كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

جامعة حلوان
Helwan University

Helwan University
Faculty of Computers and Artificial Intelligence
Computer Science Department

Sunnah QA -
Building a Question Answering System for Sunnah

A Graduation Project Dissertation by:

Mohamed Ashraf Khalifa	202000729
Ahmed Ashraf Taha	202000013
Mohammed Zakaria Kamel	202000761
Mohammed Maher Hussein	202000813
Marwa Asaad Saeed	202000867
Mostafa Kamal Fahmey	202000907

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science, at the Computer Science Department, the Faculty of Computers & Artificial Intelligence. Helwan University

Supervised by:

Dr. Ensaf Hussein

June 2024

Project Summary

Today, Islamic scholars and enthusiasts who seek to understand Hadiths, the recorded sayings and practices attributed to Prophet Muhammad (PBUH), often rely on manual methods for information retrieval. This process is both time-consuming and resource-intensive. Our project aims to streamline this process by developing a sophisticated Hadith information retrieval system that leverages advanced Natural Language Processing (NLP) techniques. This system will automate the process of finding relevant Hadiths for user queries, saving time, effort, and resources.

This project goes beyond basic information retrieval, offering a comprehensive solution for those seeking knowledge about Hadiths. By combining advanced NLP techniques and contextual understanding, the project provides accurate and contextually relevant answers to user queries. Imagine being able to find relevant Hadiths with greater efficiency and accuracy, even when dealing with concise and nuanced language. This level of automation not only saves time and resources but also ensures a deeper understanding of Hadiths, leading to improved Islamic scholarship and practice. By leveraging cutting-edge NLP techniques, this project offers a unique solution for those seeking to deepen their understanding of Islamic traditions.

The Hadith QA Chat App, which is an essential component of our project, allows users to ask questions about Hadiths and receive accurate, contextually relevant answers from a reliable database. The app provides not just the text of Hadiths but also explanations, contexts, and interpretations from credible sources.

The Hadith Authenticator, another important component of our project, independently verifies the authenticity of Hadiths by cross-referencing with established Hadith collections and scholarly works. This tool provides detailed reports on the authenticity of Hadiths, including the chain of narration (Isnad), text (Matn), and scholarly opinions.

By creating a robust Hadith information retrieval system, this project aims to democratize access to Hadith for scholars and laypeople alike. It has the potential to significantly impact Islamic research and education by streamlining the retrieval of information, facilitating knowledge acquisition, and deepening understanding of Islamic traditions for a wider audience. This project exemplifies the potential for technology to bridge the gap between traditional religious knowledge and modern needs. By creating user-friendly digital tools, the study demonstrates how technology can be harnessed to preserve and disseminate traditional Islamic knowledge in a contemporary context. This approach not only makes Hadith literature more accessible but also ensures that the authenticity and integrity of the information are maintained. This project has the potential to foster a more knowledgeable and cohesive community, grounded in authentic religious knowledge. The implementation of these tools encourages ongoing improvements and updates in the field of Islamic studies. By regularly updating the Hadith database and enhancing the verification algorithms, the project ensures that the tools remain current and relevant. This continuous improvement fosters a dynamic environment for Islamic scholarship and learning.

Keywords: Islamic Studies, Hadith Retrieval, Knowledge extraction, Information Retrieval Systems, Arabic Language Processing, Information extraction, Natural language processing (NLP), Arabic NLP, Hadith Search Engine, Contextual Retrieval, Retrieval-Augmented Generation.

Table of Contents

List of Figures

List of Abbreviations

Abbreviations	**Definition**
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
DAQAS	Deep Arabic Question Answering System
DTW	Dynamic Time Warping
GOF	Goal Oriented Fusion
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
HAQA	Holy Quran Automatic Question Answering
LARSA	Large Scale Arabic Transformer
RNNs	Recurrent Neural Networks
QA	Question Answering
SVMs	Support Vector Machines
NER	Named Entity Recognition
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
QUQA	Quran Question Answering
RAG	Retrieval-Augmented Generation
SMASH	Splitting Methods for Adaptive and Str Stratified Holdout
SVMs	Support Vector Machines
NLP	Natural Language Processing
TAQS	Text Arabic Question Similarity
eRock	ensemble Rock
ParSQuAD	Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0

Chapter 1

Introduction

1.1 Background:

Hadith, the recorded sayings and practices attributed to Prophet Muhammad (PBUH), serve as a cornerstone of Islamic scholarship and practice. However, navigating the vast collections of Hadith can be a daunting task. Traditionally, scholars have relied on manual methods for information retrieval, a process that is both time-consuming and resource-intensive.

The digital age has brought about a surge in the availability of online Hadith resources. This accessibility has paved the way for the development of advanced Hadith information retrieval systems. These systems aim to streamline information retrieval by automating the process of finding relevant Hadith for user queries. However, developing effective Hadith information retrieval systems presents unique challenges:

- **Concise and Rich Language:** Hadith texts are often concise and utilize a rich vocabulary of classical Arabic. This necessitates advanced Natural Language Processing (NLP) techniques to accurately capture the meaning and intent within the text.
- **Contextual Understanding:** Understanding the context and interpretation of Hadith is crucial for retrieving relevant information. Hadiths can have varying narrations and scholarly interpretations, requiring the system to consider these nuances when responding to user questions.

Addressing these challenges is paramount for developing robust and user-friendly Hadith information retrieval systems. Here, this project enters the scene.

This project leverages the power of NLP to create a sophisticated Hadith information retrieval system. By employing advanced NLP techniques, the system can extract key information from both Hadith texts and user queries. This enables the system to identify relevant Hadith with greater efficiency and accuracy, even when dealing with concise and nuanced language.

By creating a robust Hadith information retrieval system, this project aims to democratize access to Hadith for scholars and laypeople alike. It has the potential to significantly impact Islamic research and education by streamlining the retrieval of information, facilitating knowledge acquisition, and deepening understanding of Islamic traditions for a wider audience.

1.2 Motivation:

The vast collection of Hadith presents a significant challenge for users seeking specific information. Traditional methods of navigating these collections, relying on manual searches by scholars, are time-consuming and limit accessibility.

This project is motivated by the need for a more efficient and user-friendly approach to accessing Hadith knowledge. Natural Language Processing (NLP) offers a powerful solution for automating the process of finding relevant Hadith.

Here are some key factors driving the development of this Hadith Question Answering System:

- Enhanced Accessibility: It can empower users of all backgrounds to find relevant Hadith with greater ease. This democratizes access to Islamic knowledge, fostering deeper understanding and engagement with Hadith traditions.

- Improved Research Efficiency: By enabling users to quickly locate pertinent Hadith for their research, an Hadith QAS can significantly streamline the scholarly process. Researchers can dedicate more time to analysis and interpretation, leading to richer insights.

- Accurate Information Retrieval: NLP techniques can help filter through vast collections of Hadith and identify the most relevant ones for a given query. This reduces the risk of misinterpretations and ensures users receive accurate information.

- Preserving Islamic Knowledge: An effective Hadith QAS can serve as a valuable tool for preserving and transmitting Islamic knowledge for future generations. By making Hadith more readily accessible, the system can help safeguard this vital aspect of Islamic tradition.

This project aims to contribute to these goals by developing a robust Hadith QAS that leverages the power of NLP. By making Hadith knowledge more accessible and efficiently retrievable, the project has the potential to benefit scholars, students, and anyone interested in deepening their understanding of Islamic traditions.

1.3 Objectives:

Hadith QA Chat App

1. Provide Accurate Hadith Information: Develop a chat application that allows users to ask questions about Hadiths and receive accurate, contextually relevant answers from a reliable database.
2. User-Friendly Interface: Design a user interface that is intuitive and accessible, allowing users to easily query and navigate through the information.
3. Contextual Understanding: Ensure the app provides not just the text of Hadiths but also explanations, contexts, and interpretations from credible sources.
4. Real-Time Responses: Implement a system that offers real-time answers to user queries, making the information readily available.

