HUMAIN at IslamicEval 2025 Shared Task 1: A Three-Stage LLM-Based Pipeline for Detecting and Correcting Hallucinations in Quran and Hadith

Arwa Omayrah **Sakhar Alkhereyf** **Ahmed Abdelali**

Abdulmohsen Althubaity **Jeril Kuriakose** **Ibrahim AbdulMajeed**
HUMAIN, Saudi Arabia
aomayrah,salkhereyf@humain.com

Abstract

This paper presents HUMAIN's submission to the IslamicEval 2025 Shared Task 1, addressing hallucination detection and correction in Quranic and Hadith LLM-generated content. Our three-stage pipeline covers: (1) Span Detection via sequence-to-sequence annotation using TANL-style markup, (2) Validation with retrieval-based similarity and substring matching against reference corpora, and (3) Correction through exact matching, LCS alignment, and semantic re-ranking. On the official test set, our system achieved 87.2% F-1 for span detection, 86.1% accuracy for validation, and 68.2% accuracy for correction. While systematic detection is highly achievable, meaningful correction remains limited by semantic complexity where small textual differences can significantly impact religious understanding. This work presents a multi-stage LLM-based pipeline for Islamic content verification.

1 Introduction

Large Language Models (LLMs) enable advanced text generation but suffer from hallucination—producing linguistically fluent yet factually incorrect text. While problematic across domains, hallucinations pose critical risks in religious contexts, especially for the Quran and Hadith, where accuracy is essential. Even small errors (e.g., incorrect verse numbering, misattribution) may propagate misleading teachings or erode trust.

The **IslamicEval 2025 Shared Task** (Mubarak et al., 2025) addresses this by benchmarking hallucination detection and correction for Quranic and Hadith content. HUMAIN participated in Subtask 1 (A: Span Detection, B: Span Validation, and C: Span Correction). We propose a **three-stage** pipeline integrating sequence annotation, retrieval-based verification, and correction via semantic re-ranking. Our system achieved competitive results across all subtasks, highlighting both strengths and limitations of current LLM approaches. We made our system codes public on GitHub [1]. We have included our codes, prompts, and implementation details in our GitHub repository.

The paper is structured as follows: section 2 outlines the shared task setup. Section 3 describes our system architecture. Section 4 details experimental settings. Section 5 reports results, and section 6 concludes the paper with insights and future directions.

2 Background

The IslamicEval 2025 Shared Task (Mubarak et al., 2025) was designed to to evaluate system performance on hallucination detection and correction of Quranic and Hadith content produced by LLMs. The focus is on ensuring factual accuracy in religious texts, where even minor deviations are unacceptable.

2.1 Task Setup

Our team participated exclusively in Subtask 1, covering all three subtasks:

- **1A – Span Detection:** Identify spans in LLM outputs that correspond to Quranic verses or Hadith. This requires handling varied quotation styles, partial matches, and noise from generative models.
- **1B – Validation:** Determine whether each detected span is authentic and correctly quoted by comparing against reference corpora (Quran and six Hadith Books).
- **1C - Correction:** For spans deemed incorrect, provide the corrected version from the gold-standard texts, or indicate that the span is completely wrong.

[1] https://github.com/0xArwa/humain-islamiceval-2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 509–514
November 8-9, 2025 ©2025 Association for Computational Linguistics

All datasets are in Arabic and sourced from authentic Quran and Hadith corpora curated by the organizers. Each subtask contains 50 and 104 distinct samples in the dev and test sets, respectively. Predictions were submitted through CodaBench for official scoring on the test set.

3 System Overview

3.1 Subtask 1A – Span Detection

For span detection, we employ an LLM-based pipeline to identify and extract Quranic verses and Hadith passages. More details on the LLMs used in our experiments are shown in section 4. The process begins with preprocessing the input text to resolve formatting inconsistencies—such as irregular spacing, punctuation issues, or line breaks—ensuring that the text is normalized before being passed to the model.

The cleaned text is then provided to an LLM with a specialized system prompt and few-shot examples. These instruct the model to detect religious spans and annotate them using a bracket-based notation of the form `[span_text|tag_type]`, where `span_text` represents the identified religious content and `tag_type` specifies whether it is a Quranic verse (ق) or a Hadith (ح). For example:

Input:

وجاء في الحديث الشريف: إِنَّمَا الأَعْمَالُ بِالنِّيَّاتِ

Output:

وجاء في الحديث الشريف: [إِنَّمَا الأَعْمَالُ بِالنِّيَّاتِ | ح]

Particularly, span detection is modeled as a sequence-to-sequence translation task using the Translation between Augmented Natural Languages (TANL) framework (Paolini et al., 2021). The model regenerates the passage with special markers denoting the start, end, and type of each span. Because generative models may introduce slight variations in spacing or punctuation (or removing/adding words), the TANL framework first cleans the annotated output by removing special tokens and discarding invalid formats. After this normalization, TANL employs the Needleman–Wunsch Dynamic Programming (DP) algorithm (Needleman and Wunsch, 1970) to align the cleaned output with the original input at the token level. This alignment enables each detected span to be mapped back to its precise character positions in the source text, ensuring consistency despite formatting drift introduced during generation.

