TCE at IslamicEval 2025: Retrieval-Augmented LLMs for Quranic and Hadith Content Identification and Verification

Mohammed ElKoumy[1] **Khalid Allam**[2] **Ahmed Tamer**[2] **Mohammed Elqabalawy**[2]
[1]Rensselaer Polytechnic Institute, Department of Computer Science, Troy, NY, USA
[2]Tanta University, Computer and Control Department, Tanta, Egypt
elkoum@rpi.edu khalidallam222@gmail.com ahmad.tamer1908@gmail.com
mohammed30971488@f-eng.tanta.edu.eg

Abstract

Recent advancements in large language models (LLMs) have opened new possibilities for processing complex natural language tasks, including those involving highly regarded religious content. However, working with divine sources such as the Holy Quran and Hadith presents unique challenges. These Classical Arabic texts have, for centuries, been meticulously preserved and recited word-for-word, allowing no tolerance for errors — even a single incorrect diacritic can entirely alter the meaning. Such sensitivity demands exceptional precision, as hallucinations or inaccuracies from LLMs could lead to significant misinterpretations among general users. To address this challenge, we present an Arabic-focused, LLM-powered framework designed to identify and verify the integrity of religious text generated by widely used LLMs. Evaluation on benchmark subtasks demonstrates strong performance, achieving a Macro-Avg F1 score of **86.11%** on **Subtask 1A** and an Accuracy of **89.82%** on **Subtask 1B**.

1 Introduction

With the superior text generation capabilities of contemporary (LLMs) (Ouyang et al., 2022; OpenAI team, 2024), inaccurate yet plausible content, commonly known as hallucinations, has proliferated across various online platforms and websites (Huang et al., 2025). In response, the research community has developed fact-checking and verification methods grounded in reliable factual resources (Guo et al., 2022; Althabiti et al., 2024).

Given that languages reflect cultures, some of the content generated by LLMs in the Middle East is closely tied to the region's rich Islamic heritage, especially as these models are increasingly used for everyday tasks (Bashir et al., 2023; Mubarak et al., 2025). Consequently, there is a risk that fabricated sacred Islamic content may be generated and mistakenly treated as authentic or employed to reinforce Islamophobia or misinformation. This problem is particularly sensitive due to its significance among Muslim and Arab communities (Mubarak et al., 2025).

In this paper, we present our approach to address these challenges by focusing on the tasks of Islamic content identification and validation, namely **Subtask 1A** and **Subtask 1B** of IslamicEval (Mubarak et al., 2025), respectively.

Given the limited size of the dataset and minimal financial and time resources, our approach adopts a few-shot learning strategy powered by state-of-the-art (SOTA) LLMs to address both tasks (Liu et al., 2023; Ouyang et al., 2022). More specifically, to automatically identify divine texts in **Subtask 1A**, we leverage trigger words and common citation patterns frequently found in religious content (Bashir et al., 2023). For the verification subtask, i.e, **Subtask 1B**, we employ a retrieval-augmented LLM architecture with integrated content validation, enabling precise cross-checking of generated text against authoritative Islamic sources (Guo et al., 2022; Mubarak et al., 2025).

Our contributions are as follows:

- We achieve strong performance on both tasks using powerful multilingual LLMs such as Qwen-235B (MoE) and GPT-4o. Our results confirm that a carefully designed prompt can lead to superior performance across both tasks.
- To make verification feasible despite the large size of the authentic reference resources, we employ a lexical matching system to retrieve the most relevant verses and implement an efficient early exit strategy once verification is successful. In addition, we empirically demonstrate the effectiveness of this retrieval phase.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 515–527
November 8-9, 2025 ©2025 Association for Computational Linguistics

- We validate the consistency of our results by demonstrating strong agreement between the development set performance and the hidden final test dataset.
- We share our code[1] with the community to promote broader accessibility and encourage further exploration and improvement on this essential problem.

We organize the paper as follows. Section 2 presents the background for the task and related literature. In Section 3, we provide a detailed description of the system design for both tasks. Subsequently, Section 4 highlights the key experimental details and running configurations. Section 5 presents the results along with analysis and findings. Finally, Section 6 concludes the work.

2 Background

Attention mechanisms and the Transformer architecture have revolutionized NLP by enabling models to effectively capture long-range dependencies (Vaswani et al., 2017). Models like BERT and its Arabic variants, e.g., AraBERT (Antoun et al., 2020), have further showcased the success of these advancements for both multilingual and Arabic NLP tasks (Devlin et al., 2019). Recently, LLMs such as GPT-4 have demonstrated impressive few-shot learning capabilities (Brown et al., 2020; OpenAI team, 2024), allowing them to perform a wide range of tasks with minimal task-specific tuning. Meanwhile, prompt engineering has emerged as a crucial technique to tailor these powerful models to specialized applications (Liu et al., 2023).

Accurate processing of Islamic sacred texts is essential due to their cultural and religious significance in the Arabic world. NLP tasks targeting these texts include question answering (QA), content retrieval, morphological analysis, and recitation correction, among others (Bashir et al., 2023). Prior shared tasks, notably Qur'an QA 2022 and 2023, have laid the groundwork by focusing on QA over the Noble Quran using retrieval and comprehension techniques (Malhas et al., 2022, 2023). Central to these efforts, retrieval methods based on lexical approaches such as TF-IDF and BM25 continue to play a fundamental role in effectively locating relevant verses or narrations (Salton and Buckley, 1988).