Hadith Authenticator

1. Verify Authenticity: Create a tool that independently verifies the authenticity of Hadiths by cross-referencing with established Hadith collections and scholarly works.
2. Detailed Verification Reports: Provide detailed reports on the authenticity of Hadiths, including the chain of narration (Isnad), text (Matn), and scholarly opinions.
3. Database Integration: Maintain a comprehensive and regularly updated database of Hadiths, categorized by their authenticity (e.g., Sahih, Da'if, Hasan).
4. Ease of Use: Ensure the tool is user-friendly, allowing users to easily input Hadiths and receive verification results promptly.

1.4 Scope:

The Hadith Chat App is designed to serve as an interactive platform for users seeking knowledge about Hadith and Hadith narrations. The app will integrate the Arabic Korbora of Hadith, a respected collection of Hadith texts, to ensure authenticity and reliability in the answers provided. This document outlines the scope of the project, detailing the inclusions, exclusions, and the functionalities the app will offer.

Project Inclusions

Core Functionalities:

- Hadith Search and Retrieval: Users will be able to search for specific Hadiths using keywords or phrases and receive accurate results from the Arabic corpora of Hadith.

- Interactive Chat Interface: A chatbot feature will be included, providing real-time answers to user queries in Arabic.
- Integration with Arabic Datasets: The app will exclusively use Arabic datasets on Hadith and Hadith narrations to ensure that all content is sourced authentically.
- Language Support: The app will operate entirely in Arabic to cater to its primary user base and maintain the integrity of the original texts.
- Hadith verification…

User Experience:

- User-friendly Interface: A clean, intuitive interface will be designed to facilitate easy navigation and accessibility for all users., web app interface
- Responsive Design: The app will be accessible on various devices, including smartphones, tablets, and desktops, with a responsive design that adapts to different screen sizes.

Content Authenticity and Integrity:

- Content Verification: All Hadith content will undergo a verification process to ensure it aligns with the Arabic corpora of Hadith.
- Regular Updates: The app will receive regular updates to ensure the Hadith database is current and comprehensive.

Project Exclusions

Language Limitations:

- Non-Arabic Languages: The app will not support languages other than Arabic. No translations or transliterations will be provided.

Content Restrictions:

- Non-Hadith Religious Texts: The app will strictly limit its content to Hadith and Hadith narrations and will not include other Islamic texts such as the Quran, Tafsir, or Fiqh literature.
- Scholarly Interpretations: The app will not offer scholarly interpretations or Fatawa (Islamic legal rulings).

Functional Limitations:

- Offline Access: The app will require an internet connection to access the database and will not have offline capabilities.
- Social Networking Features: There will be no social networking or community discussion features within the app.

The Hadith QA Chat App is designed to provide users with a reliable and accessible platform to inquire about Hadiths. The scope of this application includes:

1. User Interaction:
   - Users can ask questions related to Hadiths in natural language.
   - The app provides real-time responses to user queries.
   - Users can request explanations, contexts, and interpretations of specific Hadiths.
2. Content Delivery:
   - Access to a comprehensive database of Hadiths from reputable collections.
   - Integration of scholarly interpretations and explanations to provide context and understanding.
   - Presentation of Hadith texts along with relevant classifications (e.g., Sahih, Da'if).
3. Technical Specifications:
   - Development of a robust natural language processing (NLP) system to understand and respond to user queries accurately.
   - Implementation of a user-friendly interface that is intuitive and easy to navigate.
   - Real-time processing capabilities to ensure prompt responses.
4. Maintenance and Updates:
   - Regular updates to the Hadith database to include new findings and scholarly insights.
   - Continuous improvement of the NLP system to enhance the accuracy of responses.

Hadith Authenticator

The Hadith Authenticator is an independent tool designed to verify the authenticity of Hadiths. The scope of this tool includes:

1. Verification Process:
   - Users can input the text of a Hadith for verification.
   - The tool cross-references the input with a comprehensive database of authenticated Hadith collections.
2. Output and Reporting:
   - Provides detailed verification reports including the chain of narration (Isnad), text (Matn), and classification (e.g., Sahih, Da'if).
   - Includes scholarly opinions and reasons for the classification of each Hadith.

3. Database Integration:
    - Maintains an extensive and regularly updated database of Hadiths, categorized by authenticity.
    - Incorporates data from established Hadith collections and scholarly works.
4. Technical Specifications:
    - Development of an efficient algorithm to perform quick and accurate verification.
    - User-friendly interface allowing users to easily input Hadiths and receive results.
5. Maintenance and Updates:
    - Continuous updates to the Hadith database to ensure accuracy and comprehensiveness.
    - Regular enhancements to the verification algorithm for improved performance.

1.5 Significance of the Study:

Enhancing Access to Authentic Hadith Information

The development of the Hadith QA Chat App and the Hadith Authenticator represents a significant advancement in the accessibility and verification of Hadith literature. These tools address a critical need in the Muslim community and among scholars for reliable, quick, and accurate access to Hadith information. By leveraging technology, this project makes Islamic knowledge more accessible to a broader audience, enabling users to engage with Hadiths in a more informed and meaningful way.

Facilitating Scholarly Research and Learning

For scholars and students of Islamic studies, the Hadith QA Chat App and Hadith Authenticator serve as valuable resources. The QA Chat App provides instant access to a wealth of Hadith information, complete with explanations and context from authoritative sources. This facilitates a deeper understanding and aids in academic research. The Hadith Authenticator offers a rigorous tool for verifying the authenticity of Hadiths, which is crucial for scholarly work that relies on accurate and credible sources.

Promoting Accurate Religious Practice

The authentication of Hadiths is essential for ensuring that Islamic practices are based on authentic and credible sources. The Hadith Authenticator helps users avoid weak or fabricated Hadiths that could lead to misinformed religious practices. By providing detailed verification reports, the tool supports users in making informed decisions about the Hadiths they follow and share.

Bridging the Gap Between Technology and Tradition

This project exemplifies the potential for technology to bridge the gap between traditional religious knowledge and modern needs. By creating user-friendly digital tools, the study demonstrates how technology can be harnessed to preserve and disseminate traditional Islamic knowledge in a contemporary context. This approach not only makes Hadith literature more accessible but also ensures that the authenticity and integrity of the information are maintained.

Supporting Community Education

The Hadith QA Chat App and Hadith Authenticator contribute to the broader goal of community education. By making reliable Hadith information and verification readily available, these tools empower individuals to educate themselves and others about Islamic teachings. This has the potential to foster a more knowledgeable and cohesive community, grounded in authentic religious knowledge.

Encouraging Continuous Improvement in Islamic Studies

The implementation of these tools encourages ongoing improvements and updates in the field of Islamic studies. By regularly updating the Hadith database and enhancing the verification algorithms, the project ensures that the tools remain current and relevant. This continuous improvement fosters a dynamic environment for Islamic scholarship and learning.

1.6 Outline the structure of the report:

The report structure is organized to explore the project. In Chapter 2, the Related Work section begins with an overview of the literature review of the related work papers, explaining its purpose and outlining the paper's structure. We will also discuss each paper Historical Perspective, which examines milestones in the project's subject evolution, as well as a Theoretical Framework, which clarifies the contributions of significant theories. It also includes Previous research and studies; in this part, we summarize the major findings for each paper and indicate knowledge gaps. The Current State of the Field examines current developments, trends, and prospective challenges. In Chapter 3, we will discuss Materials and Methods. We will also define our System Descriptions such as context, limits, and user views, and show all this in a context diagram. We will also discuss the system requirements and describe its functionality, interfaces, and non-functional features. We will discuss Design Constraints and Research Design, leading to Architectural Design, where system architecture and components are presented. We will discuss the Data Design section which covers our data gathering, transformation, storage, and the dataset's nature. We will also discuss algorithmic interaction, Data Flow Diagram Designs, as well as Integration with External Systems and Experimental Setup. In Chapter 4, we will discuss the Implementation and Preliminary Results, Programming Languages used in our project, Code Structure, Data Structures, and Databases, Quantitative and Qualitative Results, present findings using tables, graphs, and discussions. In Chapter 5, we will discuss the discussion and conclusion, interpret our results, compare them with previous studies, acknowledge limitations, summarize our findings, and propose future work for our project. This study ends with a References section and Appendices.

