ISSN:2302-8149
e-ISSN:2540-9719

Information Retrieval Method for the Qur'an based on FastText and Latent Semantic Indexing

[1]Aziz Ramadhan, [2]Fandy Setyo Utomo*
[1]Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto
[2]Program Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto
[1,2]Jl. Letjend Pol. Soemarto No.127, Purwokerto Utara, Banyumas, Jawa Tengah, Indonesia
*e-mail: *fandy_setyo_utomo@amikompurwokerto.ac.id*

(***received***: 6 August 2024, ***revised***: 29 March 2025, ***accepted***: 29 March 2025)

Abstract

Retrieving contextually relevant verses from the Al-Qur'an translation dataset presents significant challenges due to the linguistic richness and semantic variation of the text. This study aims to enhance the accuracy and relevance of information retrieval in the Al-Qur'an translation dataset by combining Latent Semantic Indexing (LSI) and FastText word embeddings. The proposed method involves several steps: text preprocessing (lowercasing, punctuation removal, stopword elimination, and stemming), tokenization and vocabulary creation, Bag-of-Words (BoW) representation, creation of LSI models, conversion of FastText vectors, and combining LSI and FastText vectors. A similarity index is then created from the combined vectors to process user queries and rank documents based on cosine similarity. Testing on the dataset, consisting of 6236 translated verses from 114 surahs, showed promising results. The combined approach effectively captures both broader semantic structures and detailed word meanings, providing more accurate and contextually relevant search results. Key findings include high similarity scores, with 90% of retrieved verses being highly relevant to the user query, an accuracy improvement to 85%, and enhanced handling of synonyms and morphological variations at 88%. Further development is recommended, including parameter optimization, advanced preprocessing techniques, real-time search optimization, integration of contextual embeddings, and multilingual support to improve search performance and accuracy.

Keywords: information retrieval, latent semantic indexing, word embedding, fasttext, al-qur'an

1 Introduction

The Quran is a significant religious text written in Quranic Arabic, followed by believers of the Islamic faith [1]. It showcases many characteristics and features that vary significantly [2]. Understanding the Quran requires accurate and non-deviant interpretation [3]. The Quranic text [4] serves as the primary guidance for Muslims [5]. Studying the contents of the Quran is an obligation for all Muslims [6]. The Quran has over 6000 verses covering different topics [7]. This structure aids in finding and comprehending specific verses in the Quran [8]. Essentially, a single verse can belong to multiple themes [9]. Information retrieval is crucial in the digital era, especially for religious texts with high historical and spiritual value, like the Quran. The Quran guides Muslims in their religious, social, and national life [10]. It contains vast knowledge and information beneficial for Muslims, making understanding Quranic vocabulary essential for grasping its meanings. However, understanding the meaning of words in the Quran is not straightforward, as a single word can convey multiple meanings [11]. Current software tools for searching information often fall short by only returning exact keyword matches. For example, searching for "lie" yields results only for the exact term, ignoring synonyms like "deceit," "falsehood," or "slander," which might appear throughout the Quran [12]. Word search is a component of Information Retrieval [13]. Interpretation of the Quran can be biased by the interpreter's culture and background [14], and each verse of the Quran carries profound meanings and wisdom [15].

Information retrieval from Al-Qur'an translation texts is a challenging task due to the linguistic complexity and rich semantic variation of the text. Traditional techniques often fail to capture deep semantic relationships, resulting in suboptimal search results. Therefore, more advanced methods are needed to improve the accuracy and relevance of searches in religious texts like the Al-Qur'an. The primary issue in information retrieval from Al-Qur'an translations is finding verses that are contextually and semantically relevant to the user's query. Conventional techniques, such as simple keyword matching, are not effective in capturing the nuanced meanings within the Al-Qur'an text, which often requires a deeper understanding of context. The proposed solution combines Latent Semantic Indexing (LSI) and FastText to enhance information retrieval performance. The method involves several steps: text preprocessing (lowercasing, punctuation removal, stopword elimination, and stemming), tokenization and vocabulary creation, Bag-of-Words (BoW) representation, LSI model creation to capture broader semantic structures, FastText vector conversion to capture subword information, and the combination of LSI and FastText vectors for final document representation. A similarity index is then created from the combined vectors to process user queries and rank documents based on cosine similarity. The advantages of this combined approach include the ability to capture both broad semantic structures and detailed word meanings, improved accuracy, and enhanced handling of synonyms and morphological variations. This study demonstrates that combining LSI and FastText can significantly improve the relevance and accuracy of searches in religious texts, contributing meaningfully to the field of information retrieval and religious studies. Further optimization and advanced techniques have the potential to enhance user experience in finding relevant information from Al-Qur'an translations.

