

A Multi-Layer Quranic Knowledge Graph and Hybrid KG-RAG Chatbot for Cross-Layer Semantic Question Answering

Kemas Rahmat Saleh Wiharja^{1*}, M. Arif Bijaksana¹, Kemas Muslim Lhaksana¹, Hafidzh PA¹, Aqil Anhein¹

¹*School of Computing, Telkom University, Indonesia*

*bagindokemas@telkomuniversity.ac.id

Abstract

Understanding the Al-Quran is the eternal goal of every Muslim, wherever they are. Before the emergence of the Knowledge Graph, readers of the Al-Quran needed to move from the Al-Quran to Tafsir, Hadith, scholar opinions or other resources in order to grasp the full meaning of a verse or a chapter in the Al-Quran. To help the readers of the Al-Quran in pondering the meaning of the Al-Quran and linking the verses to the meaning of the word or the interpretation of the verse in a tafsir book, we propose the first multi-layer Quranic Knowledge Graph that uses the property graph format. We also add a chatbot on top of our Quranic Knowledge Graph. And to save cost in using a Large Language Model, we deploy a vector database to memorize the previously answered user queries and their corresponding Cypher translations. We evaluate our approach using three point of views: large language model, knowledge graph, and chatbot. The result of evaluation shows that our Quranic Knowledge Graph achieves 100% correctness and 90% accuracy, despite only covering 9 parts of the Al-Quran. For the interested readers, please access <https://zentilax.github.io/quranic-chatbot-UI/> for exploring the quranic knowledge graph.

Keywords: quranic knowledge graph, multi-layer, chatbot, large language model.

I. INTRODUCTION

Al-Quran is the most widely text that read by people throughout the world and its authenticity has been maintained since the time of Rasulullah SAW until now. Statistically, Al-Quran consists of 114 chapters, 6,236 verses, and approximately 77,430 words. Even though the number of words in Al-Quran seems far little compared to a very famous Britannica encyclopedia that has approximately 44 million words, the number of meanings or semantic networks that can be constructed by the combination of words in Al-Quran is unlimited. The most popular option in documenting this semantic network is by creating a printed tafsir. Unfortunately, this option has issue with the number of pages that can go beyond 1000 (for example, tafsir Ibn Katsir that published in its complete form, typically comprises around 6,500 pages). Obviously, this will make it difficult for readers to find the meaning of the word Afwu (as example) when combined with other words in the whole tafsir. Another option is to use tabular format with many columns to store the first word, the other word that is combined with the first word, and then the meaning in different contexts. Again, this option will result in very large tables, with an unlimited number of context columns, which can confuse readers. The emergence of new digital technologies such as structured databases, Elasticsearch-based systems, index-based retrieval frameworks, and other scalable relational database implementations has opened up opportunities to store the Quran and its interpretations more efficiently and has also made it easier to search for specific verses, chapters, or issues in the Quran and its interpretations. Many popular services such as KSU electronic mushaf project¹, Quran NU online², use this newer technology. In addition to reading the Quran and its translations using web-based platforms, people now really enjoy using mobile apps to read the Quran and its translations. Popular mobile apps such as Muslim Pro³, Tarteel⁴, and others use on-device structured storage for storing the Qur'an text and also the translations. All of these new digital and mobile technologies really fast in finding specific verses, chapters, but failed

in responding to multi-hop semantic query such as: find all verses containing words derived from root *r-h-m* (رحم), and show how different tafsir sources interpret the concept of “mercy” in those verses, or identify Makki verses about patience (sabr) that are linked to trade and economic activities during the time of the Prophet SAW. To answer these multi-hop semantic queries, one can use LLM such as Gemini, Open AI, or Ollama, but two problems appear: 1. Potential bias answers, 2. Lengthy narrative explanations that make it difficult for someone to analyze the direct meaning relationship between verses and other verses.

Another direction for handling the semantic network issue is using quran ontology. Several works already propose the ontology of Al-Quran, such as [1], [2], [3], and [4]. [1] develop an ontology of Al-quran with focusing on the common concepts such as Juz/parts, Chapter, and theme. Their ontology can only be used for finding verses based on number; finding verses, surah, or based on a specific theme. [2] develop an ontology of Qur’an that consists of chapters, verses, each word of the Qur’an, and the root and lemma of each word to facilitate key word search only. [3] merged hadith and Al-quran in one ontology and then this merged ontology can be used for helping users searching a particular term. Lastly, [4] focused their ontology to be populated by nature related concepts only. They developed a simple GUI to help answering simple queries such as : *all types animals mentioned in the holy Quran, Find relation between Rain and Fruits, and Find all instances which Allah favored or punished.*

All ontological approaches mentioned above do not yet accommodate multi-hop semantic queries, therefore we propose a very natural way to represent the semantic networks of words, verses, and chapters in Al-Quran, which is knowledge graph (KG). There are two alternatives for representing a knowledge graph: the Resource Description Framework (RDF) and the Property Graph model. While RDF provides strong standardization for entity-centric knowledge bases, it exhibits notable limitations in representing the multi-layered contexts required for Qur’anic semantics. RDF’s native reasoning capabilities face significant constraints when executing deep, multi-hop contextual traversals across varying interpretive layers. Crucially, representing rich edge attributes—such as the specific contextual weight of a word translation within a particular tafsir, or the token position within a verse—requires RDF reification. Reification mechanisms transform a single conceptual edge into multiple nodes and edges, leading to graph bloat, severe performance degradation, and scalability issues during complex query execution [14]. In contrast, the Property Graph model natively allows rich attributes to be attached directly to both nodes and edges without requiring structural reification. For the Qur’anic domain, where every connection carries essential metadata like source authority and contextual bounds, the Property Graph demonstrates clear theoretical superiority. Structured comparisons in graph database benchmarking show that Property Graphs execute context-dependent local relationship traversals with significantly lower latency and memory overhead compared to highly reified RDF triple-stores [15]. Because our architecture requires tracking cross-layer dependencies without exponentially increasing schema complexity, the property graph is the most appropriate choice.

