

Embedding Search for Quranic Texts based on Large Language Models

Mohammed Alqarni

Department of Software Engineering,
University of Jeddah, Saudi Arabia
alqarni@uj.edu.sa

Abstract: *Semantic search is the process of retrieving relevant information from a large corpus of texts based on the meaning and context of the query. This paper is introduced in order to explore the use of large language models for semantic search of Quranic texts. The Quran, which is the central religious text of Islam, contains rich and complex linguistic and semantic features that pose challenges for traditional keyword-based search methods. This study investigates a semantic search approach utilizing Large Language Models (LLM) embedding and assess the performance of LLM embedding in comparison to a baseline embedding-based search method using a set of queries that represent different semantic search levels. In addition, this study will also discuss the limitations and implications of using large language models for semantic search of Quranic texts and suggest directions for future research. A significant finding in this study is the consistent effectiveness of the LLM embedding across varying semantic complexities. This suggests that embedding using LLMs can capture deep semantic connections effectively. On the other hand, as a second finding, the state-of-the-art transformer, AraT5, outperforms LLM embeddings in low-level semantic searches, indicating potential for further LLM fine-tuning on Arabic text corpora.*

Keywords: *Semantic search, quranic texts, large language models, GPT 3.5, embedding.*

Received November 7, 2023; accepted January 22, 2024
<https://doi.org/10.34028/iajit/21/2/7>

1. Introduction

The Quran holds profound significance, serving as the foremost spiritual and intellectual resource for 1.6 billion Muslims globally and appealing to both Muslims and non-Muslims seeking specific information or verses on particular subjects [1]. However, conducting effective searches within the Quran presents considerable challenges in the realm of Natural Language Processing (NLP) and semantic search due to its unique linguistic features. The Quran contains polysemous words, where a single word can convey multiple meanings, and conversely, various words can represent a single concept.

Additionally, topics may span multiple chapters and verses, further complicating the retrieval process. The Quran employs diacritical marks, which, when altered, can change a word's meaning, and it features synonyms, adding complexity to the language. Furthermore, Quranic knowledge varies in clarity, requiring supplementary explanations from Hadith and Tafsir for comprehensive understanding. Finally, a gap exists between user queries in Modern Standard Arabic and the Quran's text in Classical Arabic, posing translation and interpretation challenges, necessitating advanced solutions for effective Quranic information retrieval [1].

Semantic search plays a pivotal role in Quranic information retrieval, especially given the Quran's intricate language and semantic features, making it challenging for traditional keyword-based search

methods. Unlike lexical search, which relies on exact word matches, semantic search explores user queries, comprehending their meanings and contexts. This approach significantly enhances search accuracy by interpreting user intent and the nuanced meanings of words. Given the Quran's reliance on abstract concepts and metaphoric language, semantic search becomes indispensable, enabling exploration beyond literal text matches and facilitating a deeper comprehension of Quranic concepts and interpretations [1].

Nevertheless, existing Quran search tools face various limitations highlighted in a review by M Alqahtani and Atwell [5]. These limitations include an inability to address ambiguity in retrieved results, limited search capabilities concerning concepts, phrases, and questions, reliance on partial Quran ontologies, a lack of advanced NLP techniques like parsing and spell check in query analysis, and the absence of specialized Arabic named entities lists for Quranic text. The paper calls for future research to develop semantic search methods that overcome these limitations.

Moreover, using ontologies in Holy Quran Semantic Search has raised concerns regarding inflexibility and inconsistencies [15]. Safee *et al.* [18] address the challenge of accurately and scientifically retrieving Quranic verses. They evaluate existing methods, identifying strengths and weaknesses, and emphasize the importance of systematic Quranic information retrieval. Their research involves developing a Quran

ontology alignment model, integrating upper and domain ontologies, highlighting the necessity for ontology renovation to resolve inconsistencies.

F.S. Utomo's investigation of existing Question Answering Systems (QAS) for the Holy Quran identifies shortcomings in components like question processing, document retrieval, and answer extraction. This research underscores the need for improvements in morphology analysis, precise question classification, advanced search techniques, and the utilization of ontology resources within QAS frameworks related to the Holy Quran [24].

Given the complexities of Quranic texts, traditional keyword-based search methods prove insufficient. Existing semantic search systems for Quranic texts, while promising, often exhibit limitations in scope, accuracy, or usability. They fail to fully harness the potential of large language models, which excel in natural language understanding tasks. Therefore, there's a pressing need to explore how large language models can enhance the semantic search of Quranic texts, providing users with more relevant and comprehensive results. This research introduces a methodology to make use of LLM embeddings and compare the performance of large language model-based semantic search with traditional model-specific embedding methods. The goal is to demonstrate the feasibility and effectiveness of using these embeddings for Quranic semantic search while identifying limitations and challenges for future research.

2. The Potential of Large Language Models for Semantic Search

In the past, embedding models were limited in their scope, tailored for specific, specialized tasks. Nevertheless, the advent of modern LLMs has marked a transformative shift towards versatility. LLMs have evolved to take on a wide range of functions, including the crucial domain of semantic search. Semantic search involves the development of search engines that venture into the depths of document meaning, aiming to uncover rich, contextually relevant information. At the heart of this transformation is the intricate training process of LLMs, which inherently infuses semantic relationships throughout their layers. These embeddings constitute the foundational element in the realm of semantic search, where the focus lies on assessing the likeness between two segments of text, often referred to as 'chunks'.

In recent studies, the potential of LLMs for enhancing semantic search and information retrieval has garnered significant attention. Abuzayed and Al-Khalifa [1] explored various topic modeling techniques, including Latent Dirichlet Allocation (LDA), Non-Negative Matrix Factorization (NMF), and BERTopic, within the context of analyzing a vast collection of Arabic documents. Their investigation revealed that

BERTopic, leveraging Pre-Trained Arabic Language Models for embeddings, exhibited remarkable performance compared to conventional LDA and NMF approaches. This superiority was assessed using the Normalized Pointwise Mutual Information (NPMI) measure. However, it is worth noting that the challenge of devising precise evaluation metrics for assessing the quality of topics generated by BERTopic remains ongoing, calling for further research in this domain.

Sun *et al.* [21] delved into the application of generative Large Language Models, namely ChatGPT and GPT-4, for relevance ranking in Information Retrieval (IR). Their experiments demonstrated that appropriately instructed ChatGPT and GPT-4 outperformed supervised methods across popular IR benchmarks. GPT-4, in particular, exhibited exceptional performance, surpassing fully fine-tuned models on multiple datasets. The study also introduced a specialized model that outperformed another model trained on annotated data and introduced innovative approaches for permutation generation and distillation. These findings underscore the effectiveness and efficiency of ChatGPT and GPT-4 in passage re-ranking and offer valuable insights for code reproduction.

Petroni *et al.* [16] conducted an extensive analysis of pretrained language models, with a particular focus on BERT-large, to evaluate their capability to recall factual and commonsense knowledge without requiring fine-tuning. Their investigation revealed that in its original form, BERT-large exhibited competitive relational knowledge, showcasing proficiency in open-domain question answering, even when compared to supervised baselines. The study shed light on the acquisition patterns of various types of factual knowledge during pretraining, emphasizing the potential of pretrained language models as potent tools for unsupervised open-domain question answering. These findings contribute to the ongoing exploration of language models' knowledge representation abilities.