Chapter 2

Related Work

2.1 Introduction to Literature Review:

DTW at Qur'an QA 2022: Utilizing Transfer Learning with Transformers for Question Answering in a Low-resource Domain. [1]

Machine Reading Comprehension (MRC) is a crucial task in Natural Language Processing (NLP) aimed at evaluating the understanding of machines regarding textual content. Analogous to human examinations, where individuals must comprehend text to answer questions, MRC tasks require machines to read passages and respond to questions based on their understanding. These systems find applications across various NLP domains such as search engines and dialogue systems, eliciting significant interest from the NLP community.

Background: Traditionally, MRC tasks involve training machine learning models on annotated datasets. Researchers have explored diverse approaches, from conventional algorithms like support vector machines to cutting-edge neural approaches such as transformers. While annotated datasets are crucial, the availability of such data has been a bottleneck. Recognizing this, the NLP community has curated several datasets, with the Stanford Question Answering Dataset (SQuAD) emerging as a prominent benchmark. However, MRC research has predominantly focused on common domains like Wikipedia, neglecting low-resource domains such as religious texts.

Related Work: The history of MRC traces back to the late 1970s, with early systems like QUALM and subsequent developments in the late 1990s.. However, the field experienced a resurgence in the 2010s with the advent of large-scale supervised datasets and neural network models. Significant contributions include the creation of benchmark datasets like SQuAD, extending to various languages and domains. Notably, the Qur'an QA 2022 shared task is pioneering MRC research on religious texts, filling a critical gap in the field.

Summary: This paper addresses the Qur'an QA 2022 shared task, aiming to advance question answering and reading comprehension research on the Qur'an. Leveraging transfer learning with transformers, the DTW approach demonstrates robust performance despite challenges posed by the low-resource nature of the dataset.

HAQA and QUQA: Constructing two Arabic Question-Answering Corpora for the Quran and Hadith. [2]

The paper presents the creation of two novel datasets, HAQA and QUQA, for question-answering tasks in Arabic Islamic texts. The introduction outlines the importance of these datasets in advancing research in natural language processing (NLP) and artificial intelligence (AI) for understanding classical Arabic texts, particularly the Quran and Hadith. It emphasizes the significance of these texts for millions of Muslims worldwide and highlights the challenges in building question-answering systems due to the absence of comprehensive datasets.

Background:
This section discusses the traditional methods of building question-answering systems for Arabic texts, focusing on the limitations of existing datasets like AQQAC and AyaTEC. It highlights the scarcity of diverse and extensive datasets for both the Quran and Hadith, hindering the development of accurate and robust question-answering models. The section underscores the need for new datasets like HAQA and QUQA to overcome these challenges and enhance research in Arabic NLP and AI.

Related Work:
This section discusses the existing research efforts in building Question-Answering (QA) systems for the Holy Quran and Hadiths. It highlights the limitations of previous datasets, such as their focus on specific chapters or the lack of direct Quranic wordings in answers. Despite initiatives like the AQQAC and AyaTEC, which cover a limited portion of the Quran, there remains a shortage of comprehensive datasets for evaluation. To address this gap, the paper introduces the QUQA dataset, which integrates existing datasets, incorporates challenging

questions, and expands coverage across more Quranic verses.

Similarly, this section acknowledges the scarcity of QA datasets for Hadiths. While some studies have attempted to collect questions from Hadiths, such datasets are not publicly available. To address this gap, the paper introduces the HAQA dataset, providing a valuable resource for researchers interested in developing QA systems for Hadiths.

Summary:
In summary, the paper introduces the HAQA and QUQA datasets as valuable resources for researchers working on Arabic question-answering tasks, particularly in the context of the Quran and Hadith. It highlights the unique features of these datasets, including their size, diversity, and coverage of topics. The paper concludes by emphasizing the potential impact of HAQA and QUQA on advancing research in Arabic NLP, AI, and Islamic question-answering systems.

Retrieval-Augmented Generation for Large Language Models: A Survey . [3]

The authors do a comprehensive overview of Retrieval-Augmented Generation (RAG), a groundbreaking framework that represents a significant leap forward in the capabilities of Language Models (LLMs). RAG achieves this by integrating parameterized knowledge from language models with extensive non-parameterized data from external knowledge bases. By doing so, RAG has the potential to revolutionize natural language processing tasks, enabling models to generate more accurate, contextually relevant responses.

The outline of the literature review chapter is as follows:

Background: Traditional Language Models (LLMs) have long been the cornerstone of natural language processing (NLP) systems, enabling tasks such as text generation, summarization, and translation. However, these models often face limitations in understanding and generating contextually relevant responses. One key challenge is their reliance solely on pre-existing data within their training corpus, which may not encompass the breadth and depth of knowledge necessary for nuanced language understanding. Consequently, while LLMs excel in tasks like syntactic correctness, they may falter in capturing the subtleties of context.

Enter Retrieval-Augmented Generation (RAG), a novel framework that seeks to bridge this gap by integrating parameterized knowledge from language models with extensive non-parameterized data from external knowledge bases. This integration enables models to access a wealth of external information, ranging from factual knowledge to common-sense reasoning, thus enhancing their ability to generate more informed and contextually relevant responses. By augmenting traditional language generation processes with external knowledge retrieval mechanisms, RAG represents a paradigm shift in natural language understanding and generation, promising to unlock new capabilities for AI systems across various domains.

Related Work: The survey extensively explores the evolution of RAG technologies and their application across a wide range of natural language processing tasks. It delineates three distinct developmental paradigms within the RAG framework: Naive, Advanced, and Modular RAG. The Naive RAG approach represents an initial step in integrating external knowledge with language models, while Advanced RAG builds upon this foundation to achieve more sophisticated interactions between the model and external knowledge sources. Modular RAG further refines this integration by breaking down the model into modular components, each responsible for handling specific aspects of the input and knowledge retrieval process. Additionally, the survey discusses RAG's integration with other AI methodologies, such as fine-tuning and reinforcement learning, which further enhance its capabilities and versatility across different tasks and domains.

Summary: the survey underscores the transformative potential of RAG in advancing the capabilities of AI systems and its broader implications for various domains. The authors highlight the expanding scope of RAG's application into multimodal domains, where it can interpret and process diverse data forms such as images, videos, and code. Furthermore, the growing ecosystem of RAG-centric AI applications and supportive tools exemplifies the increasing interest from both academic and industrial sectors. However, the authors also acknowledge the need for ongoing research to improve the robustness and effectiveness of RAG, particularly in handling extended contexts and ensuring accurate performance evaluations. Despite these challenges, RAG stands as a testament to the innovative strides being made in natural language processing and AI research, offering new avenues for advancing the state-of-the-art in language understanding and generation.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks . [4]

The paper introduces Retrieval-Augmented Generation (RAG) as a novel approach to open-domain question answering (QA). RAG combines the flexibility of generative models with the effectiveness of retrieval-based approaches, aiming to achieve state-of-the-art performance across various QA tasks. The authors highlight the limitations of existing methods and the need for a more integrated approach to leverage both parametric and non-parametric memory in QA systems.

Background: This section provides context on the challenges of open-domain QA and the evolution of QA models over time. It explains the limitations of purely generative or purely retrieval-based approaches and introduces the concept of RAG as a hybrid model that addresses these limitations. The authors discuss the importance of incorporating both generative and retrieval components to enhance the accuracy and versatility of QA systems.

Related work: This section reviews the previous research in open-domain QA, including both generative and retrieval-based approaches. They discuss the strengths and weaknesses of existing models, such as T5, REALM, and DPR, and highlight the need for a unified architecture that combines the benefits of both approaches. The authors also compare RAG to other retrieval-augmented models and discuss its advantages over traditional QA systems.

Summary: the paper introduces RAG as a novel approach to open-domain QA that combines generative and retrieval-based methods. The authors provide background information on the challenges of open-domain QA and review related work in the field. They highlight the limitations of existing approaches and propose RAG as a solution to overcome these limitations. Overall, the paper presents RAG as a promising model for achieving state-of-the-art performance in open-domain QA tasks.

2.2 Historical Perspective:

DTW at Qur’an QA 2022: Utilizing Transfer Learning with Transformers for question Answering in a Low-resource Domain [1].