As a revealed book and a guidance for all humankind, the Al-Quran is commonly accessed through two complementary forms: the original Arabic text, whose wording is fixed and preserved, and translations in other languages, which are produced by qualified and trustworthy scholars to help non-Arabic-speaking readers understand its meaning. In practice, when readers who do not speak Arabic want to study a topic in the Quran, they usually read the original verse in Arabic and then consult its translation in their own language. This common reading practice suggests that Quranic knowledge should not be modeled in only one level, but in at least two connected levels of representation. The first level should remain stable and preserve the original Arabic verses, while the second level should be more flexible so that translation-related nodes and edges can be added, refined, or updated when needed. By using these two levels of knowledge graph, we hypothesize that the system can better support multi-hop queries that require cross-layer reasoning, such as linking an Arabic lexical concept to the way it is rendered in English, or grouping semantically related verses based on translation wording. In this paper, we define a multi-layer Quranic knowledge graph as a knowledge graph architecture composed of two explicitly linked semantic levels: Layer 1, the Quran verse graph, which models chapters, verses, words, roots, entities, and their relations in the source text; and Layer 2, the English translation graph, which represents how the meanings of those verses are expressed in English. To the best of our knowledge, existing studies have not explicitly modeled Al-Quran and its translation in the format of multi-layer knowledge graph. Under this definition, a single-layer Quranic KG may still return a partially correct answer, but the answer will often be incomplete because it cannot fully explain how a concept identified in the Arabic verse network is realized, compared, or grouped in the translation layer. This design choice is also consistent with broader KG research showing that complex question answering often benefits from combining structured graph knowledge with linked textual or multilingual information, although richer representation helps only when the added layers are well aligned with the reasoning needs of the query. Thus, the contribution of our approach is as follows: 1). proposes a

¹<https://quran.ksu.edu.sa/>, ²<https://quran.nu.or.id/>, ³<https://www.muslimpro.com/>, ⁴<https://tarteel.ai/>

Quranic knowledge graph in property graph format, 2). propose a multi-layer knowledge graph that contains Al-Quran and its translation, 3). develop a fast-response chatbot on top of the multi-layer quranic knowledge graph.

We will evaluate our multi-layers quranic knowledge graph by observing several aspects of our approach as follows: (1). knowledge graph construction. There are two metrics for this aspect, which is coverage (how complete our knowledge graph compared to the original sources), and correctness (how correct each information/knowledge in our knowledge graph compared to the gold standard/original sources). (2). The performance of the chatbot. There are two other metrics for this aspect, as follows: precision, recall, the accuracy of the answer, and task-based performance evaluation.

II. LITERATURE REVIEW

A. *Digital Qur'anic Resources, Linguistic Corpora, and the Limits of Conventional Retrieval*

Computational research on the Qur'an has benefited from the growing availability of standardized digital corpora, annotated resources, and task-oriented datasets. Earlier foundational resources such as the Quranic Arabic Corpus enabled large-scale linguistic study of the Qur'an, while recent work has continued to enrich this infrastructure with more detailed annotation. For example, the MASAQ line of work provides fine-grained morphological and syntactic annotation for Qur'anic Arabic, addressing the long-standing shortage of richly structured resources for downstream NLP and semantic analysis. In parallel, a systematic review by Bashir et al. shows that Qur'anic NLP has matured into a broad field covering morphology, syntax, recitation, semantic analysis, information retrieval, and question answering, while still facing major challenges due to the complexity of Classical Arabic and the scarcity of semantically rich annotated resources [4], [5].

This infrastructure has supported significant progress in semantic modeling of Qur'anic text. For instance, Alsaleh et al. used AraBERT to classify semantic relatedness between verse pairs in the QurSim dataset, demonstrating that contextual Arabic language models can capture non-trivial semantic relationships beyond exact lexical overlap [6]. Such work is important because it moves Qur'anic retrieval closer to meaning-sensitive processing. However, these approaches still primarily operate at the level of textual similarity or verse-pair classification. They do not explicitly represent the semantic network linking words, verses, translations, tafsir passages, and cross-source interpretive relations in a graph structure that can be directly traversed by a reasoning engine.

B. *Qur'anic Question Answering: From Reading-Comprehension Models to Flexible Semantic QA*

Question answering over Qur'anic text has evolved substantially in recent years, especially through benchmark-driven extractive QA and retrieval-based systems. A representative example is the Qur'an QA 2022 shared-task work by Elkomy and Sarhan, who used an ensemble of Arabic BERT-based models and reported a partial reciprocal rank of 56.6% on the official test set [7]. This line of work established Qur'anic QA as an evaluable NLP task and showed that modern neural architectures can retrieve or extract answers from Qur'anic passages with reasonable effectiveness.

More recent research has started to move beyond flat passage retrieval toward semantic question answering. Wardani et al., for example, proposed a flexible semantic Qur'an QA framework based on graph-based summarization and KNN, arguing that semantic search is more appropriate than purely lexical search for the Qur'an because of its conceptual density and linguistic specificity [8]. This is an important step, since it acknowledges that many user questions require more than keyword matching. Nevertheless, such systems still stop short of a fully graph-native, multi-layer architecture in which the Arabic source text, multilingual translations, and tafsir are represented as linked but distinct semantic layers. As a result, they only partially address the type of cross-layer multi-hop reasoning targeted in this paper.

C. *Ontology and Knowledge-Graph Approaches for Qur'anic and Islamic Knowledge*

A parallel stream of research has focused on ontology and knowledge-graph representations of Qur'anic and Islamic knowledge. Earlier Qur'anic ontology work modeled structural units such as chapters, verses, words, roots, themes, and selected semantic domains, showing that Islamic textual knowledge can be formalized in machine-readable semantic structures. More recent work has expanded this direction beyond the Qur'an alone. For example, Bendjamaa and Taleb proposed OntoDin, an Islamic ontology that integrates Qur'an and Hadith and yields a large bilingual semantic resource with 51 classes and 168,122 individuals [3]. This line of work is significant because it demonstrates the value of modeling Islamic knowledge as an interconnected semantic space rather than as isolated textual collections.

Recent advances have also shown that tafsir itself can be modeled as a structured semantic resource. SemanticTafsir introduces an OWL ontology and RDF-based knowledge graph for Tafsir al-Tabari, thereby extending semantic modeling from scriptural text to exegetical discourse [9]. Likewise, OntologyRAG-Q introduces an annotated tafsir

ontology, a collection of 15 structured tafsir books, and a benchmark of approximately 4,200 question-answer pairs for retrieval-augmented question answering in Qur’anic tafsir [10]. These works are highly relevant because they confirm that structured semantic resources are becoming central in Qur’anic and tafsir-oriented QA. However, they still differ from the present work in one crucial respect: they do not explicitly model the Qur’an and its translations as a multi-layer property graph in which the stable Arabic source-text layer is separated from, yet connected to, the more flexible translation and interpretive layer for cross-layer traversal.