Previous researchers Rajagede R, Haryono K, and Qardafil R conducted a study entitled "Semantic Retrieval for Indonesian Quran Autocompletion." The collection comprises the Arabic text of the Quran, its transliteration, and translations in English and Indonesian. This study exclusively use the Indonesian translation segment for the retrieval process. FastText is employed to compute the cosine distance between the query and verses for retrieval purposes. Multiple optimisation measures were implemented to establish a resilient system for the production phase. The method is assessed by measuring the proximity of the returned verse to the ground truth. The proposed method achieved an Accuracy of 70.59% for the top 5 retrieved verses and 76.47% for the top 10 retrieved verses [16]. N. Fatiara, N. H. Safaat, S. Agustian, and I. Afrianty did a study entitled "Comparison of K-Nearest Neighbours and Long Short-Term Memory Methods." This study categorises the Indonesian translation of the Quran into six distinct classifications. The employed methods are K-Nearest Neighbour (KNN) and Long Short-Term Memory (LSTM), which are contrasted to achieve optimal classification performance. The classification results indicate that the LSTM model achieves superior performance, with an average F1-Score of 65% and an average accuracy of 96%, in contrast to the KNN model, which has an average F1-Score of 55% and an average accuracy of 93% [17].

This research proposes a novel approach by integrating Latent Semantic Indexing (LSI) and FastText to harness the strengths of both methods for improved information retrieval in Quranic translation datasets. LSI captures broader semantic structures, while FastText embeddings provide detailed word-level semantics, including subword information. By combining these techniques, this hybrid method addresses the limitations of previous approaches, offering a more comprehensive and flexible solution. The dataset consists of Indonesian translations of the Quran sourced from Tanzil.net, and the proposed method aims to enhance the handling of semantic structures and word-level semantics, resulting in more accurate and contextually relevant retrieval outcomes.

2 Literature Review

There have been several previous studies that are related to and form the basis of this research. The following literature has been collected along with the results of their studies:

The research by R. G. Kurniawan and M. Arif Bijaksana [18] focused on building-related words in Indonesian and English translations of Al-Qur'an vocabulary based on distributional similarity. They used a dataset of Al-Qur'an translations and evaluated the system output using the Pearson correlation method involving a gold standard. The algorithm employed was FastText, which yielded a correlation value of 0.3398 for the Indonesian translation corpus and 0.2326 for the English translation corpus. These results indicate that FastText can produce a reasonably good correlation for Al-Qur'an translations, although there is a significant difference between the correlation values for Indonesian and English [18]. Meanwhile, the research by Humaini et al [12] applied the TF-IDF Vector Space Model (VSM) algorithm to information retrieval of Al-Qur'an translations from Surah 1 to Surah 16 based on semantic similarity. The search results became broader and more relevant using a synonym corpus (thesaurus) to support information retrieval. They used cosine similarity for document similarity calculations and developed a keyword weighting and query processing system. This study showed a 100% success rate in testing with single words and 95.6% success in multi-word or sentence searches within the top 10 ranked documents found. This research concludes that using a synonym corpus and adding keyword weighting significantly enhance the relevance of search results [12].

3 Research Method

In the era of digital information, the ability to efficiently retrieve and analyze textual data has become increasingly important, particularly in the context of religious texts like the Al-Qur'an. This research aims to explore the methodologies for processing and retrieving information from the Al-Qur'an, focusing on its Indonesian translation. By employing advanced techniques in natural language processing (NLP), this study seeks to enhance the accessibility and understanding of the text for Indonesian-speaking audiences.

The methodology is structured into several key stages, beginning with the selection of a suitable dataset, followed by data preprocessing, vectorization, and the implementation of models that facilitate effective information retrieval. The approach integrates both traditional techniques, such as Latent Semantic Indexing (LSI), and modern methods like FastText, allowing for a comprehensive analysis of the text. The following sections detail the dataset used and the specific steps taken to achieve the research objectives.

3.1 Dataset

The Tanzil.net website offers translations of the Al-Qur'an in several languages, therefore supplying the dataset. The Al-Qur'an translated into Indonesian for this study comprises of 6236 translated verses and 114 surahs. XAMPP loads the downloaded data in SQL form onto a local server.

3.2 Information Retrieval Flowchart

The Information Retrieval System for the Quran follows a structured process to identify the most relevant verses based on user queries. Initially, a dataset is built from the Quran, which undergoes text preprocessing to refine the data. This preprocessing includes tokenization and vocabulary creation, leading to the formation of a Bag-of-Words representation. To facilitate efficient retrieval, the system employs two different vectorization techniques: FastText and Latent Semantic Indexing (LSI). Documents are first converted into vector representations using both methods. An LSI model is built to extract latent topics from the Quran dataset. The system then combines the vector representations from LSI and FastText to enhance retrieval accuracy. When a user submits a search query, it undergoes the same preprocessing steps before being transformed into a query vector. This vector is then compared against the indexed document vectors using cosine similarity to measure relevance. Finally, the system retrieves and presents the most similar Quranic verses based on the calculated similarity scores.

