QU-NLP at QIAS 2025 Shared Task: A Two-Phase LLM Fine-Tuning and Retrieval-Augmented Generation Approach for Islamic Inheritance Reasoning

Mohammad AL-Smadi
Qatar University
Doha, Qatar
malsmadi@qu.edu.qa

Abstract

This paper presents our approach and results for SubTask 1: Islamic Inheritance Reasoning at QIAS 2025, a shared task focused on evaluating Large Language Models (LLMs) in understanding and reasoning within Islamic inheritance knowledge. We fine-tuned the Fanar-1-9B causal language model using Low-Rank Adaptation (LoRA) and integrated it into a Retrieval-Augmented Generation (RAG) pipeline. Our system addresses the complexities of Islamic inheritance law, including comprehending inheritance scenarios, identifying eligible heirs, applying fixed-share rules, and performing precise calculations. Our system achieved an accuracy of 0.858 in the final test, outperforming other competitive models such as, GPT 4.5, LLaMA, Fanar, Mistral and ALLaM evaluated with zero-shot prompting. Our results demonstrate that QU-NLP achieves near state-of-the-art accuracy (85.8%), excelling especially on advanced reasoning (97.6%) where it outperforms Gemini 2.5 and OpenAI's o3. This highlights that domain-specific fine-tuning combined with retrieval grounding enables mid-scale Arabic LLMs to surpass frontier models in Islamic inheritance reasoning.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have opened new avenues for their application across diverse domains, including specialized knowledge systems. This paper details our participation in the QIAS 2025 Shared Task, specifically focusing on Subtask 1: Islamic Inheritance Reasoning (*Ilm al-Mawārīth*) (Bouchekif et al., 2025a). This subtask challenges LLMs to navigate the intricate and highly structured field of Islamic inheritance law, which is governed by precise jurisprudential rules. The objective is to develop systems capable of comprehending complex inheritance scenarios, accurately identifying eligible and ineligible heirs, applying fixed-share rules (*farāiḍ*), managing residuary shares, and addressing advanced cases such as proportional reduction (*ʿawl*) and redistribution (*radd*), ultimately performing precise calculations to determine final shares (Mohammedi, 2012; Zouaoui and Rezeg, 2021).

The intersection of Natural Language Processing (NLP) and legal reasoning, particularly within specialized domains like Islamic law, has garnered increasing attention. Prior research has explored the application of computational methods to analyze legal texts, extract relevant information, and even automate aspects of legal decision-making. However, the unique complexities of Islamic inheritance law, with its intricate rules and diverse scenarios, present distinct challenges for traditional NLP approaches (Malhas et al., 2022, 2023).

Recent advancements in Large Language Models (LLMs) have shown promising capabilities in complex reasoning tasks, including those requiring domain-specific knowledge. Studies have demonstrated LLMs' ability to understand and generate human-like text, perform question answering, and even engage in logical inference. However, their performance in highly specialized and rule-based domains often necessitates fine-tuning or integration with external knowledge sources (Almazrouei et al., 2023; Sengupta et al., 2023; Alnefaie et al., 2023; Bari et al., 2024; Mohammed et al., 2025).

Specifically, in the context of Islamic inheritance reasoning, several works have emerged (Akkila and Naser, 2016; Tabassum et al., 2019; Zouaoui and Rezeg, 2021). For instance, (Bouchekif et al., 2025b) assesses LLMs on Islamic legal reasoning, providing evidence from inheritance law evaluation. This work highlights the potential and limitations of current LLMs in this domain, underscoring the need for more robust and accurate systems.

Furthermore, the concept of Retrieval-Augmented Generation (RAG) has gained prominence as a method to enhance LLM

Proceedings of The Third Arabic Natural Language Processing Conference, pages 892–898

performance by grounding their responses in retrieved factual information. This approach is particularly relevant for domains where accuracy and adherence to specific rules are important, as it allows LLMs to access and incorporate up-to-date or domain-specific knowledge that may not have been fully captured during their initial training. The integration of RAG with fine-tuned LLMs represents a significant step towards building more reliable and interpretable AI systems for complex reasoning tasks (Alan et al., 2024; Sayeed et al., 2025).