D. LLMs, Retrieval-Augmented Generation, and Reliability in Religious QA

Large language models have recently opened a new direction for Qur’anic and religious question answering, especially when combined with retrieval mechanisms. Alnefaie et al. showed that retrieval-augmented generation can improve GPT-4 for answering Qur’an questions, indicating that grounding the model in retrieved evidence is beneficial in this domain [11]. More broadly, recent work on religious-domain QA has also emphasized that retrieval can improve answer faithfulness and reduce hallucination compared with purely generative LLM use. For example, MufassirQAS specifically targets improved reliability in religious question answering through RAG-based grounding [12]. These findings are particularly relevant for Islamic and Qur’anic applications, where interpretive precision and provenance are critical.

At the same time, the recent literature consistently warns against relying on LLMs alone for religious interpretation. Generative models may produce plausible but weakly grounded answers, conflate distinct exegetical traditions, or provide lengthy narrative responses that obscure the actual evidence path. This is why the current trend is moving toward hybrid architectures in which the LLM is constrained by retrieval, structured evidence, or both. However, most existing RAG-style Qur’anic systems retrieve from passages, chunks, or ontology-backed text collections rather than from a graph explicitly designed for multi-hop semantic traversal across Arabic wording, verse structure, translation realization, and tafsir support. This leaves open an important design space for graph-grounded and cross-layer conversational systems.

E. Multi-Hop and Cross-Layer Reasoning as the Remaining Gap

The literature therefore suggests that Qur’anic computational research is advancing along three main directions: semantic text modeling, benchmark-oriented QA, and ontology/RAG-based knowledge access. Yet an important gap remains. Most existing systems either (1) retrieve passages from text corpora, (2) answer extractive QA benchmarks, or (3) represent Qur’anic or tafsir knowledge in ontological form without deploying a graph-native conversational interface that supports cross-layer reasoning. Even recent Islamic-domain multi-hop QA resources, such as IslamicPCQA, highlight the growing need for multi-step reasoning over religious knowledge, but they are not built as Qur’an-translation-tafsir graph systems [13].

Against this background, the present work is positioned as a **multi-layer Quranic Knowledge Graph in property-graph form**, coupled with a chatbot that combines graph retrieval and retrieval support from FAISS. Its contribution is not merely to build another Qur’anic ontology or another Qur’anic chatbot, but to explicitly separate the fixed Arabic structural layer from the more flexible translation and interpretation layer, and to use this design to support cross-layer semantic question answering. This positioning more clearly distinguishes the work from prior semantic-search systems, ontology-only models, and text-only RAG approaches.

III. RESEARCH METHOD

This study is designed as a **research-driven modeling contribution** that investigates whether a multi-layer property-graph representation can more effectively support cross-layer semantic question answering over the Qur’an and its interpretive resources than conventional data representations. Rather than merely implementing a chatbot on top of a graph, this work aims to evaluate the appropriateness of the proposed graph architecture for representing structural, lexical, translational, and interpretive relations in a way that can answer both direct factual queries and multi-hop semantic queries.

To guide the study, we formulate the following research questions:

1. Can a multi-layer property graph represent the structural and semantic relations among surahs, ayat, words, translations, and tafsir in a way that supports cross-layer reasoning?
2. Does a property-graph architecture offer practical advantages over relational databases, RDF-based knowledge graphs, and alternative retrieval systems for multi-hop Qur’anic queries?

3. Can a hybrid KG-RAG chatbot built on top of the proposed graph provide correct and grounded answers while preserving traceability to the original verse and tafsir resources?

To answer these questions, the methodology is divided into two major stages, namely the Quranic Knowledge Graph construction stage and the QA chatbot deployment stage, as illustrated in Figure 1. However, unlike a purely implementation-oriented system paper, the present work also includes a comparative methodological evaluation involving alternative data representations, query benchmarking, retrieval assessment, and scalability analysis.

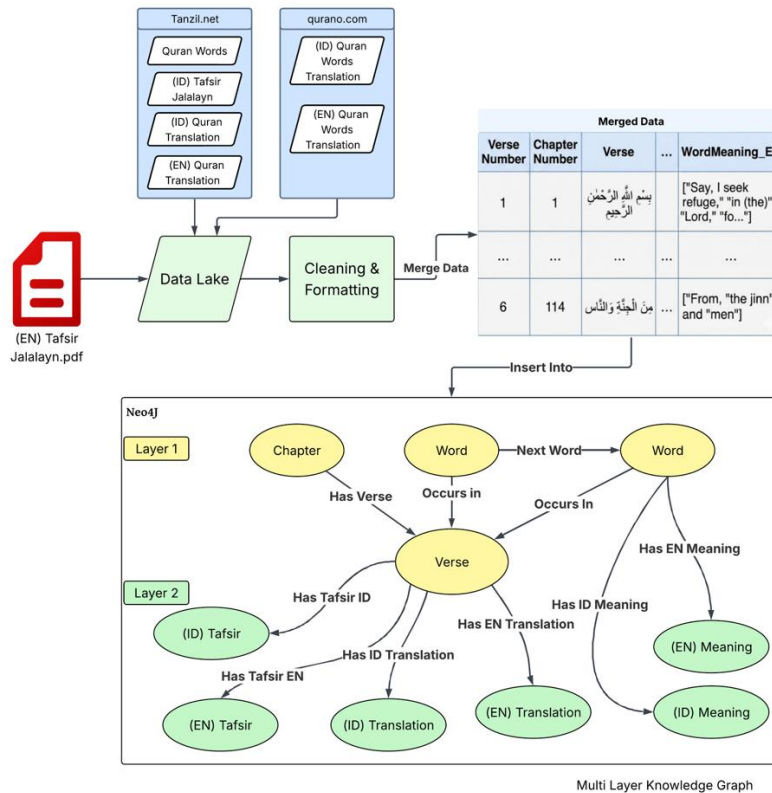


Figure 1. The Quranic Knowledge graph construction

A. Quranic KG Construction Stage

The first stage focuses on acquiring, cleaning, aligning, modeling, and inserting the data into a multi-layer Quranic Knowledge Graph. The goal of this stage is not only to build the graph, but also to ensure that the graph is semantically meaningful, structurally correct, and suitable for answering multi-hop queries.

