الجمهورية الجزائرية الديمقراطية الشعبية
The People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة محمد بوضياف بالمسيلة
University Mohamed Boudiaf of M'sila

كلية الرياضيات والإعلام الآلي
Faculty of Mathematics and Informatics

قسم الإعلام الآلي
Department of Computer Science

Domain: Mathematics and Computer Science

Thesis Presented to Fulfill the Partial Requirement
for **Master's Degree** in Computer Science

Specialty: Business Intelligence and Optimization

Prepared By: BOUZID Fatiha

Supervised By:

Dr. Bentercia Rahima

ENTITLED

TOPIC MODELING IN THE HOLY QURAN USING BERTopic

Jury Members

Dr.BOUZAAROURA AHLEM	President
Dr.Bentercia Rahima	Supervisor
Dr.CHALABI BAYA	Examiner

DEDICATION

I dedicate this thesis first and foremost to myself, for I have always believed in my ability to achieve my goals despite all challenges.

I also dedicate it to my beloved parents, each individually, in appreciation of their great role in my life.

To my caring father, who was a source of protection and support, never sparing his effort and hard work, teaching me the meaning of responsibility and patience. And to my dear mother, who has been a wellspring of kindness and generositythank you for your constant presence by my side.

To my precious grandmother, who was a fountain of affection and the heart of our family, whose love and prayers contributed to my strength and perseverance.

And I dedicate it to my dear siblings, who shared with me both joy and pain, always standing by me with love and encouragement.

To all of you, I present this achievement with pride and gratitude.

ACKNOWLEDGMENTS

All praise is due to Allah, by whose grace and bounty this achievement has been realized. I ask Him to make this work sincere for His noble face and beneficial for all.

I sincerely thank Dr.Bentercia Rahima for her continuous guidance throughout the completion of this work.

I also extend my deepest gratitude to everyone who supported me and stood by my side, whether through kind words, sincere advice, or moral support.

You all are the reason for this success, and without you, this work would not have seen the light of day.

ملخص

يتناول هذا البحث تطبيق نمذجة الموضوعات على القرآن الكريم باستخدام BERTopic وتمثيلات LaBSE لاستخراج وتصنيف موضوعاته الرئيسية بشكل آلي. وبالنظر إلى تعقيده الموضوعي، تم تنظيم الآيات إلى ثلاث فئات: التوحيد، الأحكام، والقصص. أظهرت النتائج أن هذا النهج يدعم التحليل الموضوعي للقرآن ويفتح آفاقًا لفهم أعمق باستخدام الذكاء الاصطناعي.

الكلمات المفتاحية: نمذجة الموضوعات، القرآن الكريم، المعالجة الآلية للغة، LaBSE، BERTopic.

Abstract

This research applies topic modeling to the Holy Quran using BERTopic and LaBSE embeddings to automatically extract and classify its main themes. The Qurans thematic complexity requires advanced NLP techniques. The study organizes verses into three categories: Monotheism, Legal Rulings, and Stories. Results show that this AI-based approach supports deeper understanding and thematic analysis of Quranic content.

Keywords: topic modeling, Holy Quran, Arabic NLP, BERTopic, LaBSE

Résumé

Ce travail applique la modélisation de sujets au Saint Coran en utilisant le modèle BERTopic et les représentations LaBSE, afin d'extraire et de classer automatiquement ses principaux thèmes. Étant donné la richesse et la complexité thématique du Coran, les versets ont été organisés en trois catégories : monothéisme, prescriptions juridiques et récits. Les résultats montrent que cette approche basée sur L'IA facilite lanalyse thématique et approfondit la compréhension du texte coranique.

Mots Clée : modélisation de sujets, Coran, traitement automatique de la langue arabe, BERTopic, LaBSE.

Contents

List of Figures

List of Tables

General Introduction

1 Problem Overview

The Holy Quran is the Word of Allah (God) revealed and sent down to Prophet Muhammad through the revelation of the Archangel Gabriel.It is a unique book in its style and organization. It is characterized by the multiplicity of topics within a single chapter and sometimes even within a single verse. A single verse may combine multiple themes such as doctrine, legal rulings, stories, and sermons, within a cohesive linguistic structure that does not clearly separate these subjects.

Example:

(لَّيْسَ ٱلْبِرَّ أَن تُوَلُّوا۟ وُجُوهَكُمْ قِبَلَ ٱلْمَشْرِقِ وَٱلْمَغْرِبِ وَلَٰكِنَّ ٱلْبِرَّ مَنْ ءَامَنَ بِٱللَّهِ وَٱلْيَوْمِ ٱلْآخِرِ وَٱلْمَلَٰٓئِكَةِ وَٱلْكِتَٰبِ وَٱلنَّبِيِّۦنَ وَءَاتَى ٱلْمَالَ عَلَىٰ حُبِّهِۦ ذَوِى ٱلْقُرْبَىٰ وَٱلْيَتَٰمَىٰ وَٱلْمَسَٰكِينَ وَٱبْنَ ٱلسَّبِيلِ وَٱلسَّآئِلِينَ وَفِى ٱلرِّقَابِ وَأَقَامَ ٱلصَّلَوٰةَ وَءَاتَى ٱلزَّكَوٰةَ ۖ وَٱلْمُوفُونَ بِعَهْدِهِمْ إِذَا عَٰهَدُوا۟ ۖ وَٱلصَّٰبِرِينَ فِى ٱلْبَأْسَآءِ وَٱلضَّرَّآءِ وَحِينَ ٱلْبَأْسِ ۗ أُو۟لَٰٓئِكَ ٱلَّذِينَ صَدَقُوا۟ ۖ وَأُو۟لَٰٓئِكَ هُمُ ٱلْمُتَّقُونَ)

سورة البقرة (2:177)

الموضوعات: البر الحقيقي، الإيمان، الإنفاق، الصدق، الوفاء، الصبر.

(وَٱبْتَغِ فِيمَآ ءَاتَىٰكَ ٱللَّهُ ٱلدَّارَ ٱلْآخِرَةَ ۖ وَلَا تَنسَ نَصِيبَكَ مِنَ ٱلدُّنْيَا ۖ وَأَحْسِن كَمَآ أَحْسَنَ ٱللَّهُ إِلَيْكَ ۖ وَلَا تَبْغِ ٱلْفَسَادَ فِى ٱلْأَرْضِ ۖ إِنَّ ٱللَّهَ لَا يُحِبُّ ٱلْمُفْسِدِينَ)

سورة القصص (28:77)

الموضوعات: التوازن بين الدنيا والآخرة، الأخلاق، الشكر، النهي عن الفساد.

(وَسَارِعُوٓا۟ إِلَىٰ مَغْفِرَةٍ مِّن رَّبِّكُمْ وَجَنَّةٍ عَرْضُهَا ٱلسَّمَٰوَٰتُ وَٱلْأَرْضُ أُعِدَّتْ لِلْمُتَّقِينَ (١٣٣) ٱلَّذِينَ يُنفِقُونَ فِى ٱلسَّرَّآءِ وَٱلضَّرَّآءِ وَٱلْكَٰظِمِينَ ٱلْغَيْظَ وَٱلْعَافِينَ عَنِ ٱلنَّاسِ ۗ وَٱللَّهُ يُحِبُّ ٱلْمُحْسِنِينَ)

سورة آل عمران (133–134)

الموضوعات: العمل للآخرة، الإنفاق، ضبط النفس، العفو، الإحسان.

Moreover, verses related to a specific topic are often distributed across different locations in multiple surahs, which further adds to the complexity of tracing and grouping related topics.

This diversity poses a challenge for researchers and readers in accurately identifying specific themes, despite previous manual efforts that are time-consuming and prone to interpretive variations. With the development of artificial intelligence techniques, particularly Natural Language Processing (NLP) in Arabic, there arises a need for innovative technical solutions capable of accurately and efficiently exploring these thematic relationships, in order to facilitate thematic research and analysis in the Holy Quran.

2 Motivation

The Holy Qur'an has great spiritual and intellectual value, making it an urgent priority to facilitate its understanding and objective study. The world is witnessing a remarkable increase in interest in Qur'anic studies, as reflected in the growth of academic research, cultural initiatives such as conferences and educational programs, and the growing demand for digital resources that facilitate access to the meanings of the Qur'an.

In this context, the role of artificial intelligence in developing innovative tools to analyze verses and accurately classify their themes becomes prominent. These tools will support researchers in providing deeper interpretations, enrich Qur'anic education, and enable readers to grasp the meanings of the Qur'an more easily. They will also contribute to linking Islamic heritage with modern technology, helping to preserve the Qur'anic legacy and opening new horizons for research and cultural interaction.

3 Objectives

This project seeks to harness artificial intelligence to advance the study of the Holy Quran through the following objectives:

- **To develop AI-based tools for analyzing the Quranic text to identify and extract thematic interconnections with precision.**
- **To classify Quranic themes systematically, enabling structured access to related verses across chapters.**
- **To organize Quranic verses into coherent thematic categories, facilitating efficient retrieval and study.**

- **To enhance the understanding and interpretation of the Qurans meanings for both scholars and general readers through intuitive digital tools.**
- **To support Quranic education and academic research by providing robust resources for teaching and in-depth analysis.**

4 Thesis Organization

The proposed work is organized into several chapters, each focusing on different aspects of topic modeling in the Holy Quran using topic modeling. The organization of the thesis is as follows:

- **General Introduction**

  Presents an overview of the project, defines the problem of thematic diversity and the application of topic modeling for clustering in the Holy Quran, and highlights the projects contributions.

- **Chapter1:Natural Language Processing and Topic Modeling**

  Introduces the foundational concepts of natural language processing and topic modeling techniques (e.g., LDA, BERTopic).

- **Chapter2:Related Work**

  This chapter reviews prior studies on topic modeling and clustering, divided into two main sections. The first section presents studies conducted on general Arabic texts, highlighting commonly used techniques and how recent advancements have improved topic modeling performance. The second section focuses on studies specifically applied to the Holy Quran, analyzing their approaches, and outcomes.

- **Chapter3:Proposed Methodology**

  Details the proposed topic modeling system for clustering, including data preprocessing (e.g., grouping Quranic verses by themes into separate documents), the BERTopic model implementation, and the workflow steps.

- **Chapter4:Experimental Results**

  Reports the findings, evaluates the systems performance, compares its application on the Quran with ordinary texts to address limitations (e.g., text size, meaning complexity).

- **Conclusion**

  Summarizes the research, provides recommendations, and outlines future directions.

This organization ensures a coherent progression from problem definition and literature review to methodology, results analysis, and concluding remarks.

Chapter 1

Natural Language Processing and Topic Modeling

1 Overview of Artificial Intelligence

Artificial Intelligence (AI), a transformative pillar of computer science, is dedicated to creating machines that mirror human intelligence, redefining the boundaries of technology and human potential.By replicating cognitive processes such as learning, reasoning, problem-solving, perception, and decision-making, AI has progressed far beyond its early days of rigid, rule-based systems. Today, it leverages advanced algorithms, vast datasets, and unparalleled computational power to achieve feats once thought exclusive to the human mind [1].

The impact of AI spans across virtually every sector, revolutionizing industries and reshaping modern life. In healthcare, AI enhances diagnostics and personalizes treatment plans; in finance, it powers fraud detection and algorithmic trading; in transportation, it drives autonomous vehicles; and in education, it tailors learning experiences to individual needs. From recommendation systems curating personalized content to intelligent virtual assistants streamlining daily tasks, AIs applications are as diverse as they are groundbreaking. Among its many subfields, Natural Language Processing (NLP) stands out, enabling machines to comprehend, interpret, and generate human language with remarkable fluency. This capability fuels innovations like real-time translation, sentiment analysis, and conversational AI, bridging the gap between human communication and machine understanding.

As AI continues to evolve, it promises not only to augment human capabilities but also to spark new ways of solving complex global challenges, making it one of the most pivotal technologies of our time.

2 Natural Language Processing

Natural Language Processing (NLP) is a vital part of AI that allows machines to understand, analyze, and generate human language. Using deep learning, NLP processes large amounts of text and speech, moving beyond rule-based methods to interpret syntax, semantics, context, and intent [2].

NLP has enabled technologies like machine translation, sentiment analysis, summarization, and chatbots. These are used in real-time communication, social media monitoring, and efficient information access. NLP-driven tools also enhance customer service and accessibility through voice assistants.

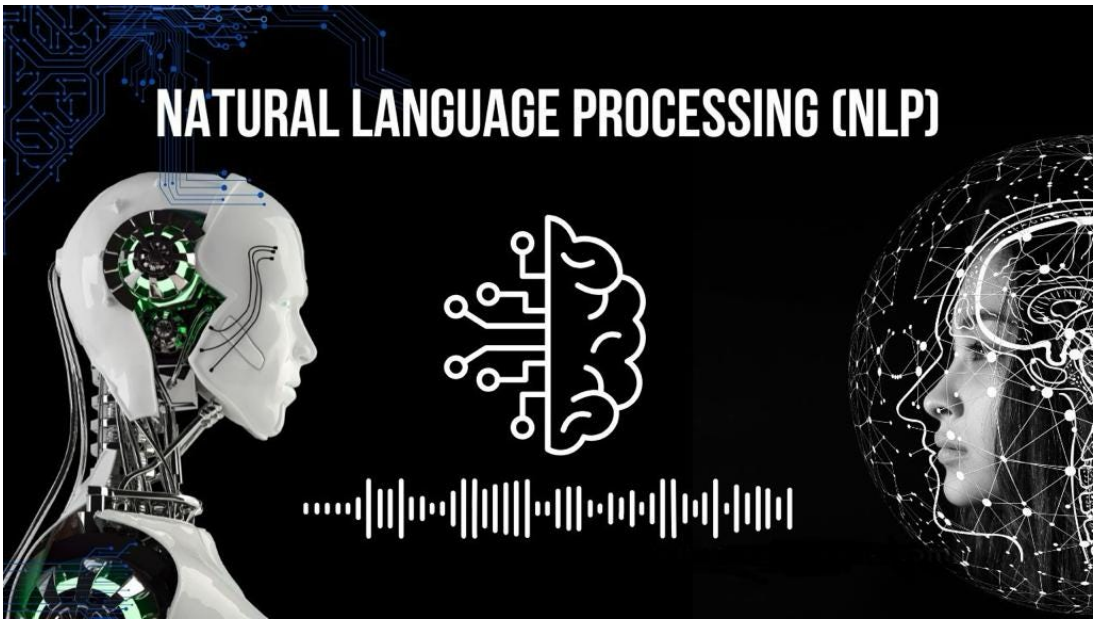


Figure 1.1: Natural Language Processing
[3]

Across industries, NLP supports healthcare diagnostics, education tools, and financial analysis. It also powers creative tools like AI storytelling. With its wide applications, NLP is central to making technology more human-like and inclusive.

