2**	44.3	22.1	**82.3**	**10.6**	**75.0**
Fanar-1-9B (Random)	**81.3**	**22.8**	**44.7**	**78.9**	6.1	**45.3**	**22.4**	**82.3**	10.5	70.4
LLaMA-3-8B	83.5	**40.5**	52.5	74.4	5.6	22.4	**61.0**	80.8	**4.6**	50.5
LLaMA-3-8B (Curriculum)	85.9	32.5	64.0	**81.6**	6.2	52.5	38.53	**86.5**	4.5	85.2
LLaMA-3-8B (Random)	**86.1**	33.6	**66.7**	81.5	**6.3**	**53.5**	37.3	**86.5**	4.4	**86.3**
Qwen3-8B	**86.9**	**47.0**	31.1	**80.8**	**6.4**	11.3	**63.8**	84.3	**4.4**	35.4
Qwen3-8B (Curriculum)	83.8	27.8	**64.2**	79.7	5.9	47.1	**63.8**	**85.2**	4.3	82.2
Qwen3-8B (Random)	83.9	28.6	**64.2**	79.6	6.0	**50.9**	**63.8**	84.9	4.3	**83.5**

Table 16: Automatic evaluation results in % (higher is better). **Bold** indicates the best score within each model family (ties are bolded).

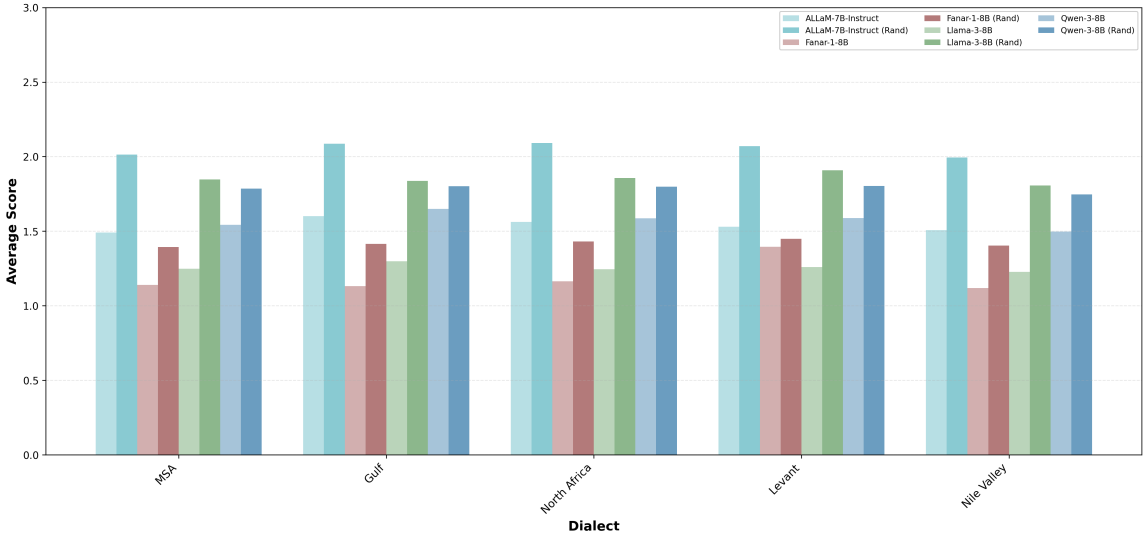

Figure 7: Average performance across base Vs. finetuned models per dialect. Lighter variant of the color = base model.