1977: The earliest known endeavor in Machine Reading Comprehension (MRC) with the development of QUALM(Question Answering Language Model), a question answering program by Lehnert. Although rudimentary by modern standards, QUALM laid the groundwork for subsequent research in MRC.
1999: Hirschman et al. constructed a reading comprehension system targeting a corpus of stories aimed at 3rd to 6th-grade material, marking an early exploration into MRC despite facing limitations due to data scarcity and model performance constraints.

Early 2010s: Interest in MRC research surged as large-scale datasets and advancements in neural networks emerged, leading to experimentation with attention-based deep neural networks for real document comprehension and answering complex questions with minimal prior knowledge of language structure.
2013: Richardson et al. introduced the MCTest dataset comprising 500 stories and 2000 questions, marking a significant milestone as one of the first large-scale MRC datasets. It provided researchers with a valuable resource for training and evaluating MRC systems.
2015: Hermann et al. pioneered a new dataset generation method, enabling the creation of large-scale supervised reading comprehension datasets, laying the groundwork for benchmark datasets like SQuAD.
2016: Rajpurkar et al introduced SQuAD (Stanford Question Answering Dataset), which emerged as a cornerstone in MRC research. Comprising over 100,000 annotated examples, SQuAD became a widely recognized benchmark for evaluating MRC systems.
2019: Mozannar et al. extended SQuAD to Arabic by creating the Arabic Reading Comprehension Dataset (ARCD) as part of the SOQAL (SQuAD in Arabic Language) project.
2020: the emergence of the Qur’an QA 2022 shared task, which aimed to promote research in MRC on religious texts, specifically the Qur’an. Malhas and Elsayed introduced a dataset containing question-passage-answer triplets extracted from the Qur’an, providing a novel challenge for the MRC community.
2022: Researchers participated in the Qur’an QA 2022 shared task, employing innovative methodologies such as transfer learning with transformers to address challenges posed by the low-resource nature of the Qur’anic dataset.

HAQA and QUQA: Constructing two Arabic Question-Answering Corpora for the Quran and Hadith. [2]

2016: Hamdelsayed and Atwell initiated the development of question-answering datasets for the Holy Quran. They manually generated a dataset comprising 263 question-answer pairs, primarily focusing on chapters such as 'Al-Baqarah' and 'Al-Fatiha'.
2017: Neamah and Saad collected 1500 questions and answers from websites.
2019: Alqahtani constructed the first available corpus of 1224 question–answer pairs called the Annotated Corpus of Arabic Al-Quran Question and Answer (AQQAC), a dataset containing questions and answers from the Quran. However, its usability for research purposes is limited due to issues with answer interpretations.

2020: Malhas and Elsayed developed AyaTEC, a specialized dataset focusing on Quranic questions and answers. This marks a significant step forward in the domain of Arabic question-answering datasets.

2022: Alnefaie, Atwell, and Alsalka embark on a comprehensive initiative to address the

shortage of reliable question-answer datasets for Islamic texts. They design and construct HAQA and QUQA, marking a significant milestone in the field.
HAQA becomes the first reusable Arabic Hadith question-answer corpus, filling a crucial gap in research resources for Hadith Sharif.
QUQA emerges as the most extensive and challenging Arabic question-answer collection on the Quran, integrating existing datasets and expanding them with additional resources and diverse questions.

Chronological Documentation:

2017
Nabeel Neamah and Saidah Saad collected hadiths and created a question answering system supporting vector machine method for hadith domain.

2019
Mohammad Mushabbab A Alqahtani
Action: Constructed a Quranic Arabic semantic search model based on ontology of concepts.

2020
Rana Malhas and Tamer Elsayed
Action: Developed the Ayatec dataset, a verse-based test collection for Arabic question answering on the Holy Quran.

2020
Rana Malhas and Tamer Elsayed
Developed QRCD, an extended version of AyaTEC intended for machine reading comprehension (MRC) tasks.

2021
Hajer Maraoui, Kais Haddar, and Laurent Romary Built an Arabic factoid question-answering system for Islamic sciences using normalized corpora.

2022
Rana Malhas, Watheq Mansour, and Tamer Elsayed Organized Qur’an QA 2022, the first shared task on question answering over the Holy Quran, providing an overview of the competition.

2022
Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka Addressed challenges in Islamic question answering corpora, presenting QUQA and HAQA, two datasets containing questions and answers about the Holy Quran and the Hadith Sharif, respectively.

The context for the evolution of ABSA is that with the rapid growth of online user-generated content, there is a high demand for analyzing and understanding the opinions and emotions of people on various topics, products, services, and events. ABSA provides a fine-grained and comprehensive way to extract and classify the sentiment elements and their relations from text, which can help businesses and individuals make better decisions and improve customer satisfaction. ABSA also poses many challenges and opportunities for natural language processing research, such as domain adaptation, language adaptation, multimodal ABSA, explainable ABSA, and aspect-level sentiment summarization.

Retrieval-Augmented Generation for Large Language Models: A Survey . [3]

2019: The authors introduce the concept of Retrieval-Augmented Generation (RAG), outlining its potential to enhance language models by integrating external knowledge bases.

2020: Initial experiments demonstrate the effectiveness of RAG in improving language understanding and generation tasks, sparking interest within the research community.

2021: RAG gains traction as researchers explore its application across diverse NLP tasks, showcasing its ability to handle complex contexts and generate more informed responses.

2022: Further developments in RAG technology lead to the emergence of three developmental paradigms: Naive, Advanced, and Modular RAG, each representing progressive enhancements over its predecessors.

2023: RAG's technical integration with other AI methodologies, such as fine-tuning and reinforcement learning, further expands its capabilities and applicability across various domains.

2024: RAG's application scope continues to expand into multimodal domains, adapting its principles to interpret and process diverse data forms like images, videos, and code. Its growing ecosystem is evidenced by the rise in RAG-centric AI applications and the continuous development of supportive tools.

The evolution of RAG stems from the need to enrich large language models with external knowledge sources for better contextual understanding. Interdisciplinary collaborations and iterative refinements have propelled RAG's advancements, attracting interest from academia and industry. Ongoing research aims to address challenges in robustness, scalability, and evaluation methodologies. Collaboration among stakeholders is crucial for maximizing RAG's impact in reshaping natural language processing and AI landscapes.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. [4]

2018: Introduction of Dense Passage Retrieval (DPR) by Karpukhin et al., focusing on leveraging large-scale retrieval for QA systems. Initial experiments with generative models like T5 (Text-to-Text Transfer Transformer) by Raffel et al., which laid the groundwork for combining generative capabilities with retrieval-based approaches.

2019: Further development of generative models like T5, demonstrating their potential for open-domain QA tasks. Exploration of retrieval-based models like REALM (Retrieval-Augmented Language Model) by Guu et al., which showed the effectiveness of incorporating retrieval mechanisms into QA systems.

2020: Introduction of the Retrieval-Augmented Generation (RAG) model by Lewis et al., which combines generative capabilities with retrieval-based methods to achieve state-of-the-art performance in open-domain QA tasks. Comparison of RAG with existing models such as T5, REALM, and DPR, highlighting its superior performance in terms of both accuracy and flexibility. Demonstration of RAG's ability to outperform traditional retrieval-based QA systems without the need for specialized pre-training techniques.

2021: Continued refinement and optimization of RAG, including enhancements to its retrieval and generation components. Experimentation with different configurations of RAG, such as RAG-Token and RAG-Sequence, to evaluate their performance across various QA tasks. Validation of RAG's effectiveness through extensive experiments and benchmarking against state-of-the-art models in open-domain QA.

2022: Further analysis of RAG's performance on specific QA tasks, including abstractive QA, fact verification, and question generation. Investigation into the underlying mechanisms of RAG, including its retrieval mechanism and generation diversity. Exploration of RAG's ability to update its knowledge dynamically at test time, showcasing its adaptability to changing information.

2023: Continued research on RAG's retrieval and generation components, aiming to optimize its performance across a wide range of QA tasks. Examination of RAG's effectiveness in handling diverse types of questions and generating accurate responses even in the absence of explicit evidence. Validation of RAG's performance through human evaluations and qualitative analysis, highlighting its superiority over existing models in terms of factuality and specificity.

2024: Comprehensive documentation of RAG's advancements and achievements in the field of open-domain QA, including its superiority over existing models in terms of accuracy, flexibility, and adaptability. Discussion of future research directions and potential applications of RAG in various domains, highlighting its potential to revolutionize the field of natural language understanding and generation.