As an alternative to TANL's alignment process, we also experimented with a guided decoding setup. In this variant, the LLM directly generates structured JSON output following a predefined schema, where each span object includes its type, textual content, and character indices. We utilize the vLLM library (Kwon et al., 2023) to enable guided decoding, which we apply only when we have direct access to the model and can deploy it on vLLM. This approach removes the need for token-level alignment altogether, since positional information is produced natively during generation.

3.2 Subtask 1B – Validation of Content Accuracy

For span validation, we developed a sophisticated verification system that handles both Quranic verses and Hadith texts through specialized processing pipelines optimized for each content type.

Hierarchical Indexing: The system employs dual indexing of reference corpora with normalized full-text indices for exact lookups and word-based inverted indices for candidate retrieval.

Verification Strategies: The core verification process implements multiple complementary matching approaches:

Multi-text Detection: The system first determines whether spans contain single or multiple verses using smart pattern detection that analyzes separators including asterisks (*), parenthetical verse numbers (e.g., (٤١)), sequences of 3+ consecutive non-Arabic characters, and contextual comma usage. This detection guides the subsequent verification approach.

Exact Matching: First-stage verification performs direct hash-based lookup in the normalized index for perfect matches after diacritic removal and character standardization.

Strict Substring Matching: For cases requiring exact textual containment, the system verifies that the normalized input appears as a complete substring within reference texts. This approach proved particularly effective for Hadith validation where authentic partial quotations are common.

Fuzzy Matching: When exact methods fail, the system applies sequence matching algorithms with experimentally-determined longest common subsequence (LCS) (Hirschberg, 1975) similarity thresholds. The process includes candidate pre-filtering using word overlap to reduce computational complexity, followed by detailed similarity scoring using LCS ratios.

Multi-word Substring Logic: For spans containing multiple words, specialized logic determines whether the entire sequence appears as a coherent substring in longer reference texts, with enhanced similarity scoring for valid substring matches.

Content-Specific Optimization: Based on empirical evaluation, we configured different verification approaches for each content type. Quranic spans use fuzzy matching with LCS similarity thresholds above 0.85 to maintain strict accuracy requirements for sacred text. Hadith spans employ strict substring matching, which better accommodates the legitimate partial quotations and paraphrasing patterns found in authentic Hadith transmission.

For multi-text spans, individual components are verified separately and aggregated using configurable consensus strategies.

3.3 Subtask 1C – Error Correction

Span correction for potentially corrupted or incomplete Quranic and Hadith texts is addressed through a multi-stage pipeline. The process begins with index-based pre-filtering, which combines a word-level inverted index with a character 3-gram index to reduce the search space. This design captures both exact word matches and partial substrings, ensuring that noisy or fragmented queries still retrieve relevant candidates.

Immediately after pre-filtering, the pipeline applies a composite fallback scoring mechanism to handle edge cases such as queries that span multiple consecutive verses presented as continuous strings without separators, or minor lexical variations that prevent standard matches. This mechanism integrates word n-gram overlap, phrase continuity, and substring containment metrics, adjusting candidate scores to ensure that these cases are retained and prioritized in subsequent processing.

Following this early edge-case handling, the candidate spans proceed to three successive matching stages. The first stage performs exact substring matching on normalized text, returning immediate matches when the query sequence appears exactly after diacritic and punctuation removal. The second stage applies LCS algorithm with source-specific similarity thresholds (Quran ≥ 0.85, Hadith ≥ 0.75). The third stage employs a multilingual semantic reranker (bge-reranker-v2-m3) (Chen et al., 2023) that applies sigmoid activation to produce normalized semantic similarity scores between 0 and 1 for the top candidates from earlier stages. The reranker evaluates semantic similarity beyond lexical overlap, combining its scores with original LCS similarities using a weighted scheme ($\alpha = 0.7$). This hybrid approach promotes semantically correct matches that may have lower lexical overlap, addressing cases where authentic content differs significantly in wording from the query.

4 Experimental Setup

4.1 Data Preprocessing

All input texts were normalized including diacritic elimination, character variant normalization (e.g. أ، إ، آ → ا), punctuation elimination, and whitespace standardization to ensure consistent matching across various text formats.