Muennighoff [12] introduced SGPT, a novel approach harnessing decoder transformers, particularly GPT models, to enhance sentence embeddings and semantic search. SGPT for Bi-Encoders (SGPT-BE) presented modifications to GPT models, including position-weighted mean pooling and fine-tuning of bias tensors, resulting in state-of-the-art sentence embeddings, even with smaller-scale models, surpassing prior methods. SGPT for Cross-Encoders (SGPT-CE) employed log probabilities from pretrained GPT models to achieve state-of-the-art, unsupervised search results in semantic search tasks. This research addresses the challenge of extending large decoder models beyond traditional language tasks, providing efficient solutions for semantic search and sentence embeddings, while making code, models, and results accessible to the research community.

In summary, LLMs offer an alternative to traditional knowledge bases for semantic searching. LLMs possess

implicit semantic knowledge embedded within their parameters, which can be further fine-tuned for specific tasks, greatly enhancing their semantic search capabilities. On the other hand, knowledge bases provide access to meticulously annotated relational data, but they require complex NLP pipelines, including entity extraction, coreference resolution, entity linking, and relation extraction, which often rely on supervised data and fixed schemas. LLMs offer several advantages, such as no need for schema engineering, support for open-ended queries, scalability, and the elimination of human annotation requirements. This makes them a promising choice for semantic searching over traditional knowledge bases.

3. Related Works

In this section, we explore various approaches to semantic search within Quranic texts, each represented by a distinct subsection. Section 3.1. investigates the use of embedding techniques for enhancing semantic searches, while section 3.2. focuses on incorporating ontological structures to improve search accuracy and context-awareness. Section 3.3. explores methods leveraging translation processes, section 3.4. delves into strategies involving indexing and query expansion, and section 3.5. discusses the application of text mining for uncovering implicit semantic relationships. Finally, section 3.6. and section 3.7. examine the utilization of knowledge representation and graph-based approaches to enrich the semantic search experience. Together, these subsections provide a comprehensive overview of diverse strategies employed in the quest for semantic excellence within Quranic textual searches.

Semantic search is essentially grounded in either explicit or implicit knowledge representation. Explicit knowledge representation involves using ontologies and other structured knowledge representations to uncover semantic relationships between stored text and queries. Implicit knowledge representation, on the other hand, utilizes embedding techniques to implicitly store semantic features of the text. In the context of knowledge representation methodologies for semantic search in the Holy Quran, explicit representation is delineated in sections 3.2. and 3.6. Conversely, implicit representation employs embedding, as outlined in section 3.1., or text mining, as demonstrated in section 3.5. Other conventional methods encompass indexing and query expansion, showcased in section 3.4., and translation, exemplified in section 3.3.

Finally, it should be noted that all the previous works in the field of semantic search for the Holy Quran have been confined to a single type of semantic search-specifically, low-level semantic search. In contrast, this study encompasses both mid-level and high-level semantic queries.

3.1. Quranic Semantic Search Based on Text Embedding

Semantic Text Similarity (STS) plays a pivotal role in enabling semantic search systems to assess the similarity in meaning between two pieces of text, a fundamental component for various applications. These applications range from plagiarism detection and improving customer service by identifying related queries to enhancing search engines and forums such as Stack Overflow and Quora. In the context of religious texts, STS facilitates the understanding of how similar knowledge is expressed across different contexts. Extensive research has been dedicated to enabling machines to measure semantic similarity, aiming to automate the extraction of pertinent information from large datasets. Since each written piece possesses its uniqueness influenced by factors like subject matter, target audience, and authorial style, STS becomes indispensable for effective semantic search and information retrieval [2].

Mohamed and shokry [11] introduced the Quranic Semantic Search Tool (QSST) as a solution for Quranic information retrieval. QSST addresses the limitations of existing Quranic search tools by integrating semantic search techniques with embedding methods. The tool comprises four essential phases: constructing a Quran dataset through manual annotation based on the Mushaf Al-Tajweed ontology, training a Continuous Bag of Words (CBOW) model on a comprehensive Quranic and Classical Arabic corpus for word embedding, computing feature vectors for user queries and Quranic topics, and retrieving relevant verses using cosine similarity. The study showcases the effectiveness of QSST, demonstrating high precision, recall, and F-scores, thus highlighting its superiority over existing tools. QSST emerges as a valuable resource for scholars and individuals seeking an enriched understanding of the Quran through enhanced semantic searching.

Saeed *et al.* [17] leveraged Word2Vec and Sent2Vec models to extract knowledge from various English translations of the Holy Quran. Sent2Vec models were trained on these translations to explore semantic relationships between words and sentences. The study evaluated the performance of custom word embeddings against pre-trained embeddings from the Spacy library and presented ongoing work on developing a text similarity computing framework. By employing the Holy Quran as a test corpus, Word2Vec was used to generate custom word embeddings from English translations, and sentence embeddings were constructed by averaging word embeddings within sentences. The experiments revealed that the Skip-Gram model achieved the highest precision score at 94%. Despite this achievement, the paper outlined plans and improvements for integrating knowledge-based models with machine learning techniques in future research.

Alshammeri *et al.* [6] ventured into the complex task of evaluating semantic similarity among Quranic verses, renowned for their religious significance. Their methodology harnessed NLP techniques to quantify semantic-based similarity by converting Arabic Quranic verses into numerical vectors capturing their semantic essence. The model's effectiveness was assessed using cosine similarity, resulting in an impressive 76% accuracy in identifying similarities. This research marks a foundational step in the ongoing exploration of Quranic semantic analysis, providing a platform for further research and development. The paper introduced an NLP model that utilized Doc2vec to predict semantic-based similarity within the original Arabic Quranic verses, achieving a 76% accuracy and a 51% F1-score. It laid the groundwork for constructing a comprehensive knowledge base for Quranic verse analysis and outlined future plans to explore various similarity measures and integrate advanced deep learning models like ELMo and BERT.

However, all the methods employed for embedding, such as CBOW, Sent2Vec, Skip-Gram model, and Doc2vec, are considered primitive compared to the advanced techniques like AraT5 and GPT embeddings. AraT5 and GPT are sophisticated transformer-based language models. To elaborate, the methods employed generate static embeddings for words or documents, lacking consideration for the context in which they appear. This absence of contextual information can restrict their capability to capture intricate relationships. In contrast, transformer architectures utilized in models such as GPT and AraT5 enable the effective capture of long-range dependencies and complex linguistic patterns. The inclusion of attention mechanisms enhances their ability to comprehend relationships across different sections of a text. Furthermore, unlike transformer-based models like GPT and AraT5, these traditional models may encounter challenges in handling polysemy and ambiguity due to their static nature.

3.2. Ontology Based Semantic Search

Beirade *et al.* [7] contribute to the advancement of semantic search for Quranic text by designing a specialized semantic search engine for the Quran, leveraging a Quranic ontology. This research recognizes the intricate nature of Arabic and the Quran's complex semantics, focusing on semantic enrichment through query expansion. The methodical construction of the Quranic ontology categorizes words hierarchically, offering a comprehensive framework for semantic relations and enhancing query capabilities. Apache Lucene serves as the search engine, incorporating ontology-based semantic treatments to bolster search outcomes. This work significantly contributes to the automatic processing of Arabic, particularly in Quranic text analysis, serving as a valuable resource for Quranic

semantic search.