As the field grows, challenges such as language bias, privacy, and ethics arise. However, with advancements in multilingual models and fairness efforts, NLP continues to improve global communication and human-machine interaction.

2.1 Importance of Natural Language Processing(NLP)

The growing volume of unstructured text data from sources like emails, social media, websites, and documents has increased the demand for NLP. It plays a vital role in enabling machines to [4]:

- **Understand Human Communication:** NLP bridges the gap between how humans communicate and how machines interpret data.
- **Automate Repetitive Tasks:** From summarizing texts to auto-completing sentences, NLP reduces manual effort.
- **Enhance User Interaction:** Smart assistants like ChatGPT, Alexa, and Siri rely heavily on NLP to provide accurate and natural responses.
- **Support Decision Making:** In business and healthcare, NLP can extract valuable insights from textual data to inform better decisions.

By leveraging NLP, organizations can improve customer service, analyze sentiment, detect fraud, and even assist in medical diagnosis. Its importance continues to grow as we move toward more human-centric AI systems.

2.2 Core Components of NLP

Natural Language Processing (NLP) enables computers to understand and generate human language, relying on two core components: Natural Language Understanding (NLU) and Natural Language Generation (NLG).

2.2.1 Natural Language Understanding (NLU)

Focuses on analyzing text or speech to extract meaning, context, and intent [5]. It involves tasks like named entity recognition (e.g., identifying "Cairo" as a location), sentiment analysis (e.g., classifying "The product is great" as positive), and intent recognition (e.g., detecting "book a ticket" as a travel request). NLU leverages techniques like transformer models (e.g., BERT) and word embeddings to understand context accurately, powering applications such as virtual assistants and search engines.

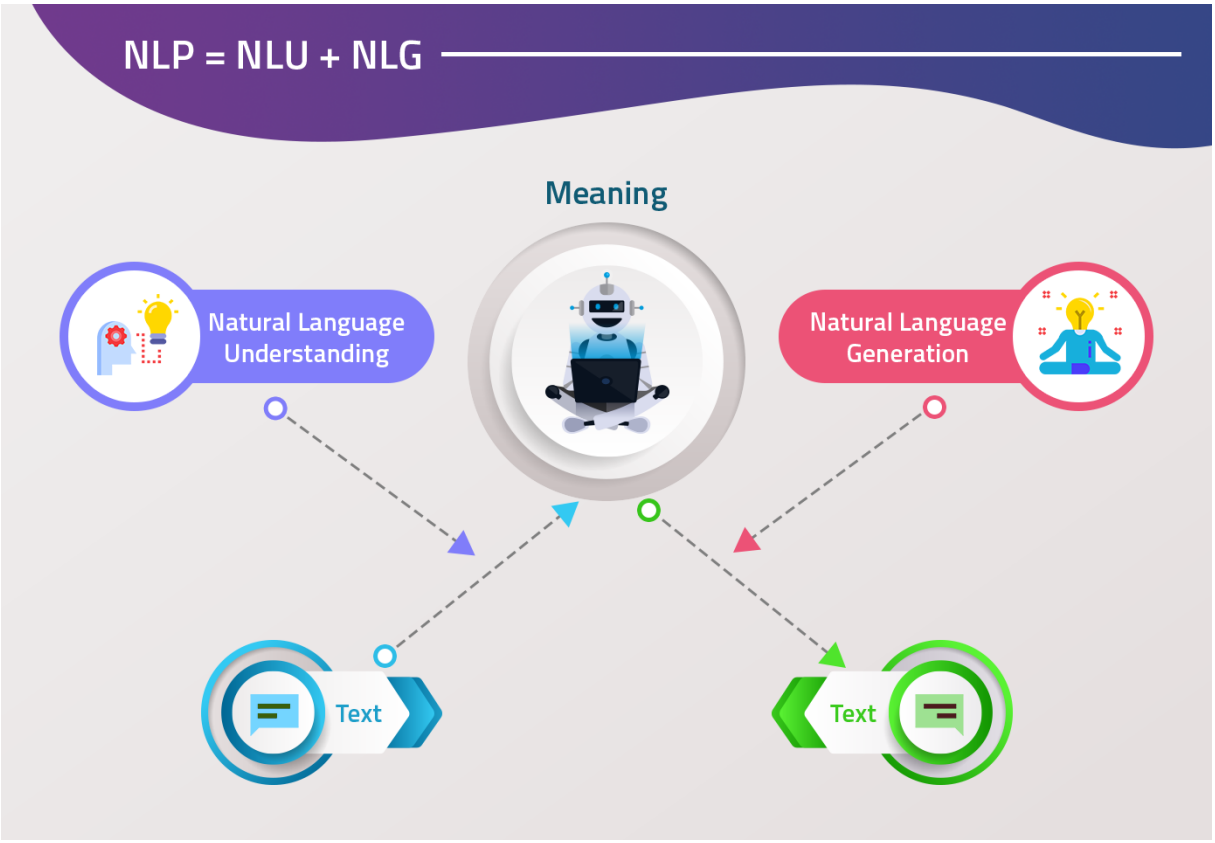


Figure 1.2: Components of NLP
[6]

2.2.2 Natural Language Generation (NLG)

Aims to produce human-like text, such as generating reports, summarizing content, or creating conversational responses. It includes tasks like data-to-text generation (e.g., converting weather data into "Its sunny with a temperature of 25řC") and dialogue generation. NLG uses models like GPT and template-based approaches to produce coherent text, applied in chatbots and automated reporting [7].

Together, NLU and NLG enable seamless human-computer interaction by interpreting user input and generating appropriate responses, making NLP vital for intelligent applications.

2.3 Applications of Natural Language Processing

Natural Language Processing (NLP) encompasses a variety of applications that have transformed how humans interact with machines. These applications are implemented in numerous sectors such as healthcare, customer service, marketing, education, and legal services [8].

2.3.1 Sentiment Analysis

Sentiment analysis involves identifying and extracting opinions, emotions, and attitudes from text data. It is particularly valuable in marketing and brand monitoring, as it helps companies understand public perception of their products or services.

For example, by analyzing customer reviews, social media posts, or feedback surveys, businesses can gauge satisfaction levels, detect dissatisfaction early, and refine their strategies accordingly.

2.3.2 Machine Translation

Machine translation focuses on automatically converting text from one language into another. It plays a crucial role in enabling cross-linguistic communication and access to information across borders. Tools like Google Translate have become widely used due to their ability to provide context-aware and fluent translations. These systems often rely on deep learning models that consider grammar, syntax, and semantics to improve translation accuracy over time.

2.3.3 Text Summarization

Text summarization aims to generate brief and informative summaries of longer texts. This is highly useful in domains that require rapid information digestion, such as journal-

ism, legal analysis, and academic research. Automatic summarizers help reduce reading time while preserving the essential meaning and structure of the source material. There are two main approaches: extractive (selecting key sentences) and abstractive (generating new sentences based on understanding).

2.3.4 Question Answering Systems

Question answering (QA) systems enable machines to respond accurately to user queries, often in natural language. These systems are at the core of virtual assistants (like Alexa, Google Assistant, and Siri) and modern search engines. They leverage a combination of information retrieval, natural language understanding, and sometimes external knowledge bases to provide direct answers instead of just suggesting documents or links.

2.3.5 Speech Recognition

Speech recognition involves converting spoken language into written text. It allows users to interact with computers using voice commands and is commonly used in virtual assistants, transcription services, and accessibility tools for individuals with disabilities. This technology supports hands-free operation of devices, making it useful in situations where typing is impractical, such as driving or medical settings.

Natural Language Processing gives us the tools to understand language by breaking sentences down, spotting patterns, and finding meanings. Topic Modeling takes this a step further by organizing the information, grouping related words together to show what a text is really about. In simple terms, NLP helps us make sense of raw language, while Topic Modeling reveals the main themes hidden in that language.

3 Topic Modeling

Topic modeling is an unsupervised technique in the field of natural language processing (NLP) used to discover latent themes or patterns within a large collection of textual documents. It aims to identify and organize semantically similar words and group related documents under the same topic without requiring any prior labeling or supervision. This capability is especially important when dealing with massive amounts of unstructured textual data, where manual classification becomes impractical or impossible. [9]

Topic modeling has been widely applied in various NLP applications such as text classification, sentiment analysis, recommendation systems, and information retrieval, where understanding the semantic structure of content is crucial.

Topic modeling contributes to simplifying the representation of textual data by reducing focus from individual words to higher-level semantic structures. This is analogous to dimensionality reduction techniques in numerical data, where unnecessary or redundant features are eliminated and only the most important components are retained, improving processing efficiency and ease of analysis. In text analysis, each word is treated as a feature, and modeling allows us to concentrate on the most meaningful patterns, thereby reducing noise and redundancy.

3.1 The Evolution of Topic Modeling

Topic modeling has evolved significantly over time. We can categorize the techniques into old and new methods, based on how they work, their strengths, and their differences:

3.1.1 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is among the earliest models used and relies on singular value decomposition (SVD) to uncover relationships between words and documents by analyzing their co-occurrence frequencies.
This model helps identify general patterns within large text corpora; for example, it might group words like Quran and book if they appear in similar contexts. However, LSA does not account for word order or polysemy (multiple meanings), limiting its ability to capture precise meanings [10].

3.1.2 Probabilistic Latent Semantic Analysis (PLSA)

Probabilistic Latent Semantic Analysis (PLSA) was proposed as a probabilistic alternative to LSA, representing each document as a mixture of topics, with each topic represented as a probability distribution over words. It uses the Expectation-Maximization (EM) algorithm to estimate parameters and provides clearer topic representations. Nevertheless, it struggles to generalize to new documents due to the lack of a true generative model [11].

3.1.3 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) addressed previous issues by introducing a generative probabilistic model that assumes documents are composed of multiple topics drawn from Dirichlet distributions. LDA has become one of the most popular topic modeling methods due to its generalizability, ability to handle large datasets, and its interpretable topic representations. However, it still ignores word order and tends to be less effective on short or sparse texts [12].

3.1.4 Topic To Vector(Top2Vec)

Top2Vec is another modern topic modeling technique that eliminates the need for prior assumptions such as the number of topics. Introduced in 2020, it works by jointly embedding documents and words in the same semantic space using models like Doc2Vec (Document To Vector) or transformer-based embeddings. Topics are then discovered by finding dense clusters of document vectors. Top2Vec offers the advantage of high topic coherence and the ability to discover topics without preprocessing steps like lemmatization or stop-word removal. However, it may require significant computational resources for large datasets and can be slower than traditional models [13].

3.1.5 BERTopic

BERTopic was developed in 2020 by researcher Maarten Grootendorst, in response to the growing need for more accurate and flexible topic models that rely on contextual understanding of text.

It was built using clustering techniques and contextual text representations through powerful language models like BERT, aiming to overcome the limitations of traditional models such as LDA, which ignore the semantic relationships between words.

The model was designed to extract coherent topic representations using a class-based variation of the TF-IDF algorithm. Its key applications include text content analysis, dynamic topic modeling, and improving search engine results and intelligent systems [14].

Chapter 2

Related Work

1 Introduction

In this chapter, we review previous works related to topic modeling and clustering in Arabic texts in general, and Quranic texts in particular. We focus on the methods used for topic modeling , clustering and evaluate how successfully they handle classical Arabic and the religious context of the Holy Quran.

2 Topic Modeling for Clustering Arabic Documents

ALHawarat and Hegazi (2018) proposed a hybrid approach that integrates K-means clustering with Latent Dirichlet Allocation (LDA) to improve the clustering of Arabic text documents, offering another perspective on topic modeling within the NLU framework. Their study utilized a dataset of 2,700 Arabic news articles categorized into nine topics, mirroring the dataset used by Alkhafaji and Al-Rashid [15].

By combining K-means, a clustering algorithm that groups documents based on feature similarity, with LDA, which extracts latent topics, the hybrid model leverages both statistical clustering and probabilistic topic modeling. The results demonstrated a Purity score of 0.933, indicating high clustering quality, and outperformed K-means alone, which lacked the semantic depth provided by LDA.

The study compares favorably with prior work, such as Abuaiadah (2016) [16], who applied Bisect K-means to Arabic texts, and Kelaiaia and Merouani (2016) [17], who compared LDA and K-means using external metrics like F-measure.

Al-Hawarat and Hijazi differs by integrating topic modeling and clustering in a single framework, evaluated on Modern Standard Arabic using external validation metrics.

Ibrahim (2019) proposed a novel hybrid approach to Arabic text clustering by integrating word embeddings from Word2Vec with semantic relations from the Arabic WordNet. The method involves preprocessing Arabic documents, extracting semantic relationships, and generating word vectors using CBOW and Skip-gram models [18].

Hierarchical clustering is then applied to group similar articles, and performance is evaluated using the Silhouette coefficient. Experiments on two datasets showed improved clustering quality, with silhouette scores reaching 0.639 on a small dataset and 0.624 on a larger one. The results demonstrated that combining semantic resources with word embeddings enhances clustering precision.

