AGS-KSU at QIAS 2026: A Comparative Study of Prompting and LLM Approaches for Structured Islamic Inheritance Reasoning

Hicham Ghazi Sidaoui
King Saud University (AGS-KSU)
hicham.gs@hotmail.com

Abstract

This paper describes our submission to the QIAS 2026 shared task on structured Islamic inheritance reasoning, based on the MAWARITH benchmark (Bouchekif et al., 2026). The task requires multi-step structured prediction for Arabic inheritance cases, including heir identification, blocking, share assignment, adjustment detection, and final distribution, evaluated with the MIR-E metric. We compare four system configurations: a QLoRA fine-tuned Qwen2.5-3B baseline, a multi-stage Fanar-Sadiq pipeline with deterministic validation and post-processing, and two GPT-5.4 prompting setups. On the official test set, the best result was achieved by GPT-5.4 with explicit inheritance rules and development examples used as in-context demonstrations, reaching a MIR-E score of 0.84, compared with 0.76 for a minimal-prompt GPT-5.4 variant. These results suggest that explicit rule conditioning and in-context demonstrations can improve performance in this setup. Since the compared systems vary in model family and prompting strategy, the findings should be interpreted as a comparison of task configurations rather than a controlled model-only comparison.

Keywords: Islamic inheritance law, Arabic NLP, structured output generation, prompting, in-context learning, legal reasoning, QLoRA

1. Introduction

Islamic inheritance law (*Ilm al-Faraid*) requires precise multi-step reasoning: identifying eligible heirs, determining which relatives are blocked, assigning Quranic fractional shares, detecting proportional adjustment, and computing the final distribution. Because each stage depends on the correctness of preceding ones, the task is a challenging structured reasoning benchmark for NLP systems.

The QIAS 2026 shared task is based on the MAWARITH benchmark (Bouchekif et al., 2026), requiring end-to-end structured output generation across the main inheritance reasoning stages, evaluated with the MIR-E metric. Unlike the earlier QIAS 2025 task (Bouchekif et al., 2025a), which was framed as multiple-choice QA, QIAS 2026 demands full structured prediction. Prior work showed a substantial performance gap between large proprietary models and Arabic-specific models (Bouchekif et al., 2025b), and that retrieval-augmented fine-tuning could yield strong results (AL-Smadi, 2025). These findings motivate our investigation of fine-tuning and prompting-based approaches.

This paper evaluates four engineered system configurations that trade off supervision type, cost, controllability, and inference-time structure enforcement: (1) a QLoRA fine-tuned Qwen2.5-3B baseline, (2) a multi-stage Fanar-Sadiq inference pipeline, (3) a GPT-5.4 configuration with a lightweight inference template, and (4) a GPT-5.4 configuration with explicit legal rules and in-context demonstrations.

2. Related Work

The MAWARITH benchmark (Bouchekif et al., 2026) introduces the MIR-E metric, which scores each reasoning stage independently with weighted aggregation. The QIAS 2025 shared task (Bouchekif et al., 2025a) attracted dozens of teams; Bouchekif et al. (2025b) show that performance gaps reflect reasoning ability rather than Arabic language proficiency alone. System papers from QIAS 2025 include Bekhouche et al. (2025), who used encoder-based relevance scoring; AL-Smadi (2025), who combined LoRA fine-tuning with RAG; and AlDahoul and Zaki (2025), who showed ensemble voting outperforms individual models. QLoRA (Dettmers et al., 2023) and in-context learning (Brown et al., 2020) are the core technical methods we employ.

3. Task and Evaluation

The MAWARITH dataset (Bouchekif et al., 2026) contains 12,500 Arabic inheritance cases. The shared-task split provides training data, 200 development examples with gold annotations, and 500 test examples. Outputs are structured JSON covering eligible heirs, blocked heirs, legal shares, adjustment status (*awl*/*radd*), and final distribution. Systems are scored with MIR-E, using set-based F1 for heir stages and value-based matching for share and distribution stages. All systems used only resources released by the shared-task organizers; no external labeled inheritance data or manual correction of test predictions was used.

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 209–212

4. System Configurations

4.1. Qwen2.5-3B with QLoRA

We fine-tuned Qwen2.5-3B-Instruct (Qwen Team, 2024) using QLoRA (Dettmers et al., 2023) on a 2,000-example subset of the training split sampled with fixed random seed 42, due to time and hardware constraints on an NVIDIA RTX 4060 Laptop GPU (8GB VRAM). Training used 4-bit NF4 quantization with bfloat16 compute. The LoRA adapter used rank $r = 16$, $\alpha = 32$, dropout 0.05, applied to all seven attention and MLP projection layers. Training ran for 3 epochs, sequence length 1,024, batch size 1 with gradient accumulation 4 (effective 4), learning rate 2×10^{-4} with cosine schedule and 20 warmup steps, paged AdamW 8-bit optimizer, and gradient checkpointing. Training completed in approximately 5.5 hours, reaching a final training loss of 0.26.