1) *Data acquisition and study boundary*: The data for this study were acquired from four main resources:

1. the Arabic Qur’anic text from <https://tanzil.net/download/>,
2. Indonesian and English verse-level translations from <https://tanzil.net/trans/>,
3. Indonesian and English word-by-word glosses from <https://qurano.com/>, and
4. verse-level Tafsir Jalalayn in Indonesian and English.

All raw resources were first collected and stored in a data lake before being normalized into a common tabular structure. In the current study, the implemented graph covers **Juz 22 until Juz 30**, corresponding to **9 Juz, 82 chapters, and 2,703 ayat**. This partial scope is due to the budget limitation of the project, especially the storage and scalability constraints of the available Neo4j deployment tier. Nevertheless, this does not constitute a conceptual limitation of the proposed architecture. The selected 9-Juz subset is sufficiently large and diverse to validate the graph-construction pattern, cross-layer linking mechanism, semantic query behavior, and chatbot integration. The same graph-construction procedure can be repeated for the remaining Juz whenever a larger infrastructure budget becomes available.

2) *Data cleaning, preprocessing, and formatting*: The raw data obtained from Tanzil were provided in a uniform format where each line consisted of a single record structured as: `chapter_number || verse_number || content`.

These single-column records were parsed into structured columns representing chapter number, verse number, and the associated content, such as Arabic verse text, translation, or tafsir. The collected resources were then standardized by unifying column names, removing irrelevant columns, and ensuring consistency across all datasets. For example, columns such as *Makna_perkata* were renamed to *Makna_perkata_ID* or *Makna_perkata_EN*, and tafsir fields were similarly standardized. As a specific cleaning procedure, the Basmalah phrase was programmatically removed from the beginning of the first verse of each surah, except chapter Al-Fatihah, in order to avoid redundancy during graph construction. After preprocessing, the data were merged into a single integrated table containing the following fields: chapter number, verse number, Arabic verse text, Indonesian translation, English translation, Indonesian word-by-word meaning, English word-by-word meaning, Indonesian tafsir, and English tafsir.

3) *Token alignment procedure and word segmentation*: Because the proposed graph aims to model not only verses but also words and their contextual meanings, special attention was given to token-level alignment. Each Arabic verse was segmented into an ordered sequence of word tokens, and each token was assigned a positional index within the verse. Thus, each word occurrence was uniquely identified using the tuple: (chapter_number, verse_number, token_position). This positional representation is important because the same surface form may appear multiple times in different verses or even within the same verse, but with different contextual meanings. Therefore, the graph models word occurrences rather than collapsing all identical word forms into a single universal lexical node.

The word-by-word glosses from Qurano were aligned to the Arabic source verse using the same chapter and verse identifiers, followed by token-order matching. If a mismatch in token count between the Arabic source and the word-by-word gloss source was detected, the verse was flagged for manual inspection and was excluded from automatic token-level alignment until the discrepancy could be resolved. This conservative alignment policy was adopted to prioritize correctness and reduce the propagation of alignment errors into the graph.

4) *Handling of Classical Arabic morphology and polysemy*: Qur'anic Arabic is morphologically rich and semantically context-sensitive. Therefore, the model does not assume that one surface form always corresponds to one fixed meaning. Instead, the graph separates the structural occurrence of a token from its contextual meaning. The structural layer stores the word occurrence as it appears in the Arabic verse, while the semantic layer stores its Indonesian and English glosses and links it to verse-level tafsir and translation nodes.

This design also provides a practical strategy for handling polysemous words. Rather than assigning one global meaning to a token type across the entire Qur'an, the proposed graph links each word occurrence to its context-specific meaning nodes. In this way, the same Arabic form may participate in different semantic neighborhoods depending on its verse context, and this variation is preserved instead of being forced into a single dictionary-level interpretation. This occurrence-level disambiguation is one of the main reasons for choosing a graph representation.

5) *Conceptual modeling and schema design*: After the data had been cleaned and aligned, the next process was to define the schema of the Quranic Knowledge Graph. The proposed graph adopts a multi-layer property graph architecture. This design is based on the observation that the Qur'an is commonly accessed through two complementary forms: the fixed Arabic source text and a more flexible layer containing translations and tafsir. The first layer therefore preserves the structural integrity of the Qur'an, while the second layer provides semantic enrichment.

Layer 1: Structural Core

The first layer represents the canonical structure of the Qur'an. It contains the following node types:

- **Chapter**, representing a chapter of the Qur'an
- **Ayat**, representing a verse of the Qur'an
- **WordOccurrence**, representing a word token at a particular position in a verse

The relationships in this layer are:

- (:Ayat) - [:PART_OF] -> (:Chapter)
- (:WordOccurrence) - [:OCCURS_IN] -> (:Ayat)
- (:Ayat) - [:NEXT_AYAT] -> (:Ayat)
- (:WordOccurrence) - [:NEXT_WORD] -> (:WordOccurrence)

This structural layer preserves the physical sequence of chapters, verses, and words, thereby supporting direct factual queries and sequential traversal.

Layer 2: Semantic Enrichment

The second layer contains the semantic and interpretive resources linked to the first layer. It contains the following node types:

- *MaknaID / MaknaEN*, representing word-by-word meanings in Indonesian and English
- *TafsirID / TafsirEN*, representing verse-level tafsir in Indonesian and English
- *TerjemahanID / TerjemahanEN*, representing verse-level translations in Indonesian and English

The main relationships in this layer are:

- (:WordOccurrence) - [:HAS_MAKNA_ID] -> (:MaknaID)
- (:WordOccurrence) - [:HAS_MAKNA_EN] -> (:MaknaEN)
- (:Ayat) - [:HAS_TAFSIR_ID] -> (:TafsirID)
- (:Ayat) - [:HAS_TAFSIR_EN] -> (:TafsirEN)
- (:Ayat) - [:HAS_TERJEMAHAN_ID] -> (:TerjemahanID)
- (:Ayat) - [:HAS_TERJEMAHAN_EN] -> (:TerjemahanEN)

The purpose of this multi-layer design is to separate the stable part of the graph, namely the Qur’anic text itself, from the editable and expandable semantic layer, which can be enriched with translations, tafsir, and potentially other interpretive resources in the future.