2.3 Theoretical Framework:

DTW at Qur’an QA 2022: Utilizing Transfer Learning with Transformers for Question Answering in a Low-resource Domain. [1]

Data

The theoretical framework of the paper revolves around leveraging transfer learning with transformers for Question Answering (QA) in a low-resource domain, specifically focusing on the Qur’an. The primary dataset used is the Qur’an QA 2022 shared task dataset, which consists of question-passage-answer triplets extracted from the Qur’an. Additionally, the paper utilizes the SOQAL dataset, comprising Arabic Machine Reading Comprehension (MRC) resources, including the Arabic Reading Comprehension Dataset (ARCD) and a machine translation of the SQuAD dataset.

Methodology

The methodology employs transfer learning, where a pre-trained transformer model is fine-tuned on the available Arabic MRC data before being applied to the Qur’anic dataset. Seven pre-trained transformer models, including camelbert-mix, camelbert-ca, mbert-cased, mbert-uncased, AraELECTRA-generator, AraELECTRA-discriminator, and AraBERTv2, are experimented with. The transformer architectures are trained on general tasks like language modeling and subsequently fine-tuned for the MRC task.

Transfer Learning

Transfer learning is a crucial component of the theoretical framework, allowing the model to leverage knowledge gained from training on a resource-rich dataset like SOQAL to improve performance on the smaller Qur’anic dataset. By initializing the training process with pre-trained weights from the SOQAL dataset, the model can better adapt to the low-resource setting of the Qur'an QA 2022 shared task dataset.

Ensemble Learning

Ensemble learning is employed as a fine-tuning strategy to enhance performance further. This technique involves combining predictions from multiple machine learning models to produce a more robust and accurate final output. Self-ensemble is specifically utilized, where the same architecture is trained using multiple random seeds, and the outputs are ensembled to mitigate the impact of randomness inherent in transformer models.

Experimental Setup

Experiments are conducted using a batch size of eight, Adam optimizer with a learning rate of 2e-5, and linear learning rate warm-up over 10% of the training data. The models are trained for five epochs using NVIDIA GeForce RTX GPUs. Various performance metrics, including partial Reciprocal Rank (pRR) scores, are used to evaluate the effectiveness of the proposed methodology.

Open-source Resources

The paper emphasizes reproducibility and transparency by releasing the code of the experiments as an open-source GitHub project. The project is available as a Python package, and the pre-trained machine learning models are made freely accessible in the HuggingFace model hub. Additionally, a Docker image of the experiments is provided, adhering to ACL reproducibility criteria.

HAQA and QUQA: Constructing two Arabic Question-Answering Corpora for the Quran and Hadith. [2]

The authors describe a detailed methodology for the development of two significant datasets, QUQA and HAQA, designed to facilitate scholarly inquiry in Islamic studies. Through a systematic approach, the authors outline the various stages of dataset creation, including design, data source identification, collection, and cleaning.

QUQA and HAQA Design (3.1):

The initial step involved defining the structure, metadata, and format for both datasets. For QUQA, the design was based on existing models like AQQAC and AyaTEC, while HAQA's design was tailored to suit the nature of Hadiths. Both datasets utilized comma-separated values (CSVs) with UTF-8 encoding, chosen for their compatibility with various systems after conversion to XML format.

Identifying Data Sources (3.2):
Data for the corpora were sourced from books and available datasets. While many books contained relevant content, not all met the project's requirements. The chosen datasets included AQQAC and AyaTEC for QUQA, providing answers from Quranic verses and interpretations, among other sources. Books like "The Doctrine of Every Muslim" and "Prayer (1770) Question and Answer" were also utilized, ensuring a diverse range of content for both QUQA and HAQA.

Data Collection (3.3):
This stage involved integrating existing datasets and incorporating new sources. Python programs were developed to convert existing datasets into the required structure and format. Additionally, new sources were processed using Optical Character Recognition (OCR) to convert text into usable formats. Manual review and extraction were conducted to ensure accuracy, followed by metadata filling using Python or manual methods.

Cleaning the Data (3.4):
Data cleaning is a crucial step to ensure the quality of the final datasets. Errors such as misspellings, missing information, and duplicate data were identified and corrected manually. Automated methods, including regular expressions, were also employed to remove extra spaces and non-Arabic characters. This meticulous cleaning process enhanced the overall quality of the datasets.

The resulting QUQA dataset comprises 3,382 records covering diverse topics related to the Holy Quran, with over 301,000 tokens extracted from 2,930 Quranic verses. Similarly, HAQA consists of 1,598 records covering various topics related to Hadith Sharif. Both datasets exhibit extensive coverage and diversity, making them valuable resources for research and analysis in Islamic studies.

Retrieval-Augmented Generation for Large Language Models: A Survey . [3]

The authors conduct an exhaustive exploration of the theoretical foundations governing Retrieval-Augmented Generation (RAG), traversing its developmental trajectory from its nascent stages to its current advanced paradigms. They meticulously dissect the theoretical frameworks underpinning the evolution of RAG, discerning distinct milestones such as Naive RAG, Advanced RAG, and Modular RAG, each representing pivotal junctures in its progression. Through an intricate examination of key technologies, the paper sheds light on the theoretical intricacies governing the augmentation of RAG capabilities, including LangChain and LLamaIndex, which play instrumental roles in enhancing retrieval mechanisms and generative models' symbiotic relationship. Furthermore, the authors delve deep into the theoretical nuances driving the optimization of retrieval efficiency, document recall enhancement, and robust data security measures embedded within the RAG framework. Within this intricate tapestry of theoretical concepts, the paper elucidates the emergence of pivotal models and approaches such as Conceptual RAG (CRAG), Unsupervised Entity-Oriented Pre-training (UEOP), and Retrieval-Based Pre-training Strategies (RBPS), each contributing uniquely to RAG's maturation and adaptability across diverse domains. Additionally, the survey meticulously examines the theoretical foundations underlying RAG's efficacy in various applications, ranging from question answering systems and dialogue generation platforms to code search engines, offering nuanced insights into the underlying mechanisms driving RAG's transformative potential in these domains. Through meticulous analysis and synthesis of findings from diverse sources, the paper provides a comprehensive understanding of the theoretical underpinnings shaping RAG's evolution and its broad spectrum of applications.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. [4]

The paper introduces a novel deep learning architecture termed Retrieval-Augmented Generation (RAG) for enhancing open-domain question answering (QA) systems. This model integrates generative capabilities with retrieval mechanisms, aiming to improve the performance and versatility of existing QA systems significantly. By combining the flexibility of generative models with the robustness of retrieval-based approaches, RAG achieves state-of-the-art results across various open-domain QA tasks.

Dense Passage Retrieval (DPR) serves as a fundamental component of the RAG model, enabling efficient retrieval of relevant passages from a large corpus of documents. This mechanism enhances the model's ability to access and utilize vast amounts of unstructured data effectively. Additionally, RAG leverages generative architectures such as T5 to generate answers directly from questions, thereby augmenting the retrieval process with flexible generation capabilities.

The theoretical framework of RAG emphasizes the synergistic integration of retrieval and generation components to address the limitations of existing QA systems comprehensively. By leveraging both retrieval and generative mechanisms, RAG can effectively handle diverse QA tasks, including those where explicit evidence may not be readily available. This hybrid approach offers improved accuracy, adaptability, and robustness, positioning RAG as a promising solution for open-domain QA challenges.

Key components of the RAG model include the retrieval mechanism, generation architecture, and training methodology, which work together to enable effective retrieval and generation of answers to questions. The model undergoes rigorous evaluation and benchmarking against state-of-the-art approaches, demonstrating superior performance across various metrics and datasets. Theoretical discussions also highlight potential future directions and applications of the RAG model, paving the way for advancements in natural language understanding and generation.

2.4 Previous Research and Studies:

Key findings:

Transfer Learning with Transformers:

- □ Utilizing transformer models like BERT and its variants (e.g., AraBERT, AraELECTRA) has significantly improved the performance of Arabic QA systems by leveraging pre-trained models and fine-tuning them for specific tasks.