4.2 Model Configurations

For **Subtask 1A**, we experimented with various LLMs: GPT-4o (via OpenAI API) and four Arabic-centric LLMs, ALLAM (Bari et al., 2024), Fanar (Team et al., 2025), Command-R7B (Alnumay et al., 2025), and Jais-13B (Sengupta et al., 2023), all without task-specific fine-tuning (*temp*=0.1, *top_p*=0.98). For **Subtask 1B**, the selected similarity thresholds are ≥ 0.9 for Quran and strict substring matching for Hadith. For **Subtask 1C**, we combine the reranker (top-20) with final similarity thresholds set to ≥ 0.85 for Quran and ≥ 0.75 for Hadith, with spans below marked as خطأ (uncorrectable).

5 Results

This section shows the results of our system on the three subtasks of IslamicEval 2025.

5.1 Subtask 1A: Span Detection

Model	**Dev**			**Test**
	P	**R**	**F1**	**F1**
GPT-4o	87.4	75.7	81.1	**87.2**
ALLAM-34B	79.5	75.0	77.2	78.1
Command-R7B-Arabic	62.6	39.1	48.1	-
Fanar(API)	32.0	23.3	27.0	-
Jais-13b-chat	16.8	10.5	12.9	-

Table 1: Subtask 1A: Span Detection Performance. (P: Precision, R: Recall).

Table 1 shows the character-level macro-averaged F-1 scores for the five LLMs on the dev set. From these, we selected only the top-performing two models for submission on CodaBench (i.e., for the test set).

Quran Performance

Configuration	Acc.	P	R	F1
Strict	88	95	85	90
Fuzzy(0.9)	**91**	93	**93**	**93**
Fuzzy(0.65)	88	86	**97**	91

Hadith Performance

Configuration	Acc.	P	R	F1
Strict	**85**	97	**76**	**85**
Fuzzy(0.8)	75	**100**	54	70
Fuzzy(0.25)	75	**100**	54	70

Table 2: Substring matching performance comparison for Quran and Hadith text verification. Parentheses indicate LCS similarity thresholds. Fuzzy matching works better for Quran due to textual variations (different Uthmani formats, with/without tashkeel diacritics), while strict matching is optimal for Hadith due to text standardization. The distinct optimal strategies reflect the nature of each corpus: Quran exists in multiple valid variants requiring flexible matching, whereas Hadith collections maintain consistent formatting.

On the test set, GPT-4o achieved 87.20% F-1, while ALLAM-34B reached 78.10%, demonstrating competitive performance under more constrained settings. Both GPT-4o and Fanar (API) employed the prompting with special markers as described earlier in subsection 3.1 as we did not have access to them. For other 3 models, we utilized the guided decoding approach to ensure structured JSON output generation. Among these, Command-R7B-Arabic was the second-best performing Arabic-centric model, though it lagged significantly behind ALLAM in overall accuracy. Jais and Fanar showed considerably lower performance, indicating that current smaller Arabic-centric models are not yet competitive for this task.

Importantly, all results were obtained without any fine-tuning of model weights, showing that our approach can generalize to different LLMs without expensive adaptation.

5.2 Subtask 1B: Validation of Content Accuracy

As described in subsection 3.2, our system supports both fuzzy substring matching (with configurable LCS similarity thresholds) and strict substring matching. Table 2 presents development set performance across different threshold configurations by content type, which guided our optimal configuration selection for test evaluation.

Table 3 shows performance of selected configurations, where "–" indicates strict substring matching (no threshold required). The repeated Hadith values demonstrate substring matching robustness across threshold combinations. Our optimal configuration achieved 86.14% test accuracy using fuzzy substring matching with a 0.90 LCS similarity threshold for Quranic content and strict substring matching for Hadith texts.

Parameters		Performance	
Quran	Hadith	Dev (%)	Test (%)
0.80	0.65	84.21	84.21
0.90	0.80	81.00	85.96
0.90	–	84.60	**86.14**

Table 3: Subtask 1B: Span Validation Overall Performance Comparison.

This hybrid approach addresses different validation requirements: Quranic verses need high LCS similarity thresholds for fuzzy matching to handle script variations between Uthmani and formal scripts while maintaining accuracy against the Uthmani reference corpus, whereas Hadith texts benefit from exact substring matching for partial quotations and paraphrasing.

5.3 Subtask 1C: Error Correction

The best configuration, which combines exact matching, LCS, and semantic reranking, achieved 68.18% test accuracy, substantially improving over simpler baseline as shown in Table 4. Overall, the system shows strong performance in span detection and validation, while error correction remains the most challenging aspect, suggesting the need for more semantically grounded approaches.

Method	Thresholds		Accuracy	
	Quran	Hadith	Dev	Test
LCS	0.70	0.65	54.60	57.48
EM+LCS	0.85	0.70	60.13	65.91
+Reranker	0.85	0.75	72.74	**68.18**

Table 4: Subtask 1C: Error Correction Performance with Different Methods (LCS: Longest Common Subsequence, EM: Exact Match).