Building upon previous endeavors, IslamicEval 2025 tackles the critical challenge of hallucination detection in LLM-generated Islamic content, emphasizing the accuracy and integrity of Quranic and Hadith references (Mubarak et al., 2025). The competition comprises the following subtasks:

- **Subtask 1A: Identification** — Detect spans of Quranic verses (Ayahs) and Hadiths within free-text responses generated by LLMs.
- **Subtask 1B: Validation** — Assess each identified utterance against authoritative sources to distinguish accurate references from hallucinated content.
- **Subtask 1C: Correction** — Generate corrected versions of any erroneously generated Ayahs or Hadiths based on authentic sources.
- **Subtask 2: Passage Retrieval** — Retrieve a ranked list of Quranic or Hadith passages that potentially answer a given question posed in Modern Standard Arabic.

As previously noted, this work presents our solutions for the **1A** and **1B** subtasks. Detailed dataset statistics for both subtasks are provided in Tables 4 and 5 in Appendices A.1 and B.1 respectively.

3 System Design

Our approach leverages few-shot learning with SOTA foundational LLMs to address both subtasks. For **1B** subtask, we propose a retrieval-augmented architecture to perform the verification procedure.

3.1 Subtask 1A: Span Extraction For Identification

For this subtask, we formulate the problem as a span extraction task, where the system identifies textual segments referencing Quranic verses and Hadith within generated responses (Mubarak et al., 2025). Our approach employs a powerful foundational LLM (Yang et al., 2025), guided by a carefully designed few-shot prompt to extract relevant spans (OpenAI team, 2024; Liu et al., 2023; Brown et al., 2020). These prompts emphasize commonly occurring trigger words and citation patterns characteristic of sacred Islamic texts, enabling effective identification given the structured nature of the citation process (Bashir et al., 2023) (See Appendix A.2 for detailed prompts in Figure 3). To ensure the input remains manageable for the LLM

[1] The code and resources are available at https://github.com/m-alqblawi/Islamic_Eval_2025

while preserving essential information, we apply chunking to segment the input into appropriately sized portions.

Driven by the limited size of the training dataset, we forego extensive task-specific fine-tuning and instead leverage the strong generalization capabilities of foundational LLMs for span extraction (Devlin et al., 2019; Vaswani et al., 2017). Subsequently, to accurately align the extracted spans with their precise locations in the generated text, the system incorporates a fuzzy matching module (Platenius et al., 2013; Salton and Buckley, 1988) that accounts for minor variations and inconsistencies. Spans with fuzzy matching scores below a predefined threshold are discarded to maintain high precision and minimize false positives. The complete system architecture is illustrated in Figure 2, with algorithmic details provided in Algorithm 1 in Appendix A.2.

3.2 Subtask 1B: Retrieval-Augmented Verification

Our approach for Subtask **1B** consists of three main phases that integrate a powerful foundational LLM with a retrieval mechanism tailored for Quranic and Hadith verification. We model this subtask as two independent few-shot binary classification problems — one for Quran verification and another for Hadith verification.

First, we retrieve relevant passages from authenticated Quranic and Hadith sources using a hybrid retrieval strategy. For Quranic material, retrieval leverages fuzzy matching based on the `py_quran` Python package (Yousef et al., 2018). Our approach performs verse-level retrieval by tokenizing the query into individual words and computing a weighted matching score for each verse based on the frequency and presence of these words. Specifically, a voting or counting map is constructed where each word match contributes to the verse's overall relevance score, allowing the system to identify the most pertinent verses despite minor textual and scripting variations.

For Hadith content, we employ a character-level TF-IDF ranking approach with character n-grams to capture fine-grained textual patterns (Salton and Buckley, 1988). After retrieval, a postprocessing algorithm is applied to the Quranic results to merge adjacent retrieved verses from the same surah into coherent contiguous segments, enhancing context and verification accuracy before input to the LLM. Subsequently, these consolidated retrieval results form the input context for the LLM, which determines the correctness of the claims through few-shot prompting. In our prompt template, we provide few-shot demonstration examples independently for Quranic and Hadith texts (detailed prompts shown in Figure 5 in Appendix B.2).

For Quran verification, the LLM is tasked with strict word-for-word matching due to the sensitivity of small textual changes on meaning (Bashir et al., 2023). In contrast, Hadith verification tolerates minor variations in the *matn* (narrative text), acknowledging authentic variations in Prophetic sayings.

It is worth mentioning that the verification LLM is invoked sequentially on each retrieved result independently. If a match is found by the LLM, the sequential process terminates early, mirroring the human strategy of stopping once sufficient evidence is found. A comprehensive system architecture is presented in Figure 4, with the detailed algorithmic implementation described in Algorithm 2 in Appendix B.2.

4 Experimental Details

For both tasks, we utilized the original development and test splits. Due to constraints in budget and time, our experiments did not involve exhaustive exploration of all possible parameters and configurations. We leave this comprehensive investigation to future work and the community.