Previous research on Arabic text clustering explored a variety of approaches. Yahya (2009) applied Frequent Itemset-based Hierarchical Clustering using N-grams [19]. Alghamdi et al. (2014) incorporated semantic class features for web page clustering [20]. Xu et al. (2017) explored deep neural networks for short-text clustering [21]. Bengio et al. (2003) [22] and Mikolov et al. (2013) [23] laid the foundational work for distributed word representations that have since powered models like Word2Vec. These models significantly outperformed traditional keyword-based representations, especially when dealing with morphologically rich languages like Arabic.

Ibrahims contribution is notable for incorporating both distributional semantics and lexical knowledge bases into a unified model, showing measurable improvements over earlier AWNet-only approaches. The methodology also supports scalability to large Arabic corpora, making it valuable for real-world applications in content categorization, topic detection, and semantic search.

Alkhafaji and Al-Rashid (2021) presented one of the first studies to apply the LDA2Vec model in analyzing and clustering Arabic documents. The model is based on a combination of Latent Dirichlet Allocation (LDA) and Word2Vec, which allows topics and words to be represented in a convergent space that supports semantic understanding. The researchers used a dataset containing 2,700 Arabic documents divided into nine categories, such as: Religion, Politics, Art, and Health [24].

The results showed that the LDA2Vec model achieved an accuracy of 82.40 % in categorizing documents by topic, outperforming the traditional LDA model, which had an accuracy of 67.96%. The study confirms the effectiveness of LDA2Vec in dealing with Arabic texts despite its linguistic challenges, such as morphology and syntax.

In the context of related work, the authors referenced several prior studies that explored the integration of LDA with Word2Vec. Moody (2016) introduced the original LDA2Vec model using datasets such as 20Newsgroups, though without support for testing new documents [25]. Xue Mu (2019) proposed a hybrid model for text retrieval based on LDA and Word2Vec, demonstrating better performance compared to baseline approaches [26].

3 Topic Modeling in Quran

Alshammeri,Atwell and Alsalka (2019) introduced a semantic similarity model using Doc2Vec to measure relatedness between verses of the Quran in Classical Arabic. They trained the model on the full text of the Quran without diacritics and evaluated it

using the Qursim dataset, which contains over 7,600 annotated verse pairs. By embedding verses as dense vectors and applying cosine similarity, their model achieved 76% accuracy and 79% precision in identifying semantically related verse pairs. The study highlights the effectiveness of document-level embeddings in capturing deep semantic structures in religious texts [27].

Earlier approaches to verse similarity in the Quran primarily relied on lexical features. Akour et al. (2014) used N-grams and TF-IDF for surah classification [28]. Hamed and Aziz (2016) applied neural networks in a question answering system using translated verses [29].

Qahl (2014) explored multiple distance metrics, including cosine and Euclidean, to compare sacred texts [30].Sharaf and Atwell (2012) developed the Qursim corpus as a resource for semantic evaluation [31]. However, most of these methods were limited to surface-level matching or translations, rather than embeddings learned directly from Arabic text.

The work of Alshammeri et al. stands out by leveraging unsupervised document embeddings on original Arabic scripture, offering a scalable and semantically aware approach to Quranic analysis. Their model not only improves similarity detection but also lays the groundwork for future applications such as thematic clustering and intelligent Quranic search.

Slamet et al. (2016) explored the application of the K-Means algorithm to cluster the verses of the Holy Quran in a text mining framework [32]. Using an English translation of the Quran as the dataset (6,236 verses), they applied standard preprocessing steps including tokenization, stemming, and stop-word removal. The algorithm grouped the verses into three clusters based on the number of steamed and unsteamed words, with the largest cluster containing 3,650 verses. Cluster membership was determined using Euclidean distance to the centroid, and the clustering results were visualized to reflect semantic similarities.

Earlier studies in Quranic data mining tackled different methods of text classification and clustering. Yauri et al. (2013) used OWL-DL ontologies to extract concepts from Quranic verses [33]. Hashimi et al. (2015) focused on evaluating text mining selection criteria [34]. Jain (2010) reviewed advancements in data clustering, including K-Means [35], while Bhatia and Khurana (2013) analyzed the effect of centroid initialization in clus- tering performance [36]. Other works, such as Bhardwaj (2016) [37] and Farooqui and Noordin (2015) addressed knowledge discovery in religious texts using a mix of rule-

based and ontology-driven models [38].

Slamet et al.s contribution provides a foundational use of unsupervised learning to explore Quranic structure, allowing thematic segmentation of verses. Although basic in its approach, this method lays the groundwork for more sophisticated clustering techniques and ontology-enhanced knowledge extraction.

Alqarni (2024) introduced a novel methodology for semantic search in Quranic texts by leveraging embeddings generated from Large Language Models (LLMs), specifically the GPT-3.5 architecture. The approach begins by chunking the Quran into verses and encoding each verse using the text-embedding-ada-002 model [39]. User queries are also embedded into the same vector space, and cosine similarity is applied to retrieve the most semantically relevant verses.

The top results are then passed to a generative LLM to produce refined and contextually grounded answers. Experimental evaluation using precision, recall, and F1-score across low-, mid-, and high-level semantic queries demonstrated that GPT-based embeddings consistently outperformed traditional models like Doc2Vec and Universal Sentence En- coder, particularly in high-level semantic tasks.

Previous research in Quranic semantic search has utilized various techniques, including ontology-based retrieval (Zouaoui and Rezeg, 2021) [40], Word2Vec sentence embeddings (Saeed et al., 2020) [41], and translation-based query expansion (Afzal and Mukhtar, 2019) [42]. While transformer-based models like AraT5 have shown promising results in low-level semantic matching, traditional methods often lacked flexibility in capturing abstract or indirect meanings within Quranic texts.

Alqarnis contribution lies in demonstrating that LLM embeddings can effectively model complex semantic relationships across verses, enabling deeper understanding of Quranic concepts. The system also accommodates diverse query types and supports explainable answer generation. Despite limitations such as lack of diacritic handling and reliance on GPT models not extensively trained on Arabic corpora, this work marks a significant advancement in Arabic semantic search and paves the way for future enhancements through fine-tuning and integration with Tafsir resources.

4 Conclusion

In summary, previous research on topic modeling and clustering of Arabic texts reveals a progressive shift from traditional statistical models toward embedding-based and neural approaches. While early methods offered foundational insights, recent advances, especially those involving large language models and semantic resources, demonstrate significant potential for deeper understanding and better organization of Quranic content. These developments lay a solid foundation for future work in thematic clustering, intelligent search, and semantic interpretation of Classical Arabic texts.

Chapter 3

Methodology

1 Introduction

This chapter presents the methodology adopted in this study for modeling the topics of the Holy Quran. The methodology is designed to extract, preprocess and analyze Quranic texts in an effective manner, aiming to uncover the main themes addressed in the verses.

The process begins with the collection and preparation of the Quranic dataset, including the verses. It then proceeds to the text preprocessing stage, which involves cleaning operations.

Finally, the structure and components of the selected model are explained, with clarification on why it is suitable for processing Arabic texts, particularly within the unique linguistic and religious context of the Quran.

2 Working environment

- **Hardware Environment:**

  The implementation was performed on a laptop with the following specifications:

  - Intel Core I5 7th Generation CPU
  - 8 GB RAM
  - NVIDIA GeForce MX130 (2 GB), and Intel HD Graphics 620

- **Softwar environment**

  - **Google Colab:**

    Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education [43].

  - **Python(3.13):**

    Python is a high-level programming language that is popular in many domains, including web development, machine learning, artificial intelligence, and data analysis [44].

- **Libraries**
  - **RE(Regular Expressions):**

    Regular expressions are a powerful tool for text processing in many programming languages, and Python is no exception. The re library in Python provides a way to work with regular expressions. It allows you to search, match, replace, and split text based on specific patterns [45].
  - **NLTK (Natural Language Toolkit):**

    NLTK is a popular Python library for natural language processing. It provides a suite of tools for tokenization, classification, morphological and syntactic analysis, stemming, and parsing. In this project, it is used for analyzing Arabic texts and extracting roots and meanings [46].
  - **Arabic-Stopwords:**

    A library that provides a list of common Arabic words, such as prepositions, conjunctions, and articles, which are frequently used but have little semantic value. These words often introduce noise into text data, which can hinder the performance of natural language processing (NLP) models. The library improves data quality [47].
  - **PyArabic:**

    A specific Arabic language library for Python, provides basic functions to manipulate Arabic letters and text, like detecting Arabic letters, Arabic letters groups and characteristics, remove diacritics [48].
  - **Sentence Transformers:**

    Sentence Transformers is the go-to Python module for accessing, using, and training state-of-the-art embedding and reranker models. It can be used to compute embeddings using Sentence Transformer models or to calculate similarity scores using Cross-Encoder models . This unlocks a wide range of applications, including semantic search, semantic textual similarity, and paraphrase mining [49].
  - **Pandas:**

    Pandas is a widely-used open-source library for data manipulation and analysis in Python [50]. It offers a user-friendly and efficient way to handle structured data using its DataFrame data structure. With Pandas, you can easily clean, filter, transform, and aggregate data. It also provides robust tools for data visualization and seamless integration with other libraries. Pandas is a popular choice among data scientists and analysts for working with tabular data in Python.

- **Sklearn:**

  Scikit-learn (sklearn) is a comprehensive machine learning library for Python [51], providing a wide range of algorithms and tools for various machine learning tasks, such as classification, regression, clustering, and dimensionality reduction.

- **BERTopic(0.16.0) :**

  BERTopic is a library for topic modeling that uses deep learning and clustering algorithms. Version 0.16.0 offers improved support for multilingual text, including Arabic [52] .

- **UMAP:** UMAP (Uniform Manifold Approximation and Projection) is a powerful dimensionality reduction technique in Python, ideal for visualizing high-dimensional data [53].

- **HDBSCAN(Hierarchical Density-Based Spatial Clustering of Applications with Noise):**

  HDBSCAN is a density-based clustering algorithm that groups data points without the need to predefine the number of clusters. It is used in BERTopic to automatically cluster verses into shared thematic topics [54].

- **KeyBERTInspired:**

  KeyBERTInspired (from bertopic.representation) This module is part of the BERTopic library and enhances topic representation by identifying the most informative keywords for each topic. It improves the clarity and interpretability of the extracted themes [55].

3 Dataset

The Holy Quran is the word of Allah Almighty, revealed to His Prophet Muhammad (peace and blessings be upon him). It is miraculous in its wording, and it is worshipped through its recitation. It is written in the Mushafs and has been transmitted to us through continuous, reliable transmission (tawatur) [56].
Muslims were so interested in the Quran that they counted the number of its verses and words. It consists of 60 parties and 114 surahs It begins with Surah Al-Fatiha and ends with Surah Al-Nasas.

﴿ * وَقَضَىٰ رَبُّكَ أَلَّا تَعْبُدُوا إِلَّا إِيَّاهُ وَبِالْوَالِدَيْنِ إِحْسَانًا ۚ إِمَّا يَبْلُغَنَّ عِندَكَ الْكِبَرَ أَحَدُهُمَا أَوْ كِلَاهُمَا
فَلَا تَقُل لَّهُمَا أُفٍّ وَلَا تَنْهَرْهُمَا وَقُل لَّهُمَا قَوْلًا كَرِيمًا﴾
[الإسراء: 23]
سورة : الإسراء - Al-Isra - الجزء : (15) - الصفحة: (284)

Figure 3.1: Surat al-Isra

According to the most common version, the number of verses is 6236 verses, while some sources differ and state that the number of verses is 6200 verses due to the difference in counting the basmalah as an independent verse, as well as due to the disagreement over the letters of the tide. The number of words is about 77,437 words, and the number of letters ranges from 321,000 to 323,015 letters according to different counting methods.

The miracle of the Qur'an's vocabulary and themes is one of the most prominent manifestations of its uniqueness; its words are carefully selected, expressing meanings in the shortest and most eloquent phrases, and carrying deep linguistic, educational, and spiritual connotations.

As for its topics, they include doctrine, legislation, stories, ethics, the universe, the human being, and the resurrection and rebirth(العقيدة، التشريع، القصص، الأخلاق، الكون، الإنسان، والبعث والحياة الآخرة) making it a comprehensive book that deals with man in all his conditions and relationships with his Lord, himself, and his society. This thematic diversity and verbal richness make the Qur'an a rich and ideal source for research.

3.1 Data Collection

This study relied on the book **Al-Jami' li-Mawdu'at Ayat al-Quran al-Karim** [57] (The Compendium of the Topics of the Verses of the Holy Quran) by Muhammad Faris Barakat, published in 1959. This book represents one of the earliest scholarly efforts to thematically classify Quranic verses. It is the result of the authors personal initiative, supported by the contributions of various scholars and researchers in the field of Quranic studies. The author organized the verses into 20 primary themes, each comprising several subtopics.

To align this classification with the objectives of this study in topic modeling, a thematic framework was adopted based on a well-known interpretive approach among classical Quranic commentators. According to this view, the content of the Quran revolves around three core categories:

1-Monotheism (Tawheed): Including verses related to the oneness of God, such as those on the creation of the heavens and the earth, resurrection, divine names and

attributes, and the Day of Judgment.

2-Rulings (Ahkam): Including verses that deal with legal rulings, acts of worship, financial and social transactions, ethics, and obligations such as prayer, zakat, and Hajj.

3-Stories (Qasas): Including narratives about the prophets, previous nations, and significant historical events mentioned throughout the Quran.