Our work builds upon these foundations by specifically addressing the challenges of Islamic inheritance reasoning within the framework of a shared task. By combining parameter-efficient fine-tuning with a Retrieval-Augmented Generation (RAG) pipeline, we aim to demonstrate a robust and effective approach for tackling this specialized legal domain, contributing to the broader discourse on applying advanced NLP techniques to complex, rule-governed knowledge systems.

2 Research Methodology

Our research methodology for QIAS 2025 SubTask 1 involved a comprehensive approach to address the complexities of Islamic inheritance reasoning using Large Language Models. This section details the task definition, dataset characteristics, the models employed, and our training and inference setup.

2.1 Task: Islamic Inheritance Reasoning (*Ilm al-Mawārīth*)

SubTask 1 of QIAS 2025 focuses on evaluating the capabilities of LLMs in understanding and reasoning within Islamic inheritance law (Bouchekif et al., 2025a). The subTask is framed as a multiple-choice question (MCQ) classification problem, where each question has exactly one correct answer. Questions are categorized into two difficulty levels with balanced representation: Beginner (identifying eligible heirs, basic shares, and non-eligible heirs) and Advanced (dealing with multiple heirs, addressing multi-generational cases, fixed estate constraints, and intricate fractional distributions) (Bouchekif et al., 2025b).

The dataset provided for SubTask 1 consists of a total of 22,000 examples, split into 20,000 examples for model training and 1,000 examples for each validation and testing datasets. Each example is an MCQ related to Islamic inheritance, with question text and up to six answer options (A–F).

2.2 Models

We finetune our primary model **Fanar-1-9B-Islamic-Inheritance-Reasoning**[1] based on **Fanar-1-9B**[2], a 9-billion parameter causal decoder-only transformer specifically designed for Arabic and Islamic domain text (Abbas et al., 2025).

In addition to the fine-tuned Fanar-1-9B, we integrated it into a **Retrieval-Augmented Generation (RAG)** pipeline (Lewis et al., 2020) for inference. The RAG setup utilizes the `all-MiniLM-L6-v2`[3] embedding model as a retriever to encode questions and retrieve top-k relevant passages from a **FAISS** index (Johnson et al., 2021; Douze et al., 2024). These retrieved passages are then combined with the question and options to form an enriched Arabic chat prompt, which is fed to the fine-tuned Fanar-1-9B model.

2.3 Training Setup

Our training setup focused on parameter efficiency and memory optimization. To adapt Fanar-1-9B LLM efficiently for our task, we employed **Low-Rank Adaptation (LoRA)** (Hu et al., 2021). LoRA injects trainable rank-decomposition matrices into specific layers while keeping the original weights frozen. This significantly reduces the number of trainable parameters and computational cost. We also applied **4-bit NormalFloat (NF4) quantization** (Dettmers et al., 2023) to reduce GPU memory consumption and enabled **gradient checkpointing** (PyTorch Team, 2025) to reduce peak memory usage. The attention implementation was set to *eager* for improved training stability, and `use_cache` was disabled when gradient checkpointing was enabled. Table 1, provides the key hyperparameters used during model fine-tuning.

Training data were serialized as *system–user–assistant* turns, where the assistant's target output is a single gold letter (A–F). LoRA adapters are applied to attention projection and MLP modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`) with $r = 32$, $\alpha = 64$, and dropout of 0.1.

For the RAG pipeline, the retrieval k was set to 5, meaning the top 5 relevant passages were

[1]available on HuggingFace:https://huggingface.co/msmadi/Fanar-1-9B-Islamic-Inheritance-Reasoning

[2]available on HuggingFace:https://huggingface.co/QCRI/Fanar-1-9B

[3]available on HuggingFace:https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Hyperparameter	Value
Epochs	4
Batch size (per device)	2 (train and eval)
Gradient accumulation steps	32
Learning rate	310^{-4}
Weight decay	0.01
Warmup ratio	0.1
Max gradient norm	1.0
Optimizer	`adamw_torch`
Scheduler	Cosine decay
Precision	FP16

Table 1: Key hyperparameters for fine-tuning.

retrieved. The maximum input length for the RAG inference was 10,000 tokens, and the maximum new tokens generated by the model was 15. A low temperature of 0.05 was used for decoding, along with a greedy decoding strategy to ensure short, deterministic outputs. Answer extraction was performed using a regex-based procedure to select a single choice letter (A–F), see Appendix A for more information about prompting template and template and decoding settings.