For Subtask **1A**, we employ various multilingual Qwen and LLaMA3 LLMs (Yang et al., 2025; Grattafiori et al., 2024), accessing all open-source models via the Hugging Face API. The LLM is prompted to output the extracted spans in a structured JSON format. The fuzzy matching threshold is set to a high value of 90% for precise matching and robustness against hallucination. Chunking was applied consistently throughout all experiments using sentence-aware segmentation with a 800-character limit, preserving semantic boundaries at Arabic and standard punctuation marks. To assess the impact of this technique, we conducted a minor ablation study by disabling chunking for our top-performing model.

In Subtask **1B**, for Quran retrieval, since citations must be exact word-by-word matches, we combine a proximity score between matched words to preserve their relative order, along with a coverage score representing the proportion of matched words within each potential ayah. For Hadith retrieval, we employ a TF-IDF module configured with character n-grams up to 7-grams to capture fine-grained textual patterns. For verification, we experimented with two distinct models; we utilized the open-source Gemma model (Team et al., 2024), accessed through OpenRouter, alongside GPT-4 via the OpenAI API (OpenAI team, 2024).

5 Results and Analysis

Subtask 1A: Table 1 presents the validation set performance for span extraction across different LLMs. The Qwen3-235B-A22B-Instruct (MOE) model achieves the best overall performance with an accuracy of **0.860** and a macro-average F1 score of **0.765**, demonstrating superior capability in identifying Islamic content spans. Notably, the comparison between the chunked and non-chunked versions of the same model reveals the significant impact of preprocessing: the model without chunking achieves substantially lower performance (0.795 accuracy vs. 0.860), confirming the importance of chunking preprocessing for maintaining model performance on longer text inputs. Among smaller models, Qwen14B shows competitive precision (0.807), while Llama-3.3-70B-Instruct lags behind other models across all metrics.

Subtask 1B: Table 2 shows the binary classification results for Islamic content verification. GPT-4o with Arabic diacritics achieves the highest performance with an accuracy of **0.9** and F1 score of **0.92**, significantly outperforming all Gemma variants. Among the Gemma models, the 12B variants consistently outperform 4B variants, with Gemma-12B-IT (with diacritics) achieving 0.737 accuracy compared to 0.676 for Gemma-4B-IT.

Our deeper analysis of these results reveals several critical insights: **(1)** The high recall rates achieved by the full pipeline across all experimental conditions (consistently above 95% as shown in Table 2) indicate that our hybrid retrieval architecture effectively captures relevant Islamic content from authoritative sources. However, as evidenced in Tables 6, 7, and 8, **(2)** we observe a consistent pattern toward Type I errors (false positives are underlined and italicized in all confusion matrices for clarity), suggesting that LLM verifiers are occasionally deceived by similar Islamic content generated by powerful language models.

(3) Removing diacritics generally reduces performance across all model sizes, with accuracy drops of 2-3 percentage points (e.g., Gemma-12B drops from 0.737 to 0.709). This performance degradation is particularly pronounced in Quranic content compared to Hadith content, especially for GPT-4o, suggesting that diacritical marks are essential for understanding nuanced Quranic text where subtle diacritical differences significantly impact meaning. **(4)** Verification errors are significantly more prevalent in Quranic content than in Hadith content, indicating that Quranic language presents greater verification challenges. This disparity stems from two key factors: first, the strict word-for-word preservation requirements in Quranic text compared to the relatively acceptable variations in Hadith transmission; and second, the precise linguistic requirements and rich diacritical structure inherent to Quranic Arabic. In contrast, Hadith content allows for authentic variations in transmission across different narrations, making it inherently more tolerant of minor textual discrepancies. Given GPT-4o's superior discriminative capabilities compared to open-source Gemma variants, these structural differences between Quranic and Hadith content explain why GPT-4o consistently produced the fewest errors across all verification tasks.

Official Test Set Performance: Table 3 reports the final results on the hidden test set as provided by the IslamicEval 2025 organizers. Our best-performing models, **Qwen3-235B-A22B-Instruct for Subtask 1A** and **GPT-4o for Subtask 1B**, achieved strong performance on the official evaluation: **0.861 macro-average F1** for span identification and **0.898 accuracy** for verification, respectively. These results demonstrate the effectiveness of our hybrid approach combining large language models with domain-specific preprocessing and retrieval strategies for Islamic content processing tasks.

Task	**Metric**	**Score**
1A (Qwen3-235B)	Macro F1	0.861
1B (GPT-4o)	Accuracy	0.898

Table 3: Official Test Results from IslamicEval 2025

6 Conclusion

We present a framework for identifying and verifying Islamic content in LLM-generated text, addressing hallucination detection in sacred Arabic texts. Our approach combines SOTA multilingual LLMs with domain-specific preprocessing and retrieval-augmented verification strategies.

Index	Model	Accuracy	Precision	Recall	Macro-F1
1	Qwen3-8B	0.836	0.778	0.766	0.751
2	Qwen14B	0.835	**0.807**	0.781	0.765
3	Qwen3-32B	0.804	0.795	0.772	0.758
4	Llama-3.3-70B-Instruct	0.731	0.743	0.698	0.700
5	Qwen3-235B-A22B-Instruct (MOE)	**0.860**	0.801	**0.789**	**0.765**
6	Qwen3-235B-A22B-Instruct (MOE)†	0.795	0.769	0.748	0.719

†Without chunking preprocessing step.