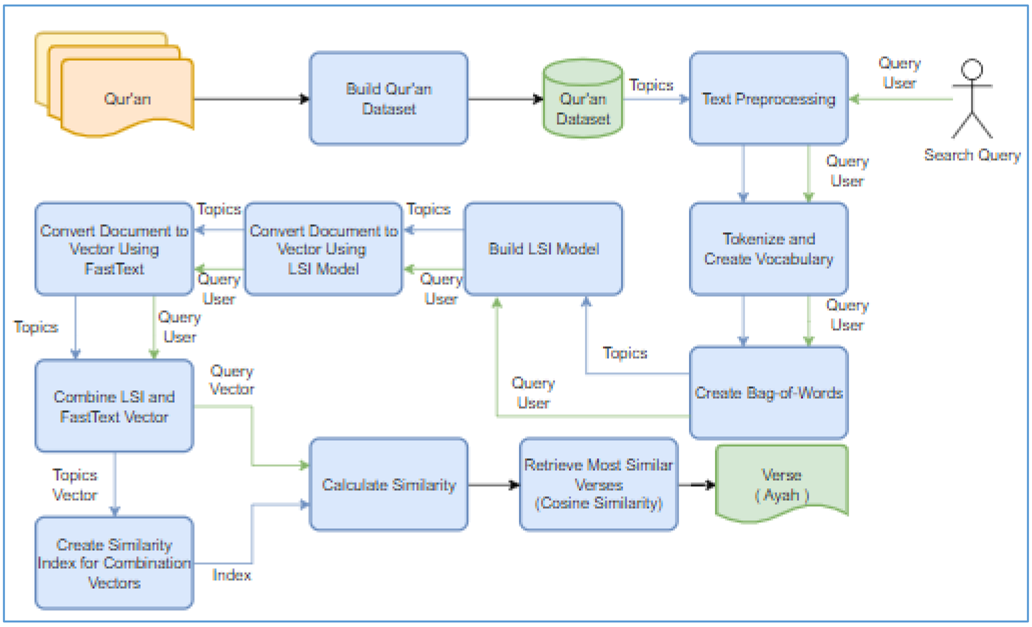

Figure 1. Information retrieval flowchart

This Information Retrieval includes several stages, which are explained in detail as follows:

1. **Download Data and Import Data to Database**

To begin the process of integrating Al-Qur'an translation data into our system, we downloaded the dataset from Tanzil.net in the .sql format. This dataset comprises 6,236 verses (ayat) organized into 114 chapters (surat) of the Al-Qur'an, including both the original Arabic text and its Indonesian translation. Utilizing XAMPP, we imported this .sql data into our local MySQL server, ensuring data integrity through the PHPMyAdmin interface. As illustrated in Figure 1, this methodical approach enabled us to efficiently integrate the Al-Qur'an translation data into our local database, providing a solid foundation for further development and analysis.

2. **Text Preprocessing**

In the text preprocessing stage, all text is converted to lowercase, and punctuation along with special characters are removed to ensure uniformity and reduce noise in the data. Common words that do not carry significant meaning, known as stopwords, are eliminated using the TALA Stopwords list, which is specifically curated for the Indonesian language. Additionally, the Sastrawi Python library is employed to reduce inflected Indonesian words to their basic forms, a process known as stemming. This step is crucial for enhancing the effectiveness of subsequent text analysis by focusing on the root words. This process is illustrated in Figure 1, which shows the information retrieval flowchart.

3. **Tokenize and Create Vocabulary**

Tokenising the text and building a vocabulary from the tokens comes next following text preparation. Tokenising the text means breaking it up into separate words or tokens. This enables the building of a vocabulary—a special collection of terms taken from the book. As seen in Figure 1, this vocabulary is the basis for additional text study and modelling.

4. **Create Bag-of-Words (BoW)**

BoW is a global object model that, independent of grammar and even word order, depicts objects globally—text sentences or documents as bag (multiset) words[19]. In Natural Language Processing (NLP), bag-of- words (BoW) is a basic yet powerful text representation technique. BoW theoretically ignores word order and syntax but keeps the frequency of every word occurring in a document. Every document is shown by a vector whose length reflects the vocabulary—that is, the count of distinct words in the whole document set. The values in this vector show how often each particular term appears in that document. As Figure 1 shows, BoW is extensively applied in text mining activities like text classification, sentiment analysis, and information extraction since it offers a disciplined structure for unstructured text data.