Construction of Arabic QA Corpora:

- □ Developing high-quality, domain-specific datasets such as those for the Qur'an and Hadith is crucial for improving the accuracy and relevance of QA systems.

Retrieval-Augmented Generation (RAG):

- □ Combining retrieval mechanisms with generative models enhances QA performance by incorporating relevant external knowledge into the answer generation process.

Ensemble and Post-Processing Techniques:

- □ Using ensembles of BERT-based models and applying post-processing steps like majority voting can enhance the robustness and accuracy of QA systems.

Survey and Literature Reviews:

- □ Comprehensive surveys on RAG and custom domains provide insights into the adaptability of RAG frameworks for various QA tasks, highlighting their potential for specialized knowledge bases.

Optimization of BERT Models:

- □ Heavy optimization and fine-tuning of BERT models lead to significant improvements in answering accuracy for religious texts.

Neural Networks and Deep Learning Approaches:

□ Integrating contemporary deep neural networks with existing models enhances understanding and answering capabilities for Arabic religious texts.

Text-to-Text Transformer Models:

□ Using text-to-text transformers has proven effective in capturing and generating relevant answers by understanding the relationships between questions and texts.

Challenges and Techniques in Arabic QA:

□ Addressing specific challenges faced in processing and understanding Arabic texts through advanced NLP techniques improves the overall performance of QA systems.

Duplicate Question Detection and Machine Reading Comprehension:

□ Implementing advanced detection mechanisms can improve the relevance and accuracy of answers in Arabic QA systems.

Factoid QA Systems:

□ Developing factoid QA systems for Islamic sciences using normalized corpora ensures accurate and contextually appropriate answers.

Comparative Studies on Transformer Models:

□ Comparative studies reveal that certain transformer models outperform others in terms of accuracy and efficiency when applied to Arabic texts.

Question Similarity Systems:

□ Utilizing hybrid architectures like BERT with BiLSTM for question similarity can enhance QA performance by managing question similarity effectively.

Early Insights into QA System Development:

□ Initial research on reading comprehension programs provides foundational insights into the development of QA systems, emphasizing the importance of statistical language processing.

2.5 Current State of the Field:

RAG

Current State:

- **Transfer Learning and Pre-trained Models:** The use of models like BERT, AraBERT, and ELECTRA has proven effective in understanding the nuances of Arabic texts and generating accurate answers.

- **Dataset Development:** High-quality, domain-specific datasets such as those for the Qur'an and Hadith are critical for training robust QA systems.
- **Retrieval-Augmented Generation (RAG):** Combining retrieval mechanisms with generative models has enhanced the ability to handle complex, knowledge-intensive queries.
- **Optimization and Fine-tuning:** Heavy optimization and fine-tuning of models are essential to achieve high accuracy and relevance in QA tasks.
- **Neural Networks and Hybrid Architectures:** The integration of neural networks and hybrid models like BiLSTM with BERT has shown significant improvements in managing question similarity and generating contextually appropriate answers.
- **Limited Hadith-specific Datasets:** While datasets like HAQA and QUQA exist for Quran and Hadith, resources specifically tailored for Hadith remain scarce.
- **Focus on Deep Learning Approaches:** Recent research explores transfer learning, fine-tuning pre-trained models, and ensemble methods to improve QA performance.

RAG's Potential:

- **Promising for Low-Resource Domains:** RAG can be particularly effective for domains with limited training data, potentially addressing the scarcity of Hadith-specific datasets.
- **Improved Accuracy:** Studies suggest RAG can enhance answer accuracy compared to traditional generative models.

However, challenges remain:

- **Adapting RAG for Hadith:** While RAG shows promise, research on its application to Hadith-specific tasks is limited.
- **Faithfulness and Bias:** Ensuring answers accurately reflect Hadith content and minimizing bias requires careful consideration.

Additional Insights from References:

- Studies at Qur’an QA 2022 showcase various approaches to Arabic religious text QA, offering valuable insights for Hadith .
- Research on Arabic Question Answering explores challenges, techniques, and the potential of transformer models.

While RAG offers promise for Hadith QA, there are significant challenges and unresolved issues to address:

- **Adapting RAG for Hadith:** Current research hasn't extensively explored how to adapt RAG for the specific characteristics of Hadith text. Hadith may have unique stylistic elements, vocabulary, and references compared to general Arabic text, requiring tailored RAG implementations.
- **Faithfulness and Bias Mitigation:** Ensuring answers accurately reflect the content and meaning of Hadith is crucial. Bias in training data or the RAG model itself could lead to inaccurate or misleading responses. Developing methods to evaluate and mitigate bias in Hadith QA systems is necessary.
- **Limited Hadith-specific Datasets:** As mentioned earlier, the lack of large, high-quality datasets specifically designed for Hadith QA hinders the development and evaluation of RAG models in this domain. Creating or curating such datasets is essential for effective training and testing.
- **Evaluation Metrics:** Standard QA evaluation metrics like BLEU score or ROUGE score might not be sufficient for assessing the faithfulness and nuance of answers in the context of Hadith. Developing domain-specific evaluation methods that consider religious and cultural sensitivity is important.

Additionally, some unresolved issues relevant to the broader field of Arabic NLP also apply to Hadith QA with RAG:

- **Explainability and Interpretability:** Understanding how RAG models arrive at their answers, particularly in the context of religious texts, is crucial for building trust and ensuring transparency.
- **Incorporating Context and Reasoning:** Current models often struggle to understand the broader context and reasoning behind Hadith teachings. Developing methods to integrate these aspects into RAG models could improve answer quality.

Chapter 3

Materials and Methods

3.1 System Description:

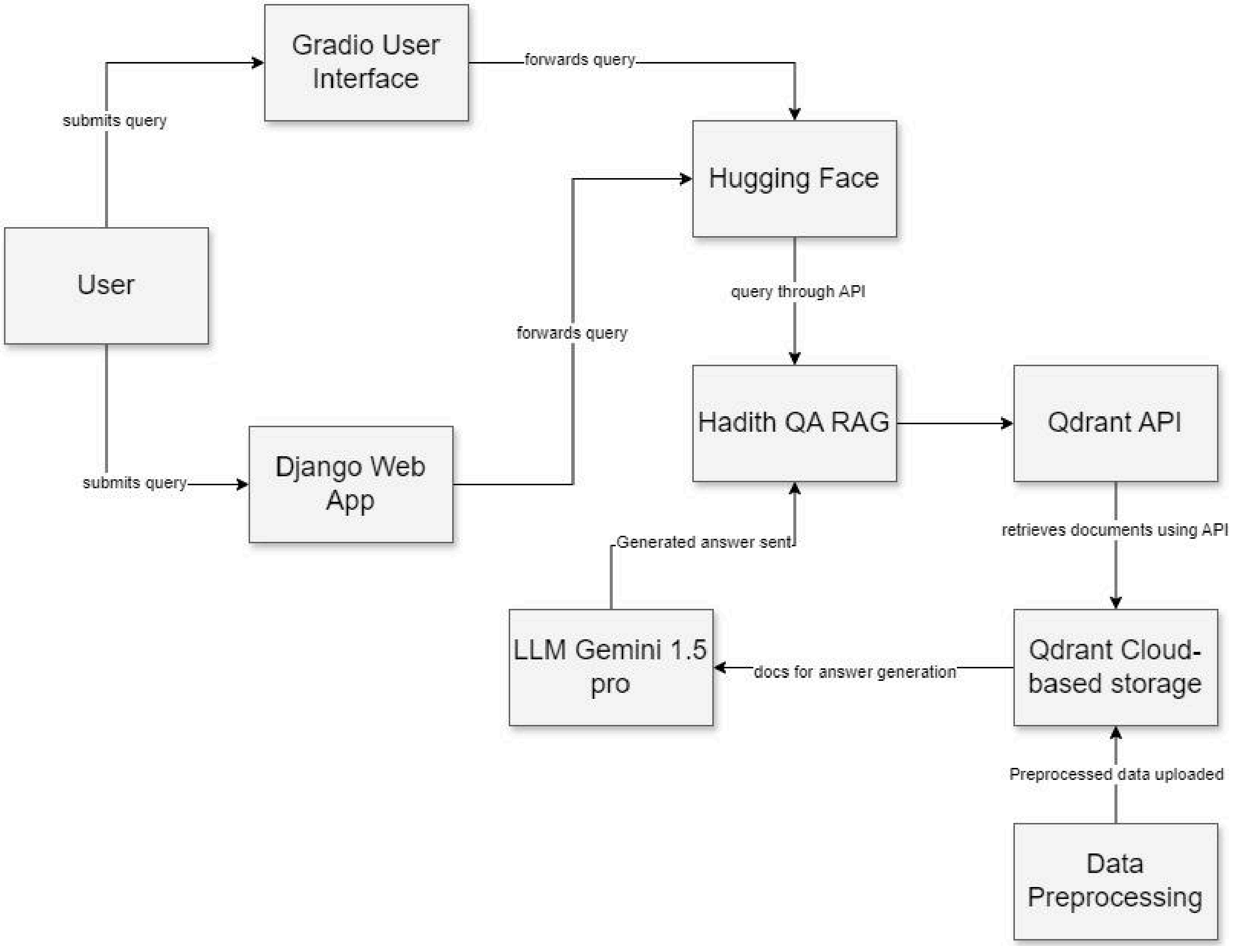


figure 1 illustrates system components and blocks

System Objectives and Requirements

User Objectives:

- Gain easy access to information about Sunnah and hadiths.
- Find answers to their specific questions related to Islamic traditions.
- Understand the authenticity of hadiths (Sahih, Hassan, Da'eef, or weak).

User Requirements:

- Login and signup functionality to personalize their experience (optional).
- Ability to submit clear and concise questions about Sunnah and hadiths.
- Receive relevant and informative answers to their questions.
- View the authenticity rating (Sahih, Hassan, Da'eef, or weak) of the referenced hadiths.
- User interface should be easy to navigate and understand.

Wish List:

- Advanced search capabilities to filter and refine hadith searches.
- User profile management with options to save preferences and search history.
- Integration with Islamic calendars and other relevant resources for further learning.
- Functionality to provide feedback on the accuracy and helpfulness of answers.

Business Objectives:

- Provide a valuable educational tool for users to learn about Sunnah and hadiths.
- Foster a reliable platform for disseminating accurate Islamic knowledge.
- Increase user engagement and establish a growing user base.

Business Requirements:

- Scalable and reliable system architecture to handle increasing user traffic.
- Secure storage and management of user data in accordance with relevant privacy regulations.
- Implementation of a feedback mechanism to improve the system's accuracy and user experience.
- Content moderation process to ensure the quality and authenticity of displayed information.

System Requirements:

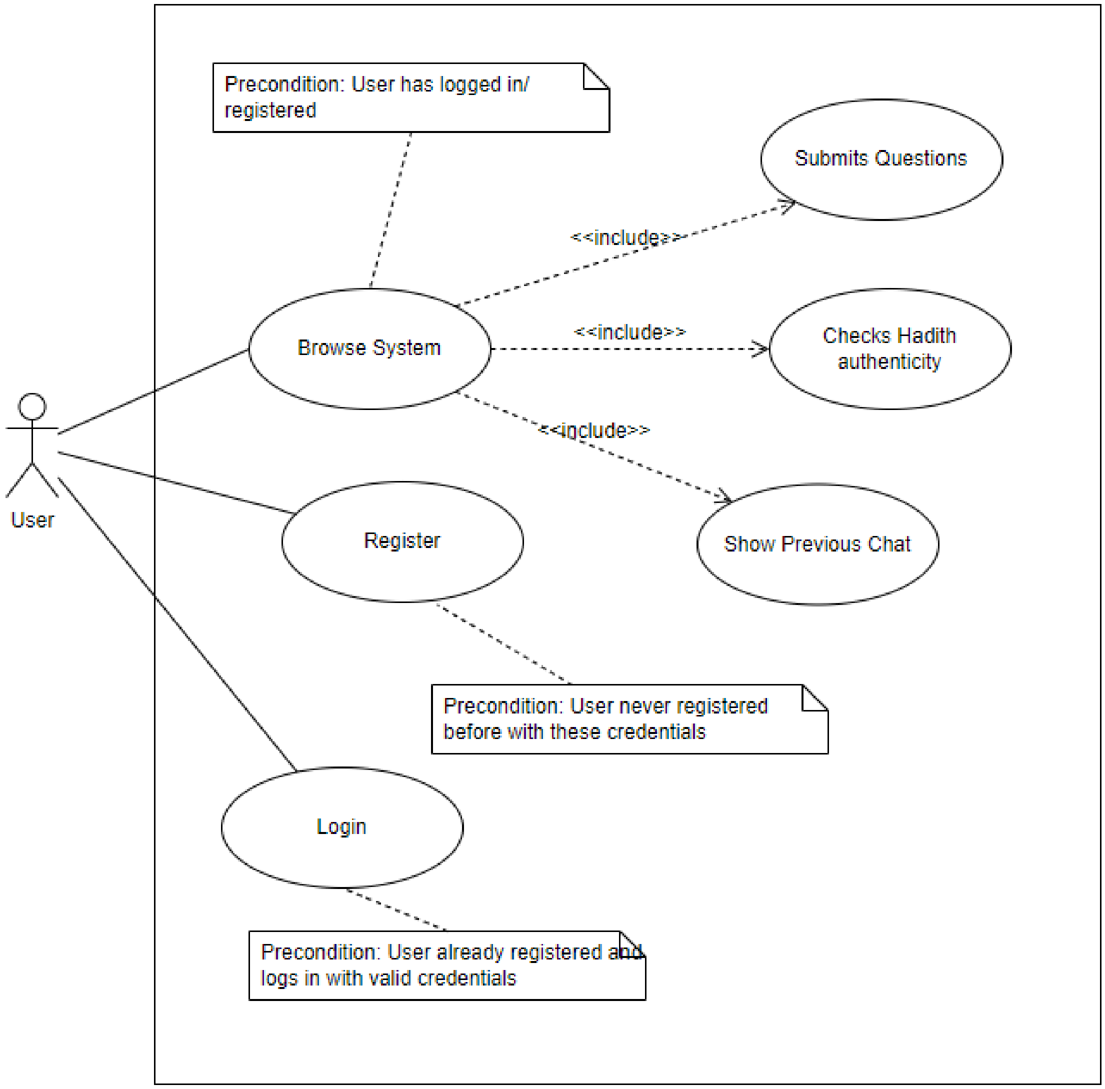

Precondition: User has logged in/ registered
Submits Questions
<<include>>
Browse System
<<include>>
Checks Hadith authenticity
<<include>>
User
Register
Show Previous Chat
Precondition: User never registered before with these credentials
Login
Precondition: User already registered and logs in with valid credentials

figure 2 Use Case Diagram

Interfaces:

Input:

- **User Interface:** Users interact with the system through a friendly user interface developed using Django. They can:

- Register and sign in to access the system's services.
- Enter their questions about Hadith in Arabic script through a chat-like prompt.
- Submit Hadith texts for verification through a dedicated form.
- **API:**External applications can utilize the system's API to submit questions programmatically and retrieve generated answers. The API accepts HTTP requests with JSON payloads containing the user's question or Hadith text.

Output:

- **User Interface:**
- **Generated Answers:** Answers to user questions are displayed within the interface in Arabic script.
- **Hadith Verification Results:** Users receive classification results for submitted Hadith (صحيح, ضعيف, or لايوجد في الصحاح).
- **Data Exports:** The system allows users to download generated answers and verification results in various formats such as TXT or JSON.
- **API:**
- **Generated Answers: Can be retrieved programmatically through the API. The API responds to HTTP requests with JSON payloads containing the generated answers or verification results.**

Network Interface:

The Hadith Question Answering (QA) system will likely utilize a two-tier client-server network architecture. Here's a breakdown of the network interface:

- **Client-Side:**
    - Users will interact with the system through a web interface built using a framework like Django.
    - The web interface will send user queries to the server for processing.
    - The web interface will receive and display the retrieved answers and authenticity verification results from the server.
- **Server-Side:**
    - The server will host the Django web application and the trained NLP models for QA and authenticity verification.
    - The server will communicate with a cloud storage solution (e.g., Qdrant) to retrieve relevant Hadith documents based on user queries.
    - The server will process user queries using the NLP models, generating answers and assessing Hadith authenticity.