Table 1: Validation Set Performance for Official Split on Subtask 1A. Models are ordered by parameter size from the smallest to largest.

Index	Model	Accuracy	Precision	Recall	F1
1	Gemma-4B†	0.664	0.642	**0.986**	0.777
2	Gemma-4B	0.676	0.652	0.980	0.783
3	Gemma-12B†	0.709	0.674	0.986	0.801
4	Gemma-12B	0.737	0.697	0.986	0.817
5	GPT-4o†	0.87	0.82	**0.986**	0.9
6	**GPT-4o**	**0.9**	**0.87**	**0.986**	**0.92**

†Without diacritics.

Table 2: Validation Set Performance for Subtask 1B

Our results demonstrate strong performance: **86.11%** macro-average F1 on Subtask **1A** and **89.82%** accuracy on Subtask **1B**. Key findings include the critical importance of chunking preprocessing for longer text inputs. The retrieval-augmented approach enables precise cross-checking against authoritative sources while maintaining computational efficiency through early termination strategies.

This work contributes to the broader effort of ensuring accuracy and integrity in AI-generated religious content, addressing a critical need for the Muslim community. We hope our publicly available code and findings facilitate further exploration and improvement in this essential domain.

Limitations

As noted in prior studies (Farghaly and Shaalan, 2009; Bashir et al., 2023), NLP for Islamic content is challenged by the limited availability of sizable datasets and constrained computational resources. Our work similarly faces these limitations, as it requires more extensive experimentation across a diverse range of LLMs to fully assess performance and robustness. Furthermore, the development of a reasonably sized, well-annotated dataset representative of the varied nature of Islamic texts would be instrumental in enabling more effective learning-based approaches. Such datasets could facilitate the use of smaller, more efficient LLMs to perform Islamic content processing and classical Arabic language tasks with higher accessibility and lower computational cost. Addressing these limitations remains an important direction for future research.

Acknowledgments

We sincerely thank the organizers for their dedication to advancing Islamic NLP and Information Retrieval research, especially in this groundbreaking era of powerful LLMs.

References

Saud Althabiti, Mohammad Ammar Alsalka, and Eric Atwell. 2024. Ta'keed: The first generative fact-checking system for arabic claims. *Preprint*, arXiv:2401.14067.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Muhammad Huzaifa Bashir, Aqil M. Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and

Junaid Qadir. 2023. Arabic natural language processing for qur'anic research: a systematic review. *Artificial Intelligence Review*, 56(7):6801–6854.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*, 8(4).

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. *Preprint*, arXiv:2407.21783.

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in large multilingual translation models. *Preprint*, arXiv:2303.16104.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

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 *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, 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 (Hybrid). Association for Computational Linguistics.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Darwish, and Walid Magdy. 2025. IslamicEval 2025: The First Shared Task of Capturing LLMs Hallucination in Islamic Content. In *Proceedings of ArabicNLP 2025*, TBD. Association for Computational Linguistics. To appear.

OpenAI team. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Marie C. Platenius, Markus von Detten, Steffen Becker, Wilhelm Schäfer, and Gregor Engels. 2013. A survey of fuzzy service matching approaches in the context of on-the-fly computing. In *Proceedings of the 16th International ACM Sigsoft Symposium on Component-Based Software Engineering*, CBSE '13, page 143–152, New York, NY, USA. Association for Computing Machinery.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

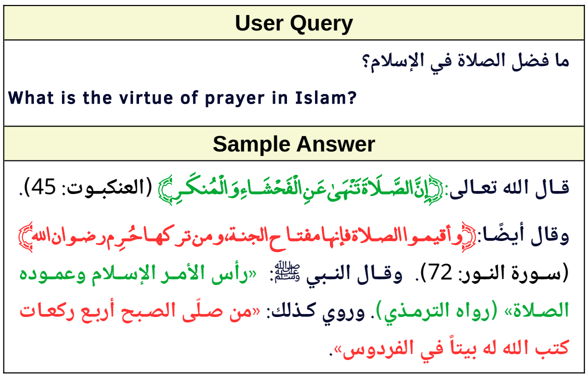
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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. *Preprint*, arXiv:2505.09388.

Waleed A. Yousef, Taha M. Madbouly, Omar M. Ibrahime, Ali H. El-Kassas, Ali O. Hassan, and Abdallah R. Albohy. 2018. Pyquran: The python package for quranic analysis. https://hci-lab.github.io/PyQuran-Private.

Appendix

Figure 1 illustrates a sample hallucinated output generated by GPT-4o, demonstrating that even SOTA models can produce inaccurate Arabic Islamic content (Guerreiro et al., 2023).



User Query

ما فضل الصلاة في الإسلام؟

What is the virtue of prayer in Islam?