3 Evaluation and Results

For the evaluation of our methodology, we compare our final test results with results reported by the task organizers in (Bouchekif et al., 2025b,a) for testing LLMs with zero-shot prompting on the same test set. The evaluation metric for this task is accuracy.

Model	Overall	Beginner	Advanced
o3	93.4	94.4	92.4
Gemini 2.5	90.6	91.6	89.6
QU-NLP	**85.8**	**74.0**	**97.6**
GPT-4.5	74.0	86.8	61.2
LLaMA3	48.8	57.8	39.8
Fanar 7B	48.1	60.4	35.8
Mistral	44.5	58.6	30.4
ALLaM7B	42.9	58.0	27.8

Table 2: Accuracy (%) for each model across difficulty levels. Other models results are based on zero-shot setting using Arabic prompts as reported in (Bouchekif et al., 2025b,a)

As presented in Table 2, QU-NLP, achieved an overall accuracy of 85.8%, outperforming other competitive models such as, GPT 4.5, LLaMA 3 70B[4], Fanar (Islamic-RAG[5]), Mistral-Saba-24B[6] and ALLaM-7B[7] and achieving competitive results behind state of the art commercial LLMs in reasoning capabilities, such as: Gemini 2.5 (flash-preview), OpenAI's o3. While our system did not achieve the top rank, QU-NLP (with RAG) surpassed all models on the advanced subset of the testing dataset (500 MCQs) with accuracy of 97.6%. This result demonstrates the effectiveness of our approach, which combines LoRA fine-tuning of the Fanar-1-9B model with a Retrieval-Augmented Generation (RAG) pipeline, in addressing the complex reasoning challenges posed by Islamic inheritance law. Our model's performance indicates a strong capability in comprehending inheritance scenarios, identifying heirs, and applying the intricate rules required for accurate share calculation.

4 Discussion

We evaluate a multiple-choice inheritance reasoning system on 1,000 items with an overall accuracy of 85.8%. Performance differs sharply by level: *Beginner* = 74.0% (n=500) vs. *Advanced* = 97.6% (n=500). Two phenomena account for most residual errors at the Beginner level. First, items whose correct answer indicates a **محجوب** ("blocked") heir are substantially harder (64.5%, n = 299) than all other cases (94.9%, n = 701), suggesting the model sometimes assigns shares despite the presence of higher-priority heirs. Second, questions containing explicit negation or exception cues (e.g., **بدون/غير/لن/لم/ليس/لا**) yield lower accuracy (83.5%, n = 807) compared to those without negation (95.3%, n = 193), indicating occasional polarity flips.

To further investigate QU-NLP's limitation on blocked cases, we analyzed the count of questions whose gold answer is **محجوب** in the development and training splits. We found that blocked items constitute only 1.70% of development set (17/1,000) but 17.46% of train (3,491/20,000), whereas (for reference) they account for 29.90% of Test (299/1,000). This mismatch—especially the severe under-representation in Development

[4]Available via the Groq API: https://console.groq.com/keys

[5]Available via a free public API: https://api.fanar.qa/request/en

[6]Available via the Groq API: https://console.groq.com/keys

[7]Arabic model hosted on Hugging Face: https://huggingface.co/Abdelaali-models/ALLaM-7B-Instruct-preview

set—helps explain the degraded Test performance on blocked questions (64.55% vs. 94.86% on non-blocked).

A further class of errors results from near-duplicate answer options where orthographic differences (e.g., باقى vs. باقي) leave the semantics unchanged but map to different label IDs. We found 10 such cases (about 7% of all errors). These are dataset artifacts rather than modeling deficiencies. After normalizing Arabic orthography (removing diacritics and unifying letter forms), gold and predicted options collapse to the same string. For transparency, Appendix B lists two misclassified examples across the three categories: (A) blocked heirs (محجوب), (B) negation/exception cues, and (C) near-duplicate option texts, and Table 3 demonstrates the counts of misclassified questions per category of error and level.

Category	Advanced	Beginner	Total
Blocked (محجوب)	0	106	106
Negation-Exception	3	14	17
Near-duplicate options	0	10	10
Other	9	0	9
All errors	12	130	142

Table 3: Misclassification counts by category and level (total errors = 142).