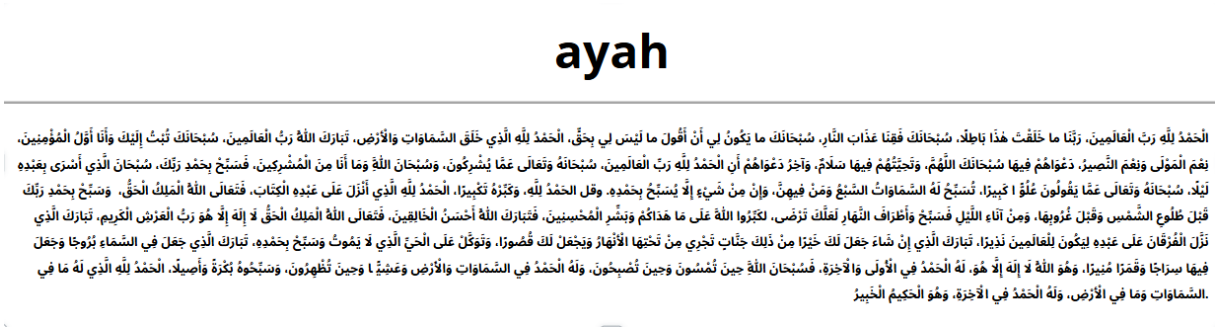


Figure 3.2: Structure and Distribution of the Thematically Grouped Quranic Dataset

Based on this tripartite structure,I reorganized the verse groups from the original source accordingly. The verses under each theme were then distributed across multiple documents of varying and randomly assigned lengths.This restructuring was done to ensure a balanced dataset that facilitates subsequent preprocessing and modeling tasks.

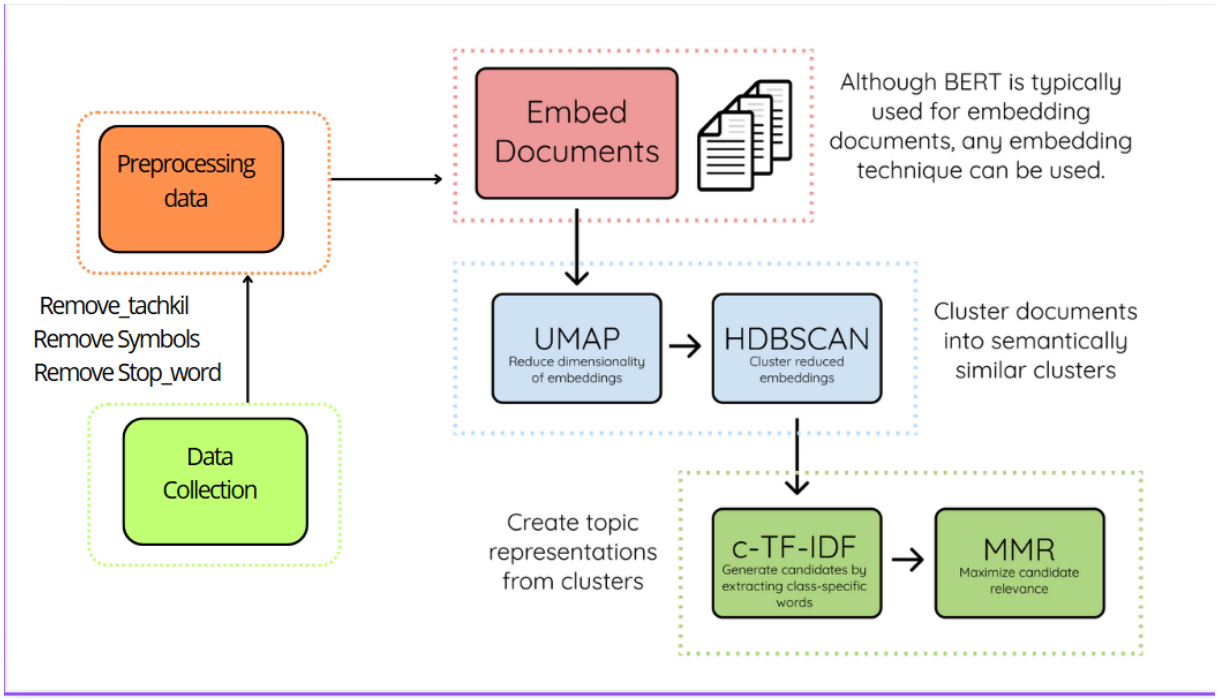


Figure 3.3: Structure and Distribution of the Thematically Grouped Quranic Dataset [58]

3.2 Data Preprocessing

Text preprocessing is a fundamental step in Natural Language Processing (NLP) and text analysis that involves cleaning, transforming, and structuring raw textual data into a format that is more suitable for computational processing [59].

Given that text data often contains inconsistencies, noise, and unnecessary information, preprocessing enhances data quality and improves the efficiency of NLP models, including those used for machine learning, deep learning, sentiment analysis, machine translation, and information retrieval.

Text preprocessing typically consists of multiple techniques, each addressing different challenges associated with raw text. These techniques include :

3.2.1 Removal of Diacritics (Tashkeel)

Arabic diacritics are short vowel marks that help with pronunciation but do not affect the core meaning of words. They were removed to reduce textual variation and ensure consistency, as many Qur'anic verses are recorded with diacritics that may differ in formatting or usage.

Example:

Original Text (With Diacritics):

الَّذِينَ يُقِيمُونَ الصَّلَاةَ وَيُؤْتُونَ الزَّكَاةَ وَهُم بِالْآخِرَةِ هُمْ يُوقِنُونَ

After Removing Diacritics:

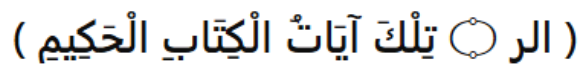
الذين يقيمون الصلاة ويؤتون الزكاة وهم بالآخرة هم يوقنون

3.2.2 Removal of the Symbol

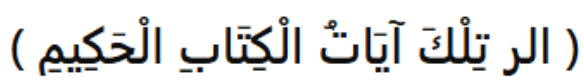
This symbol is a decorative separator used in the printed Mushaf to denote the start of specific sections or rubrics. It holds no linguistic value and was removed to avoid interference with textual analysis and tokenization.

Example:

Original Text (With Symbol):



After Removing Symbol:

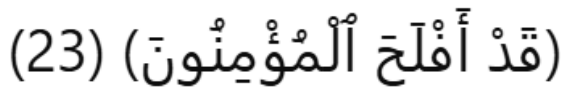


3.2.3 Removal of Other Symbols and Numbers

All punctuation marks, special symbols, and numeric characters were stripped out. These elements do not contribute to the semantic meaning of the verses and can introduce noise into the analysis.

Example:

Original Text:



After Removing:

3.2.4 Removal of Stop Words

Stop words are commonly used words in a language (such as articles, prepositions, pronouns, and conjunctions) that do not contribute significant meaning to a sentence. Therefore, they are typically removed during text preprocessing.

Examples:

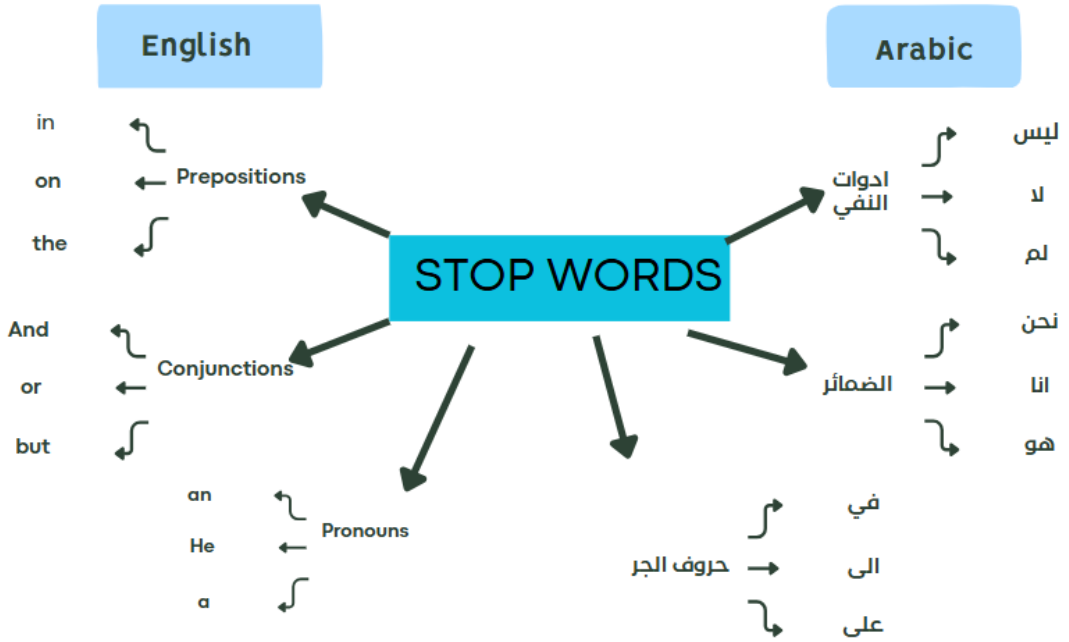


Figure 3.4: Example of Stop Words Removed During Preprocessing

Why Remove Stop Words?

- **Reduces Text Size:** Eliminates unnecessary words, making text processing faster.
- **Improves NLP Models:** Enhances machine learning tasks like sentiment analysis and classification.
- **Optimizes Search Engines:** Focuses on relevant keywords and improves search accuracy.

3.2.5 Tokenization (Splitting Text into Words)

The cleaned text was segmented into individual tokens or words. This step is essential for further computational analysis such as vectorization, embedding, and topic modeling, where each word is treated as a discrete feature.

Each of these preprocessing steps contributed to producing a cleaner and more semantically-focused dataset, better suited for modeling the topics of the Quranic text using the chosen NLP techniques.

4 Text Embedding

After completing the preprocessing of the Quranic texts, it became essential to transform these textual data into numerical representations that machines can process.

In the context of Quranic texts, embeddings can help capture semantic relationships between verses, enabling tasks like semantic search or interpretation analysis.

Since algorithms cannot process human language directly, text must be converted into numerical vectors that capture its semantic meaning. This process is referred to as text embedding, which denotes the representation of high-dimensional data as vectors in a lower-dimensional space.

Such transformation is crucial, as it enables machine learning algorithms to process and interpret complex inputs, including words, sentences, images, and even graphs.

For instance:
words like "king" and "queen" can be represented as vectors that are close to each other in the embedding space, reflecting their semantic similarity. This is in contrast to one-hot encoding, where each word is represented as a sparse binary vector, failing to capture any semantic relationships.

Embeddings are typically learned through various machine learning techniques, and the resulting vectors can be used in downstream tasks such as classification, clustering, and recommendation systems.

4.1 Types of Embeddings

In Natural Language Processing (NLP), embeddings are vector representations of words, sentences, or documents that capture their semantic meaning. Various types of embeddings have been developed over the years, each with different characteristics and use cases. This section outlines the main categories: [60]

4.1.1 Word Embeddings

Word embeddings are vector representations of individual words that capture their semantic meaning. Popular models for generating word embeddings include Word2Vec, GloVe, and FastText.

These embeddings form the backbone of many NLP tasks, as they allow models to recognize relationships and similarities between words.

Example:

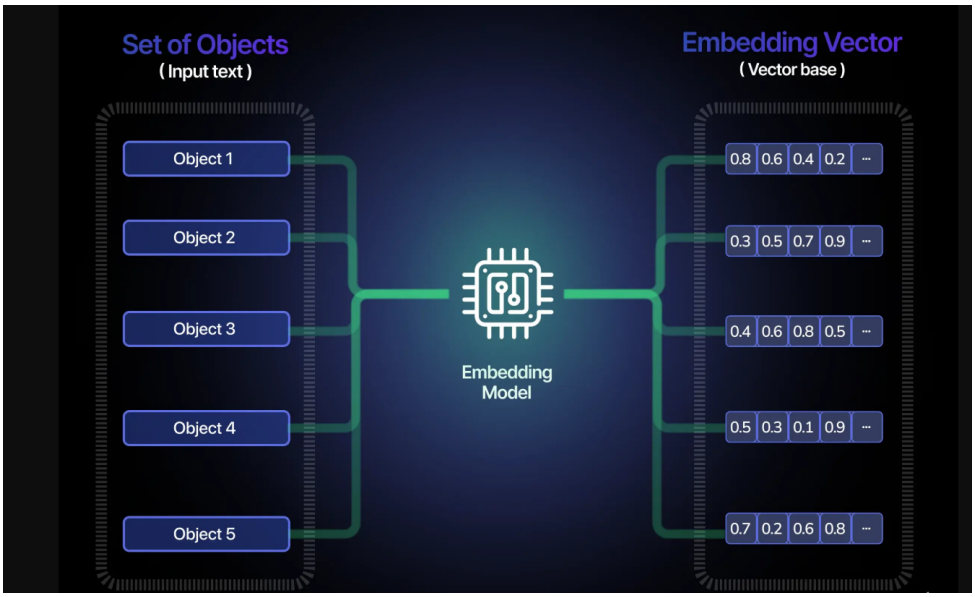


Figure 3.5: Visual Representation of Word Embeddings

4.1.2 Sentence Embeddings

Sentence embeddings represent the meaning of entire sentences or paragraphs in a single vector. Models such as the Universal Sentence Encoder and Sentence-BERT (SBERT) generate these embeddings by aggregating word vectorseither through averaging, pooling, or more complex techniquesto capture the overall semantic content of the sentence.

Example:

The sentence "My phone is good" would be converted into a vector that reflects the complete meaning of the sentence as a whole, rather than focusing on individual words in isolation.