Zouaoui and Rezeg [25] introduces an innovative Quranic semantic search engine driven by an ontology index, addressing the limitations of existing Quranic search engines. The creation of a Quranic ontology rooted in traditional Arabic grammar, enriched with concepts and relationships from existing ontologies, underpins this solution. Leveraging word links within the ontology, the engine enables precise verse retrieval, with user queries converted into SPARQL queries. Performance evaluations underscore substantial improvements in Quranic information retrieval, making it indispensable for automated Quranic processing and opening avenues for Question Answering (QA) in the Quran.

Khan *et al.* [10] Take on the challenge of enhancing semantic search within the Holy Quran, advocating for the application of semantic web principles, particularly ontologies. They illustrate this through the creation of a domain ontology focused on living creatures mentioned in the Quran, utilizing the Protégé ontology editor. The paper emphasizes the potential for achieving comprehensive semantic search coverage across the entire Quran, including proposals for a Quranic WordNet, ontology integration, and mapping under an overarching upper ontology. This research paves the way for enhancing topic-based search capabilities within the Quranic text.

Shoab *et al.* [19] addresses the limitations of conventional keyword-based Quranic search techniques, introducing a novel model for semantic search within the Quran. The model utilizes WordNet relationships in a relational database structure and focuses on Al-Baqrah S, showcasing superior precision compared to simple keyword searches. This model holds promise not only for Quranic text analysis but also for potential applications in other holy books and legal texts, with future adaptations considered for Hadith and Fiqh books.

Alqahtani and Atwell [4] conducts a comprehensive review of Quranic search techniques and ontologies, culminating in the development of an Arabic-English Quran ontology aimed at comprehensive Quranic knowledge coverage and semantic tagging. The research also introduces an Annotated Corpus of Arabic Questions and Answers on the Quran, containing labeled question-answer pairs from trusted Islamic sources. Furthermore, it presents a semantic search model for the Arabic Quran based on this ontology, integrating information retrieval and semantic search techniques to improve Quranic search applications.

Sultana *et al.* [20] presents a semantic-based approach for retrieving food-related verses and concepts from the Quran using natural language queries. This method employs ontology technology and a triplet extraction algorithm, demonstrating effectiveness in extracting domain-specific information from the Quran. The approach achieved high recall and precision in

experimental queries related to the food domain, showcasing the potential of ontological semantic search for specific Quranic knowledge extraction and suggesting future enhancements through the integration of Quran and Hadith ontologies.

It is crucial to note that ontology-based representation lacks the ability to discover unknown relations unless already stored in its knowledge base, making it weak in uncovering indirect or embedded semantic relations.

3.3. Semantic Search Based on Translation

Nuhu *et al.* [13] introduce an innovative query expansion approach aimed at augmenting the performance of Quranic web search engines. This approach harnesses explicit relevant feedback, word synonyms, and semantic relatedness to enhance search results. Through rigorous experiments conducted on Quran datasets, specifically Yusuf Ali, Arberry, and Sarwar Quran datasets, the proposed approach, termed Query Expansion with Explicit-relevant feedback and Synonyms (QEES), exhibits superiority over alternative methods. This superiority is substantiated by notable enhancements in metrics such as Mean Average Precision (MAP), mean reciprocal rank, average precision, and recall. Notably, QEES offers the advantage of adaptability across various search engine algorithms for ranking documents by relevance. As part of future work, the paper suggests exploring the integration of distributed term representations with semantic metadata and potentially incorporating machine learning algorithms, such as neural networks, to further elevate query expansion performance and enhance the quality of search results.

Afzal and Mukhtar [2] present an encompassing framework designed to enhance English search capabilities within the Holy Quran. This comprehensive framework addresses both concept- and keyword-based search methods. It introduces a Qur'anic English WordNet, an enriched resource built upon the WordNet structure, tailored to Quran-specific information. Furthermore, it automates the extraction of a Vocabulary of Qur'anic Concepts, effectively creating a conceptual hierarchy derived from English Quran translations. The framework culminates in the development of the Qur'an Enhanced Search Tool (QuEST), a potent information retrieval tool enabling semantically enriched Quranic searches based on these resources. QuEST's performance surpasses that of comparable tools, particularly in terms of recall. Significantly, the approach is semi-automatic, incorporating initial automatic extraction and data preprocessing, followed by manual validation by domain experts to ensure fidelity to the Quranic message. The paper underscores the scalability of this knowledge repository creation process compared to manual methodologies and underscores its potential application to other religious

texts, such as Islamic Jurisprudence and Hadeeth, with the overarching goal of facilitating semantically enriched Quranic searches using linguistically rich resources containing semantic annotations of English Quran translations.

In a quest to elevate the performance of Quran translation search, with a specific focus on French text retrieval, Nuhu *et al.* [14] introduce a novel query expansion methodology rooted in semantic search principles. This approach underscores the significance of considering the context of terms within sentences. The experimentation involves 6236 verses of the Holy Quran, translated by Hamidullah. Notably, the results demonstrate a substantial 36% improvement in performance compared to other models. As a future research direction, the paper suggests the exploration of Quran ontology development to further enhance query expansion capabilities for Quranic text retrieval. In summary, this paper contributes by proposing a semantic search query expansion approach, conducting performance comparisons with analogous methods, and augmenting Quranic French text retrieval through improved query expansion enriched by latent semantic analysis.

In conclusion, this approach relies on the existence of explicit linguistic relations that can be translated into Arabic, presenting the same drawbacks of inflexibility in adapting and discovering unseen or indirect relations.

3.4. Using Indexing and Query Expansion

Nuhu *et al.* [15] introduces an innovative approach to enhance Quranic search engines by integrating semantic search and Lucene ranking techniques for query expansion. This novel method introduces semantic search principles to expand Quranic search queries with word synonyms and employs a Quran ontology to establish concept relationships within the query. The Lucene ranking algorithm is then customized to enhance query expansion, considering stemming, stop words, and derivatives. Comprehensive experiments conducted on eight Quran datasets reveal the effectiveness of the proposed approach, named Lucene Query Expansion with Synonyms and Lucene Ranking (LQA) and Semantic Search Lucene Ranking (SSLR). These approaches yield substantial improvements in MAP, with enhancements of 55% and 48%, respectively, particularly notable on the Arberry dataset. Future research directions should prioritize the refinement of the Quran ontology and the exploration of distributed semantic representations to further enhance query expansion.

Karim [9] embarked on creating the first Semantic Search and Intelligence System for the Quran, enabling advanced semantic searches, comprehensive Quranic text analysis, and effective data visualization for users and scholars. This ambitious project also aimed to consolidate prior research efforts from Leeds University

and establish an open-source framework for Quranic analysis, fostering innovation in this field. Key research objectives encompassed the construction of a Semantic Search Engine, implementation of an Intelligence and Analytics system, enhancement of result visualization, and incorporation of a question answering component. The remarkable outcomes of this research include the development of a specialized semantic search website for the Quran, the contribution of 20 accessible research tools catering to scholars, the establishment of an extensive Quranic Ontology, the initiation of an open-source Quranic research framework, the introduction of innovative visualization techniques, advancements in ontology extraction from Arabic text, and the provision of new Quranic corpora and resources.

Ishkewy and Harb [8] introduce the Islamic Semantic Web Search Engine (ISWSE) as a dedicated system to improve information retrieval within the Holy Quran and Islamic texts. Addressing the limitations of traditional keyword-based search engines, this work advocates for the adoption of semantic search principles in an Islamic context. The ISWSE is constructed upon an Islamic Ontology, incorporating fundamental concepts from Islamic Sharia and utilizing Azhary as a lexical ontology for Arabic. The paper outlines the architecture of the system, presents experimental findings, and illustrates its effectiveness in retrieving Quranic information. Future endeavors involve expanding the Islamic ontology and the inclusion of additional Islamic texts such as the Sunnah and Sharia books, aiming for a more comprehensive and context-aware search system.