6) *Ontological abstraction and inter-verse semantic relations*: To move beyond purely structural linking, the revised schema also allows semantic abstraction through additional conceptual node types, such as **Theme**, **Concept**, **NamedEntity**, and **RevelationContext**. These nodes are not required for basic graph construction, but they are part of the conceptual design of the graph and support future extensions for deeper semantic reasoning. The intended semantic relations intended to support thematic grouping, concept-based traversal, and inter-verse semantic relations. The following are the intended semantic relations:

- (:Ayat) - [:ABOUT_THEME] -> (:Theme)
- (:Ayat) - [:MENTIONS_ENTITY] -> (:NamedEntity)
- (:Ayat) - [:HAS_REVELATION_CONTEXT] -> (:RevelationContext)
- (:Ayat) - [:SEMANTICALLY_RELATED_TO] -> (:Ayat)

7) *Why property graph*

The decision to use a property graph rather than a relational database or RDF triple store is motivated by the query workload targeted in this study. The property graph model is more suitable when the domain requires:

1. rich attributes attached directly to nodes and edges,
2. efficient traversal across multiple relation types, and
3. context-dependent local relationships such as ordered word sequences, verse adjacency, and word-specific meanings.

In the context of the Qur’an, many intended queries are not simple lookup queries, but path-based queries that move from words to verses, from verses to tafsir, and from Arabic lexical items to translated meanings. Compared with normalized relational tables, the property graph reduces the complexity of expressing such traversal-oriented queries. Compared with RDF, it offers more direct support for local attributes and Cypher-based path exploration. To help the interested reader to compare between RDF and Property Graph in more detail, we present Table I.

TABLE I
STRUCTURED COMPARISON OF RDF VS. PROPERTY GRAPH FOR MULTI-LAYER QUR’ANIC MODELING

Feature	RDF (Triple Store)	Property Graph	Impact on Qur’anic KG
Edge Attributes	Requires Reification, creating dummy nodes to store metadata [14], [16]	Native support via key-value pairs on edges [16]	Property Graph easily stores specific contextual weight, token positions, and tafsir metadata on the relationships
Multi-Hop Traversal	Slower; requires complex, multi-way JOIN operations in SPARQL [15]	Faster; utilizes index-free adjacency for graph traversal [17]	Property Graph enables real-time cross-layer reasoning from verse to translation to tafsir without joining large tables.
Schema Complexity	High graph bloat and scalability issues when handling highly contextual semantic networks [14], [15].	Compact, intuitive and prevents structural explosion [15], [16].	Property Graph prevents the exponential structural explosion caused by modeling multi-layered Islamic texts.

8) *Comparative baselines*

To justify the choice of a property graph empirically, the study includes three comparative baselines:

1. **Relational baseline**

The same Qur'anic data are represented in normalized tables such as Surah, Ayat, WordOccurrence, Translation, and Tafsir. This baseline is expected to perform well for direct factual lookup but requires increasingly complex joins for multi-hop semantic queries.

2. **RDF-based baseline**

The same entities and relations are represented as RDF triples. This baseline supports formal semantic interoperability, but sequence-sensitive word occurrences and richly attributed local relations are less natural to represent.

3. **Alternative retrieval baseline**

Verse-level translations and tafsir are indexed in a vector-retrieval system. This baseline supports semantic similarity search, but it does not natively preserve explicit graph constraints such as verse order, structural hierarchy, or multi-hop evidence paths.

9) *Graph insertion*

After the schema was defined, the cleaned and aligned data were inserted into Neo4j. Each node and relationship was created with stable identifiers so that the resulting graph could be checked against the original source tables. The final graph, covering Juz 22 to Juz 30, contains **85,805 nodes** and **110,692 relationships**. This graph size already demonstrates that the architecture is non-trivial and that the graph supports a substantial portion of the Qur'an together with its bilingual semantic layer.

B. *QA Chatbot Deployment Stage*

The second stage of the methodology concerns the deployment of a chatbot and retrieval mechanism on top of the Quranic Knowledge Graph. The purpose of this stage is to evaluate whether the graph can serve as an effective foundation for direct factual question answering and for broader semantic queries.

1) *Intent recognition*

The first process in this stage is recognizing the intent of the user query. User questions are classified into two categories:

- **DIRECT queries**, which explicitly reference structured elements such as surah number, verse number, verse order, or chapter-level facts;
- **SEMANTIC queries**, which focus on themes, meanings, conceptual relations, or interpretive context.

DIRECT queries are sent to the Cypher generation module, whereas SEMANTIC queries are first routed to the vector retrieval layer before graph expansion.

2) *Cypher generator architecture and constraints*

The Cypher generator is implemented as an LLM-assisted query-construction module. Its purpose is to translate natural-language user queries into executable Cypher queries over the Quranic Knowledge Graph. To reduce hallucinated graph queries and to preserve safety, the generator is constrained by the following mechanisms:

1. a fixed schema prompt containing only the allowed node labels and edge types,
2. template-guided generation for common question patterns,
3. read-only query restrictions, and
4. post-generation validation that rejects unknown labels, invalid relations, or non-permitted query clauses.

In this way, the Cypher generator functions as a controlled semantic parser rather than as an unrestricted text generator.

3) *FAISS vector database*

For semantic retrieval, a FAISS vector database is used to store embeddings of verse-level tafsir and related historical query patterns. When a user submits a SEMANTIC query, the system retrieves the top-*k* most relevant verse candidates from FAISS before expanding the answer through graph retrieval.

The following are the parameters that we use for the final implementation:

- The embedding model used is OpenAI *text-embedding-3-small*, which provides a good balance between semantic representation quality and computational efficiency.

¹<https://quran.ksu.edu.sa/>, ²<https://quran.nu.or.id/>, ³<https://www.muslimpro.com/>, ⁴<https://tarteel.ai/>

- The embedding dimensionality is 1536, allowing the model to capture detailed semantic relationships within the data.
- The similarity metric is cosine similarity, as it effectively measures the semantic closeness between vectors regardless of their magnitude.
- The FAISS index type is IndexFlatL2, which performs exact nearest neighbor search to ensure high retrieval accuracy.
- The value of k used during retrieval is 5, chosen to provide sufficient relevant results while maintaining efficiency.

