Gumball at QIAS 2025 Shared Task: Arabic LLM Automated Reasoning in Islamic Inheritance

Eman Elrefai*[1], Aml Hassan Esmail*[2], and Mohamed Lotfy Elrefai*[3]

[1]Alexandria University, eman.lotfy.elrefai@gmail.com
[2]Benha University, aml.hassan.esmil@gmail.com
[3]Ain Shams University, mohamed.lotfy.elrefai@gmail.com

Abstract

In this paper, we present a system for solving Islamic inheritance problems using large language models (LLMs), focusing on accurate reasoning in Arabic based on fara'id rules. Our approach is built on the Qwen3-4B model, quantized, and trained using the Unsloth framework for efficiency. We explore multiple training strategies: (1) retrieval-augmented generation (RAG) using fatwas from Islamweb, (2) supervised fine-tuning (SFT) on annotated inheritance datasets, (3) instruction tuning of a base Qwen model followed by GRPO training for multiple choice question solving, and (4) a two-stage pipeline involving SFT on a classical Islamic inheritance book followed by MCQ fine-tuning. Among these, the fourth approach achieved 97.2% accuracy, outperforming all other submissions and ranking our team first in the competition.

1 Introduction

Islamic inheritance laws are complex and highly nuanced, and vary significantly depending on factors such as Islamic sect, national legislation, and cultural practices. Due to this complexity, accurately determining inheritance shares often requires the expertise of scholars well-versed in both jurisprudence and contextual legal systems. This intricate structure makes the domain of Islamic inheritance particularly well-suited for developing reasoning tasks in the Arabic language, offering a rich and challenging environment for natural language understanding and logical inference.

In this work, we present a system that leverages large language models (LLMs) to solve Islamic inheritance problems with high accuracy. We base our approach on the Qwen3-4B (Yang et al., 2025a) model, using the Unsloth framework (Daniel Han and team, 2023) for efficient quantisation and training. To tackle the complexity of the domain, we explore several strategies: retrieval-augmented generation using real-world fatwas, supervised fine-tuning on curated inheritance scenarios, instruction tuning followed by reinforcement training (GRPO) (Shao et al., 2024), and a two-stage pipeline that first fine-tunes on classical texts before solving multiple-choice questions. Our best-performing system as of 2025-08-20, which follows the two-stage pipeline approach, achieved 97.2% accuracy and ranked first on the leaderboard of the Arabic-NLP conference (Bouchekif et al., 2025a,b).

2 Related Work

Prior work in automated Islamic inheritance question answering has been limited, with most systems focusing on rule-based reasoning (Powers, 2017). While these approaches achieve perfect accuracy on explicitly encoded cases, they lack generalisation to unseen problems. Recent advances in Arabic NLP have enabled transformer-based models (Antoun et al., 2020) to tackle domain-specific MCQ tasks, yet most studies address general knowledge or educational exams rather than deep legal reasoning. Our contribution is novel in two aspects: (1) applying a small language model fine-tuned on a large-scale, domain-specific dataset of Islamic inheritance MCQs, and (2) integrating reasoning traces in the training phase (via the reasoning-augmented subset) to improve interpretability and accuracy in complex cases.

3 Dataset

The dataset used in this work consists of Arabic multiple-choice questions (MCQs) in the domain of Islamic heritage. The task involves predicting the correct answer option (A–F) for each question, given six possible choices. This problem requires a combination of reading comprehension, domain-specific legal knowledge, and numerical reasoning.

*These authors contributed equally to this work.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 953–959

The system takes as input a question q in Modern Standard Arabic, typically formulated in formal jurisprudential language, and a set of six possible answer options $\{o_1, o_2, \ldots, o_6\}$. The output is the index of the correct option. For example:

Question:
مات وترك: زوجة (٣) و ابن عم الأب (٥) و بنت و ابن أخ لأب (٥) و بنت ابن (٥)، كم إجمالي عدد الأسهم الذي تقسم عليه التركة قبل تصحيح المسألة؟
Options: A. 27 B. 22 C. 25 D. 26 E. 24 F. 23
Answer: E. 24

This setup differs from generic MCQ tasks because the reasoning process often involves applying formal rules from Islamic law, understanding exception cases, and performing share calculations.