Tarawneh and Al-Shawakfa [23] propose a hybrid approach for searching and retrieving information from the Holy Quran, addressing the intricacies of searching Arabic text. This innovative method combines syntactic (keyword-based) and word semantic techniques, processing the Quranic text in XML format and implementing a user-friendly Python-based system featuring a Graphical User Interface. Preliminary results showcase the potential of this approach, demonstrating efficient information retrieval from Quranic text. The authors suggest that this approach could be extended to handle other Islamic texts like Hadith and Fiqh, emphasizing the need for continual advancements in Arabic information retrieval methodologies. This pioneering research contributes significantly to the domain of Arabic information retrieval, particularly focusing on the word-level analysis.

Using indexing and query expansion provides means to expand the query based on the existence of open linked data. However, similar to ontology-based search, it lacks the ability to adapt, discover, and infer indirect or contextual semantic relations.

3.5. Search Based on Text Mining

Alhawarat *et al.* [3] initiates a series of research studies focused on extracting valuable information and knowledge from the Holy Quran, catering to both specialized scholars and the wider audience. The aim is to establish a framework for analyzing Arabic text, particularly by preprocessing the Holy Quran and applying various text mining techniques to uncover insights about its terms. The study employs Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) methods to calculate term frequencies, leading to findings such as important words, word clouds, and chapters with high term frequencies. This research serves as a starting point for future work in natural language processing, data mining, and text mining applied to the Holy Quran text, providing a glimpse of potential insights that can be gleaned from this sacred text. The study acknowledges the need for more advanced stemming algorithms for Arabic to enhance accuracy in future research endeavors.

Text mining in this work relies on the statistical TF-IDF methods, lacking the ability to discover contextual dependencies beyond its scope or handle out-of-vocabulary words. It treats each word as an independent entity based on its frequency and lacks the semantic richness provided by embeddings.

3.6. Search Using Knowledge Representation

Ta'a *et al.* [22] aims to enhance the knowledge retrieval process from the Al-Quran by constructing an ontology that aligns with its inherent structure of Division (Juz), Chapter (Surah), and Verse (Ayah). Traditional knowledge management methods, such as Taxonomy and hierarchy, are deemed insufficient for capturing the depth of Al-Quran knowledge, as they don't connect it to broader themes in Islamic knowledge. The authors propose the development of an Al-Quran ontology using a semantic-based approach to ensure accurate representation and retrieval of verses related to Al-Quran themes. The ontology, comprising 14 themes, is validated by experts, and a semantic-based searching method is introduced for efficient content retrieval. Evaluation tests indicate promising accuracy levels, with the goal of improving Al-Quran understanding. Future work will focus on user acceptance of the Al-Quran ontology applications.

Similar to ontologies, this approach may struggle with capturing the subtleties of semantic relationships in the Holy Quran. Adapting the knowledge base to new themes or dealing with unstructured data can be challenging. It faces issues in handling ambiguity and polysemy, relying on predefined relationships and meanings. Additionally, it may not effectively capture contextual variations, especially in tasks requiring an understanding of highly semantic contexts similar to the Holy Quran.

4. The Proposed Methodology

This paper addresses the problem of exploring semantic search of Quranic texts with large language models embedding. The purpose of this research is to investigate how embedding using large language models can improve the accuracy and relevance of semantic search results for Quranic texts. As previously described, the utilization of robust LLMs and embedding libraries represents a significant advancement in the practical implementation of semantic search and various natural language processing tasks. These models possess extensive language understanding and generation capabilities, enabling us to extract context and meaning from text data effectively. The integration of these readily available components offers a rapid path to application development. There is a wealth of open-source frameworks and a diverse array of large language models, including both closed and open-source alternatives, at our disposal.

The population of this study is the Quran, which is the central religious text of Islam. The sample size is a publicly available Quranic corpus that contains 114 chapters, 6236 verses, and 77,430 words in Arabic. The sampling technique is purposive sampling, as the Quranic corpus is selected based on its availability, accessibility, and suitability for semantic search tasks.

This research aims to compare the performance of large language model-based semantic search with a baseline of specific embedding methods on a set of queries that reflect different types of semantic search tasks, such as finding verses related to a specific topic, finding verses with similar or opposite meanings, and finding verses that contain specific words or phrases. The research methodology is quantitative, as it will use

numerical data and statistical analysis to measure and compare the precision, recall, and F1-score of the two search methods.

The inclusion criteria for this study are queries that are relevant to Quranic texts and can be answered by semantic search methods. The exclusion criteria are queries that are irrelevant, ambiguous, or too complex for semantic search methods. These criteria will ensure that the data collection and analysis are focused on the research question and objectives.

The functionality of our methodology is as follows: When a user poses a question like 'What are the rewards for the believers?' the system initiates by generating an embedding for the question.

The functionality of our methodology is as follows: When a user poses a question like 'What are the rewards for the believers?' the system initiates by generating an embedding for the question. Subsequently, we reference our collection of embeddings to identify the relevant verses or sections from policy documents connected to this question. We aggregate all these sections, in conjunction with the user's initial question, into a prompt for our LLM, instructing it to respond based on this data. The LLM then furnishes us with an answer.

The proposed methodology has 5 steps as shown in Figure 1.

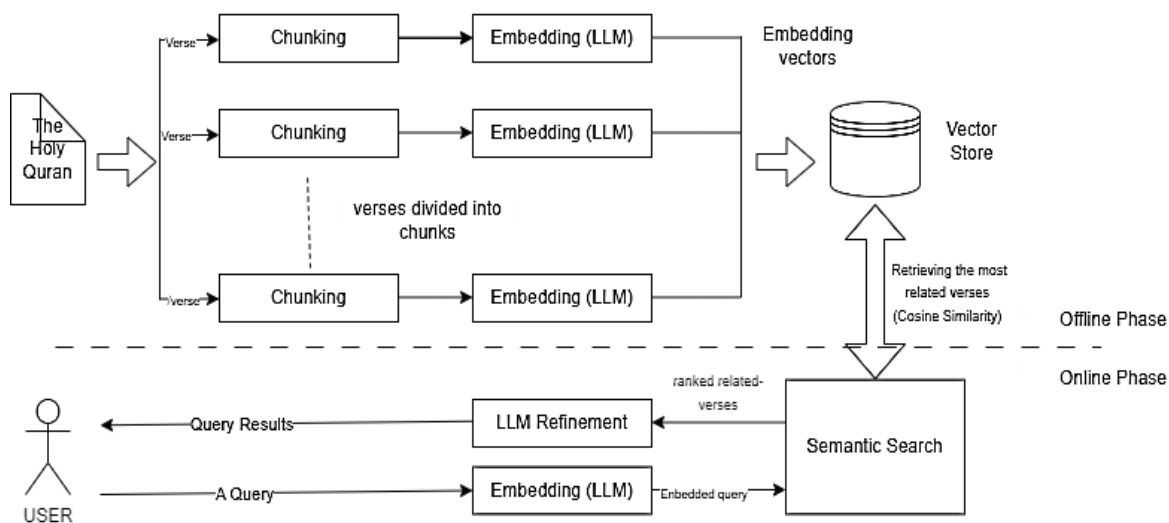


Figure 1. The proposed methodology.