To mitigate these errors, we suggest: (i) adding explicit post-rules or contrastive training focused on hijb (محجوب) cases; (ii) augmenting training with negation/exception rewrites; and (iii) normalizing and deduplicating answer options during dataset curation and evaluation to avoid orthography-induced label mismatches.

Model	All	Beginner	Advanced
Fanar-1-9B (Base)	18.6	22.6	14.6
Fanar-1-9B + LoRA	**86.5**	**76.2**	96.8
Fanar-1-9B + LoRA + RAG	85.8	74.0	**97.6**

Table 4: Results for ablation analysis with accuracy (%) for each model across question difficulty levels.

5 Ablation Analysis Study

We ablate the contributions of (i) the base model (**Fanar-1-9B**), (ii) parameter-efficient specialization via **LoRA** (Hu et al., 2021), and (iii) **RAG** (Lewis et al., 2020) using the same test set and decoding settings.

Table 4 summarizes accuracies. Moving from *Base* to *LoRA (no RAG)* achieved the highest gain of **+67.9** points overall (18.6→86.5), including **+53.6** on *Beginner* (22.6→76.2) and **+82.2** on *Advanced* (14.6→96.8). Adding RAG (*LoRA+RAG*) leads to a small drop overall (**-0.7** points; 86.5→85.8), with a slight decrease on *Beginner* (76.2→74.0) and a slight increase on *Advanced* (96.8→97.6). Hence, RAG helps in answering the advanced cases but can add noise to easy ones. Further investigation on RAG affect can be conducted in future research. The dominant effect in this ablation is therefore the finetuning process using LoRA.

6 Conclusion

This paper presented our system, QU-NLP, for Sub-Task 1: Islamic Inheritance Reasoning at the QIAS 2025 Shared Task. We demonstrated the application of a LoRA fine-tuned Fanar-1-9B causal language model integrated within a Retrieval-Augmented Generation (RAG) pipeline to address the intricate challenges of Islamic inheritance law. Our methodology focused on parameter-efficient fine-tuning and leveraging external knowledge retrieval to enhance the model's reasoning capabilities and factual accuracy in this specialized domain.

Our system achieved an accuracy of 0.858 in the final test, securing a competitive position among the participants. This result highlights the significant potential of combining advanced LLM architectures with retrieval mechanisms for complex, rule-based legal reasoning tasks. We successfully navigated challenges related to memory constraints through techniques like 4-bit NF4 quantization and gradient checkpointing, making the deployment of such large models more feasible.

Future work will explore further enhancements to the RAG pipeline, including more sophisticated retrieval strategies and the potential incorporation of explicit symbolic reasoning components to handle the highly structured nature of Islamic jurisprudence. Additionally, investigating methods for generating interpretable justifications for the model's predictions could provide deeper insights into its

reasoning process and build greater trust in its applications.

References

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, and 22 others. 2025. Fanar: An arabic-centric multimodal generative ai platform.

Alaa N. Akkila and Samy S. Abu Naser. 2016. Proposed expert system for calculating inheritance in islam. *World Wide Journal of Multidisciplinary Research and Development*, 2(9):38–48.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *arXiv preprint*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint*.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is gpt-4 a good islamic expert for answering quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133.

M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Alrashed, Faisal Mirza, Shaykhah Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint*.

Abdessalam Bouchekif, Samer Rashwani, Emad Mohamed, Mutaz Al-Khatib, 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. 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.

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, pages 10088–10115. Curran Associates, Inc.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2022. Qur'an qa 2022: Overview of the first shared task on question answering over the holy qur'an. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, pages 79–87, Marseille, France. European Language Resources Association.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an qa 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: Enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Omar T Mohammedi. 2012. Sharia-complaint wills; principles, recognition, and enforcement. *NYL Sch. L. Rev.*, 57:259.

PyTorch Team. 2025. Gradient checkpointing for large models. `https://pytorch.org/docs/stable/checkpoint.html`.

Mohammad Amaan Sayeed, Mohammed Talha Alam, Raza Imam, Shahab Saquib Sohail, and Amir Hussain. 2025. From rag to agentic: Validating islamic-medicine responses with llm agents. *arXiv preprint*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming

Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint*.

Sadia Tabassum, A. H. M. Hoque, Sharaban Twahura, and Mohammad Osiur Rahman. 2019. Developing an islamic farayez system applying software engineering. *Jurnal Kejuruteraan*, 31(1):25–38.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University-Computer and Information Sciences*, 33(1):68–76.