3.1 Dataset Splits

The annotated dataset is divided into three splits:

- **Training:** 20,000 questions (10,000 Beginner, 10,000 Advanced)
- **Development:** 1,000 questions (500 Beginner, 500 Advanced)
- **Test:** 1,000 questions (500 Beginner, 500 Advanced, with gold labels for evaluation)

Each question contains six answer options (A–F), exactly one of which is correct. The label distribution in the training set is moderately imbalanced, with option C being the most frequent (21.7%) and option F the least frequent (13.4%) as shown in figure 1 . Table 3 summarises the main statistics.

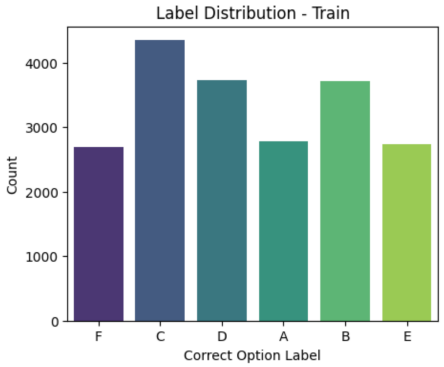

Figure 1: Label distribution in the training set.

3.1.1 IslamWeb Dataset

The IslamWeb corpus contained a total of 3,166 questions distributed across four batches. The dataset was structured as JSON arrays containing detailed fatwa objects with the following fields:

- **ID**: Unique identifier for each fatwa; **URL**: source link on IslamWeb.
- **Category**: Jurisprudential classification.
- **Dates**: Gregorian and Hijri publication dates.
- **Question**: User query; **Answer**: scholar's response with Quran and Hadith references.

3.2 Label and Difficulty Distributions

The label frequencies and difficulty level proportions are illustrated in Figure 1. These reveal a slight imbalance in label frequencies, which may influence model bias toward more frequent options.

This work was conducted in the context of the QIAS 2025– SubTask 1: Islamic Inheritance Reasoning, where participants developed models to predict the correct answer choice. Our submission was evaluated which considers both Beginner and Advanced difficulty levels.

4 System Overview

4.1 Two-Stage Fine-Tuning of SLM (Continual Pretraining + SFT)

Our end-to-end Figure 2 is organised as three distinct stages that are executed in a pipeline:

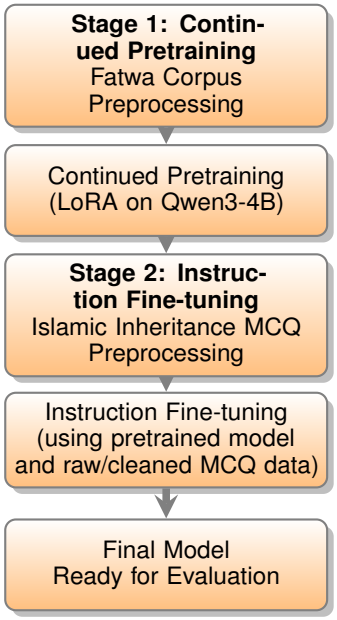

Figure 2: Two-Stage Fine-Tuning Pipeline for Islamic Legal Text Modelling.

1. **Domain Continual Pretraining:** We perform LoRA-based (Hu et al., 2022) continual pretraining of a Qwen3 family base model on a curated IslamWeb fatwa/article corpus to adapt the model to jurisprudential registers, domain phrases, and common reasoning patterns. The pretraining objective is standard autoregressive next-token likelihood.

2. **Supervised Fine-Tuning (SFT).** We fine-tune the adapted model on the MCQ inheritance dataset using instruction-style prompts (question + choices → answer token or short explanation). SFT enforces the mapping from the problem statement to the correct option and optionally to an intermediate reasoning trace.

3. **Cleanup & Post-processing.** A lightweight script normalises Arabic diacritics (tashkeel), punctuation, and simple orthographic variants – both as a preprocessing step for training and as a post-processing step on model outputs prior to scoring. This reduces spurious surface mismatches between model outputs and gold labels.

4.2 Reinforcement Learning Fine-Tuning

We fine-tuned the base `Qwen3-4B` model on an Islamic inheritance reasoning dataset using supervised fine-tuning (SFT) with instruction-style prompts, where each input was a question and the output contained step-by-step reasoning.

Following the DeepSeek reasoning framework (Shao et al., 2024), we applied reinforcement learning fine-tuning (RLFT) with the GRPO algorithm, training the model to produce both detailed reasoning traces and final multiple-choice answers.