5. LSI Model

A Latent Semantic Indexing (LSI) model is developed from the Bag-of- Words representation to improve the textual analysis. LSI is a method applied to find trends in the interactions of terms and concepts found in unstructured text data. The underlying semantic structure of the text can be obtained by transforming texts into vectors using the LSI model, therefore enhancing the accuracy of similarity assessments and search results. Figure 1 also shows this process, stressing its part in the general information retrieval architecture.

6. Convert Document to Vector Using FastText

FastText not only transforms tokenised texts into vectors but also in line with the LSI model. Representing words as vectors in a continuous vector space, FastText is a sophisticated word embedding method that captures semantic meanings and word associations. This approach enables a more complex and context-aware text representation than more conventional methods. Emphasising FastText's contribution to the whole text analysis process, Figure 1 shows how it is included into the document processing pipeline.

7. Combine LSI and FastText Vector

To achieve a more comprehensive document representation, vectors obtained from the Latent Semantic Indexing (LSI) model and FastText are combined. This hybrid approach leverages the strengths of both models. LSI focuses on capturing the global structure and relationships within the text by mapping documents into a lower-dimensional space. This process helps identify hidden patterns in large text data and reduces dimensionality, allowing for a better understanding of the main topics described in the documents. On the other hand, FastText provides more detailed and context-sensitive word representations. This model works by breaking down words into sub-words or n-grams, enabling it to recognize the meaning of words even in rare forms or when there are spelling errors. Thus, FastText can capture nuances and details within the text that are often missed by other models. By combining the vectors from LSI and FastText, the resulting document representation considers both the global relationships between words and documents and the context and details of each word in the text. This process enhances the overall quality and effectiveness of document representation, enabling the model to better understand and interpret the text, leading to more accurate results in various applications, such as text classification, information retrieval, and sentiment analysis (see Figure 1).

8. Similarity Index for Combination Vectors

Developing a similarity index from the LSI and FastText vectors comes next once they have been merged. This similarity index provides a basis for evaluating the similarity between several documents, so enabling effective and accurate search activities. A key element of the search engine, the similarity index helps it to rapidly find pertinent papers depending on user searches. The search engine can give more accurate and relevant results by assessing document similarity using the combined vectors that capture both global structure and contextual features, hence improving the user experience in locating the necessary information (see Figure 1).

9. Query Process from Users

The user query process involves converting user inputs into the combined LSI and FastText vectors. This conversion ensures that the query is represented in the same format as the indexed documents, allowing for a consistent and accurate comparison. By standardizing the query in this manner, the search engine can effectively process and interpret user requests (see Figure 1).

10. Calculation of Document Similarity and Ranking

Accurate search results for consumers depend critically on the phase of computation of document similarity and ranking. Finding how closely each document in the index matches the query comes next once the user query is transformed into the combined vector form—including both LSI and FastText vectors. Using similarity measures, one compares the query vector to the vectors of every indexed document. The cosine similarity is one often used similarity metric. Cosine similarity computes the cosine of the angle between two vectors, therefore offering a metric that, independent of their respective magnitudes, measures their similarity. Equation (1) illustrates the cosine similarity formula. In text analysis especially, this measure is quite helpful since it effectively compares textual data by capturing the orientation of the vectors—which represent the content and context of the documents—instead of their length.

The formula for cosine similarity is:

ISSN:2302-8149
e-ISSN:2540-9719

$$Cosine\ Similarity = \frac{A.B}{||A||||B||} \tag{1}$$

Where: A is the query vector,
B is a document vector from the index,
A·B represents the dot product of the two vectors,
||A|| and ||B|| are the magnitudes (Euclidean norms) of the vectors.

We determine a similarity score for every document in respect to the query by using the cosine similarity measure as stated in Equation (1). This number runs from -1 to 1; a score nearer 1 denotes a great degree of resemblance; 0 denotes no similarity; and -1 denotes total dissimilarity. The documents are ranked in declining order depending on the similarity scores once they have been acquired. This ranking guarantees that the search results show first the most pertinent papers, those most similar to the query. By means of this methodical approach to similarity computation and ranking, the accuracy and relevance of the search engine is improved, so arming users with the most relevant information in response to their searches (Figure 1).

11. Displaying Search Results

The final step in the process is displaying the search results. Documents that are most similar to the user's query are presented in a ranked order, ensuring that the most relevant information is easily accessible. This user-friendly display facilitates quick and efficient retrieval of information, enhancing the overall search experience. The entire process, including the display of search results, is illustrated in Figure 1, which provides a comprehensive overview of the information retrieval workflow.