A Prompting Template

This appendix documents the exact message templates and decoding settings used in all experiments. Unless otherwise noted, *the assistant must output one uppercase letter only* from the set of available options.

1.1 System Message (Arabic)

Content: أنت خبير متخصص في أحكام الميراث الإسلامي والفرائض الشرعية. أجب بدقة واختصار اعتماداً على القواعد الفقهية المعتبرة (القرآن الكريم والسنة والإجماع). ستكون الأسئلة على شكل إختيار من متعدد (ا–س). أعد إجابة نهائية مكونة من حرف واحد فقط من بين الحروف المتاحة دون أي شرح إضافي.

1.2 User Message — No-RAG (Question + Options)

Template:

السؤال: {QUESTION}

الخيارات:
A) {OPTION_A}
B) {OPTION_B}
C) {OPTION_C}
D) {OPTION_D}
E) {OPTION_E}
F) {OPTION_F}

أَعِد حرف الإجابة الصحيحة فقط من الخيارات المتاحة({A,B,C,D,E,F}

1.3 User Message — RAG (Retrieved Evidence + Question + Options)

Template:

المعلومات المرجعية (مختصرة):

- {DOC_1_SNIPPET}
- {DOC_2_SNIPPET}
- {DOC_3_SNIPPET}

ملاحظة: تجاهل أي سياق غير ذي صلة بالمسألة المعروضة.

السؤال: {QUESTION}

الخيارات:
A) {OPTION_A}
.
.
F) {OPTION_F}

أَعِد حرف الإجابة الصحيحة فقط من الخيارات المتاحة({A,B,C,D,E,F}

Parameter	Value
Decoding	Greedy (no sampling)
Temperature	0.05
Top-p	1.0
Max new tokens	15
Input length	5k (No-RAG), 10k (RAG)
Repetition penalty	1.0

Table 5: Decoding parameters used in a all runs.

1.4 Tokenization / Chat Template Notes

We construct messages as (*system*, then *user*). When using HuggingFace chat templates, we call *apply_chat_template(..., add_generation_prompt=true, tokenize=false)* and subsequently tokenize the resulting string with `add_special_tokens=false` to avoid duplicating special tokens.

1.5 Decoding Settings (All Runs)

Table 5 demonstrates the decoding parameters used in a all runs. Given the model text output, we extract the first valid letter from the allowed set. If the first character of the response is already a valid letter, it is taken directly; otherwise we scan for the first occurrence of any valid option. Outputs other than a single letter are truncated to the extracted letter.

We fix decoding to greedy with the settings above. For RAG, we retrieve top-k=5 passages and include their snippets exactly as shown. All ablations use the *same* prompt shape, differing only by (i) the presence/absence of the المعلومات المرجعية block and (ii) the model (base vs. LoRA).

B Misclassified Examples

As presented in Table 6, this appendix explains misclassified examples across different categories.