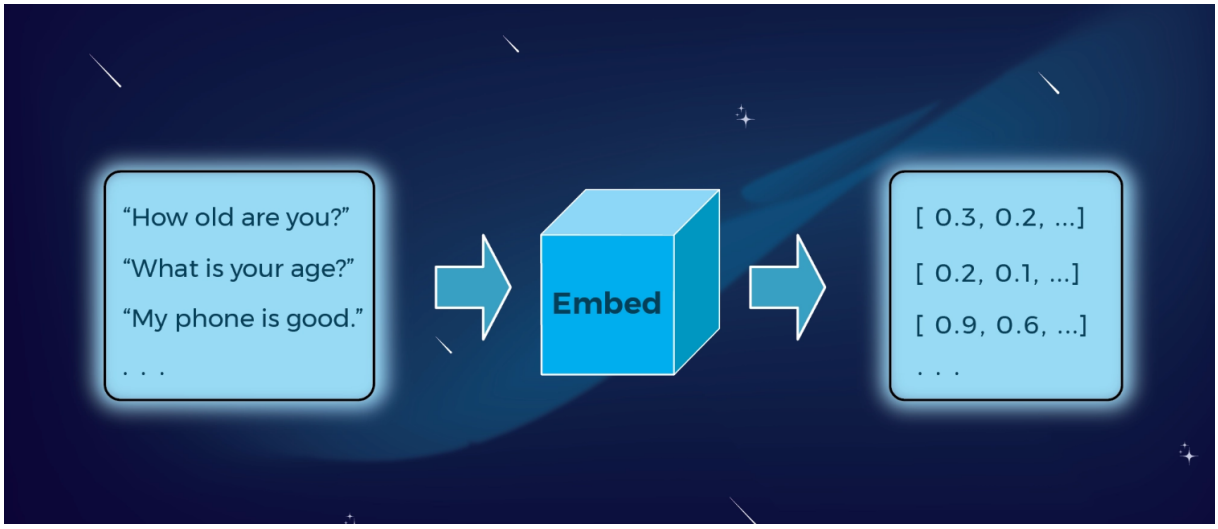


Figure 3.6: Illustration of Sentence Embeddings Capturing Semantic Meaning

4.1.3 Contextual Embeddings:

Contextual embeddings generate dynamic vector representations that vary based on a words or sentences context. Models like BERT and Arabic BERT produce these embeddings by leveraging transformer-based architectures, capturing nuanced meanings in complex texts.

For example, consider the Arabic word ") "هدىguidance):

In the verse:

(إِنَّ هَٰذَا ٱلْقُرْآنَ يَهْدِي لِلَّتِي هِيَ أَقْوَمُ) (الاسراء: 9)

The word "يهدي "refers to spiritual and moral guidance provided by the Qur'an.

In contrast, in the verse:

(وَجَعَلْنَا لَهُمْ سَمْعًا وَأَبْصَارًا وَأَفْئِدَةً)(السجدة: 9)

The focus is on the intellectual faculties (hearing, sight, and hearts),

implying a form of intellectual or rational guidance.

Although the concept of guidance appears in both verses,

contextual embeddings would assign different vector representations to the word "هدى"or its derivatives like "يهدي" in each case, reflecting the nuanced difference in meaning.

4.2 Applications of Embedding Models for Quranic Texts:

Following the discussion of embedding types, their applications in analyzing Quranic texts are diverse and impactful. Embedding vectors serve as the input for several downstream tasks, including:

- **Semantic Search:** Using sentence embeddings to retrieve verses with similar meanings, such as finding verses about mercy like

  "ورحمتي وسعت كل شيء" (الأعراف: 156).

- **Text Classification:** Employing word and sentence embeddings to categorize verses by themes, such as faith or charity, for automated organization.

- **Interpretation Analysis:** Leveraging contextual embeddings to compare tafsir texts, identifying differences in interpretations of verses like

  "إنما المؤمنون إخوة" (الحجرات: 10)

- **Topic Modeling:** Applying embeddings to uncover themes like paradise or prophethood across the Quran.

  By transforming Quranic texts into numerical vectors while preserving their semantic richness, embedding models revolutionize the analysis and interaction with the Quran.

Traditional representations such as One-hot Encoding or TF-IDF relied on statistical measures (e.g., word frequency counts) but ignored the context in which words appear, thus failing to provide accurate information about meaning. In this research, however, an advanced numerical representation derived from a Transformer-based model, namely LaBSE (Language-agnostic BERT Sentence Embedding).

4.3 LaBSE for Quran Embedding

LaBSE (Language-agnostic BERT Sentence Embedding) is a multilingual deep learn ing model developed by Google in 2020. It is based on a bidirectional Transformer ar chitecture. Its main purpose is to generate precise and unified numerical representations (embeddings) for sentences across multiple languages. [61]

LaBSE can convert sentences from different languages into vectors that lie within the same numerical space, enabling direct comparison and understanding regardless of the language used.

The model leverages advanced techniques, including self-attention mechanisms, to capture the full contextual meaning of each word within a sentence, allowing it to grasp deep semantic information.

These features make LaBSE highly suitable for applications such as translation, mul tilingual search, and topic modeling. It is particularly effective in analyzing linguistically rich texts, including religious scripture like the Quran.

It is especially suitable for studies requiring accurate analysis of linguistically complex texts, including religious texts like the Quran.

4.3.1 LaBSE Workflow and Technical Mechanism

This section describes the detailed technical workflow of LaBSE, highlighting each key processing step it performs to generate high-quality multilingual sentence embeddings.

- **Tokenization:** The input sentence (e.g., a Quranic verse) is first tokenized using a subword tokenizer such as WordPiece. This process breaks words into smaller subword units to ensure even rare or previously unseen words are effectively encoded.
- **Input Embedding:** Each token is mapped to a dense continuous vector through an embedding layer. In addition, positional encodings are added to these vectors to preserve the order of words in the sentence, which is essential for understanding context.
- **Transformer Encoding with Self-Attention:** The token embeddings are then passed through multiple layers of Transformer encoders. Each layer employs a self attention mechanism, allowing every token to attend to all other tokens in the sequence.

  This enables the model to capture both local and global contextual relationships between words, which is crucial for comprehending semantically rich sentences such as Quranic verses.
- **Sentence Embedding Extraction:** After processing through Transformer layers, each token has a contextually enhanced representation. To obtain a single fixed-length vector representing the whole sentence, LaBSE applies average pooling across all token vectors, producing a 768-dimensional embedding that encapsulates the sentences overall semantic meaning.
- **Multilingual Alignment (During Training Phase):** During training, LaBSE uses a dual-encoder architecture. It independently encodes pairs of parallel sentences (i.e.,translations) and minimizes the distance between embeddings of sentences with equivalent meanings. This alignment ensures that semantically similar sentences in different languages are projected close to each other in the shared embedding space.
- **Output and Usage:** The resulting embeddings serve as powerful numerical repre sentations ready for downstream natural language processing tasks such as clustering,semantic similarity measurement, topic modeling (e.g., BERTopic), and cross-lingual in formation retrieval.
- **Relation to This Study:** In this study, LaBSE was used solely to encode Quranic verseswithout their interpretationsinto dense, context-sensitive embeddings. These vectors provided the semantic foundation for applying the BERTopic model to cluster verses and extract latent themes.

  These embeddings formed the semantic foundation for the BERTopic model, which clusters semantically similar verses and extracts underlying themes. By leveraging LaBSEs.

5 Bidirectional Encoder Representations from Transformers(BerTopic)

After obtaining the sentence embeddings using the LaBSE model, a modern topic modeling approach was employed BERTopic which leverages embeddings generated by transformer-based models.

BERTopic surpasses traditional topic modeling techniques like LDA, which depend mainly on word co-occurrence and frequency. Instead, it captures deeper contextual and semantic relationships within the text, making it particularly suitable for semantically rich content such as Quranic verses.

The workflow begins with dimensionality reduction using UMAP (Uniform Manifold Approximation and Projection), which transforms high-dimensional sentence embeddings into a lower-dimensional space to enable effective clustering. Next, clustering is performed using HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), a robust algorithm capable of discovering dense clusters while identifying and excluding noise or outliers.

Once clusters are identified, class-based TF-IDF (c-TF-IDF) is used to extract representative keywords for each cluster, thus defining the underlying topic. This process results in coherent thematic groupings that reflect the shared semantic essence of the verses.

In this study, BERTopic was applied using precomputed embeddings generated from LaBSE, which had encoded both the Quranic verses.

BERTopic, introduced by Maarten Grootendorst in 2020, integrates deep contextual understanding with unsupervised learning. It has gained popularity for its flexibility and effectiveness in handling complex, multilingual, and semantically diverse textual data.

By applying this method, the study successfully identified topics such as divine power, stories of prophets, the afterlife, and moral teachingsproviding a more nuanced thematic analysis of the Quranic content [62].

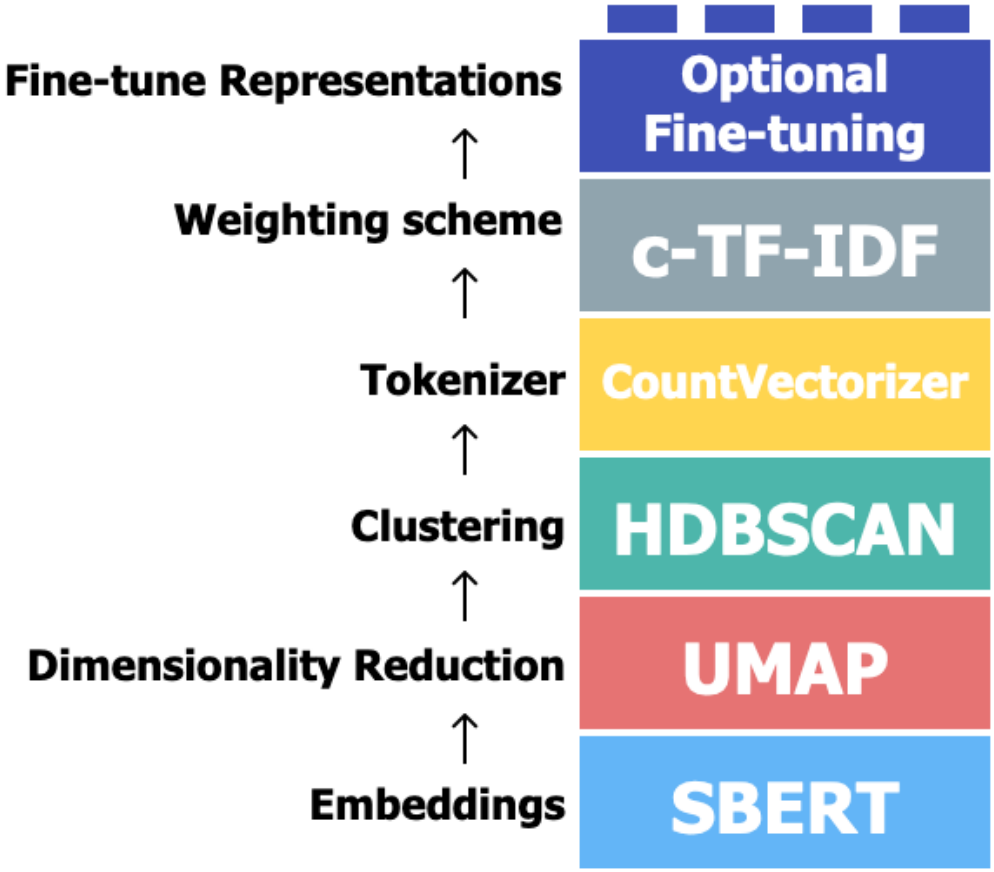


Figure 3.7: Workflow of BERTopic Topic Modeling Process [63]

5.1 Steps of BERTopic Model

The process of BERTopic model follows several key steps as illustrated in Figure 3.7

5.1.1 Dimensionality Reduction

In advanced machine learning models like BERTopic, dimensionality reduction plays a crucial role in facilitating data analysis and accurately identifying topics. After obtaining numerical representations (embeddings) of texts from a model such as LaBSE, these embeddings are typically high-dimensional (e.g., 768 dimensions per sentence).

This high dimensionality complicates clustering operations and makes it difficult to visualize or explore the data efficiently [64].

To address this, advanced dimensionality reduction techniques are used, such as UMAP (Uniform Manifold Approximation and Projection), a nonlinear method designed

to reduce the number of dimensions while preserving the local and global structure of the data.

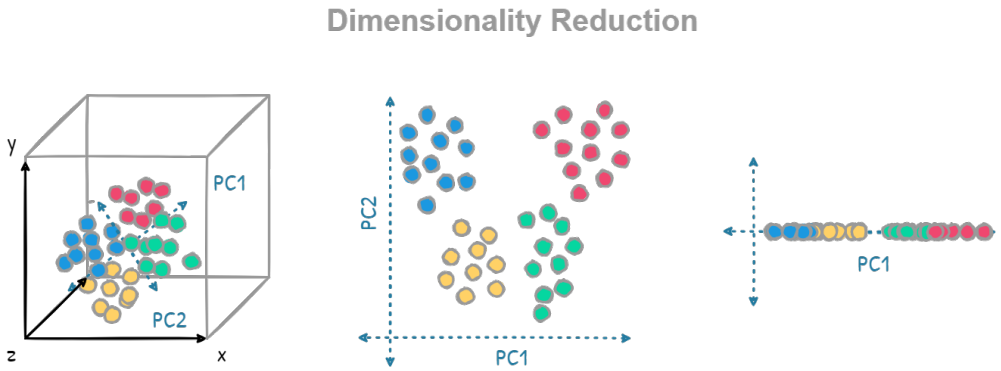


Figure 3.8: Dimensionality Reduction:
[63]

- **General Concept of Dimensionality Reduction:** The goal of dimensionality reduction is to project high-dimensional numerical representations into a lower-dimensional space (e.g., from 768 to 5 dimensions). This projection maintains the contextual relationships and semantics between sentences, ensuring that semantically similar sentences remain close together in the new space.

  Without this step, clustering becomes slower and less effective, especially with large datasets such as collections of Quranic verses.

- **Common Dimensionality Reduction Techniques:** Before selecting the appropriate technique for dimensionality reduction, it is important to understand the various available methods. Several widely used techniques exist, each with its advantages and limitations depending on the nature of the data:

  – **Principal Component Analysis (PCA):** A linear technique that projects data into a lower-dimensional space based on the greatest variance. However, it assumes linear relationships between variables, making it less effective in representing the complex contextual relationships found in textual data such as Quranic verses.