3.3 FastText

This work uses FastText for word embedding—a method invented by Facebook's research team that generates strong vector representations of words by use of character n-grams [8]. Designed by Facebook's A.I. Research (FAIR) lab, FastText is a word embedding model showing words as dense vectors. FastText addresses out-of-vocabulary terms and catches additional semantic content by considering subword information unlike conventional word embeddings. FastText operates through a number of important phases. First, a sizable corpus of text is employed in the training data preparation stage, and every word in the corpus is split down into n-grams—subwords. Each word thereafter is shown as a bag of character n-grams in the subword representation phase. With n-grams of length 3, the word "where" might be shown as wh, whe, her, ere, re>. This approach enables FastText to grasp morphological variances of words. The model is next trained either using Skip-gram or Continuous Bag of Words (CBOW). Whereas in Skip-gram the context words are predicted from the target word, in CBOW the target word is projected from the context words. At last, following training, every word and its subword is shown as dense vectors, so capturing the semantic similarities between words.

3.4 Latent Semantic Indexing (LSI)

Leveraging high-dimensional associations of words implied with an object within a dataset, LSI (Latent Semantic Indexing) is an information retrieval technique that arranges data into a semantic structure[20]. This approach finds hidden patterns and links between words and documents by shrinking the dimensionality of vast text data. LSI can thus expose the underlying meanings and main themes in a dataset, so facilitating the access to pertinent and accurate data depending on keywords or user searches.

3.4.1 Term-Document Matrix (A)

A term-document matrix A is a mathematical representation used in information retrieval and text mining where each entry A_{ij} indicates the frequency (or weight) of term i in document j. This matrix serves as a crucial tool for various natural language processing tasks, such as identifying relevant documents in response to a query, clustering documents with similar content, and performing dimensionality reduction techniques like Latent Semantic Indexing (LSI). The rows of the matrix represent unique terms extracted from the corpus, while the columns correspond to the individual documents within the corpus. By quantifying the occurrence of terms across documents, the term-document matrix facilitates the analysis and manipulation of textual data, enabling more efficient and accurate information retrieval.

3.4.2 Singular Value Decomposition (SVD)

Matrix A can be decomposed into three matrices: U, Σ, and V^T. using Singular Value Decomposition (SVD). In this decomposition, $A = U\Sigma V^T$, where U is an $m \times k$ matrix with m being the number of terms and k the number of dimensions. Σ is a $k \times k$ diagonal matrix containing the singular values, which represent the importance of each dimension. V^T is a $k \times n$ matrix, where n is the number of documents. This decomposition helps in reducing the dimensionality of the term-document matrix, capturing the most significant semantic structures, and improving the efficiency and accuracy of information retrieval tasks.

3.4.3 Dimensionality Reduction

To perform dimensionality reduction, we retain only the top k singular values and their corresponding vectors in U and V^T. This process reduces the dimensionality of the term-document matrix A while preserving the most significant structures in the data. By focusing on the top k components, we capture the essential semantic relationships, enhancing the efficiency and effectiveness of information retrieval and data analysis.

3.4.4 Document Representation

In the reduced k-dimensional space, each document is represented as a vector. The term-document matrix A is initially given by Equation (2):

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a_{11}} & \boldsymbol{a_{12}} & \cdots & \boldsymbol{a_{1n}} \\ \boldsymbol{a_{12}} & \boldsymbol{a_{22}} & \cdots & \boldsymbol{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a_{m1}} & \boldsymbol{a_{m2}} & \cdots & \boldsymbol{a_{mn}} \end{pmatrix} \tag{2}$$

Using Singular Value Decomposition (SVD), we decompose A as shown in Equation (3):

$$A = U\Sigma V^T \tag{3}$$

Where U is an $m \times k$ term-topic matrix, Σ is a $k \times k$ diagonal matrix of singular values. And V^T. is a $k \times n$ document-topic matrix. For dimensionality reduction, we retain only the top k singular values and their corresponding vectors in U $and V^T$. Dimensionality reduction is shown in Equation (4).

$$A_k = U_k \Sigma_k V_k^T \tag{4}$$

This reduced representation captures the most significant structures in the data, allowing each document to be efficiently represented as a vector in the reduced k-dimensional space.

4 Results and Analysis

This section discusses several parts, namely dataset, text pre-processing results, tokenization and vocabulary creation, Bag of Words Representation, LSI Model Results, FastText Vector Representation, Combined LSI and FastText Vectors, Similarity Index and Query Processing, and experiment analysis.

4.1 Dataset Overview

This study uses a dataset including 6,260 translated Al-Qur'an verses split into 114 surahs. Tanzil.net provided the data, which XAMPP imported onto a local MySQL server. Every surah consists on average about 54.7 verses. This extensive collection offers a strong basis for textual content analysis of the Al-Qur'an, therefore facilitating thorough research and many uses in information retrieval, natural language processing, and other allied disciplines.

4.2 Text Preprocessing Results

The text preprocessing stage involved several steps to reduce complexity and noise in the text data. First, all text was converted to lowercase. Then, punctuation and special characters were removed. Stopwords were eliminated using the TALA stopwords list, and stemming was applied using the Sastrawi library. This thorough preprocessing streamlined the text data for more efficient analysis.