- *Step 1. Chunking:* in the context of developing applications involving LLMs, “chunking” is the practice of breaking down lengthy text into smaller segments. This is a crucial step to improve the

relevance of information retrieved from a vector store after text embedding using LLMs. These embedded text segments are then used to build context for a conversational agent, which relies on a knowledge

base for reliable information. The first step in our proposed model is to chunk a chapter of the Holy Quran into verses. This can be done using a variety of methods, such as using regular expressions or a Quranic text processing library.

In the text of the Holy Quran, verses are conventionally divided using a symbol referred to as an "ayah marker" or "verse marker." The most commonly employed marker in the Quran is the Arabic symbol "◌," also known as the "U+06DD: ARABIC END OF AYAH" character within the Unicode standard. In digital formats of Quranic text, the presence of this character, U+06DD, serves to indicate the conclusion of a verse. Pseudocode (1) presents a Python-like code snippet that demonstrates how to transform Quranic text into a list of verses by identifying and utilizing this character.

Pseudocode 1: Performing Chunking.

```
def Chunking(Quran_text):
# Converts a text into a list of sentences.
args:
Quran_text: A string containing the Quran text, which is
usually a chapter.
returns:
A list of strings, where each string is a sentence in the
paragraph.
# Split the Quran text into verses based on the separator "verse
marker."
# Or the "U+06DD: ARABIC END OF AYAH" character
verses = Quran_text.split('◌')
# Remove any empty strings from the list of sentences.
verses = [verse for verse in verses if verse]
return verses
# Example usage:
verses_list = Chunking(Quran_text)
print(verses_list)
Output:
//// a list of separated verses
```

- **Step 2. Chunks Embedding:** vector embeddings encode data by preserving vital semantic details, transforming text into a machine-comprehensible structure that establishes dimensions for individual text components. This is essential for identifying text similarities. Chunk Embedding has key features, such as assigning a unique numerical identifier to each segment and utilizing LLM-generated embeddings, which can cover a wide range of topics, span different concepts, and capture subtle language nuances to enhance language comprehension and application performance.

Once we have chunked the chapter into verses, we need to embed each verse into a vector space. This can be done using a variety of embedding models, such as text-embedding-ada-002. Verse or chunk embedding demonstrates crucial features, such as assigning a unique numerical label to individual chunks, providing adaptability through embeddings generated by LLMs capable of encompassing a wide range of concepts.

We employed the 'text-embedding-ada-002' embedding model developed by OpenAI for our implementation. This second-generation model represents a significant advancement over its predecessors. Text-embedding-ada-002 is categorized as a unified embedding model, designed to handle various tasks, including text and code search, sentence similarity assessment, and text classification. It notably outperforms OpenAI's previous embedding models in text search, code search, and sentence similarity. One of the model's key strengths lies in its ability to generate contextual word representations, allowing it to discern word meanings based on their surrounding context. The Pseudocode (2) is the Python-like algorithmic coding for performing the embedding.

Pseudocode 2: Chunks Embedding.

```
import OpenAI
# Create an OpenAI client
openai.api_key = "My_API_Key"
# Generate embeddings for all of the documents in the search
corpus
verses_embeddings = []
for verse in verses_corpus:
    verse_embedding =
OpenAI.create_embedding(input=verse_text)
    verses_embeddings.append(verse_embedding)
# End of embedding phas
```

This code will generate a 1536-dimensional embedding vector for the text string *verse_text*. The embedding vector can be used for the next stages of the model.

- **Step 3. Query Embedding:** the next step is to embed the user query into the same vector space as the verses. This allows us to compare the similarity between the query and the verses. A code segment that can perform the query embedding is as shown in Pseudocode (3).

Pseudocode 3: Query Embedding.

```
# Generate an embedding for the query string
query_embedding =
openai.create_embedding(input=user_query)
```

- **Step 4. Semantic Search:** this step uses this embedding to find the most similar verses based on cosine similarities with their embedding. Cosine similarity is a common method for measuring the similarity between vectors, often used in information retrieval, including Quranic verse retrieval based on semantic vectors. It quantifies how closely aligned the directions of two vectors are, with a value between -1 and 1. A cosine similarity of 1 implies identical vectors, 0 indicates orthogonal vectors, and -1 signifies vectors in opposite directions. This technique is instrumental in *assessing* the relevance of Quranic verses to specific search queries. We can calculate the cosine similarity between the query embedding and the embedding of each verse.

The verses with the highest cosine similarities to the

query will be the most similar verses. The Python code, as shown in Pseudocode (4), will generate a list of the top n search results for the query `user_query`. The search results are ranked by their cosine similarity to the query embedding.

Pseudocode 4: Semantic Search.

```
# Calculate the cosine similarity between the query embedding
and all of the verses embeddings
scores = []
for verse_embedding in verses_embeddings:
    score =
openai.calculate_cosine_similarity(query_embedding,
verse_embedding)
    scores.append(score)
# Sort the verses by their scores
sorted-verses = [verse for score, verse in sorted(zip(scores,
verse_corpus), reverse=True)]
# Print the top n search results
printf("Top n search results for query '{query_embedding}':")
for verse in sorted-verses[:n]:
    print(verse)
```

- **Step 5. LLM-based refinement:** this step leverages LLMs to generate responses based on related verses, enabling the construction of appropriate answers. The process involves identifying the most relevant verses to a query using LLMs and subsequently utilizing these LLMs for answer generation, which can take the form of summarizing verses meaning or explanations, answering questions about them, or generating a subject with similar content. By adopting this approach, you gain flexibility and adaptability in addressing various issues using a single LLM. It simplifies updates and accommodates extensive inputs, including entire text. This strategy, termed “grounding,” encourages the LLM to rely on provided context, reducing the risk of inaccurate responses. This framework extends beyond mere question answering, offering broader capabilities.

Pseudocode (5) is Python-like code that uses the top n search results from the previous example and feeds them to the LLM in order to get an enriched answer:

Pseudocode 5: Query response refinement.

```
# Get the top n search results for the query
for i in range(n):
    ranked-verses.append(sorted-verses[i])
# Feed the top ranked search results to the LLM
enriched_answer = openai.complete(
    prompt="Here are the top verses results for the query What
is 'Muhammad's mission in the Quran? '\n\n"
    + "\n".join(ranked-verses)
    + "\n\nWhat is the enriched answer to this query?",
    temperature=0.7,
    max_tokens=1024,
)
# Print the enriched answer
print(enriched_answer["choices"][0]["text"])
```

5. Results and Analysis

The used dataset is a tensor of two-dimensional array, as explained earlier, that consists of four columns. The first column is the verse embedding and the other three columns are Verse Text (without Diacritics), Surah ID, Verse ID. The Verse Text are imported from the “Arabic Original Qur'an” dataset (Tanzil Documents). The original Arabic Original Qur'an dataset is a CSV file. The Quran is divided into 30 parts (juz'), 114 chapters (surah) and 6236 verses (ayat).

The performance of the model is evaluated using precision, recall and F-score. Precision and recall are fundamental performance metrics in the field of information retrieval, including semantic search. They are used to evaluate the quality and effectiveness of search systems in retrieving relevant information from a corpus or database.