- **t-distributed Stochastic Neighbor Embedding (t-SNE):** A nonlinear technique commonly used for visualizing data. It produces good results for 2D visualization but lacks scalability for large datasets and is sensitive to parameter settings, making it less stable for automated clustering.

- **Uniform Manifold Approximation and Projection (UMAP):** A relatively recent technique based on manifold learning. It preserves both local and global data structure and is more efficient than t-SNE in terms of speed and scalability. It also produces more consistent results in automated clustering scenarios.

- **Choosing UMAP as the Preferred Method:** UMAP was selected for this work due to its superior ability to preserve local data structure compared to traditional methods like PCA. Unlike PCA, which assumes linear relationships among features, UMAP can represent complex, nonlinear relationships, making it more suitable for deep contextual embeddings such as those produced by LaBSE.

  UMAP organizes points in such a way that sentences with similar meanings in the original high-dimensional space remain close in the reduced space. This is ideal for grouping verses with similar themes, such as those discussing Paradise or verses describing punishment.

- **Applying UMAP in BERTopic:** First, the numerical embeddings generated by the LaBSE model are passed to the UMAP algorithm. The target number of dimensions is specified (typically between 2 and 5), along with key parameters such as:

  - **n neighbors:**

    Determines the number of local neighbors used to estimate the data structure.

  - **min dist:**

    Controls the minimum distance between points in the low-dimensional space (a smaller value results in tighter clusters).

  The result is a new set of low-dimensional representations, ready for use in the clustering stage.

– **Practical Example of Using UMAP:**

Consider the following three verses:

A Paradise as vast as the heavens and the earth	جنة عرضها السماوات والأرض
Indeed the righteous will be in a secure place	إن المتقين في مقامٍ أمين
And He has prepared for them a painful punishment	وأعدّ لهم عذابًا أليمًا

After passing them through LaBSE, we obtain numerical embeddings that are not directly suitable for clustering. By applying UMAP, these embeddings are projected into a 2D space where the first two verses appear close together (as they both refer to Paradise), while the third verse appears further away due to its focus on punishment.

– **Benefits of Dimensionality Reduction:**

* Improves the performance of the HDBSCAN clustering algorithm by reducing computational overhead.
* Facilitates data visualization for thematic maps or result analysis.
* Reduces noise and increases the coherence of resulting clusters, making topic extraction more accurate and meaningful.

5.1.2 Clustering in BERTopic

In the context of BERTopic, clustering refers to the process of grouping together textual data points (sentences or documents) that share semantic similarity. After the dimensionality of the sentence embeddings is reduced using UMAP, clustering algorithms are applied to identify coherent groups that reflect underlying thematic structures.

The primary purpose of clustering in BERTopic is to enable the identification of latent topics based on the proximity of sentence meanings. Sentences that discuss similar themes (e.g., verses about Paradise or punishment) tend to form distinct clusters, each representing a topic.

- **Types of Clustering Algorithms:**

  There are several clustering methods in machine learning, and each has its own strengths and ideal use cases. The most common types include: [65]

  - **K-means:** A centroid-based algorithm that partitions data into k clusters by minimizing intra-cluster variance. It requires predefining the number of clusters and assumes that clusters are spherical and of similar size.

  - **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** A density-based method that identifies clusters based on the density of data points. It handles noise and outliers well but struggles with clusters of varying density.

  - **HDBSCAN (Hierarchical DBSCAN):** An extension of DBSCAN that constructs a hierarchy of clusters and selects the most stable ones. It does not require specifying the number of clusters beforehand and can detect clusters of varying densities and shapes.

- **Why HDBSCAN Was Used in BERTopic:** BERTopic employs HDBSCAN for clustering because it offers flexibility, scalability, and robustnessespecially for real-world, noisy, and high-dimensional textual data. Unlike k-means, HDBSCAN does not require setting the number of clusters, which is advantageous for unsupervised topic modeling where the number of topics is unknown.

  HDBSCAN is also more robust than DBSCAN when handling datasets with varying density, such as Quranic verses that differ in length and semantic richness. Its ability to mark outliers (e.g., verses not clearly belonging to any topic) adds value in improving topic coherence.

- **Advantages of HDBSCAN Over K-means and DBSCAN:** Compared to k-means, HDBSCAN does not assume spherical clusters and is better suited for the irregular shapes common in semantic spaces. It is also non-parametric regarding the number of clusters. Unlike DBSCAN, HDBSCAN introduces a hierarchy and selects clusters based on stability, allowing it to adapt better to different data distributions. Additionally, HDBSCAN can handle noise, ignore ambiguous points, and produce more meaningful and fine-grained topic groupings.

- **How HDBSCAN Works (Brief Overview):** HDBSCAN begins by building a minimum spanning tree based on the distance between points in the reduced space (typically UMAP output). It then converts this into a hierarchy of clusters based on the relative density between data points. From this hierarchy, the algorithm selects clusters that are the most stablethat is, those that persist across multiple levels of the hierarchy.

This stability-based selection helps ensure that resulting clusters are not artifacts of the algorithm's parameters but reflect inherent structures in the data.

- **Key Parameters Affecting HDBSCAN:** Several parameters influence the output of HDBSCAN. The most important include:

  - **min cluster size:**The minimum number of data points required to form a cluster. Smaller values allow finer topic distinctions, while larger values produce broader topics.
  - **min samples:** Influences the definition of core points and affects sensitivity to noise.
  - **metric:** Determines how distances between points are calculated (e.g., Euclidean or cosine). In textual data, cosine similarity is typically used.

  These parameters can significantly affect the quality and granularity of the resulting topics.

- **Impact of Clustering on Topic Identification :** Clustering is a central component of BERTopic, as it defines the boundaries between different topics. Accurate clustering ensures that sentences grouped together share meaningful semantic themes, which improves the relevance and coherence of extracted topics.
  Poor clustering, by contrast, leads to mixed or incoherent topics. By using HDBSCAN, BERTopic is able to form nuanced clusters of similar texts, allowing for the extraction of detailed and specific themes from large collections of textual data such as the Quran.

5.1.3 Tokenization Using CountVectorizer

In the context of topic modeling with BERTopic, the tokenization step is essential after clustering the verses using HDBSCAN. The objective of this phase is to convert the grouped texts (Quranic verses categorized under a given topic) into a numerical format that can later be used to extract the most representative keywords for each topic. To achieve this, the **CountVectorizer** tool is employed.

- **CountVectorizer:**

  CountVectorizer is a component from the **scikit-learn** library. It transforms textual data into a matrix of token counts. The process includes:

  - **Tokenization**: Each verse or document is split into individual tokens (words).
  - **Vocabulary building**: A unique vocabulary is constructed from all words across all topics.

- **Vectorization**: Each group of verses (topic cluster) is represented by a vector indicating the frequency of each word in the vocabulary.

- **Important of Tokenization:**

This numerical representation is crucial for computing the `c-TF-IDF` weighting scheme, which determines how representative each word is for a given topic compared to others. Without tokenization and count-based vectorization, the topic model cannot identify distinguishing terms effectively.

Example: Consider a topic cluster on "Paradise" "EL jana" with the following words appearing in its verses: **Gardens "EL JANAT", Bliss "EL NAIME", Reward "EL JAZAA", Rivers "ANHAR"**.

After applying `CountVectorizer`, we might get the following frequency table:

Word	Frequency
gardens	5
bliss	3
reward	2
rivers	4

Table 3.1: TF-IDF Example

This output is then used in the `c-TF-IDF` calculation to extract top keywords for the topic.

- **Relation to Other Steps:**

  This tokenization stage:

  - Comes **after** clustering with HDBSCAN,
  - Precedes and enables the **c-TF-IDF** weighting step,
  - Plays a vital role in forming interpretable and semantically coherent topic labels.

- **Importance for Quranic Texts**

For Quranic verses, where meaning is often conveyed through powerful and concise expressions, tokenization helps capture frequently used theological, ethical, or spiritual terms within each topic cluster. This leads to more accurate identification of core themes across the Quran, enhancing interpretability and supporting thematic analysis for both academic and religious research.

"CountVectorizer" is thus a foundational step in bridging textual data and structured semantic understanding.

5.1.4 Topic Representation

After completing the clustering phase, where Quranic verses or input texts are grouped into semantically similar clusters, the next essential step is topic representation. The objective of this phase is to identify the most representative and distinctive words within each cluster so that they reflect the core thematic content of the group. These keywords are later used to label or describe the topics accurately [62].

BERTopic uses a technique called class-based TF-IDF (c-TF-IDF), which is inspired by the traditional TF-IDF method but adapted specifically to extract representative keywords at the class or topic level rather than for individual documents.

- **Concept of c-TF-IDF:**

  This method treats each cluster of documents (e.g., verses grouped under the same topic) as a single document and computes the importance of each term within that "topic document" compared to other clusters. The more frequently a word appears in one topic and the less it appears in others, the more representative it is of that topic.

- **Steps of Topic Representation Using c-TF-IDF:**

  1. **Cluster Document Aggregation:** All texts that belong to the same topic are concatenated into a single large document.
     For example:
     if a cluster contains five verses about Paradise (Jannah), they are combined into one document that represents that topic.
  2. **Term Frequency (TF):** The number of times each word appears in the topic document is counted.
     For example:
     if the word "Paradise" appears five times in the cluster document, then
     TF = 5. This frequency indicates the words importance within its own topic.
  3. **Inverse Document Frequency (IDF):**

     The number of topic documents that contain the same word is calculated.
     If "Paradise" appears only in one topic out of ten, it is highly distinctive.
     In contrast, if it appears in most or all clusters, it is considered a generic term.
     The IDF is calculated using the formula:

     IDF = log(Number of topics /(Number of topics containing the word+1))

4. **c-TF-IDF Score Calculation:**

   The final c-TF-IDF score for each word is calculated by multiplying its TF by its IDF.

   A higher score means the word is both frequent in the topic and rare in others thus, more representative.

5. **Top Terms Selection:**

   Words within each topic are ranked based on their c-TF-IDF scores.

   A fixed number of top words (e.g., top 5 or top 10) are selected to represent each topic.

   These selected words are used to label and interpret the topics meaning.

- **Applied Example:**

  Assume that a cluster contains verses that speak about Paradise. After applying the c-TF-IDF process, the following top representative words might be extracted:

  - "Paradise": c-TF-IDF = 8.21
  - "Bliss": c-TF-IDF = 6.15
  - "Rivers": c-TF-IDF = 4.89
  - "Righteous": c-TF-IDF = 3.97
  - "Divine pleasure": c-TF-IDF = 3.52

  These results allow us to interpret and label this cluster as related to the theme of "Paradise.

- **Importance of This Phase:**

  This step is essential for making the results of BERTopic interpretable and meaningful.

  It enables researchers to provide semantic descriptions for each discovered topic.

  It also facilitates visualization when presenting topic maps or analyzing thematic structures.

- **Advantages of Using c-TF-IDF:**

  Captures topic-specific context and performs well even with short texts like Quranic verses.

  Fast to compute and easy to implement.

  Produces stable and reproducible results, which is crucial in academic research.

5.1.5 Visualizing Topics in BERTopic

After completing the processes of topic extraction and defining their textual representation, the visualization phase becomes essential for deepening the understanding and interpretability of the results. Far from being a mere aesthetic addition, topic visualization in BERTopic provides powerful analytical tools that allow researchers to:

1. Comprehend the overall structure of discovered topics.
2. Assess the coherence and quality of clustering.
3. Detect overlaps or similarities between different topics.
4. Evaluate whether the clusters reflect genuine semantic distinctions in the textual data.
5. Interact with the data in a visually intuitive way to better interpret the findings.

- **Main Visual Tools Provided by BERTopic:**

  1. **2D Document Scatter Plot:**
     This tool projects each documentsuch as a Quranic verseinto a two-dimensional space using UMAP to reduce the high-dimensional embeddings. Each point is colored based on the topic it belongs to.
     This representation allows the researcher to visually identify clusters of documents that belong to the same topic and to assess how well-separated or overlapping the topics are. It is especially useful for visually validating the performance of the topic modeling.
     To generate this visualization, you can use the following command:

     topic model.visualize documents(docs, embeddings)

     Here, docs represents the list of verses, and embeddings are the numerical representations using LaBSE.
  2. **Topic Bar Chart (Representative Keywords):**
     This bar chart displays the most representative words or phrases for each topic, based on the c-TF-IDF algorithm. It can also display multiple topics at once for comparison.
     It is valuable for interpreting the semantic core of each topic and distinguishing between closely related topics. You can generate it with:
     topic model.visualize barchart(top n topics=5)
     You can change the number of topics shown by modifying the value of top n topics.

3. **Topic Similarity Matrix / Hierarchical Clustering:**
   This visualization computes the similarity between topics based on their vector representations and displays it as either a heatmap or a hierarchical dendrogram.

   It helps identify semantically close topics and supports decisions such as merging or filtering topics that are overly similar. You can use:
   topic model.visualize heatmap()
   topic model.visualize hierarchy()

- **Applied Example: Quranic Verses**

Assume that we have clustered a collection of Quranic verses into five main topics. When plotting the 2D document scatter plot, we observe that verses related to the topic "Paradise" (Al-Jannah) form a tight cluster in one region, while those related to "Punishment" (Al-Adhab) form a distinct cluster in another.

Using the bar chart, we see that the topic of "Paradise" is associated with representative keywords like:

- **gardens** (جنات)
- **rivers** (أنهار)
- **pleasure of God** (رضوان الله)
- **bliss** (نعيم)

In contrast, the "Punishment" topic features terms such as:

- **fire** (نار)
- **torment** (عذاب)
- **Hell** (جهنم)

These visualizations provide tangible confirmation of the model's effectiveness and help fine-tune topic definitions if necessary.

- **Importance of the Visualization Phase:**
  - Offers a visual evaluation of topic coherence and distribution.
  - Helps refine topic modeling by identifying weak or redundant topics.
  - Supports final interpretation and narrative construction of extracted topics.
  - Facilitates presenting the results to non-technical stakeholders or in research reports.
  - Enables interactive analysis when building dashboards or tools for scholars or language experts.