Sample Answer

قال الله تعالى: ﴿إِنَّ الصَّلَاةَ تَنْهَىٰ عَنِ الْفَحْشَاءِ وَالْمُنكَرِ﴾ (العنكبوت: 45).
وقال أيضًا: ﴿وأقيموا الصلاة فإنها مفتاح الجنة، ومن تركها حُرِم رضوان الله﴾
(سورة النور: 72). وقال النبي ﷺ: «رأس الأمر الإسلام وعموده
الصلاة» (رواه الترمذي). وروي كذلك: «من صلّى الصبح أربع ركعات
كتب الله له بيتاً في الفردوس».

Figure 1: Sample generated content by GPT-4o with color-coded verification: green indicates correct content, while red highlights invented Quran or Hadith. Some irrelevant content was truncated for clarity.

A Subtask 1A: Islamic Content Identification

A.1 Dataset Details

Table 4 presents the statistical analysis of the dataset for subtask **1A**. The dataset demonstrates varying annotation densities and imbalanced label distributions across the identification task.

A.2 System Design Details

Figure 2 provides an overall view of the system design for subtask **1A**. Algorithm 1 demonstrates the algorithmic pseudocode for the span extraction problem. Figure 3 shows the few-shot prompt template used for Islamic content identification.

Metric	Value
Unique Questions	50
Annotations per Question	4.20 ± 4.30
Ayahs per Question	2.36 ± 3.26
Hadiths per Question	1.52 ± 2.47
Label Distribution	
Ayah	118
Hadith	76
NoAnnotation	16

Table 4: Subtask **1A** dataset statistics: annotation density and class distribution for span extraction task.

B Subtask 1B: Islamic Content Verification

B.1 Dataset Details

Table 5 presents the statistical analysis of the dataset for subtask **1B**. The dataset demonstrates imbalanced label distributions across the binary classification verification task.

Metric	Value
Number of samples	247
Number of verses	4940
Number of unique questions	50
WrongAyah	70
CorrectAyah	110
WrongHadith	30
CorrectHadith	37

Table 5: Subtask **1B** dataset statistics: sample distribution and verification labels for binary classification task.

B.2 System Design Details

Figure 4 provides an overall view of the system design for subtask **1B**. Algorithm 2 demonstrates the algorithmic pseudocode for the verification problem. Figure 5 shows the few-shot prompt template used for binary classification in content verification.

B.3 Additional Results

Tables 6, 7, and 8 present comprehensive confusion matrices for different model configurations, evaluating performance across overall metrics, Quranic content verification, and Hadith content verification respectively.