Precision, often referred to as positive predictive value, measures the accuracy of the retrieved results based on Equation (1).

$$Precision = (Number\ of\ Relevant\ Documents\ Retrieved) / (Total\ Number\ of\ Documents\ Retrieved) \quad (1)$$

In semantic search or information retrieval, precision is crucial because users expect highly relevant information when querying for specific concepts or topics. A system with high precision ensures that users receive accurate and valuable content in their search results.

Recall, also known as sensitivity, measures the completeness of the retrieval system. Recall assesses the ability of the search system to find all relevant items in the corpus based on Equation (2).

$$Recall = (Number\ of\ Relevant\ Documents\ Retrieved) / (Total\ Number\ of\ Relevant\ Documents\ in\ the\ Corpus) \quad (2)$$

Finally, F-Score provides a balanced measure of precision and recall and is measured as a harmonic mean of precision and recall as shown in Equation (3).

$$F - Score = 2 \times Precision \times Recall / (Precision + Recall) \quad (3)$$

In semantic search, recall is essential to ensure that users are not deprived of relevant information. Failing to retrieve relevant documents can lead to incomplete search results, which can be frustrating for users seeking comprehensive information.

In semantic search, achieving an optimal balance between precision and recall is crucial. Users expect both highly relevant results and comprehensive coverage. Search systems may employ various techniques, such as query expansion, relevance ranking algorithms, and filtering, to enhance both precision and recall simultaneously.

The evaluation of the suggested approach's effectiveness involves three distinct types of semantic searches: low-level, mid-level, and high-level. In the low-level semantic search, the focus is on searching for a term or its associated linguistic terms or synonyms. In contrast, the mid-level semantic search centers around

querying for concepts, and in the high-level semantic search, the focus is on searching for topics or subjects, the used queries and their categorizations is shown in Table 1 the relevant verses for each query were reviewed by three experts.

Table 1. The list of the queries.

Category	Query	Examples of related semantics	Number of relevant verses
Low-level semantic query	مكة المكرمة Mecca the Honored	بكة، أم القرى، البلد البند الأمين البلدة التي حرمها الله	9
	الإحسان Doing good	المحسنون، الذين أحسنوا	57
	المدينة المنورة Al Madina	المدينة يثرب	
	ضوء القمر The moonlight	والقمر نورا، والقمر بازغا، وجعل القمر فيهن نورا وخسف القمر	26
	المسجد الحرام the Grand Mosque	البيت، البيت العتيق بيتك المخرم	28
	الأشجار The trees	الشجر، الشجرة، النخل،	26
Mid-level semantic query	معاملة اليتيم dealing with orphans	اكرام اليتيم، اليتامى، حق اليتيم	17
	فضل الصدقة the virtue of charity	الصدقات، لعنا نتصدق، اطعام المسكين، تصدق، المتصدقين	21
	فضل الاستغفار the virtue of seeking forgiveness	استغفروا ربكم، رب اغفر لي المستغفرين بالاسحار	55
	الطير في القرآن الكريم Birds in the Holy Quran	السلوى، والبعض، والذباب، والنحل، والعنكبوت، والجراد، والهدده، والغراب، وأبائيل،	33
	بر الوالدين kindness to parents	بالوالدين احسانا، ووصينا الانسان بوالديه	20
	أنواع القلوب Types of hearts	أقنعة قلوب سليم، قلوب كالحجارة	137
High-level semantic query	الدليل على غنى قارون What is the evidence of Qarun's wealth?	ما أوتي قارون إنه لنذو حظ عظيم وأثنيته من الكنوز	4
	كيف وصف المولى عز وجل سيدنا اسماعيل How did God describe Ismael?	ويشراه بعلام حلیم، إنه كان صادق الوعد وكان رسولا نبيا وإسماعيل وإدريس وذا النفل كل من الصابرين واتكر إسماعيل واليسع وذا النفل وكل من الأخيار	15
	وظائف الرياح Roles of the winds	يرسل الرياح بشرا ريح عاصف قاصفا من الريح وتصريف الرياح، الريح لواقع، ريحا صرصر	26
	حركة الأجرام السماوية motion of celestial objects	كل في فلك يسبحون والنجم اذا هوى	25
	من هم المحسنون who are the benefactors	اولئك هم المحسنون، احسانا	45
	لماذا خلق الله القمر Why did God create the moon?	لتعلموا عدد السنين والقمر ضياء	16

The performance of the proposed model was assessed in comparison to three other models: an averaging model called Doc2Vec, a predictive model known as Universal Sentence Encoder (USE), and a transformer-based model called AraT5. These models represent the chronological progression of embedding techniques for Arabic text, ranging from the less advanced technique, Doc2Vec, to the more advanced AraT5. Subsequently, the precision, recall and the harmonic mean of precision, recall, F-Score, were computed for each semantic level.

In Figure 2 the precision scores for different levels of semantics are compared across various embedding models. Notably, the GPT Embedding demonstrates superior performance, particularly for queries with higher-level semantics.

Figure 3 showcases the recall scores for each semantic level across different embedding models. Here, the GPT Embedding stands out, demonstrating its remarkable ability for high recall and outperforming other methods as the semantic complexity of the queries increases.

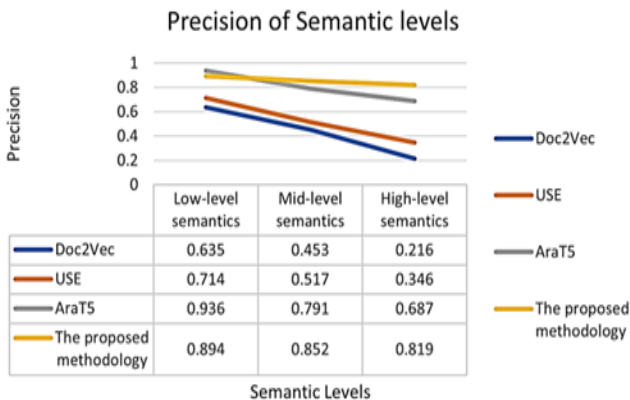


Figure 2. The precision of each level of semantics versus the embedding models.

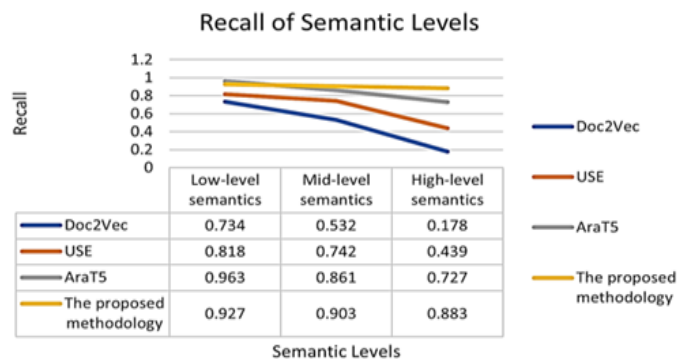


Figure 3. The recall of each level of semantics versus the embedding models.

Finally, Figure 4 illustrates the F-Score for each semantic level in relation to the different embedding models. As the harmonic mean of precision and recall, the F-Score highlights that the GPT Embedding performs exceptionally well, especially as the semantic complexity of the queries intensifies.

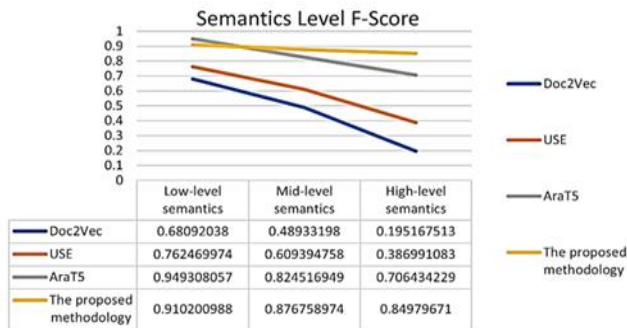


Figure 4. The F-Score of each level of semantics versus the embedding models.