To guide RLFT, we implemented the following custom reward functions:

- **Template Matching (Exact/Approximate)** – enforce reasoning and solution structure using predefined tokens.
- **Answer Format Validation** – ensure answers match valid multiple-choice options.
- **Numerical Accuracy Check** – reward exact matches to ground truth values.
- **Fuzzy Matching** – grant partial credit for near-correct outputs in format or structure.

These rewards balanced structural consistency, factual accuracy, and reasoning quality during training.

4.3 Retrieval-Augmented Generation

We implemented a Retrieval-Augmented Generation (RAG) system as an initial baseline, combining the competition-provided domain-specific corpus with additional external resources to expand coverage and improve retrieval quality.

For the retrieval component, we employed dense vector embeddings and evaluated several multilingual models: `e5-base` (Wang et al., 2024), `MiniLM-L12-v2` (Reimers and Gurevych, 2019), and `Matryoshka` (Nacar and Koubaa, 2024). Among these, the `Matryoshka` model consistently achieved the highest retrieval accuracy in our experiments.

The generation component was powered by `Qwen2.5-7B` (Yang et al., 2025b) using the Ollama v0.11.10 (Ollama Team, 2023) inference framework.

5 Experimental Setup

5.1 Two-Stage Fine-Tuning of SLM Pipeline

Training configuration and hyperparameters: The key training hyperparameters used across experiments are listed in Table 5 is provided in the Appendix.

5.1.1 Two-Stage Fine-Tuning Prompt

We used the following prompt format for training the first stage:

```
### الموضوع: {}
### السؤال:{}
<think>
### الإجابة:{}
</think>
```

Second stage multiple-choice questions, we formatted the prompts as:

System Prompt:

```
أنت مساعد ذكي تتحدث باللغة العربية الفصحى.
كن مهنيًا، لبقا وودودا في ردودك.
أجب بوضوح وتفصيل، وتجنب الردود المختصرة.
```

User Prompt:

```
"ما هي الإجابة الصحيحة على السؤال التالي؟
.\n\n"
"السؤال: {example['question']}\n"
```

الاختيارات :

```
"A. {example['option1']}\n"
"B. {example['option2']}\n"
"C. {example['option3']}\n"
"D. {example['option4']}\n"
"E. {example['option5']}\n"
"F. {example['option6']}\n"
```

جاوب برمز الاجابة الصحيحة فقط

5.1.2 Hardware Configuration

Experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3090 Ti GPU with 24GB VRAM. This hardware provided sufficient memory for efficient training of the 4B parameter models using the Unsloth framework with LoRA fine-tuning.

5.2 Reasoning Pipeline

We fine-tuned the `Qwen3-4B-Base` model using the `Unsloth` framework for efficient training and inference. The model was configured with a 2048-token context window and LoRA rank 32 applied to projection and feed-forward layers. Gradient checkpointing and a 70% GPU memory cap were used to reduce resource usage.

5.2.1 Supervised Fine-Tuning

SFT was performed for 2 epochs with batch size 1 using the AdamW (Loshchilov and Hutter, 2017) 8-bit optimizer, learning rate $2e^{-4}$, weight decay 0.01, and linear scheduling with 5 warm-up steps.

5.2.2 Reinforcement Learning Fine-Tuning

We then applied (GRPO) with `vllm` for fast sampling. Settings included 4 generations per prompt, temperature 1.0, `top_p` = 1.0, and learning rate $5e^{-6}$ for 100 steps. Multiple reward functions were used to enforce output format and correctness.

5.3 RAG Pipeline

5.3.1 Data Sources & Preprocessing

We combined the corpus provided by the competition with external inheritance resources. JSON files were converted into structured Q&A pairs, while unstructured documents were segmented using two strategies:

- Q&A extraction: regex-based identification of question–answer patterns.
- Semantic chunking: splitting long passages into 400-token segments with guiding questions.

All text was normalized through diacritic removal, character unification, stopword filtering, and whitespace cleanup, reducing noise and improving retrieval quality.

5.3.2 Retrieval

Documents were embedded using dense vector models from the SentenceTransformers library. We evaluated `e5-base`, `MiniLM-L12-v2`, and `Matryoshka`.

The last of these achieved the best retrieval accuracy in our domain. Retrieval employed cosine similarity, and for the best results, we used a top-3 selection strategy and a minimum similarity threshold of 0.7.

6 Results

6.1 Two-Stage Fine-Tuning of SLM Pipeline

6.1.1 Main results

Table 8 summarises the most relevant submissions (sorted by test accuracy). For each run, we report whether the cleanup script was applied (preprocessing and/or post-processing), the development accuracy (noting whether the dev split was cleaned), and the test accuracy used in the leaderboard submission.