Chapter 4

Experimental Results

1 Introduction

In this chapter, we present the results and evaluation of the proposed model for clustering Quranic verses according to different topics using the BERTopic algorithm. The model was used to group verses into semantically coherent topic clusters. The evaluation was conducted based on a set of metrics to assess how effectively the model classifies the verses in a logical and consistent manner.

In addition, we perform a comparative analysis between the results of clustering Quranic verses and the results of clustering regular Arabic texts. This comparison aims to explore the impact of the unique nature of the Quranic text on the performance of topic modeling algorithms, particularly in terms of topic coherence, diversity, stability, and accuracy.

2 Evaluation Metrics

In this section, we present a detailed overview of the metrics used to evaluate the performance of the proposed model in classifying verses of the Quran into different topics. Four main indicators were adopted to assess the quality of topic clustering: Coherence Score, Diversity Score, Stability Score, and Accuracy.

2.1 Coherence Score

To measure the semantic cohesion within each topic, a simplified method was employed based on calculating the average cosine similarity between the top keywords. The top 10 words for each topic were extracted, and the verses were converted into numerical representations using a Bag-of-Words model restricted to these words.

The similarity between each word pair was then computed based on their distribution across different verses, and the arithmetic mean of these similarities served as an indicator of topic coherence. The overall average was then calculated for all topics [66].

Pointwise Mutual Information (PMI):

In addition to cosine similarity, Pointwise Mutual Information (PMI) was used to evaluate how strongly two words are associated within a given context compared to what would be expected if they were independent. It is defined as:

$$PMI(x, y) = \log(\frac{PMI(x, y)}{P(x) \times P(y)})$$

Where:

P(x,y) is the probability that both words x and y co-occur.

P(x) and P(y) are the probabilities of each word occurring independently.

For example:

if the word "Allah" appeared in 100 out of 200 verses, and "Rahim" (the Merciful) in 80 verses, and they co-occurred in 60 verses, the probabilities are calculated as follows:

- P("Allah") = 0.5
- P("Rahim") = 0.4
- P("Allah", "Rahim") = 0.3

PMI is a statistical measure used to evaluate how strongly two words are associated within a given context compared to what would be expected if they were independent. A positive PMI value suggests that the words co-occur more frequently than expected by chance, enhancing topic coherence. Values close to zero indicate weak or no association, while negative values suggest a rare co-occurrence. This measure has been widely adopted in computational linguistics for tasks like topic modeling and word similarity [67].

$$PMI(\text{"}Allah\text{"},\text{"}Rahim\text{"}) = \log \frac{0.3}{0.5 \times 0.4} = log(1.5) \approx 0.176$$

This value indicates a higher-than-random association, which is a positive indicator of coherence.

2.2 Diversity Score

This metric aims to measure how distinct each topic is from the others in terms of key vocabulary. It is calculated through the following steps:

- Extract the top 10 words for each topic.
- Aggregate all these words into one list.
- Count the number of unique words in the list.
- Calculate the percentage of unique words relative to the total using the formula:

  Diversity = (Number of unique words / Total number of words) Œ 100

  A higher diversity score indicates that topics are more lexically distinct, reducing redundancy and increasing interpretability. This approach is commonly used to evaluate clustering quality and topic distinctiveness in text mining tasks [68].

Example: If there are 10 topics, the total number of words is 100, and the number of unique words is 80, then:

$$Diversity = \frac{80}{100} * 100 = 80\%$$

A high diversity score indicates a clear distinction between topics, which is desirable in topic modeling.

2.3 Stability Score

This metric evaluates how stable the results are when the model is rerun on the same data with slight changes in initialization or ordering.
It is measured by comparing the resulting keywords from each run using the **Jaccard similarity coefficient [69]:**

$$Jaccard = \frac{|A \cap B|}{|A \cup B|}$$

Example: If the keywords in the first run are "Allah", "Rahim", "Malik" and in the second run "Allah", "Rahman", "Malik", then:

$$Jaccard = \frac{2}{4} = 0.5$$

The closer this average is to 1, the more stable the model is in reproducing similar topics with each run.

2.4 Accuracy

Accuracy measures the degree to which BERTopics classifications align with manually prepared reference classifications, using the formula [70]:

Accuracy = Number of correctly classified verses / Total number of verses
Example: If there are 100 verses and 82 are correctly classified:

$$Accuracy = \frac{82}{100} = 82\%$$

A higher value indicates better performance of the model in identifying the correct topic for each verse.

3 Results Presentation and Evaluation

This section presents the results of applying the BERTopic model, supported by LaBSE embeddings, to a set of Quranic verses manually categorized into three main topics. The model was evaluated using the four criteria mentioned earlier.

Metric	**Value**
Number of Extracted Topics	3 topics
Coherence Score	0.2110
Diversity Score	0.8500
Stability Score	0.8333
Accuracy	36.81%

Table 4.1: Evaluation metrics of the BERTopic model

The average coherence score reached 0.2144, a moderate value reflecting partial semantic connectivity among words within each topic. The relatively low score is attributed to the complex rhetorical nature of the Quranic text, which does not follow an explicit vocabulary structure like general texts. Moreover, the use of a simplified method rather than a formal coherence metric like C V may also play a role.

When the model was rerun with minor changes in initialization, the stability score reached 0.8056, which is a high value indicating that the model is consistent and capable of reproducing similar topics.

Out of 144 groups of verses., 53 were correctly classified according to the manual labels, yielding an accuracy of 36.81%. Although the percentage is relatively low, it can be explained by several factors:

- The reference classifications were based on documents containing multiple verses, which caused overlap between topics.
- A "noisy" topic labeled "-1" was included, representing an undefined category.
- The model relies on unsupervised clustering, making it inherently less accurate than supervised models.

4 Analysis of Topics Extracted by the BERTopic Model

In this chapter, the topics extracted by the model were not previously discussed. Based on the model results, the following topics were identified:

	Topic	Count	Name	Representation	KeyBERT
0	-1	4	أنزلنا_أنزلناه_أقسم_عربيا_1-	[...أنزلنا, أنزلناه, أقسم, عربيا, قرآنا عربيا, أن]	[...ورحمة لقوم يؤمنون, آمنوا وعملوا الصالحات, وهد
1	0	96	قال_خير_الدنيا_قوم_0	[...قال, خير, الدنيا, قوم, شيئا, أصحاب, الآخرة, ا]	[...آمنوا وعملوا الصالحات, آمنوا وعملوا, الله ورس
2	1	28	قال_إبراهيم_مريم_قوم_1	[... ,قال, إبراهيم, مريم, قوم, ربه, كنت, قال موسى]	[...آمنوا وعملوا الصالحات, إبراهيم, عبادنا, وعملو
3	2	20	الليل_ماء_قدير_السماوات الأرض_2	[...الليل, ماء, قدير, السماوات الأرض, السماء ماء]	[...السماوات والأرض الله, والأرض الله, الله شيء ق

Figure 4.1: Overview of Extracted Topics from Quranic Verses

4.1 Topic -1: Undefined Topic (Number of Verses = 4)

index	Topic	Count	Name	Representation	KeyBERT
0	-1	4	أنزلنا_أنزلناه_أقسم_عربيا_1-	أنزلنا,أنزلناه,أقسم,عربيا,قرآنا عربيا,أنزل الكتاب,مصدقا,ذكرا,مصدقا يديه,مبارك	ورحمة لقوم يؤمنون,آمنوا وعملوا الصالحات,وهدى ورحمة لقوم,القرآن لأنذركم,يخافون يحشروا ربهم,أوحينا قرآنا عربيا,آمنوا وعملوا,الله الكتاب,ورحمة لقوم,وأقاموا الصلاة

Figure 4.2: Topic -1: Undefined Theme (4 Verses)

Analysis:

This group includes verses referring to the revelation of the Quran and its linguistic characteristics. However, the small size and varied contexts of the verses caused the model to classify them under an undefined topic

4.2 Topic 0:Guidance and the Hereafter(Number of Verses = 96)

Index	Topic	Count	Name	Representation	KeyBERT
0	0	96	0_قال_خير_الدنيا_قوم	قال,خير,الدنيا,قوم,شيئا,أصحاب,الآخرة,الحياة الدنيا,الشيطان,الحياة	آمنوا وعملوا الصالحات,آمنوا وعملوا,الله ورسوله,المؤمنون,الظالمين,الله والله,وعملوا الصالحات,كانوا يعملون,ظلموا,المنافقين

Figure 4.3: Topic 0: Guidance and the Hereafter (96 Verses)

Analysis:

This cluster contains verses that include divine guidance and comparisons between worldly life and the hereafter. The large size of the group suggests potential overlap between subtopics. However, it can be classified under the theme of legal rulings (Ahkam) due to its underlying legislative and educational content.

4.3 Topic 1: Stories of the Prophets Ibrahim, Maryam, and Others (Number of Verses = 28)

index	Topic	Count	Name	Representation	KeyBERT
0	1	28	1_قال_إبراهيم_مريم_قوم	قال,إبراهيم,مريم,قوم,ربه,كنت,قال موسى,رسولا,أوحينا,فقال	آمنوا وعملوا الصالحات,إبراهيم,عبادنا,وعملوا الصالحات,رسول الله,آمنوا وعملوا,وإسماعيل,النبيين,مسلمون,قالوا آمنا

Figure 4.4: Topic 1: Stories of Prophets Ibrahim, Maryam, and Others (28 Verses)

Analysis:

This cluster clearly focuses on the narratives of the prophets and notable figures, with repeated mentions of key characters, demonstrating the models ability to distinguish narrative patterns. Accordingly, it can be classified under the theme of Quranic stories (Al-Qasas) due to its narrative nature and content structure.

4.4 Topic 2: Signs of Divine Power in Creation (Number of Verses = 20)

index	Topic	Count	Name	Representation	KeyBERT
0	2	20	2_الليل_ماء_قدير_السماوات_الأرض	الليل,ماء,قدير,السماوات,الأرض,السماء,ماء,شيء,قدير,خلقكم,خلقنا,لآيات,يريد	السماوات والأرض الله,والأرض الله,الله شيء قدير,وعد الله حق,الله ربكم,الله إله,السماء ماء,السماوات الأرض,الأرض جميعا,الله شيء

Figure 4.5: Topic 2: Signs of Divine Power in Creation (20 Verses)

Analysis:

This group includes verses that encourage reflection on creation and divine power, showing clear semantic coherence. Based on the content, it can be categorized under the theme of Tawheed (Monotheism), as it emphasizes the oneness, greatness, and creative power of Allah.

5 Topic Visualization and Hierarchical Clustering

5.1 Top Words per Topic

The bar chart in Figure 4.6 shows the top keywords in each of the three extracted topics. It is clear that Topic 0 centers around worldly terms , Topic 1 is more related to prophetic narratives and Topic 2 focuses on cosmological and existential concepts .

```
] topic_model.visualize_barchart()
```

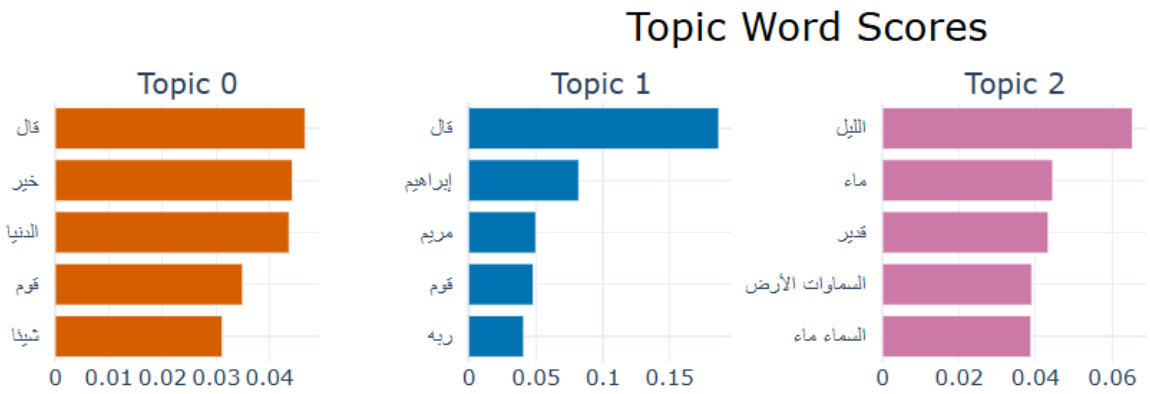


Figure 4.6: Top c-TF-IDF Keywords per Topic Extracted from Quranic Verses

5.2 Hierarchical Clustering of Topics

Moreover, the dendrogram in Figure 4.7 illustrates the semantic distances between these topics. It shows that Topic 0 and Topic 1 are relatively closer, suggesting they might share contextual themes, while Topic 2 is more distant, indicating distinct thematic content.

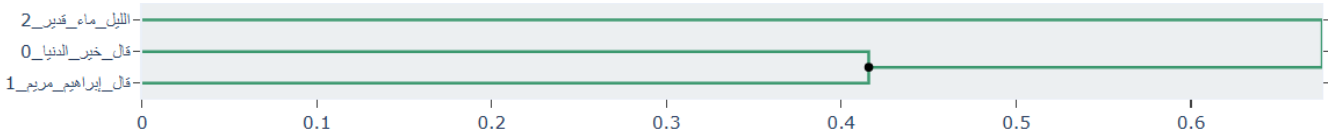


Figure 4.7: Hierarchical Clustering of Topics Based on Semantic Similarity

The experiment using BERTopic on Quranic text showed that the model was able to extract 3 topics from a total of 144 groups of verses. A total of 53 groups were correctly classified within these topics, yielding a classification accuracy of 36.81%. While some clusters reflected clear semantic coherence among the verses, the stylistic complexity and conceptual interweaving of the Quranic text posed challenges to the model in achieving high topical homogeneity across all outputs.