One of the key findings is that the effectiveness of the proposed approach remains consistent even as the semantic complexity increases. This outcome suggests that the LLM embedding method can effectively capture profound semantic connections and relationships. It also provides insights into the methodology used for this embedding, which relies on a large corpus and a stack of transformers with attention mechanisms capable of capturing contextual dependencies between concepts.

The second noteworthy observation is that in low-level semantic searches, the state-of-the-art transformer AraT5 demonstrates better performance compared to LLM embedding. This suggests that LLM might not have been sufficiently trained on an extensive Arabic text corpus and may require additional efforts for fine-tuning.

5.1. Analysis of the Results

Doc2Vec, an extension of Word2Vec, relies on averaging word embeddings to create an embedding for an entire document. Consequently, it may fall short in capturing nuanced relationships with far-reaching contextual and indirect connections.

On the contrary, the USE excels in generating embeddings for short sentences. It performs well in capturing semantic relations within short distances, making it suitable for contexts where relations between entities are recognizable in brief text. Consequently, it might not perform optimally in tasks requiring a high level of semantic discovery.

In contrast, AraT5, a transformer-based model, is trained on a substantial volume of Arabic texts. Utilizing an attention mechanism, AraT5 can effectively capture relations between semantic entities even in distant contexts.

However, when compared to generative pre-trained models like GPT, GPT stands out for its exceptional contextual understanding, achieved through a larger

training set and an autoregressive nature that allows it to capture longer-range dependencies in text. Additionally, GPT is distinguished by its substantial parameter size, enabling it to capture a broader range of semantic dependencies.

In summary, while Doc2Vec may struggle with complex contextual relationships, USE excels in short-context scenarios, AraT5 proves effective in capturing distant semantic relations, and GPT, with its exceptional contextual understanding and massive parameter size, stands out in capturing intricate semantic dependencies. The choice among these models should be driven by the specific requirements of the task at hand.

5.2. Concluding Remarks

The superior performance of AraT5 over GPT in low-level semantic similarity tasks for Arabic text can be attributed to several factors. Firstly, AraT5 is specifically trained on tasks related to low-level semantic similarity in Arabic text, enabling it to demonstrate enhanced proficiency in this particular domain. Secondly, AraT5 is meticulously designed with a focus on Arabic text, potentially resulting in a more effective capture of language-specific features. Lastly, AraT5 has been dedicatedly trained on a diverse and extensive Arabic text corpus, contributing to its comprehensive understanding of the linguistic nuances present in the data.

The study has certain limitations, including:

- The study focuses only on a non-diacritics Quranic text and specific large language models, which means that the findings may not be applicable to other Quranic collections or different large language models.
- The accuracy of the semantic search results is dependent on the quality of the pre-trained GPT model and its fine-tuning on Quranic texts. Consequently, the results may not fully demonstrate the capabilities of large language models for semantic search tasks.
- The assessment of the semantic search results could be subjective and influenced by individual preferences or interpretations.

The study encounters several challenges, which include:

- The Quranic texts possess intricate and diverse linguistic and semantic elements that present obstacles for natural language processing techniques. Consequently, effectively capturing and representing the meaning and context of Quranic texts using large language models may prove challenging.
- The semantic search tasks necessitate a profound comprehension of Quranic texts and their connection to Islamic beliefs, practices, and historical context. Consequently, formulating appropriate queries and evaluating their relevance using large language models may pose difficulties.

To enhance the performance of Arabic GPT Embedding for a basic semantic similarity task with the aim of surpassing AraT5, it is recommended to undertake a series of tasks:

1. Fine-Tuning GPT: fine-tune the GPT model on a dataset explicitly crafted for low-level Arabic semantic similarity, encompassing aspects like synonyms and lemma relations.
2. Diverse Training Data: ensure that the training dataset used for GPT encompasses a broad spectrum of contexts and linguistic variations. This diversification can enhance the model's adaptability to different semantic scenarios.
3. Transfer Learning from AraT5: leverage transfer learning from AraT5 by initializing the GPT model with weights pretrained on AraT5. This transfer can contribute to a deeper understanding of Arabic semantics within the GPT framework.

6. Conclusions

This research explores the use of large language models for embedding and semantic search of Quranic texts. Traditional specific-model based embedding methods face challenges due to the complex linguistic and semantic features of Quranic texts. The application of these methods for information retrieval and semantic search in Quranic texts are limited in scope and effectiveness, especially for retrieving high-level semantics queries. To address this, the research uses a publicly available Quranic corpus and the embedding APIs of GPT 3.5 model to compare their performance with the baseline methods. The study aims to demonstrate the feasibility and effectiveness of large language models for semantic search of Quranic texts and identify areas for improvement. This research has implications for semantic search, Islamic studies, and digital humanities, enhancing accessibility and usability of Quranic texts and providing insights into their linguistic and semantic aspects. The combination of semantic search and large language models has transformative potential for information retrieval and natural language processing tasks. Results demonstrate the ability of LLM models to embed and retrieve appropriate answers and it could be enhanced using finer tuning for Arabic texts and more feeding with verses explanation in order to augment information and knowledge that are related to these verses and then increase the performance of our approach.

References

- [1] Abuzayed A. and Al-Khalifa H., "BERT for Arabic Topic Modeling: An Experimental Study on BERTopic Technique," *Procedia Computer Science*, vol. 189, pp. 191-194, 2021. <https://doi.org/10.1016/j.procs.2021.05.096>
- [2] Afzal H. and Mukhtar T., "Semantically Enhanced Concept Search of the Holy Quran: Qur'anic English WordNet," *Arabian Journal for Science and Engineering*, vol. 44, pp. 3953-3966, 2019. <https://doi.org/10.1007/s13369-018-03709-2>
- [3] Alhawarat M., Hegazi M., and Hilal A., "Processing the Text of the Holy Quran: A Text Mining Study," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, pp. 262-267, 2015. DOI:10.13140/RG.2.1.3025.3608
- [4] Alqahtani M. and Atwell E., "Arabic Quranic Search Tool Based on Ontology," *Natural Language Processing and Information Systems*, pp. 478-485, 2016. https://doi.org/10.1007/978-3-319-41754-7_52
- [5] Alqahtani M. and Atwell E., "Evaluation Criteria for Computational Quran Search," *International Journal on Islamic Applications in Computer Science and Technology*, vol. 5, no. 1, pp. 12-22, 2017. https://doi.org/10.1007/978-3-319-41754-7_52
- [6] Alshammeri M., Atwell E., and Alsalka M., "Detecting Semantic-based Similarity between Verses of the Quran with Doc2vec," *Procedia Computer Science*, vol. 189, pp. 351-358, 2021. <https://doi.org/10.1016/j.procs.2021.05.104>
- [7] Beirade F., Azzoune H., and Zegour D., "Semantic Query for Quranic Ontology," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 6, pp. 753-760, 2021. DOI: 10.1016/j.jksuci.2019.04.005
- [8] Ishkewy H. and Harb H., "ISWSE: Islamic Semantic Web Search Engine," *International Journal of Computer Applications*, vol. 112, no. 5, pp. 37-34, 2015. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2824d65e2716fe759a4d4c9a47ebf11ca09b5c2c>
- [9] Karim O., QuranAnalysis: A Semantic Search and Intelligence System for the Quran, Master Theses, University of LEEDS, 2015. DOI:10.13140/RG.2.1.3165.7681
- [10] Khan H., Saqlain S., Shoaib M., and Sher M., "Ontology-Based Semantic Search in Holy Quran," *International Journal of Future Computer and Communication*, vol. 2, no. 6, pp. 570-575, 2013. DOI:10.7763/IJFCC.2013.V2.229
- [11] Mohamed E. and Shokry E., "QSST: A Quranic Semantic Search Tool Based on Word Embedding," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 934-945, 2022. <https://doi.org/10.1016/j.jksuci.2020.01.004>
- [12] Muennighoff N., "SGPT: GPT Sentence Embeddings for Semantic Search," *arXiv Preprint*, arXiv:2202.08904v5, 2022. <https://doi.org/10.48550/arXiv.2202.08904>