4) Graph retrieval module

After candidate verses are obtained either from the Cypher generator or from FAISS retrieval, the graph retrieval module extracts the relevant subgraph from Neo4j. This subgraph contains the ayat, associated word occurrences, translations, tafsir nodes, and any relevant neighboring nodes needed for answering the query.

To ensure reproducibility, we configured the graph retrieval stage using bounded parameters such as follows:

- no maximum traversal depth,
- no maximum subgraph size, and
- all edge types are allowed during traversal.

These constraints prevent over-expansion of irrelevant graph neighborhoods and make the retrieval behavior more stable.

5) Result humanizer

The subgraph returned by the graph retrieval module is then passed to a result humanizer. The role of this component is to reformat the graph-based evidence into a more human-readable response without adding, removing, or altering the semantic content provided by the graph. Thus, the final answer remains grounded in retrieved graph evidence rather than in the free-form knowledge of the LLM.

C. Benchmark Query Set and Evaluation Protocol

To evaluate the methodological advantages of the proposed graph architecture, the study uses three classes of benchmark queries.

1) Structural lookup queries

These queries test direct factual retrieval and include questions such as:

- “How many verses are in Surah Muhammad?”
- “What is the last verse of Surah At-Tin?”

These questions are expected to be answerable by all baseline representations.

2) Cross-layer semantic queries

These queries require movement between the Arabic structural layer and the semantic layer, for example:

- “Find verses containing a lexical item related to a particular root and show how it is rendered in English.”
- “List verses where an Arabic word has different contextual meanings across translations or glosses.”

3) Multi-hop interpretive queries

These queries require traversal across multiple relation types and levels of abstraction, for example:

- “Identify Makki verses about patience that are linked to trade and economic activities.”
- “Find verses where a mercy-related lexical item is interpreted in tafsir as divine forgiveness.”

These are the most discriminative queries for the proposed graph model and are the main justification for using a multi-layer property graph.

For each query class, the following aspects are evaluated:

- answer correctness,
- query formulation complexity,
- execution latency, and
- traceability of evidence paths.

D. Evaluation Metrics

The proposed methodology evaluates the system from three perspectives.

1) Knowledge graph construction metrics

To evaluate the graph-construction stage, we use:

- Coverage, which measures the proportion of source verses, chapters, Juz, and word-meaning pairs that are represented in the graph;
- Correctness, which measures whether the graph content faithfully preserves the information contained in the original sources.

2) Retrieval metrics

To evaluate the retrieval stage, we use:

- Precision@k
- Recall@k
- Hit@k or MRR
- Grounding rate, i.e., whether the final answer can be traced to the retrieved verse and tafsir nodes

3) QA and Cypher-generation metrics

To evaluate the chatbot and Cypher generation, we use accuracy as the metric. We measure the accuracy of the answer of a chatbot by asking a domain expert to check whether the answer that is generated by a chatbot is correct or not.

E. Scalability and Stress-Test Analysis

Although the current implementation covers only 9 Juz, the methodology also considers scalability. The purpose is to demonstrate that the graph design remains robust even though the full 30 Juz were not included in the current deployment. Scalability is examined by observing:

1. node and edge growth patterns per additional Juz,
2. query latency under repeated workloads,
3. projected memory and storage requirements for full-corpus expansion, and
4. the stability of the same graph-construction pattern when extended to additional data.

Therefore, the selected 9-Juz subset is sufficient to demonstrate the correctness of the construction pipeline, the adequacy of the multi-layer schema, and the usefulness of the hybrid KG-RAG retrieval strategy. Full-corpus deployment is left for future work when larger infrastructure resources become available.

IV. RESULT AND DISCUSSION

In this section, we present the evaluation of our approach from three points of view.

A. LLM Evaluation

We use OpenAI GPT-5-Nano as our LLM. The primary aspect we evaluate is its ability to translate a user's query or prompt into Cypher queries. Across all test prompts (see Table II for details), 100% of the generated queries were syntactically valid and executable. The correctness of the generated SPARQL queries was manually evaluated to determine whether they were logically appropriate for the given knowledge graph. The strong performance of GPT-5-Nano in translating user queries into Cypher can be attributed to its ability to identify nodes and edges from each query. To reduce the cost of using the LLM, we store previously answered user queries and their corresponding Cypher translations in a FAISS vector database. The LLM is therefore used only for entirely new questions.

B. Knowledge Graph Evaluation

There are two aspects of knowledge graph that needs to be evaluated, which is correctness and coverage. Coverage measures the proportion of entities from the source (Al-Quran, tafsir, hadith) that are represented in the ontology, while correctness evaluates whether the instantiated individuals conform to the ontology schema (Figure 1).

We put the result of our evaluation for coverage of our quranic KG in Table III. From table III we can see that for all categories, the mean coverage of our quranic KG for verses, surah, and Juz are 43,35%, 71,93%, and 30% respectively. The reason for these percentages is due to the limitation of the free version of Neo4J aura, we only can accommodate 9 parts from Al-Quran which is from part 22 until part 30. And as we know that the last 9 parts from Al-Quran contain many short chapters and verses.

¹<https://quran.ksu.edu.sa/>, ²<https://quran.nu.or.id/>, ³<https://www.muslimpro.com/>, ⁴<https://tarteel.ai/>