At inference, the model tended to generate extended Arabic reasoning before producing JSON, exhausting the generation budget. Prefilling the assistant turn with an opening brace and disabling the thinking template reduced generation time from $\sim$170s to $\sim$25s per example. Despite this, structured-generation failures resulted in a MIR-E score of 0.30. We include this configuration as a resource-constrained fine-tuning baseline rather than a performance-optimized local model.

4.2. Fanar-Sadiq Pipeline

Fanar-Sadiq (Abbas et al., 2025) is an Arabic-focused LLM accessed via API. We implemented a three-stage inference pipeline: (1) heir extraction, (2) fiqh reasoning for share assignment and adjustment, and (3) constrained structured JSON extraction. All three stages used the same Fanar backend with temperature 0.0.

The pipeline processed all 500 test examples sequentially, with each example requiring three API calls. The heir-extraction prompt asked the model to identify all relatives mentioned in the case text and normalize their names to canonical forms. The reasoning prompt then took the extracted heir list and requested full fiqh reasoning covering share assignment, blocking decisions, and adjustment detection. Finally, the structured-extraction prompt took the reasoning output and asked for a JSON response conforming to the six-field task schema, with no legal reasoning included in the output.

After structured extraction, we applied a deterministic validation-and-repair layer, including heir-name normalization, completeness enforcement for the heirs field, recomputation of `post_tasil` percentages, and deterministic computation of `awl_stage` when needed. If the extracted JSON failed validation, the structured-extraction stage was retried once using validation errors as corrective feedback. At the batch-execution level, API requests used exponential-backoff retry handling for rate-limit failures. Fanar-Sadiq achieved MIR-E 0.46.

4.3. GPT-5.4

We evaluated two inference configurations of GPT-5.4, a commercial large language model accessed via API, under otherwise identical decoding settings: temperature 0, top_p 1.0, frequency and presence penalty 0.0, with batches of five test cases per request.

The **minimal-prompt** variant used a brief instruction to solve each inheritance case and return the answer in the required structured JSON format, including heirs, blocked heirs, shares, `awl_or_radd`, optional `awl_stage`, and `post_tasil`. It did not include explicit legal rules or in-context examples. This configuration achieved MIR-E 0.76.

The **full-prompt** variant used the same target JSON schema, but added two components: (1) an explicit rule block covering eight hard-case families (grandfather–sibling share selection, sister-with-daughters agnatic transformation, *radd* exclusion of spouses, multiple grandmother blocking, *al-Akdariyyah*, *al-Mushtarakah*, son's-daughter interactions, and the *Umariyyatan* rule), and (2) 20 gold development examples per batch as fully solved JSON demonstrations, selected by fixed circular offset: batch i used examples $[(20i) \bmod 200, \ldots, (20i + 19) \bmod 200]$. The exact prompt is embedded in the code in the repository. The development examples were used only as inference-time in-context demonstrations and were not used for parameter updates. We use "in-context learning" to refer to inference-time prompting with fully solved development examples, without any parameter updates. Following standard shared-task practice, the released development set was used only for inference-time prompting and not for parameter updates or test-set selection. This configuration achieved MIR-E 0.84.

Post-processing was deterministic and restricted to schema-level normalization, including count-annotation removal from heir names, required-field completion, and fraction-format verification. No legal predictions were modified.

5. Results and Analysis

Table 1 summarizes the official test-set results. The GPT-5.4 full-prompt configuration achieved the highest MIR-E of 0.84. Because the two GPT-5.4 configurations differed only in inference tem-

System	Method	MIR-E
Qwen2.5-3B (fine-tuned)	QLoRA	0.30
Fanar-Sadiq pipeline	Zero-shot	0.46
GPT-5.4 + minimal prompt	Prompting	0.76
GPT-5.4 + rules + dev examples	In-context	0.84

Table 1: Official test-set MIR-E for the four submitted system configurations.

plate content, the improvement from 0.76 to 0.84 provides a more controlled within-model comparison of prompt design, since model family and decoding settings were held fixed across the two GPT-5.4 runs. In this narrower sense, the observed improvement is consistent with a positive contribution from explicit rule conditioning and in-context demonstrations in our setup. These results should be interpreted as a comparison of submitted system configurations rather than a controlled cross-model study, because prompting regimes and supervision levels were not matched across model families.