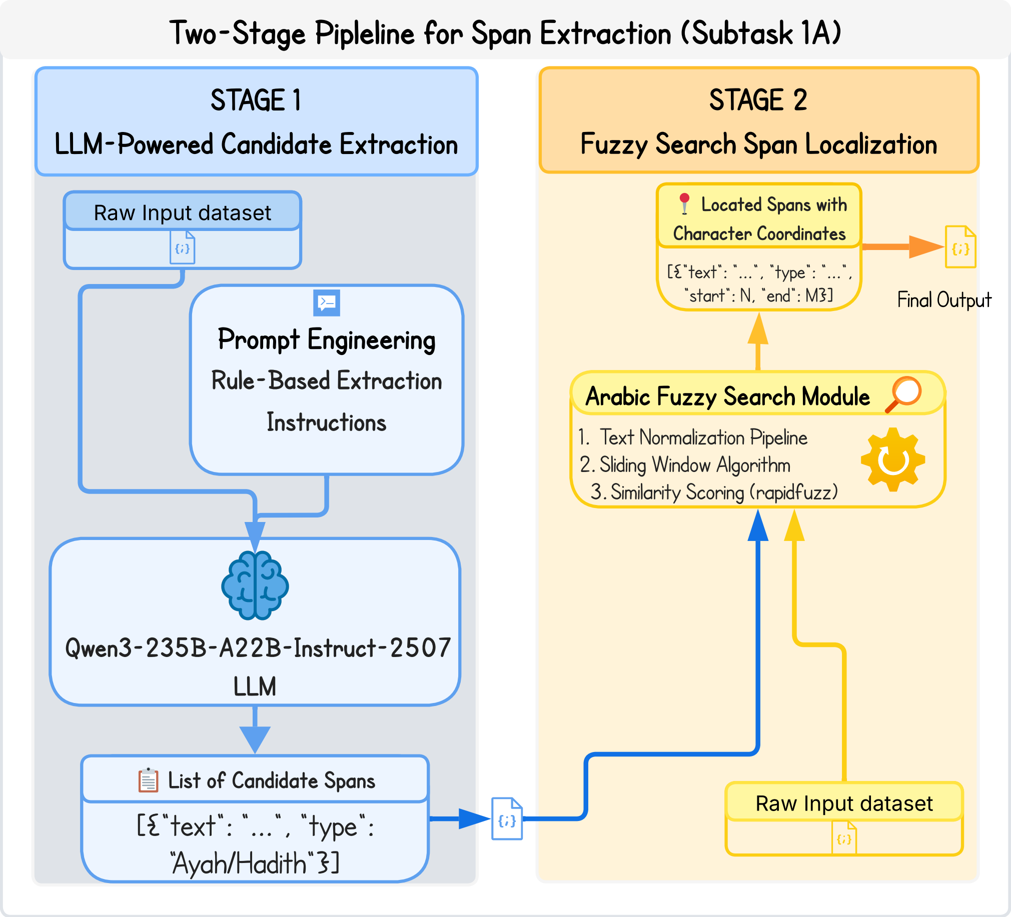

Two-Stage Pipleline for Span Extraction (Subtask 1A)
STAGE 1
LLM-Powered Candidate Extraction
Raw Input dataset
Prompt Engineering
Rule-Based Extraction
Instructions
Qwen3-235B-A22B-Instruct-2507
LLM
List of Candidate Spans
[{"text": "...", "type": "Ayah/Hadith"}]
STAGE 2
Fuzzy Search Span Localization
Located Spans with
Character Coordinates
[{"text": "...", "type": "...", "start": N, "end": M}]
Final Output
Arabic Fuzzy Search Module
1. Text Normalization Pipeline
2. Sliding Window Algorithm
3. Similarity Scoring (rapidfuzz)
Raw Input dataset

Figure 2: Overall system architecture for Islamic content Identification (Subtask 1A).

Algorithm 1 Span Extraction with Fuzzy Matching

```
Require: Generated response text $T$, pretrained LLM, prompt template $P$, fuzzy matching threshold $\theta$
Ensure: Extracted and verified spans $S$
Define $\mathcal{F}(s, T)$ as fuzzy matching function returning set of matched entries with similarity scores
$T_{chunks} \leftarrow$ chunk text $T$ into manageable segments for LLM processing
Construct few-shot prompt $P$ emphasizing trigger words and citation patterns
$S_{\text{raw}} \leftarrow \emptyset$
for all chunk $c$ in $T_{chunks}$ do
    $S_{chunk} \leftarrow$ output spans extracted by LLM using prompt $P$ on chunk $c$
    $S_{\text{raw}} \leftarrow S_{\text{raw}} \cup S_{chunk}$
end for
$S \leftarrow \emptyset$
for all span $s$ in $S_{\text{raw}}$ do
    $M_s \leftarrow \mathcal{F}(s, T)$                                  ▷ Get matching results
    $m_s \leftarrow \max_{(e,\text{score}) \in M_s}$ score               ▷ Select highest similarity score
    if $m_s \geq \theta$ then
        $S \leftarrow S \cup \{s\}$                                     ▷ Add span to verified set
    end if
end for
return $S$
```

System Prompt for A1 Subtask

هذه مهمة استخراج مقاطع نصية (Span Extraction Task).

سأعطيك مقطعًا نصيًا باللغة العربية أنتجه نموذج لغة كبير (LLM).

مهمتك أن تقرأ النص بعناية وتحدد أي مقاطع منه هي:

- استخرجها آيات قرآنية حقيقية أو منسوبة إلى القرآن الكريم (حتى لو كانت منسوبة بشكل غير صحيح)
- أستخرج كل الاحاديث نبوية صحيحة أو منسوبة إلى النبي صلى الله عليه وسلم (حتى لو كانت منسوبة بشكل غير صحيح)

شروط الاستخراج:

1. لا تعتبر النص آية أو حديث إلا إذا وردت عبارة تمهيدية صريحة قبلها مباشرة.

أمثلة لعبارات تمهيدية لآيات قرآنية:

- قال الله تعالى
- قوله تعالى
- قوله تعالى (
- كما قال تعالى
- يقول الله عز وجل
- كما جاء في القرآن الكريم
- وقد ورد في القرآن الكريم
- وأنزل الله تعالى
- في قوله تعالى
- كما ورد في كتاب الله
- في آية من كتاب الله
- كما قال في القرآن
- جاء في القرآن
- نصت الآية الكريمة
- فبالرجوع إلى الآية الكريمة
- جاء في آية من القرآن
- كما نص القرآن الكريم
- كما تضمنته آية من القرآن

أمثلة لعبارات تمهيدية لأحاديث نبوية:

- قال رسول الله صلى الله عليه وسلم
- قال النبي ﷺ
- كما جاء عن النبي صلى الله عليه وسلم
- كما ذكر الحديث الشريف
- كما روى مسلم وأبو داود وابن ماجه
- وفي الحديث الشريف
- عن النبي صلى الله عليه وسلم
- فقال لها النبي صلى الله عليه وسلم
- كما في الحديث
- كما صح عن النبي
- فيما رواه النبي صلى الله عليه وسلم
- جاء في الحديث الشريف
- ورد في الحديث الشريف
- كما ورد عن رسول الله
- قال عليه الصلاة والسلام
- في قول النبي صلى الله عليه وسلم

2. يجب أن يأتي نص الآية أو الحديث مباشرة بعد العبارة التمهيدية، مع السماح فقط بعلامات ترقيم بسيطة أو كلمة وصل مثل 'أن'.
3. تجاهل أي نصوص أو أمثلة أو إعادة صياغة أو شروحات حتى لو كانت مشابهة في الأسلوب.
4. لا تصحح أو تكمل أو تعدل النصوص؛ استخرجها كما هي في النص.
5. لا تتضمن أي أقواس أو محتوياتها مثل () أو [] أو {}.