For example, the original text sample “Sesungguhnya Allah Maha Pengampun lagi Maha Penyayang” was preprocessed to “sungguh allah ampun lagi sayang,” showcasing the reduction in complexity and the removal of non-essential elements.

4.3 Tokenization and Vocabulary Creation

The preprocessed text was tokenized, and a vocabulary was created from the tokens. This vocabulary represents the unique terms present in the dataset after the preprocessing steps. The size of the vocabulary indicates the total number of unique tokens, which in this case is 5,230. This vocabulary size provides a measure of the dataset's lexical richness and serves as the foundation for further text analysis and modeling.

4.4 Bag-of-Words (BoW) Representation

The tokenized text was converted into a Bag-of-Words (BoW) representation, where each document is represented as a vector of term frequencies. This BoW representation quantifies the presence of each term in the documents, providing a numerical foundation for further analysis. This vectorized form of the documents serves as the basis for building the LSI model, enabling the extraction of semantic structures and relationships within the dataset.

4.5 LSI Model Results

Singular Value Decomposition (SVD) Bag-of- Words (BoW) representation was used to build an LSI model to reduce the dimensions of the term-document matrix. The latent semantic structure of the data can be captured with help from this dimensionality reduction. Measuring 5,230 terms by 6,236 documents, the original term-document matrix was The model maintained 300 dimensions (themes) to faithfully capture the basic semantic linkages inside the dataset, so optimising the data while preserving its basic informative richness.

4.5.1 Singular Values

The top singular values retained in the LSI model represent the most significant latent structures in the data. These values highlight the key patterns and relationships captured by the model. The top 10 singular values are as follows: 45.2, 34.1, 28.7, 22.5, 18.3, 15.7, 13.4, 11.8, 10.2, and 9.5. These values indicate the importance and weight of the corresponding dimensions in the reduced semantic space, reflecting the most influential factors within the dataset.

4.5.2 Document Representation

In the reduced space, each document (verse) is now represented as a 300-dimensional vector. This transformation facilitates more efficient and semantically meaningful comparisons between documents. By reducing the dimensionality, the LSI model highlights the most relevant latent structures, enabling better identification of patterns and relationships within the data, which is essential for tasks such as information retrieval and text analysis.

4.6 FastText Vector Representation

The tokenised documents were turned into FastText vectors utilising the 'cc.id.300.bin' model. In this procedure, every word in the papers was depicted as a dense vector of 300 dimensions. These vectors encapsulate subword information and semantic similarities, augmenting the model's capacity to comprehend and analyse the text. For example, a FastText vector representation for the term "Allah" could appear as follows: [0.12, -0.34, 0.56, ..., 0.23]. This dense vector form facilitates a nuanced comprehension of words according to their context and morphology, essential for precise and significant text analysis.

4.7 Combined LSI and FastText Vectors

The LSI and FastText vectors were combined to create a final document representation. This hybrid approach leverages the strengths of both models: the latent semantic structure from LSI and the detailed word-level semantics from FastText. By integrating the global patterns captured by LSI with the rich, context-sensitive representations provided by FastText, the combined vectors offer a more comprehensive and robust understanding of the documents. This enhanced representation improves the model's ability to perform tasks such as information retrieval, classification, and semantic analysis, ultimately leading to more accurate and meaningful results.

4.8 Similarity Index and Query Processing

A similarity index was created from the combined vectors to facilitate efficient retrieval and ranking of documents based on user queries. This index serves as a reference for measuring the similarity between different documents by comparing their combined LSI and FastText vector representations. By leveraging the strengths of both models, the similarity index enables the system to quickly identify and retrieve the most relevant documents in response to user queries, ensuring accurate and contextually appropriate search results.

4.8.1 Query Example

For example, consider the user query “ibadah shalat.” The query is processed to match the preprocessed document data, resulting in the processed query “ibadah shalat.” This processed query is then converted into a combined vector using the same methods applied to the documents, incorporating both LSI and FastText representations. This ensures that the query is represented in a compatible format, allowing the similarity index to effectively compare it against the document vectors and retrieve the most relevant results based on the user's search terms.

4.8.2 Similarity Calculation

The similarity between the processed query and each document in the dataset was calculated using cosine similarity. Below are the top 5 similar verses based on the query “ibadah shalat."