6.1.2 Ablation: cleanup vs. no-cleanup

We compare matched runs where the only difference is whether the evaluation is performed on cleaned or raw data. The most illustrative matched pair is experiment F (raw training, evaluated on the cleaned test set) versus experiment H (raw training, evaluated on the raw test set):

- **Exp F (Raw → Clean Test):** Test 95.1%.
- **Exp H (Raw → Raw Test):** Test 94.3%.

This indicates that applying the deterministic cleanup procedure during evaluation yields a measurable improvement in final test accuracy (+0.8 percentage points in this pair).

6.2 Reasoning Pipeline

- **Baseline RLFT performance:** Applying RLFT directly to the `Qwen3-4B` model yielded **15%** accuracy.
- **Domain-adapted initialisation :** Initialising RLFT (500 steps) from a checkpoint fine-tuned on the Islamic inheritance MCQ dataset achieved **57%** accuracy.

Table 1: Accuracy of Different Inheritance Reasoning Pipelines on the test dataset

Pipeline	Accuracy (%)
RAG (Qwen2.5-7B + best embedding model)	35.33
Instruction SFT + GRPO	57.00
SFT on Annotated Dataset	87.00
Two-Stage Fine-Tuning of SLM (Continual Pretraining + SFT)	**97.20**

These results highlight the advantage of starting from a domain-adapted model for improving reasoning performance. Further optimisation of RLFT was not pursued due to time and resource constraints.

6.3 RAG Pipeline

We conducted a two-stage evaluation process on the development dataset:

1. **Pre-RAG Evaluation:** As shown in Table 2, `Qwen2.5-7B` (Yang et al., 2025b) achieved the highest standalone accuracy (31.5%), clearly outperforming both `Qwen3-4B` and `Qwen3-8B`. This established it as the strongest baseline model prior to retrieval integration.

2. **RAG Integration:** Building on this superior baseline, we integrated `Qwen2.5-7B` with our retrieval pipeline. Table 4 shows that combining the model with different embedding backbones led to further improvements, with the best accuracy (44.0%) obtained using the `Arabic-all-nli-triplet-Matryoshka` embeddings.

The RAG pipeline delivered an absolute gain of 12.5% (about 39.7% relative) over the standalone baseline, highlighting the value of targeted retrieval in knowledge-intensive tasks. While it did not surpass our fine-tuned models, it remains a strong, resource-efficient option for settings with limited computational budgets.

7 Analysis of Result

The model demonstrates strong performance overall, but a detailed analysis of its failures is crucial for future improvements. The overall error rate is low, though it is notably higher for questions categorised as "Advanced" (5.0%) compared to those labelled "Beginner" (0.6%). This suggests that the model struggles more with complex inheritance scenarios. Such performance gaps align with the concerns raised by (Fawzi et al., 2025; Sibaee et al., 2025), who highlight that errors in large language models in Arabic and religious contexts, particularly in complex reasoning tasks, can have serious consequences. The following tables 6 and 7 present representative failure cases, followed by a detailed analysis of the underlying reasons for the incorrect predictions.

7.1 Statistics

The model was evaluated on 1000 test questions, equally split between 'Beginner' and 'Advanced' levels. Overall accuracy was 97.2%, with a total error rate of 2.8%. Errors were more frequent in 'Advanced' questions (5.0%) compared to 'Beginner' ones (0.6%), indicating strong performance on basic rules but reduced accuracy in complex cases involving multiple heirs, distant kinship, and share correction (*tas'hih*).

8 Conclusion

We built an Arabic system for solving Islamic inheritance problems using large language models, achieving first place in QIAS 2025 with 97.2% accuracy. Our two-stage fine-tuning—domain continual pretraining plus supervised fine-tuning—was most effective, aided by targeted preprocessing. While basic cases were nearly flawless, complex scenarios require improvements in reasoning, symbolic integration, and interpretability for reliable real-world use.
For reproducibility, the implementation and code are available at `Gumball at QIAS 2025 | GitHub`.

9 Acknowledgments

We thank the QIAS 2025 shared task organisers for providing this valuable evaluation framework. We also acknowledge the anonymous reviewers for their constructive feedback.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9*. Association for Computational Linguistics.