تنسيق الإخراج المطلوب:

- أعد قائمة JSON صالحة تمامًا (قابلة للتحويل بـ json.loads) تحتوي على عناصر، كل عنصر كائن له:
  - **'text':** نص المقطع.
  - **'type':** إما 'Ayah' أو 'Hadith'.
  - **مثال صحيح:**

[{"text": "...", "type": "Ayah"},
{"text": "...", "type": "Hadith"}]

- إن لم تجد أي مقاطع، أعد: []
- لا تضف أي شرح أو نص خارج القائمة.
- تأكد من شمول جميع الايات و الاحاديث في النص

النص (صادر عن **LLM**): {text}

Figure 3: Few-shot prompt template for span extraction in Subtask 1A: Islamic content identification using trigger words and citation patterns.

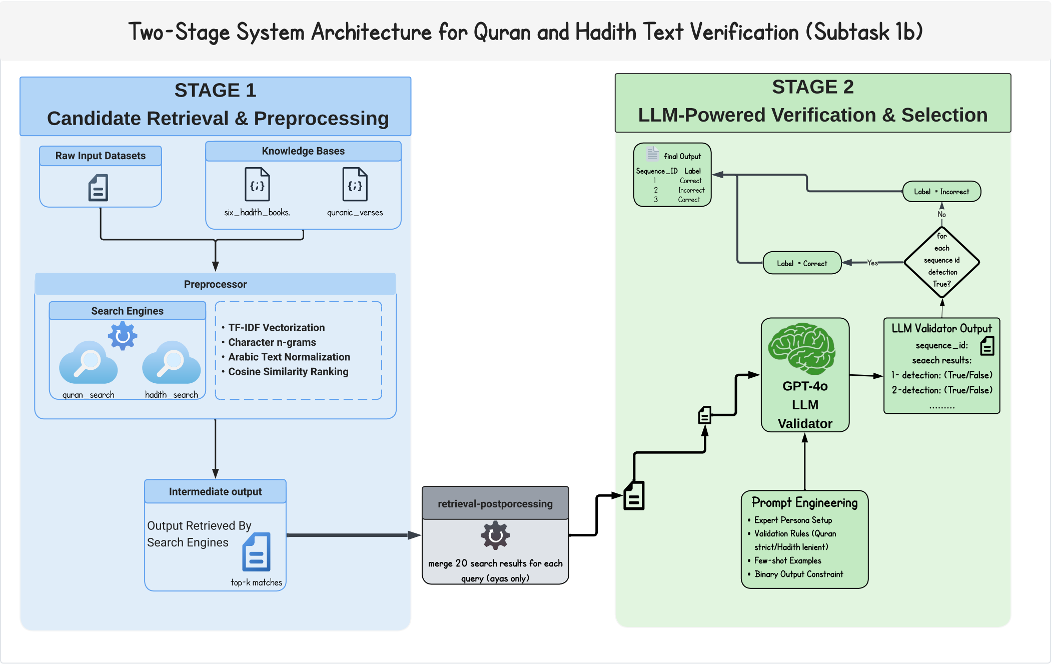


Figure 4: Overall system architecture for Islamic content Verification (Subtask 1B).

Algorithm 2 Verification with Hybrid Retrieval for Subtask 1B

```
Require: Extracted span $s$, Quranic database $DB_Q$, Hadith database $DB_H$, LLM, prompt templates
    $P_Q, P_H$, retrieval threshold $k$
Ensure: Verification result: Verified or Not Verified
Content type $t$ is provided from input file (Quranic or Hadith)
if $t$ is Quranic then
    Tokenize span $s$ into words $W = \{w_1, w_2, \ldots, w_n\}$
    $R \leftarrow$ retrieve top $k$ verses from $DB_Q$ using word-level voting
    $R_{merged} \leftarrow$ merge adjacent verses from same surah in $R$
    $P \leftarrow P_Q$                                   ▷ Strict word-for-word matching
else
    $R \leftarrow$ retrieve top $k$ Hadith entries from $DB_H$ using char-level TF-IDF
    $R_{merged} \leftarrow R$                            ▷ No merging for Hadith
    $P \leftarrow P_H$                                   ▷ Allow minor variations
end if
for all retrieved result $r$ in $R_{merged}$ do
    result $\leftarrow$ LLM($P$, $s$, $r$)              ▷ Few-shot binary classification
    if result is Verified then
        return Verified                                  ▷ Early termination
    end if
end for
return Not Verified
```

System Prompt for B1 SubTask

You are a highly knowledgeable expert in Quranic and Hadith text verification.
You will be given two texts:

- **“query_text”:** This text may contain errors, partial phrases, or slight variations and is NOT guaranteed to be an exact excerpt from the Quran or Hadith.
- **“candidate_text”:** This is a literal, exact excerpt taken from either the Quran or Hadith, free from errors.

Your task:

1. Ignore all Arabic diacritics **(tashkeel)** in both texts during comparison.
2. For Quranic verses **("ayah_text")**, require strict literal substring matching ignoring diacritics and spacing.
3. For Hadith texts **("hadithTxt"**), allow slight leniency in wording or conversational phrasing—small paraphrases or reordering are acceptable-but the core meaning and most of the key phrases should be clearly present.
4. Respond ONLY with a single word:
    - **"True"** if the candidate text validly matches the query according to the above criteria.
    - **"False"** otherwise.