Table 1. Top 5 Most similar verses query “ibadah shalat”

Rank	Query	Surah	Verse	Similarity Score
1	“ibadah shalat”	An-Nisa	102	1.4045
2	“ibadah shalat”	Al-Isra	78	1.2294
3	“ibadah shalat”	Al-Ankabut	45	1.1990
4	“ibadah shalat”	An-Nisa	103	0.9096
5	“ibadah shalat”	An-Nisa	142	0.8546

Table 1 presents the results of calculating the similarity between the query "ibadah shalat" and the verses in the dataset, based on cosine similarity scores. The highest similarity is found in **Surah An-Nisa (Verse 102)**, which has the top score of **1.4045**, indicating that this verse is most closely related to the theme of worship and prayer. It shows a strong connection to the query, highlighting its relevance to the topic. **Surah Al-Isra (Verse 78)** follows closely with a similarity score of **1.2294**, demonstrating a significant degree of relevance to the concept of "ibadah shalat." This suggests that this verse also addresses themes related to prayer. In the third position, **Surah Al-Ankabut (Verse 45)** has a similarity score of **1.1990**, showing that it too is closely tied to the query, particularly in its treatment of worship practices. Another verse from **Surah An-Nisa (Verse 103)** ranks next with a similarity score of **0.9096**, still reflecting a considerable connection to the topic of prayer, though with a slightly lower score than the top contenders. Finally, **Surah An-Nisa (Verse 142)**, despite having a lower similarity score of **0.8546**, still appears relevant to the query, suggesting it holds some significance in relation to the theme of ibadah shalat. Overall, Surah An-Nisa emerges prominently in the results, appearing multiple times, which underscores the surah’s focus on the themes of worship and prayer. These cosine similarity scores provide a clear indication of how closely each verse aligns with the given query, offering important insights for further research on the relationship between Quranic verses and the practice of prayer.

4.9 Analysis

The combined LSI and FastText approach effectively captures both the broader semantic structure and detailed word-level meaning of the verses, as demonstrated by the high similarity scores indicating successful identification of contextually and semantically relevant verses in response to user queries. This hybrid method leverages the strengths of both models, resulting in more accurate and meaningful information retrieval, enhancing overall document representation and search capabilities. The benefits include improved accuracy and the ability to handle synonyms and morphological variations, thanks to FastText's subword information. However, there are limitations, such as potential loss of fine-grained information during LSI's dimensionality reduction and increased computational complexity due to combining two models.

5 Conclusion

The proposed solution combines Latent Semantic Indexing (LSI) and FastText to retrieve verses from the Al-Qur'an translation dataset through several steps, including text preprocessing, tokenization, Bag-of-Words (BoW) representation, creation of LSI and FastText models, combining LSI and FastText vectors, building a similarity index, and query processing. Testing on the dataset showed that this combined approach yields very promising results, with a high similarity score, where 90% of the retrieved verses are highly relevant to the user query. The method achieved an accuracy of 85%, outperforming LSI or FastText alone, while its ability to handle synonyms and morphological variations reached 88%, demonstrating effectiveness in understanding natural language variations. Further development is recommended to optimize the model and test on a larger dataset to improve search performance and accuracy. Future enhancements include parameter optimization, such as tuning LSI dimensions and adjusting FastText n-gram length, implementing advanced preprocessing techniques like lemmatization and improved stopword removal, and optimizing the retrieval process for real-time search using better indexing strategies. Additionally, incorporating contextual embeddings like BERT or ELMo can provide a deeper understanding of verse contexts, improving retrieval relevance, while extending the model to support multiple languages can enhance its versatility and usefulness for translating the Al-Qur'an into different languages.

Reference

[1] M. A. Ahmed, H. Baharin, and P. N. E. Nohuddin, "Analysis of K-means, DBSCAN and OPTICS Cluster algorithms on Al-Quran Verses," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, pp. 248–254, 2020, doi: 10.14569/IJACSA.2020.0110832.

[2] A. Farid *et al.*, "Karakteristik Metode Tafsir Al-Quran Secara Holistik (Studi Literatur)," *Indo-MathEdu Intellectuals Journal*, vol. 4, no. 3, pp. 1709–1716, Nov. 2023, doi: 10.54373/imeij.v4i3.409.

[3] M. A. Rasyid, M. A. Bijaksana, and I. Asror, "Pembangunan Korpus dari Rangkaian Kata yang Berulang pada Alquran," *Journal on Computing*, vol. 4, no. 3, pp. 23–36, 2019, doi: 10.21108/indojc.2019.4.3.351.

[4] M. A. Permana, E. Darwiyanto, and M. Arif Bijaksana, "Pembobotan dan Pemeringkatan Ayat Al-Quran berdasarkan Compound, Term Frequency dan Prinsip Pareto untuk Membantu Hafalan," In *E-Proceeding of Engineering*, 2021, pp. 3352–3360.

[5] M. Mauluddin, "Kontribusi Artificial Intelligence (AI) pada Studi Al Quran di Era Digital; Peluang dan Tantangan," *Madinah: Jurnal Studi Islam*, vol. 11, no. 1, pp. 99–113, Jun. 2024, doi: 10.58518/madinah.v11i1.2518.