6 Conclusion

This paper presents a comprehensive three-stage pipeline for detecting and correcting hallucinations in Quranic and Hadith content generated by LLMs, addressing a critical challenge where factual accuracy carries profound religious and cultural significance. Through our participation in the IslamicEval 2025 shared task, we demonstrate that specialized approaches can effectively handle the unique requirements of Islamic textual verification.

The results highlight several key insights. GPT-4o outperformed other models overall in span detection, while ALLAM showed the strongest performance among Arabic-centric models, indicating the growing maturity of regional LLMs. Importantly, our system achieves strong results without any fine-tuning, showing that the approach can be applied to different models without modifying their weights —an advantage in terms of cost and scalability. Different similarity thresholds are needed for Quran versus Hadith validation, and semantic reranking provides modest but consistent improvements over exact matching and LCS in correction tasks.

The relatively modest correction accuracy underscores the complexity of this task and the need for continued research. Our analysis reveals that the most challenging correction cases involve contextual misattributions where the hallucinated span shares thematic content with the correct reference but differs substantially in wording. For instance, spans discussing the same Quranic narrative may require corrections that are semantically related but lexically distant. In addition, some fabricated content is so disconnected from authentic sources that determining whether any meaningful correction exists presents significant challenges for automated systems, particularly when minimal lexical overlap (e.g., sharing only one or two common words) may result in questionable matches, where providing no correction might be more appropriate (subsection 1.2). This limitation highlights the need for context-aware correction methods that consider not just the isolated span but also its surrounding discourse and thematic coherence. Future work should focus on proactive hallucination prevention, integration of Islamic scholarly expertise, and development of more sophisticated retrieval-augmented generation systems. This research represents a crucial step toward building trustworthy AI systems for religious texts, where accuracy is not merely a technical requirement but a matter of profound cultural and spiritual significance. Our publicly available code contributes to ongoing efforts in this critical domain.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback, the shared task organizers for their efforts, and HUMAIN for providing access to various models and infrastructure.

References

Yazeed Alnumay, Alexandre Barbet, Anna Bialas, William Darling, Shaan Desai, Joan Devassy, Kyle Duffy, Stephanie Howe, Olivia Lasche, Justin Lee, and 1 others. 2025. Command R7B Arabic: A small, enterprise focused, multilingual, and culturally aware Arabic llm. *arXiv preprint arXiv:2503.14603*.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan AlRashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024. ALLaM: Large language models for Arabic and English. *Preprint*, arXiv:2407.15390.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

Daniel S. Hirschberg. 1975. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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 the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *Preprint*, arXiv:2101.05779.

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 arXiv:2308.16149*.

Fanar Team, 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, and 23 others. 2025. Fanar: An Arabic-centric multimodal generative ai platform. *Preprint*, arXiv:2501.13944.

A Appendix

1.1 Guided Decoding: Schema Definition

As an alternative to TANL's post-processing alignment, we experimented with guided decoding to constrain the model to produce valid JSON conforming to our span detection schema.

The model generates spans with explicit character positions, formally described by the following JSON Schema:

```
{
  "type": "object",
  "properties": {
    "spans": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "type": {
            "enum": ["q", "h"]
          },
          "text": {
            "type": "string"
          },
          "start": {
            "type": "integer"
          },
          "end": {
            "type": "integer"
          }
        },
        "required": [
          "type", "text",
          "start", "end"
        ]
      }
    }
  }
}
```

where `type` denotes Quran (q) or Hadith (h), and `start`/`end` specify character indices.

1.1.1 Example

For input text إنا أنزلناه في ليلة القدر والله أعلم,
the model generates:

```
{
  "spans": [
    {
      "type": "q",
      "text": "إنا أنزلناه في ليلة القدر",
      "start": 0,
      "end": 25
    }
  ]
}
```

1.1.2 Limitation

This approach relies entirely on the model's ability to generate accurate character positions during inference. The guided decoding constraints (via vLLM's `guided_json` parameter) ensure structural validity but cannot prevent hallucination of non-existent text spans.

1.2 Challenging Correction

This section illustrates cases where automated correction systems face significant challenges in determining appropriate mappings between fabricated content and authentic sources.

Example: Hallucinated verse with minimal lexical overlap

LLM-Generated (Hallucinated):

قوله تعالى: وْإذا مرْضت فلا ركب علي ولا
حمية وأصبح بياضْ

Annotation Label:

`WrongAyah (correctable)`

Correction:

وإذا مرضت فهو يشفين

Analysis: The fabricated content shares minimal lexical overlap with the proposed correction (primarily the words إذا مرضت). The hallucinated verse contains nonsensical elements and bears no meaningful semantic relationship to the authentic verse. This example demonstrates the challenge of determining when shared vocabulary constitutes sufficient grounds for correction versus when providing no correction (خطأ) might be more appropriate.