TABLE II
THE RESULT OF LLM EVALUATION

No	User Question	Generated Cypher	Result
1	how many verses are in surah muhammad	MATCH p=(a1:Ayat {id:'47:1'})-[:NEXT_AYAT*0..37]->(a38:Ayat {id:'47:38'}) OPTIONAL MATCH (a1)-[:HAS_TAFSIR_EN]->(t1:TafsirEN) RETURN p, size(nodes(p)) AS ayats_in_surah47, t1.text AS tafsir_EN_ay1	Surah Muhammad (Surah 47) contains 38 verses. The dataset explicitly lists ayats_in_surah47 as 38, confirming that Surah Muhammad has 38 verses.
2	What is verse 50:1?	MATCH p=(a:Ayat {id:'50:1'})<[:OCCURS_IN]-[:HAS_MAKNA_ID]->(m:MaknaID) OPTIONAL MATCH p2=(a)-[:HAS_TAFSIR_EN]->(t:TafsirEN) RETURN p, w.text AS word, m.text AS makna_ID, p2, t.text AS tafsir_EN	Here is what verse 50:1 says, based on the provided information. Arabic text : ق وَالْفُرَّانِ الْمَجِيدِ - Translation and explanation (from the provided Tafsir EN): "Qāf: God knows best what He means by this [letter]. By the glorious Qur'ān, [by] the noble [Qur'ān], the disbelievers of Mecca have [certainly] not believed in Muhammad (s)." - Last verse of Surah At-Tin (Surah 95): Ayah 8
3	What is the last verse of Surah At-tin	MATCH p_a=(a:Ayat {id:'95:8'}) OPTIONAL MATCH p_en=(a)-[:HAS_TAFSIR_EN]->(t:TafsirEN) RETURN COALESCE(p_en, p_a) AS path, a.text AS ayah_text, t.text AS tafsir_EN	- Last verse of Surah At-Tin (Surah 95): Ayah 8 بِأَحْكَمِ الْأَحْكَامِينَ يُالْتَمِزُ Translation: "Is not God the fairest of judges?" (also rendered as "Is not God the most just of all judges?") Context from the retrieved information: This is the concluding verse (Ayah 8) of Meccan Surah At-Tin, describing God as the fairest/most just of those who judge.
4	Retrieve the tafsir of verse 60:2	MATCH p=(a:Ayat {id:'60:2'})-[:HAS_TAFSIR_EN]->(t:TafsirEN) RETURN p, t.text AS tafsir_EN	Tafsir (English translation): If they were to prevail over you, they would be your enemies, and would stretch out against you their hands, to kill and assault you, and their tongues with evil [intent], with insults and reviling; and they long for you to disbelieve.
5	What is verse 60:2	MATCH p=(w:Word)-[:OCCURS_IN]->(a:Ayat {id:'60:2'})-[:HAS_TAFSIR_EN]->(t:TafsirEN) RETURN p, a.text AS ayat_text, w.text AS word, t.text AS tafsir_EN	- Ayat (Arab): إِنْ يَنْفَعُكُمْ يُكُونُوا لَكُمْ أَعْدَاءُ وَيَسْطُورُ إِلَيْكُمْ أَيْدِيَهُمْ وَأَلْسِنَتُهُمْ بِأَلْسِنَةٍ سَوِيَّةٍ وَوَدُّوا لَوْ تَكْفُرُونَ - English translation (tafsir EN): If they were to prevail over you, they would be your enemies, and would stretch out against you their hands, to kill and assault you, and their tongues with evil [intent], with insults and reviling; and they long for you to disbelieve.

TABLE III
COVERAGE FOR QURANIC KG

Category	Source	Quranic KG	% covered
Al-Quran verses, chapters, and parts (https://tanzil.net/docs/resources) and Tafseer (Tafsir Jalalayn) Indonesia	Verses: 6,236	Verses: 2,703	Verses: (2703 / 6236) * 100 = 43.35%
	Chapters/Surah: 114	Chapters/Surah: 82	Surah: (82 / 114) * 100 = 71.93%
	Parts/Juz: 30	Parts/Juz: 9	Juz: (9 / 30) * 100 = 30.00%
Tafseer (Tafsir Jalalayn) English and Quran Translation in Indonesia	Verses: 6,236	Verses: 2,703	Verses: (2703 / 6236) * 100 = 43.35%
	Chapters/Surah: 114	Chapters/Surah: 82	Surah: (82 / 114) * 100 = 71.93%
	Parts/Juz: 30	Parts/Juz: 9	Juz: (9 / 30) * 100 = 30.00%
Meaning word by word in Indonesia and English	77,430 words	29,512 word- meanings	(29,512 / 77,430) * 100 = 38.11%

To assess correctness, a domain expert in Al-Quran was consulted. The expert evaluated a random sample of 5 chapters from the Knowledge Graph and then compared it with Al- Quran (see Table IV). From the result, we can see that despite the coverage of parts of Al- Quran being only 30%, our KG has 100% correctness. This is due to the zero loss transformation that we do from the dataset to Neo4J.

TABLE IV
 THE EVALUATION OF CORRECTNESS OF QURANIC KG

Chapter (and verses)	Expert's comment
Chapter Al-lail (21 verses)	I found all verses from Al-Lail in this Quranic KG, and all verses are correct.
Chapter Al-Ikhlash (4 verses)	I found all verses from Al-Ikhlash in this Quranic, KG and all verses are correct.
Chapter Al-Mulk (verse 1-6)	I found all verses from Al-Mulk in this Quranic KG, and all verses are correct.
Chapter Al-Jumuah (verse 6-8)	I found all verses from Al-Jumuah in this Quranic KG, and all verses are correct.
Chapter Al-Fath (verse 17-21)	I found all verses from Al-Fath in this Quranic KG, and all verses are correct.

C. Model Evaluation

To evaluate the quality of the chatbot's responses, we conducted a Q&A-based assessment. For this purpose, an external domain expert, a domain expert is engaged to interact with the system. The expert was instructed to formulate and submit a total of 20 domain-specific questions to the chatbot, covering a broad range of relevant topics. The objective of this procedure was to assess not only whether the chatbot could provide correct and coherent answers, but also whether its responses were contextually appropriate and aligned with expert knowledge. The evaluation process revealed that the chatbot can answer correctly 18 expert's queries. In other words, for 2 out of 20 questions, the system returned the standard "No results found" response. Of the 20. questions that received an answer, 90% were judged as correct by the domain expert. A typical example of an incorrectly answered question is presented in table V:

TABLE V
 INCORRECT ANSWERED QUESTION

Question / Query	Respond from Chatbot
What is meant by goodness in Surah Az-Zalzalah, Juz 30?	I can't answer your specific question about the meaning of goodness in Surah Az-Zalzalah based on the information you provided. If you'd like, I can try to help with additional information about Surah Az-Zalzalah, or you can direct me to the relevant section.
What themes are discussed in the letter Al-Kauthar?	The available data does not cover the contents of Surah Al-Kauthar. If you would like a specific explanation of Surah Al-Kauthar, I can assist you based on general explanations found in other sources, but that information is not included in the information you provided here.