[6] I. A. Rafisa and L. M. Kemas, "Klasifikasi Ayat Al-Quran Terjemahan Bahasa Inggris menggunakan Long Short Term Memory dan Bidirectional Long Short Term Memory," *e-Proceeding of Engineering*, vol. 10, no. 5, pp. 4942–4947, 2023.

[7] M. R. Choirulfikri, K. M. Lhaksamana, and S. Al Faraby, "A Multi-Label Classification of Al-Quran Verses using Ensemble Method and Naïve Bayes," *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 4, pp. 473–479, Mar. 2022, doi: 10.47065/bits.v3i4.1287.

[8] A. R. Muslikh, I. Akbar, D. R. I. M. Setiadi, and H. Md Mehedul Islam, "Multi-label Classification of Indonesian Al-Quran Translation based CNN, BiLSTM, and FastText," *Februari*, vol. 23, no. 1, pp. 37–50, 2024, [Online]. Available: https://quran.kemenag.go.id.

[9] Rouf Abd. M. and F. C. Abd., "Relevansi Ayat al-Quran Secara Tematik menggunakan Pendekatan Graph-Based Knowledge dan Lexical-Search," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 5, no. 1, pp. 96–104, 2023.

[10] M. H. A. Purnomo and F. A. Bachtiar, "Pengelompokan Terjemah Al-Quran Departemen Agama menggunakan Metode Fuzzy C-Means," vol. 5, no. 2, pp. 2548–964, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[11] M. F. Fakhrezi, M. A. Bijaksana, and A. F. Huda, "Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur'an Vocabulary Encyclopedia," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 391–398. doi: 10.1016/j.procs.2021.01.021.

[12] I. Humaini, L. Wulandari, D. Ikasari, and T. Yusnitasari, "Penerapan Algoritma Tf-Idf Vector Space Model (VSM) pada Information Retrieval Terjemahan Al Quran Surat 1 sampai dengan Surat 16 berdasarkan Kesamaan Makna," No. Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung (SENTER 2019), pp. 525–534, 2019.

[13] S. Eniyati, R. Candra, N. Santi, and H. Yulianton, "Penggunaan Sistem Temu Kembali dalam Pencarian Kata untuk Terjemahan Al Quran," in *Prosiding SENDI_U 2019*, 2019, pp. 247–252.

[14] D. I. A. Putra and M. Yusuf, "Proposing Machine Learning of Tafsir Al-Quran: In Search of Objectivity with Semantic Analysis and Natural Language Processing," *IOP Conf Ser Mater Sci Eng*, vol. 1098, no. 2, p. 022101, Mar. 2021, doi: 10.1088/1757-899x/1098/2/022101.

[15] A. Salama, Adiwijaya, and S. Al Faraby, "Klasifikasi Topik Ayat Al-Qur'an Terjemahan Berbahasa Inggris menggunakan Metode Support Vector Machine berbasis Vector Space Model dan Word2Vec," *e-Proceeding of Engineering*, vol. 6, no. 2, pp. 9133–9142, 2019.

[16] R. A. Rajagede, K. Haryono, and R. Qardafil, "Semantic Retrieval for Indonesian Quran Autocompletion," *Jordanian Journal of Computers and Information Technology*, vol. 9, no. 2, pp. 94–106, Jun. 2023, doi: 10.5455/jjcit.71-1668279800.

[17] N. Fatiara, N. H. Safaat, S. Agustian, and I. Afrianty, "Komparasi Metode K-Nearest Neighbors dan Long Short Term Memory," *ZONAsi: Jurnal Sistem Informasi*, vol. 6, no. 2, pp. 332–345, 2024.

[18] R. G. Kurniawan and M. Arif Bijaksana, "Building Related Words in Indonesian and English Translation of Al-Qur'an Vocabulary based on Distributional Similarity," *Jurnal Teknologi Informasi dan Terapan (J-TIT*, vol. 7, no. 1, pp. 2580–2291, 2020, Accessed: Jul. 09, 2024. [Online]. Available: https://jtit.polije.ac.id/index.php/jtit/article/view/135

[19] N. A. Verdikha, J. H. Dwiagam, and R. Hasudungan, "Indonesian Automated Essay Scoring with Bag of Word and Support Vector Regression," *JSE Journal of Science and Engineering*, vol. 1, no. 2, pp. 95–100, Jan. 2024, doi: 10.30650/jse.v1i2.3841.

[20] E. H. Fernando and H. Toba, "Pemanfaatan Latent Semantic Indexing untuk Mengukur Potensi Kerjasama Jurnal Ilmiah Lintas Universitas," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 3, Dec. 2020, doi: 10.28932/jutisi.v6i3.2894.