Examples:

Quran Example 1:
query_text: "يسرنا القرآن للذكر"
candidate_text: "ولقد يسرنا القرآن للذكر فهل من مدكر"
Answer: True
Explanation: Literal substring present ignoring diacritics.

Quran Example 2:
query_text: "لقد أرسلنا من قبلك رسلا و آتيناهم أيات ودافعنا عنهم الذين كفروا و كنا لهم عضد"
candidate_text: "ولقد أرسلنا من قبلك في شيع الأولين"
Answer: False
Explanation: No exact substring match.

Hadith Example 1:
query_text: "ما يصيب المؤمن من شوكة فما فوقها إلا رفعه الله بها درجة، أو حط عنه بها خطيئة"
candidate_text: " حدثنا محمد بن عبد الله بن نمير قال رسول الله صلى الله عليه وسلم لا تصيب المؤمن شوكة فما فوقها إلا قص الله بها من خطيئته."
Answer: True
Explanation: Despite slight wording differences, core meaning and key phrases are clearly present with acceptable phrasing variations.

Hadith Example 2:
query_text: "إذا مات المؤمن انتقل إلى الجنة مباشرة"
candidate_text: "عن النبي صلى الله عليه وسلم قال: المؤمن إذا قبض توضع روحه في تاج من نور ينير ما بين المشرق والمغرب."
Answer: False
Explanation: Candidate text does not contain the key content or meaning of the query.

Now evaluate:
query_text: {query}
candidate_text: {text}
Answer:

Figure 5: Few-shot prompt template for binary classification in Subtask 1B: Quranic and Hadith content verification against authoritative sources.

Gemma-4B-IT (with diacritics)			**Gemma-4B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	144 (58.3%)	3 (1.2%)	Correct	145 (58.9%)	2 (0.8%)
Incorrect	*77 (31.2%)*	23 (9.3%)	Incorrect	*81 (32.9%)*	19 (7.7%)
Gemma-12B-IT (with diacritics)			**Gemma-12B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	145 (58.8%)	2 (0.8%)	Correct	145 (58.7%)	2 (0.8%)
Incorrect	*63 (25.5%)*	37 (15.0%)	Incorrect	*70 (28.3%)*	30 (12.1%)
GPT-4o (with diacritics)			**GPT-4o (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	145 (58.7%)	2 (0.81%)	Correct	145 (58.7%)	2 (0.81%)
Incorrect	*22 (8.9%)*	78 (31.57%)	Incorrect	*31 (12.5%)*	69 (27.93%)

Table 6: Confusion Matrices for Gemma and GPT Models (Overall Performance)

Gemma-4B-IT (with diacritics)			**Gemma-4B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	108 (60.0%)	2 (1.1%)	Correct	109 (60.6%)	1 (0.6%)
Incorrect	*57 (31.7%)*	13 (7.2%)	Incorrect	*62 (34.4%)*	8 (4.4%)
Gemma-12B-IT (with diacritics)			**Gemma-12B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	109 (60.6%)	1 (0.6%)	Correct	109 (60.6%)	1 (0.6%)
Incorrect	*49 (27.2%)*	21 (11.7%)	Incorrect	*52 (28.9%)*	18 (10.0%)
GPT-4o (with diacritics)			**GPT-4o (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	109 (60.6%)	1 (0.6%)	Correct	109 (60.5%)	1 (0.6%)
Incorrect	*19 (10.6%)*	51 (28.33%)	Incorrect	*28 (15%)*	42 (23.33%)

Table 7: Confusion Matrices for Gemma and GPT Models (Quranic Content)

Gemma-4B-IT (with diacritics)			**Gemma-4B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	36 (53.7%)	1 (1.5%)	Correct	36 (53.7%)	1 (1.5%)
Incorrect	*20 (29.9%)*	10 (14.9%)	Incorrect	*19 (28.4%)*	11 (16.4%)
Gemma-12B-IT (with diacritics)			**Gemma-12B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	36 (53.7%)	1 (1.5%)	Correct	36 (53.7%)	1 (1.5%)
Incorrect	*14 (20.9%)*	16 (23.9%)	Incorrect	*18 (26.9%)*	12 (17.9%)
GPT-4o (with diacritics)			**GPT-4o (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	36 (53.7%)	1 (1.5%)	Correct	36 (53.7%)	1 (1.5%)
Incorrect	*3 (4.5%)*	27 (40.3%)	Incorrect	*3 (4.5%)*	27 (40.3%)

Table 8: Confusion Matrices for Gemma and GPT Models (Hadith Content)

Category	**Accuracy**	**Precision**	**Recall**	**F1 Score**
GPT-4o (no diacritics)				
Overall	86.6%	82.4%	98.6%	89.8%
Quran	83.9%	79.6%	99.1%	88.3%
Hadith	94.0%	92.3%	97.3%	94.7%
GPT-4o (with diacritics)				
Overall	90.3%	86.8%	98.6%	92.4%
Quran	88.9%	85.2%	99.1%	91.6%
Hadith	94.0%	92.3%	97.3%	94.7%

Table 9: GPT-4o Performance Metrics for Subtask 1B