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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Mahmoud Fawzi, Björn Ross, and Walid Magdy. 2025. Fabricating holiness: Characterizing religious misinformation circulators on arabic social media. *Preprint*, arXiv:2508.07845.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu et al. Chen. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Omer Nacar and Anis Koubaa. 2024. Enhancing semantic similarity understanding in arabic nlp with nested embedding learning. *Preprint*, arXiv:2407.21139.

Ollama Team. 2023. Ollama: An open-source framework for local llm inference. `https://github.com/ollama/ollama`. Accessed: Aug 1, 2025.

David S Powers. 2017. The islamic inheritance system: a socio-historical approach. In *Issues in Islamic Law*, pages 165–181. Routledge.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and Yang et al. Wu. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Serry Sibaee, Omer Nacar, Adel Ammar, Yasser Al-Habashi, Abdulrahman Al-Batati, and Wadii Boulila. 2025. From guidelines to practice: A new paradigm for arabic language model evaluation. *Preprint*, arXiv:2506.01920.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, and Chenxu et al. Lv. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, and Jingren et al. Zhou. 2025b. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

A Appendix

1.1 Tables

Table 2: Performance of different base Qwen Models on the development dataset

Model	Accuracy (%)
Qwen3-4B	10.8
Qwen3-8B	15.0
Qwen2.5-7B	**31.5**

Table 3: Dataset statistics by split.

Split	# Questions	Beginner	Advanced
Train	20,000	10,000	10,000
Dev	1,000	500	500
Test	1,000	500	500

Table 4: Performance of different embedding models on the development dataset

LLM Model	Embedding Model	Accuracy (%)
Qwen2.5-7B	paraphrase-multilingual-MiniLM-L12-v2	38.0
	multilingual-e5-base	42.6
	Arabic-all-nli-triplet-Matryoshka	**44.0**

Table 5: Key hyperparameters (representative values).

Hyperparameter	Pretraining	SFT
Base model	Qwen3-4B	PreTrain Model Qwen3 with LoRA
LoRA rank (r)	128	128
LoRA α	16	16
Context length	2048	2048
Batch size (per GPU)	12 (accumulation)	12 (accumulation)
Optimizer	AdamW	AdamW
Learning rate	5e−5	5e−5
Embedding Learning rate	1e−5	1e−5
Warmup steps	5	5
Weight decay	0.01	0.00
Epochs	3	4 (SFT)
Precision	bf16	bf16

Table 6: Analysis of Failure Case 1: Complex Kinship

Failure Case 1: Complex Kinship	
ID	4232_nq7p3f6g_18
Level	Advanced
Question	مات وترك: أم أب الأب و ابن ابن أخ لأب (٢) الأم و عم شقيق (٣) و، وام كم عدد الأسهم بعد التصحيح التي يحصل عليها لكل ابن ابن أخ لأب؟
Prediction	A: 3 shares
Ground Truth	F: 1 share
Analysis	**Key Error**: Miscalculated *tas'hih* (correction) for agnatic heirs **Reason**: Incorrect priority order determination **Fix**: Improve share correction logic

Table 7: Analysis of Failure Case 2: Exclusion Error

Failure Case 2: Exclusion Error	
ID	1981_nm1l6g8b_1
Level	Advanced
Question	مات وترك: أم الأم و أخ لأب و ابن عم شقيق (٣) و ابن (٥) و أب الأب كم النصيب الأصلي لكل صنف من الورثة من التركة؟
Prediction	B: أم الأم: السدس، أب الأب: محجوب، ...
Ground Truth	F: ... ،أم الأم: السدس، أب الأب: السدس
Analysis	**Key Error**: Excluded grandfather **Reason**: Core rule misunderstanding **Fix**: Correct exclusion principles

Table 8: Two-Stage Fine-Tuning of SLM pipeline: results with cleaned vs. raw training data. **Base model:** Qwen pretrained on IslamWeb.

Exp	Data	Pre Steps	Cleanup Steps	Eval Set	Dev (%)	Test (%)
A	Cleaned	5500	3000	Clean	81.9	97.2
B	Cleaned	5500	4500	Clean	82.4	97.0
C	Cleaned	5500	4834	Clean	82.5	96.8
D	Cleaned	5500	2500	Clean	82.0	96.8
E	Cleaned	5500	1500	Clean	80.7	96.4
F	Raw	2500	–	Clean	–	95.1
G	Raw	5500	–	Raw	80.7	95.8
H	Raw	2500	–	Raw	78.9	94.3