V. CONCLUSION

This paper has presented the design, development, and evaluation of the Quranic Knowledge Graph that aims to preserve the semantic networks of words, verses, and chapters in the Al-Quran. The type of knowledge graph we used is a property graph with a multi-layer architecture. By choosing this type, our Quranic Knowledge Graph is able to store rich relationships among words and verses from the Quran, along with the meaning and *tafsir* for each word or verse. To make our Quranic Knowledge Graph more useful, we developed a chatbot on top of the Quranic KG.

We performed a very complete evaluation, ranging from the LLM evaluation, knowledge graph evaluation, to the chatbot evaluation. All evaluations revealed the following facts: 1). Our approach can translate all user queries into correct Cypher language, 2). Due to the limitations of the free version of Neo4J, our Quranic KG only consists of $\pm 30\%$ of the Al- Quran, even though the correctness level is 100%, 3. The chatbot on top of our Quranic Knowledge Graph can achieve 90% accuracy in answering user questions.

As for future work, we want to investigate other free property knowledge graph platforms that can accommodate the entire Al-Quran and also the *tafsir*. We also want to enlarge the scope or increase the level of our Quranic KG by considering Hadith and other Islamic documents. Lastly, to increase the accuracy of our chatbot, we will try incorporating our system with advanced mechanisms, such as Retrieval Augmented Generation (RAG) or Reinforcement Learning from Human Feedback (RLHF).

DATA AND COMPUTER PROGRAM AVAILABILITY

Data and program used in this paper can be accessed in the following site <https://zentilax.github.io/quranic-chatbot-UI/>.

¹<https://quran.ksu.edu.sa/>, ²<https://quran.nu.or.id/>, ³<https://www.muslimpro.com/>, ⁴<https://tarteel.ai/>

REFERENCES

- [1] I. Safiraa, I. G. A. Wibawaa, N. A. S. ERa, I. B. Gede, and L. G. A. Dwidasmaraa, "Aplikasi pencarian informasi Al-Quran dengan ontologi web semantik," *Jurnal Elektronik Ilmu Komputer Udayana*, p-ISSN 2301-5373.
- [2] A. Hakkoum and S. Raghay, "Ontological approach for semantic modeling and querying the Qur'an," *Int. J. Islam. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 37–45, 2016.
- [3] F. Bendjamaa and N. Taleb, "OntoDin: An Islamic ontology of Quran and Hadith," *Int. Arab J. Inf. Technol.*, vol. 21, no. 5, pp. 773–785, 2024.
- [4] M. H. Bashir, A. M. Azmi, H. Nawaz, W. Zaghouani, M. Diab, A. Al-Fuqaha, and J. Qadir, "Arabic natural language processing for Qur'anic research: A systematic review," *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 6801–6854, 2023.
- [5] M. Sawalha, F. Alshargi, S. Yagi, A. T. AlShdaifat, and B. Hammo, "MASAQ parser: A fine-grained morphosyntactic analyzer for the Quran," in *Proc. New Horizons in Comput. Linguistics for Religious Texts*, Jan. 2025, pp. 67–75.
- [6] A. Alsaleh, E. Atwell, and A. Altahhan, "Quranic verses semantic relatedness using AraBERT," in *Proc. 6th Arabic Nat. Lang. Process. Workshop*, Apr. 2021, pp. 185–190.
- [7] M. Elkomy and A. M. Sarhan, "TCE at Qur'an QA 2022: Arabic language question answering over Holy Qur'an using a post-processed ensemble of BERT-based models," in *Proc. 5th Workshop Open-Source Arabic Corpora Process. Tools Shared Tasks Qur'an QA Fine-Grained Hate Speech Detection*, Jun. 2022, pp. 154–161.
- [8] D. Wardani, H. A. Syah, A. Wijayanto, and B. Harjito, "Flexible semantic Qur'an question answering using graph-based summarization and KNN," *JOIV: Int. J. Inform. Vis.*, vol. 8, no. 4, pp. 2155–2162, 2024.
- [9] A. Basharat, A. B. Kamran, and M. Rehman, "SemanticTafsir: Building a cultural heritage ontology and knowledge graph from the Quranic exegesis of al-Tabari," manuscript submitted to *Semantic Web J.*, 2025.
- [10] S. Al-Azani, M. Alowaifeer, A. Alhunief, and A. Abdelali, "OntologyRAG-Q: Resource development and benchmarking for retrieval-augmented question answering in Qur'anic tafsir," in *Proc. 2025 Conf. Empir. Methods Nat. Lang. Process.*, Nov. 2025, pp. 15551–15569.
- [11] S. Alnefaie, E. Atwell, and M. A. Alsalka, "Using the retrieval-augmented generation technique to improve the performance of GPT-4 in answering Quran questions," in *Proc. 6th Int. Conf. Nat. Lang. Process. (ICNLP)*, Mar. 2024, pp. 377–381.
- [12] A. Y. Alan, E. Karaarslan, and Ö. Aydın, "Improving LLM reliability with RAG in religious question-answering: MufassirQAS," *Turk. J. Eng.*, vol. 9, no. 3, pp. 544–559, 2025.
- [13] A. Ghafouri, M. A. Asl, H. Naderi, and M. Firouzmandi, "IslamicPCQA: A dataset for Persian multi-hop complex question answering in Islamic text resources," *IEEE Trans. Audio, Speech, Lang. Process.*, 2025.
- [14] F. Orlandi, D. Graux, and D. O'Sullivan, "Benchmarking RDF metadata representations: Reification, singleton property and RDF*," in *Proc. 2021 IEEE 15th Int. Conf. Semant. Comput. (ICSC)*, 2021, pp. 178–185, doi: 10.1109/ICSC50631.2021.00038.
- [15] T. Hahn, M. Hofer, and E. Rahm, "Benchmarking the RDF and property graph model in the temporal dimension: A case study with FinBench," in *BTW 2025: 21. Fachtagung für Datenbanksysteme für Business, Technologie und Web*, Lecture Notes in Informatics (LNI), 2025, pp. 1043–1054, doi: 10.18420/BTW2025-69.
- [16] R. Angles, H. Thakkar, and D. Tomaszuk, "Mapping RDF databases to property graph databases," *IEEE Access*, vol. 8, pp. 86091–86110, 2020, doi: 10.1109/ACCESS.2020.2993117.
- [17] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases: New Opportunities for Connected Data*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2015. (*This is the canonical academic and industry reference for defining and proving the performance benefits of "index-free adjacency"*).