6. Error Analysis

To better understand the remaining gap between high MIR-E performance and exact structured correctness, we compared the 0.84 configuration's predictions against the released gold file across four components (Table 2). The system is strongest on `awl_or_radd` (93.6%) and weaker on exact heir and distribution matching. Table 3 summarizes the main error categories.

Component	Matches	Rate
Heirs list	350/500	70.0%
Blocked list	345/500	69.0%
`awl_or_radd`	468/500	93.6%
Normalized final dist.	355/500	71.0%
All four jointly	284/500	56.8%

Table 2: Diagnostic exact-match statistics (more stringent than MIR-E; diagnostic only).

Error category	Cases	Rate
Grandmother blocking errors	63	12.6%
False *radd* (residual agnate removed)	37	7.4%
`awl_or_radd` mismatches	32	6.4%
Collateral continuation errors	20	4.0%

Table 3: Top error categories out of 500 test cases.

Error categories were assigned by automated comparison of predicted outputs against gold annotations, with each case labelled according to its primary observed failure type. The dominant errors include **grandmother-line confusion** (63 cases), where grandmother eligibility or blocking differs between the gold and predicted outputs; **false *radd*** cases, where the prediction redistributes the remainder despite the gold output preserving a residual agnate (e.g., case `nh3a2q4h_4`); **`awl_or_radd` mismatches** (32 cases), mostly involving *radd* with non-spousal heirs; and **collateral continuation errors**, where the gold output preserves a collateral agnatic heir but the prediction blocks that heir (e.g., case `nh2s1u5s_7`).

Case	Gold	Prediction	Error
`nh3a2q4h_4`	Fixed shares + paternal uncles (residual)	Uncles blocked; *radd* applied	False *radd*
`nh2s1u5s_7`	Mother, paternal half-sister, and residual collateral heir	Residual collateral heir blocked	Collateral error

Table 4: Representative prediction errors made by the 0.84 system on the test set.

These comparisons suggest that the system's remaining weaknesses are concentrated in legally difficult remainder-allocation decisions. In many cases, the prediction captures part of the heir structure correctly, but fails in the final legal resolution step: deciding whether a residual agnate remains, whether *radd* is permitted, and whether the remainder should continue to a collateral heir. Future improvements should focus on explicit rule enforcement for grandmother precedence, residual agnate detection, and remainder transfer.

7. Conclusion

This paper described our submitted system configurations for the QIAS 2026 shared task. Among them, GPT-5.4 with explicit inheritance rules and in-context demonstrations achieved the highest MIR-E of 0.84. More broadly, structured Islamic inheritance reasoning appears to be a demanding benchmark for Arabic legal AI, as it requires compositional rule application, exact arithmetic, and multi-stage output consistency simultaneously. The results suggest that rule-aware prompting, in-context demonstrations, and schema-aligned inference design were plausible contributing factors in the strongest evaluated configuration. Because the evaluated systems were not matched for supervision level or prompting regime, the results should be interpreted at the configuration level rather than as evidence about

model family in isolation. Future work should investigate matched cross-model comparisons, targeted handling of hard cases (grandmother blocking, residual agnates, *radd* interactions), and generalization to other *madhahib*.

8. Ethics Statement

This work involves Islamic inheritance law, a sensitive religious and legal domain. The system is intended for research purposes only and should not be used as a substitute for qualified legal or religious advice. All data were provided by the QIAS 2026 shared-task organizers.

9. Data and Code Availability

All materials are publicly available at https://github.com/xAGS1/qias2026-ags-ksu, including prompt templates, inference scripts, post-processing code, and final prediction files. The prediction file can be independently scored using the official MIR-E evaluator to verify the 0.84 result.

10. Acknowledgements

The author thanks the QIAS 2026 shared task organizers for providing the MAWARITH benchmark and evaluation infrastructure.

11. Bibliographical References

Hanan Abbas et al. 2025. Fanar: A large language model for arabic. Technical report, Qatar Computing Research Institute.

Mohammad AL-Smadi. 2025. QU-NLP at QIAS 2025 shared task: A two-phase LLM fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning. *arXiv preprint arXiv:2508.15854*.

Nouar AlDahoul and Yasir Zaki. 2025. Benchmarking the legal reasoning of LLMs in arabic islamic inheritance cases. *arXiv preprint arXiv:2508.15796*.

Salah Eddine Bekhouche et al. 2025. CVPD at QIAS 2025 shared task: An efficient encoder-based approach for islamic inheritance reasoning. *arXiv preprint arXiv:2509.00457*.

Abdessalam Bouchekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. MAWARITH: A dataset and benchmark for legal inheritance reasoning with LLMs. *arXiv preprint arXiv:2603.07539*.

Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025a. QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860.

Abdessalam Bouchekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al Khatib, and Mohammed Ghaly. 2025b. Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257.

Tom B. Brown et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.