In contrast, the study by Abuzayed and Al-Khalifa (2021) [71] on general Arabic texts (articles, tweets, news) demonstrated that BERTopic could extract a greater number of topics (up to 20 in some experiments), with stronger internal coherence within each topic. This performance difference is attributed to the nature of the texts: general texts have a clear thematic structure and direct style, making them more suitable for topic modeling

algorithms like BERTopic. Quranic text, however, requires more sophisticated techniques to capture its unique rhetorical and semantic structure.

This comparison highlights that the effectiveness of topic modeling tools depends not only on the strength of the underlying algorithm but also on the characteristics of the input text. Therefore, more tailored tools and methods are needed when dealing with religious or highly stylized language.

6 Example of Topic Assignment for a Quranic Verse

To evaluate the performance of the topic model, we provided a test input consisting of a Quranic verse that describes the story of Prophet Yunus (peace be upon him). The input text was processed using **the transform()** function of BERTopic to determine the most probable topic cluster and the associated confidence score.

```
story = """
وَإِنَّ يُونُسَ لَمِنَ الْمُرْسَلِينَ، إِذْ أَبَقَ إِلَى الْفُلْكِ الْمَشْحُونِ، فَسَاهَمَ فَكَانَ مِنَ الْمُدْحَضِينَ، فَالْتَقَمَهُ الْحُوتُ وَهُوَ مُلِيمٌ، فَلَوْلَا أَنَّهُ كَانَ مِنَ الْمُسَبِّحِينَ، لَلَبِثَ فِي بَطْنِهِ إِلَى يَوْمِ يُبْعَثُونَ"""
_topic, _prob =  topic_model.transform([story])
```

Figure 4.8: Inputting the Quranic Verse and Using BERTopic for Topic Prediction

```
_topic, _prob
```

```
([np.int64(1)], array([0.45025654]))
```

Figure 4.9: BERTopic Classification Result

The model assigned the input to Topic 1 with a confidence score of 0.45
in Figure 4.9,indicating a moderate level of certainty in the topic classification. This result suggests that the content of the verse is semantically aligned with the verses grouped under Topic 1, which likely corresponds to themes such as Prophetic stories or Divine trials and deliverance.

Such an assignment demonstrates the models ability to generalize from learned topics and classify unseen or individual verses, reinforcing the effectiveness of BERTopic in organizing and thematically labeling Quranic content.

General conclusion

General Conclusion

This research aimed to explore the effectiveness of the BERTopic algorithm in performing unsupervised clustering of Quranic verses according to their semantic similarity. The LaBSE model was used to generate text embeddings for the verses, followed by the application of the BERTopic algorithm to group them into semantically related topics.

A manual reference clustering of the verses was carried out based on the book "Al-Jami' li Mawdu'at Al-Qur'an Al-Karim" (The Compendium of Quranic Topics), where similar verses were grouped into separate documents representing three main topics:**Monotheism (Tawheed), Stories,** and **Legal Rulings.** Interpretations (tafsir) were not used at this stage, making the experiment dependent solely on the raw text, allowing for an initial evaluation of the models ability to capture the semantic structure of the verses without external guidance.

The model demonstrated a notable capacity to uncover latent semantic structures, identifying three main topics:**Monotheism, Stories,** and **Creation.** Despite the partial overlap with the reference classification, there were differences in some conceptual areasmost notably, the absence of the "Legal Rulings" topic and the emergence of "Creation"reflecting the unsupervised nature of the model and its complete reliance on the semantic distribution of the verses.

The evaluation of the models results revealed a moderate coherence score (0.2110), a high stability score (0.8056), and an accuracy of 36.81% compared to the manual clustering. This relatively low accuracy is justifiable in light of the unsupervised approach and the Qurans unique nature, which is characterized by dense semantics and complex rhetorical structures.

This study's main strength is its novel use of the BERTopic model to analyze Quranic texts, showcasing how advanced NLP techniques can handle complex religious language. It also benefits from a semi-supervised clustering approach based on a credible scholarly source, ensuring alignment with established interpretations. Furthermore, it highlights the unique challenges machine learning models face when processing quranic and contextually rich texts.

Future Research Directions

- Integrating Quranic interpretations (tafsir) to provide a broader linguistic and contextual background for the model.

- Experimenting with Arabic embedding models specifically designed for religious or Quranic texts.
- Developing evaluation metrics better suited to the unique nature of Quranic language compared to general texts.
- Testing hybrid models that combine supervised and unsupervised techniques to improve clustering accuracy.

In conclusion, this work demonstrates that topic modeling tools like BERTopic are capable of uncovering semantic structures within the Quranic text. However, they require careful handling and the integration of external sources (such as tafsir or religious lexicons) to achieve results that more closely align with established Islamic concepts. This makes the field both promising and rich with future research potential.

Bibliography

[1] Jonas Schuett et al. A legal definition of ai. *arXiv preprint arXiv:1909.01095*, 4, 2019.

[2] Salvatore Claudio Fanni, Maria Febi, Gayane Aghakhanyan, and Emanuele Neri. *Natural Language Processing*, pages 87–99. Springer International Publishing, Cham, 2023.

[3] Annie. Nlp: How machine understands human language. https://medium.com/@annie_/nlp-how-machine-understands-human-language-009ec4000a66, 2019.

[4] IBM. What is natural language processing? https://www.ibm.com/think/topics/natural-language-processing, 2024.

[5] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[6] Annie. Nlp: How machine understands human language. https://medium.com/@annie_/nlp-how-machine-understands-human-language-009ec4000a66, 2019. Accessed: 2025-06-08.

[7] Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. A survey of natural language generation. *ACM Comput. Surv.*, 55(8), December 2022.

[8] IBM. What is natural language processing? https://www.wgu.edu/blog/12-applications-natural-language-processing2108.html, 2021.

[9] Subhashini Gupta and Grishma Sharma. Topic modeling in natural language processing. *International Journal of Engineering Research & Technology (IJERT)*, 10(06), 2021.

[10] Thomas Landauer, Peter Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 01 1998.

[11] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[13] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

[14] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[15] Mohammad Alhawarat and Mohamed Hegazi. Revisiting k-means and topic modeling, a comparison study to cluster arabic documents. *IEEE Access*, 6:42740–42749, 2018.

[16] Diab Abuaiadah. Using bisect k-means clustering technique in the analysis of arabic documents. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 15(3):1–13, 2016.

[17] Abdessalem Kelaiaia and Hayet Farida Merouani. Clustering with probabilistic topic models on arabic texts: a comparative study of lda and k-means. *Int. Arab J. Inf. Technol.*, 13(2):332–338, 2016.

[18] NM Ibrahim. A new model for arabic text clustering by word embedding and arabic word net. *J. Eng. Technol.*, 4:401–406, 2019.

[19] Haytham S Al-sarrayrih and Riyad Al-Shalabi. Clustering arabic documents using frequent itemset-based hierarchical clustering with an n-grams. In *Proc. Int. Conf. Inf. Technol.(ICIT)*, page 113, 2009.

[20] Hanan M Alghamdi, Ali Selamat, and Nor Shahriza Abdul Karim. Arabic web pages clustering and annotation using semantic class features. *Journal of King Saud University-Computer and Information Sciences*, 26(4):388–397, 2014.

[21] Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31, 2017.

[22] Y. Bengio, Holger Schwenk, Sofian Audry, F. Morin, and Jean-Luc Gauvain. *Neural Probabilistic Language Models*. 01 2005.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[24] Doaa Wahhab Alkhafaji and Sura Al-Rashid. A topic modeling for clustering arabic documents. In *2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, pages 76–81. IEEE, 2021.

[25] Christopher E. Moody. Mixing dirichlet topic models and word embeddings to make lda2vec. *CoRR*, abs/1605.02019, 2016.

[26] Mu Xue. A text retrieval algorithm based on the hybrid lda and word2vec model. In *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 373–376. IEEE, 2019.

[27] Menwa Alshammeri, Eric Atwell, and Mhd ammar Alsalka. Detecting semantic-based similarity between verses of the quran with doc2vec. *Procedia Computer Science*, 189:351–358, 2021.

[28] Mohammed Akour, Izzat M Alsmadi, and Iyad Alazzam. Mqvc: Measuring quranic verses similarity and sura classification using n-gram. 2014.

[29] Suhaib Kh Hamed and Mohd Juzaiddin Ab Aziz. A question answering system on holy quran translation based on question expansion technique and neural network classification. *J. Comput. Sci.*, 12(3):169–177, 2016.

[30] Salha Hassan Muhammed Qahl. *An automatic similarity detection engine between sacred texts using text mining and similarity measures*. Rochester Institute of Technology, 2014.

[31] Abdul-Baquee M Sharaf and Eric Atwell. Qursim: A corpus for evaluation of relatedness in short texts. In *LREC*, pages 2295–2302, 2012.

[32] Cepy Slamet, Ali Rahman, Muhammad Ali Ramdhani, and Wahyudin Darmalaksana. Clustering the verses of the holy qur'an using k-means algorithm. *Asian Journal of Information Technology*, 15(24):5159–5162, 2016.

[33] Aliyu Rufai Yauri, Rabiah Abdul Kadir, Azreen Azman, and MA Azmi Murad. Quranic verse extraction base on concepts using owl-dl ontology. *Research Journal of Applied Sciences, Engineering and Technology*, 6(23):4492–4498, 2013.

[34] Hussein Hashimi, Alaaeldin Hafez, and Hassan Mathkour. Selection criteria for text mining approaches. *Computers in Human Behavior*, 51:729–733, 2015.

[35] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[36] MPS Bhatia and Deepika Khurana. Analysis of initial centers for k-means clustering algorithm. *International journal of computer applications*, 71(5), 2013.

[37] Bhavna Bhardwaj. Text mining, its utilities, challenges and clustering techniques. *International Journal of Computer Applications*, 135(7):975–8887, 2016.

[38] NK Farooqui and MF Noordin. Document details. *Journal of Theoretical and Applied Information Technology*, 72(3):385–393, 2015.

[39] Mohammed A Alqarni. Embedding search for quranic texts based on large language models. *Int. Arab J. Inf. Technol.*, 21(2):243–256, 2024.

[40] Samia Zouaoui and Khaled Rezeg. A novel quranic search engine using an ontology-based semantic indexing. *Arabian Journal for Science and Engineering*, 46(4):3653–3674, 2021.

[41] Sumaira Saeed, Sajjad Haider, and Quratulain Rajput. On finding similar verses from the holy quran using word embeddings. In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pages 1–6. IEEE, 2020.

[42] Hammad Afzal and Tayyeba Mukhtar. Semantically enhanced concept search of the holy quran: Quranic english wordnet. *Arabian Journal for Science and Engineering*, 44:3953–3966, 2019.

[43] Google colaboratory. https://colab.google/.

[44] Bijay Kumar. Is python a high level language? https://pythonguides.com/is-python-a-high-level-language/, 2025. Accessed: 2025-05-27.

[45] Bijay Kumar. Python re library: A comprehensive guide. https://coderivers.org/blog/python-re-library/, 2025.

[46] Natural language toolkit. https://www.nltk.org/, 2024.

[47] Amal Alajmi, E Mostafa Saad, and RR Darwish. Toward an arabic stop-words list generation. *International Journal of Computer Applications*, 46(8):8–13, 2012.

[48] Taha Zerrouki. Pyarabic: A python package for arabic text. *Journal of Open Source Software*, 8(84):4886, 2023.

[49] Sentencetransformers documentation. https://www.sbert.net/, 2025.

[50] Wes McKinney. *pandas: powerful data analysis tools for Python*, 2020. Version 1.3.0.

[51] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. *scikit-learn: Machine Learning in Python*, 2021. Version 1.2.1.

[52] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. https://arxiv.org/abs/2203.05794, 2022.

[53] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. https://arxiv.org/abs/1802.03426, 2020.

[54] Leland McInnes, John Healy, Steve Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[55] Material for MkDocs. Representation models. https://maartengr.github.io/BERTopic/getting_started/representation/representation.html#partofspeech, 2024.

[56] Quran.com. About the quran. https://quran.com/about-the-quran, 2025.

[57] Mouhamed fares barkat. *Eljamia li mawathia ayat el quran el karim.* 1959.

[58] Vibhu Jawa and Mayank Anand. Accelerating topic modeling with rapids and bert models, March 2022. Accessed: 2025-06-27.

[59] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[60] GeeksforGeeks. What is text embedding? https://www.geeksforgeeks.org/what-is-text-embedding/, 2022. Accessed: 2025-06-08.

[61] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[62] Maarten Grootendorst. ctfidf bertopic api documentation. urlhttps://maartengr.github.io/BERTopic/api/ctfidf.html, 2025.

[63] MLGuru. Dimensionality reduction - mlguru, 2023.

[64] Laurens Van Der Maaten, Eric O Postma, H Jaap Van Den Herik, et al. Dimensionality reduction: A comparative review. *Journal of machine learning research*, 10(66-71):13, 2009.

[65] Deepti Sisodia, Lokesh Singh, Sheetal Sisodia, and Khushboo Saxena. Clustering techniques: a brief survey of different clustering algorithms. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 1(3):82–87, 2012.

[66] Federico Bianchi, Silvia Terragni, and Dirk Hovy. Pretrained language models for topic modeling: A comparative study. *arXiv preprint arXiv:2104.07586*, 2021.

[67] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[68] Jerome Friedman Trevor Hastie, Robert Tibshirani. *The Elements of Statistical Learning*. Springer, 2009.

[69] Derek Greene, Derek OCallaghan, and Pádraig Cunningham. How many topics? stability analysis for topic models. In *Machine Learning and Knowledge Discovery in Databases*, pages 498–513. Springer, 2014.

[70] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (3rd Edition Draft)*. Draft available online, 2023.

[71] Abeer Abuzayed and Hend Al-Khalifa. Bert for arabic topic modeling: An experimental study on bertopic technique. *Procedia Computer Science*, 189:191–194, 2021. AI in Computational Linguistics.