- [13] Nuhu Y., Yunus M., and Wahid N., "Query Expansion Based on Explicit-Relevant Feedback and Synonyms for English Quran Translation Information Retrieval," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, pp. 227-234, 2019. DOI:10.14569/IJACSA.2019.0100530
- [14] Nuhu Y., Yunus M., and Wahid N., "Query Expansion for Quran French Text Retrieval Using Semantic Search," *Journal of Soft Computing and Data Mining*, vol. 1, no. 2, pp. 26-30, 2020. <https://doi.org/10.30880/jscdm.2020.01.02.003>
- [15] Nuhu Y., Yunus M., Wahid N., Nawi N., and Samsudin A., "Query Expansion Method for Quran Search Using Semantic Search and Lucene Ranking," *Journal of Engineering Science and Technology*, vol. 15, no. 1, pp. 675-692, 2020.
- [16] Petroni F., Rocktäschel T., Lewis P., Bakhtin A., and Wu Y., "Language Models as Knowledge Bases," *arXiv Preprint*, arXiv:1909.01066v2, 2019. <https://doi.org/10.48550/arXiv.1909.01066>
- [17] Saeed S., Haider S., and Rajput Q., "On Finding Similar Verses from the Holy Quran Using Word Embeddings," in *Proceedings of the International Conference on Emerging Trends in Smart Technologies*, Karachi, pp. 1-6, 2020. doi:10.1109/ICETST49965.2020.9080691.
- [18] Safee M., Afifi M., Pitchay S., and Ridzuan F., "A Systematic Review Analysis for Quran Verses Retrieval," *Journal of Engineering and Applied Sciences*, vol. 11, no. 3, pp. 629-634, 2016. DOI:10.3923/jeasci.2016.629.634
- [19] Shoaib M., Yasin M., Hikmat U., Saeed M., and Khiyal M., "Relational WordNet model for semantic search in Holy Quran," in *Proceedings of the International Conference on Emerging Technologies*, Islamabad, pp. 29-34, 2009. Doi:10.1109/ICET.2009.5353208.
- [20] Sultana Z., Rahman M., Uddin M., and Arfat M., "Developing a Semantic Search Method for Retrieving Food Related Verses and Concepts from Holy Quran Using Ontology," in *Proceedings of the 5th International Conference on Electrical Engineering and Information Communication Technology*, Dhaka, pp. 1-6, 2021. doi:10.1109/ICEEICT53905.2021.9667817
- [21] Sun W., Yan L., Ma X., Wang S., and Ren P., Chen Z., Yin D., and Ren Z., "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent," *arXiv Preprint*, arXiv:2304.09542v1, 2023. <https://doi.org/10.48550/arXiv.2304.09542>
- [22] Ta'a A., Abed A., Ali B., and Ahmad M., "Ontology-Based Approach for Knowledge Retrieval in Al-Quran Holy Book," *International Journal of Computational Engineering Science*, vol. 6, pp. 8-15, 2016. <http://dx.doi.org/10.4314/jfas.v9i5s.57>
- [23] Tarawneh M. and Al-Shawakfa A., "A Hybrid Approach for Indexing and Searching the Holy Quran," *Jordanian Journal of Computers and Information Technology*, vol. 1, no. 1, pp. 42-51, 2015. Doi: 10.5455/jjcit.71-1445981961
- [24] Utomo F., Suryana N., and Azmi M., "Question Answering Systems on Holy Quran: A Review of Existing Frameworks, Approaches, Algorithms and Research Issues," *Journal of Physics: Conference Series*, vol. 1501, no. 1, pp. 012022. 2020. DOI:10.1088/1742-6596/1501/1/012022
- [25] Zouaoui S. and Rezeg K., "A Novel Quranic Search Engine Using an Ontology-Based Semantic Indexing," *Arabian Journal for Science and Engineering*, vol. 46, pp. 3653-3674, 2021. <https://doi.org/10.1007/s13369-020-05082-5>



Mohammed Alqarni is an associate professor at the Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah. He was awarded a Ph.D. in Computer Science from the Department of Computing and Software, McMaster University, Hamilton, Canada, in 2016 and an MSc in Computational Sciences from the Department of Mathematics and Computer Science, Laurentian University, Sudbury, Canada, in 2012. He researches various topics, including Cyber-Physical Systems, Security by Design, and Artificial Intelligence.

Appendix (1)

In Figure 5 an example of using GPT for refinement and generate answers based on the ranked and selected verses. This example demonstrates the power of GPT in generating an answer that is based on focusing the search to the ranked and selected verses:

الآيات التي تتحدث عن قارون وثرائه على النحو التالي:

1. آية 76 من سورة القصص: "إن قارون كان من قوم موسى فيغى عليهم وآتيناه من الكنوز ما إن مفاتحه لتنوء بالعصبة أولي القوة إذ قال له قومه لا تفرح إن الله لا يحب الفرحين".

- في هذه الآية، يُشير ذكر قارون إلى شخص من قوم موسى. وقد منّ الله عليه بثروة كبيرة من الكنوز والمال.

- عندما فتح قارون خزانته، كانت كمية الثروة التي يمتلكها هائلة بحيث كادت أن تثقل الأشياء.

- قومه نصحوه بعدم الفرح والتباهي بثروته، حيث قالوا له: "لا تفرح إن الله لا يحب الفرحين".

- هذا يشير إلى أن الغنى والثراء يجب أن يتمتع بهما الإنسان بتواضع وبدون تكبر وتباهي.

2. آية 79 من سورة القصص: "فخرج على قومه في زينته قال الذين يريدون الحياة الدنيا يا ليت لنا مثل ما أوتي قارون إنه لذو حظ عظيم".

- في هذه الآية، قارون خرج إلى قومه متباهياً بثرائه وزينته.

- بعض الناس الذين يتمنون الحياة الدنيا والثروة أعربوا عن تمنيتهم أن يحظوا بمثل ما أوتي قارون من ثروة، واعتبروه محظوظاً بسبب ثرائه.

- هذا يظهر الاهتمام بالثروة والحياة الدنيا في قلوب بعض الأشخاص ورغبتهم في الحصول على حظ مماثل.

بشكل عام، يُظهر القرآن هذه الآيات للتأمل في موضوع الثراء والغنى، وكيفية تصرف الأفراد تجاههما وتأثيرهما على سلوكهم وعلاقتهم بالله.

- Figure 5. Query response for "ما الدليل على غنى قارون".