Category	Question (excerpt)	Gold / Predicted
Blocked (محجوب)	مات وترك: بنت ابن ابن (٢) و بنت (٤) و ابن ابن عم لأب و زوجة و أخ لأب (٣) و أخ شقيق (٣) كم النصيب الأصلي لـ ابن ابن عم لأب من التركة، وما الدليل على ذلك؟	نصيبه هو محجوب، والدليل: لا يرث ابن (C) ابن عم لأب فى وجود الفرع الوارث المذكر - مثل الإبن أو ابن الإبن وإن نزل - ولا الأصل المذكر - مثل الأب وأب الأب وإن علا- ولا فى وجود الإخوة الأشقاء أو لأب ولا عند إجتماع الأخت مع أحد البنات
		نصيبه هو لا شيء، والدليل: لا يرث ابن (D) ابن عم لأب فى وجود الفرع الوارث المذكر - مثل الإبن أو ابن الإبن وإن نزل - ولا الأصل المذكر - مثل الأب وأب الأب وإن علا- ولا فى وجود الإخوة الأشقاء أو لأب ولا عند إجتماع الأخت مع أحد البنات
Blocked (محجوب)	مات وترك: ابن ابن أخ لأب (٤) و أخ شقيق (٢) و عم الأب لأب (٢) و ابن عم شقيق (٤) و أم الأب كم النصيب الأصلي لكل صنف من الورثة من التركة؟	أم الأب: السدس، أخ شقيق (٢): باقى (F) التركة، ابن ابن أخ لأب(٤): محجوب، عم الأب لأب(٢): محجوب، ابن عم شقيق(٤): محجوب
		أم الأب: السدس، أخ شقيق (٢): باقى (A) التركة، ابن ابن أخ لأب(٤): عصبة، عم الأب لأب(٢): محجوب، ابن عم شقيق(٤): محجوب
Negation/Exception	مات وترك: أخت شقيقة (٣) و أخت لأم (٢) و ابن أخ لأب (٢) كم النصيب الأصلي لـ أخت شقيقة (٣) من التركة، وما الدليل على ذلك؟	نصيبه هو الثلثان، والدليل: الأخت الشقيقة (F) - عند عدم الأخ الشقيق - مثلها مثل البنت - إذا لم يكن هناك بنات صلبيات أو بنات ابن - فتأخذ الشقيقة النصف ان كانت واحده والثلثان ان كانتا اثنتين أو أكثر ... وإلا حجبت بهم
		نصيبه هو كل التركة، والدليل: الأخت (B) الشقيقة - عند عدم الأخ الشقيق - مثلها مثل البنت ... وإلا حجبت بهم
Negation/Exception (in explanation)	مات وترك: بنت ابن (٣) و أخ لأب (٢) و ابن أخ لأب (٤) و أب الأب و ابن عم لأب (٢) و ابن عم الأب (٣) كم النصيب الأصلي لـ بنت ابن (٣) من التركة، وما الدليل على ذلك؟	نصيبه هو الثلثان، والدليل: بنات الإبن - (E) مثل بنت الإبن وبنت ابن الإبن - مثلهن مثل البنت بشرط عدم وجود بنت صلبيه أوابن صلبى أو ابن ابن أعلى منهن فيحجبهن. فترث الواحدة من بنات الابن النصف إذا لم يكن هناك ابن ابن فى درجتها يعصبها وترث الأكثر من واحدة الثلثين . قال تعالى (يُوصِيكُمُ اللَّهُ فِي أَوْلادِكُمْ لِلذَّكَرِ مِثْلُ حَظِّ الأُنثَيَيْنِ فَإِنْ كُنَّ نِسَاءً فَوْقَ اثْنَتَيْنِ فَلَهُنَّ ثُلُثَا مَا تَرَكَ وَإِنْ كَانَتْ وَاحِدَةً فَلَهَا النِّصْفُ)
		نصيبه هو لا شيء، والدليل: بنات الإبن - (A) مثل بنت الإبن وبنت ابن الإبن - مثلهن مثل البنت بشرط عدم وجود بنت صلبيه أوابن صلبى أو ابن ابن أعلى منهن فيحجبهن. فترث الواحدة من بنات الابن النصف إذا لم يكن هناك ابن ابن فى درجتها يعصبها وترث الأكثر من واحدة الثلثين . قال تعالى (يُوصِيكُمُ اللَّهُ فِي أَوْلادِكُمْ لِلذَّكَرِ مِثْلُ حَظِّ الأُنثَيَيْنِ فَإِنْ كُنَّ نِسَاءً فَوْقَ اثْنَتَيْنِ فَلَهُنَّ ثُلُثَا مَا تَرَكَ وَإِنْ كَانَتْ وَاحِدَةً فَلَهَا النِّصْفُ)
Near-duplicate options	مات وترك: عم الأب لأب (٤) و أخت لأب (٥) و عم لأب (٢) و أم أم الأب و أم أم الأم كم النصيب الأصلي لـ عم لأب (٢) من التركة، وما الدليل على ذلك؟	نصيبه هو باقى التركة، والدليل: لأنه (B) عصبة
		نصيبه هو باقي التركة، والدليل: لأنه (E) عصبة
Near-duplicate options	مات وترك: أب أب الأب و أخت لأب (٥) و عم الأب (٥) و أم الأب كم النصيب الأصلي لـ أخت لأب (٥) من التركة، وما الدليل على ذلك؟	نصيبه هو باقى التركة، والدليل: لأنه (A) عصبة
		نصيبه هو باقي التركة، والدليل: لأنه (C) عصبة

Table 6: Illustrative misclassified examples across three categories: (A) blocked heirs (محجوب), (B) negation/exception cues, and (C) near-duplicate option texts.