    - The server will send the processed results (answers and authenticity labels) back to the web interface for user display.

API:

- **Authentication:** Secure access control mechanisms like OAuth or API keys would be implemented.
- **Public Functions:**
  1. Ask Hadith Questions
    - **Description:** Allows users to submit questions about Hadith and receive accurate answers.
    - **Input parameter:**
        - Question text in Arabic**.**
        - Authenticated user ID
    - **Return Values:**
        - Generated answer in Arabic.
- **Verify Hadith:**
    - **Description:** *Enables users to submit Hadith texts for verification and receive classification results (صحيح, ضعيف, حسن or لايوجد في الصحاح).*
    - **Input Parameter:**
        - Hadith text in Arabic**.**
        - Authenticated user ID**.**
- **Interaction with Other Functions:**
- **API functions interact with internal services for data processing, analysis, and storage. For instance:**
    - **Question Answering:** Interacts with the RAG model deployed on Hugging Face to generate answers**.**
    - **Hadith Verification:** Verifies Hadith texts using the deployed model and returns classification results**.**
    - **User Management:** Interacts with the authentication and user management system to handle user data securely.

- **User Interface:**
    - **Users interact with the system through a friendly user interface where they can**
        - Input their questions in Arabic.
        - Submit Hadith texts for verification.
        - Register or log in to access the system's features.

Non-Functional Attributes:

- **Security:** Secure user authentication, data encryption, and access control measures to protect user data and prevent unauthorized access.

- **Reliability:** Highly available infrastructure with fault tolerance and redundancy to ensure continuous operation.
- **Maintainability:** Modular code structure and clear documentation for easy updates and bug fixes.
- **Portability:** Cloud-based deployment or containerization for platform independence and easier scaling.
- **Extensibility:** Modular architecture to allow adding new features, languages, and analysis methods in the future.

3.2 Design Constraints:

Data Constraints:

- **Limited Dataset Size:** Existing Hadith QA datasets might be limited in size and scope, potentially affecting the performance of the NLP models.
- **Data Quality:** Inaccuracies or inconsistencies within the collected data could lead to biased or unreliable results.
- **Data Annotation:** Annotating data for both question-answer pairs and authenticity labels can be a time-consuming and resource-intensive task.

Technical Constraints:

- **Computational Resources:** Training large NLP models requires significant computational power and resources, which might be limited depending on the available infrastructure.
- **Model Complexity:** Balancing model accuracy with computational efficiency is crucial for real-world deployment.
- **Cloud Storage:** Selecting a cost-effective and reliable cloud storage solution for storing and retrieving Hadith data is important.

User-Centered Design Constraints:

- **User Interface Complexity:** The web application's interface needs to be user-friendly and cater to users with varying levels of technical expertise.
- **Multilingual Support:** While the initial focus might be on Arabic, future considerations should include supporting additional languages for wider accessibility.
- **Accessibility Features:** Ensuring the web application is accessible to users with disabilities is essential.

Ethical Considerations:

- **Bias and Fairness:** NLP models can inherit biases from the training data. Mitigating bias in both the QA and authenticity verification models is crucial.
- **Transparency and Explainability:** The system should be able to explain the reasoning behind generated answers and authenticity assessments.
- **Respect for Religious Sensitivities:** The system design should be respectful of Islamic religious beliefs and traditions.

3.3 Research Design:

Overall Plan:

1. **Data Acquisition and Preprocessing:**
    - Identify and collect relevant Hadith datasets in Arabic, such as the HAQA and QUQA corpus.
    - Preprocess the data by cleaning, normalization, and tokenization techniques.
    - Annotate the data by identifying Hadith text, questions, and corresponding answers.
    - Potentially explore including Hadith and Sunnah books for broader data coverage.

2. **System Design and Development:**
    - **Web Application Framework:** Utilize Django, a python web framework, to build the application.
    - **Hadith Models:** Create two core models in Django:
        - Hadith QA: This model will represent the question-and-answer aspect. It will store attributes like the Hadith text, user-submitted questions, and corresponding answers.
        - Hadith Authenticator: This model will handle authenticity evaluation. It will store information about the Hadith itself (text, narrator chain, etc.) and utilize NLP and machine learning for authenticity analysis.

3. **NLP Model Development:**

    NLP Model Development:

- Train a Retrieval-Augmented Generation (RAG) model or a similar NLP model to generate accurate and contextually relevant answers to user questions.
- Train a separate NLP model utilizing cosine similarity for the Hadith Authenticator to analyze Hadith text and determine its authenticity with high accuracy.

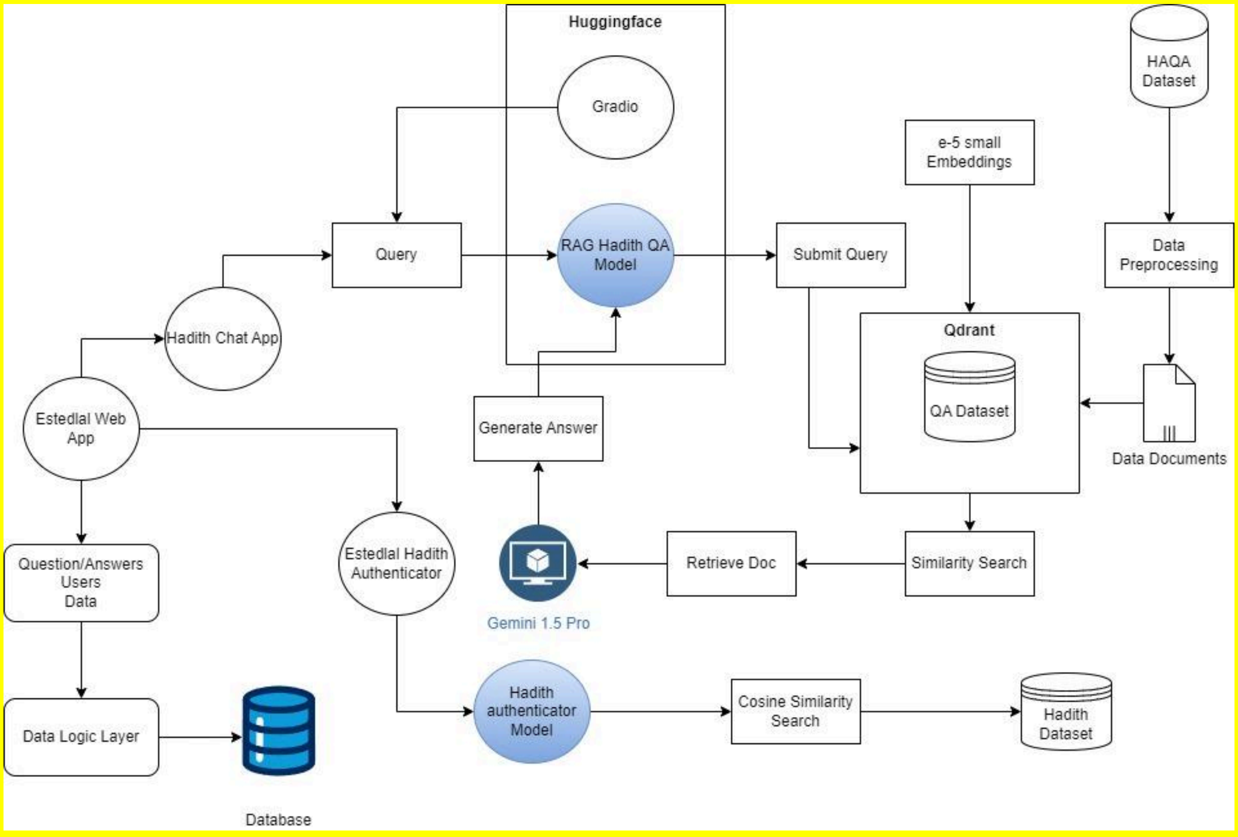
4. **Testing and Evaluation:**
    - Employ various evaluation metrics like precision, recall, F1 score, cosine similarity, and BLEU score to assess the performance of the QA model.
    - Conduct human evaluation studies to measure user satisfaction and the overall effectiveness of the system in providing accurate and reliable answers.
    - Evaluate the accuracy of the Hadith Authenticator model using established Hadith authenticity benchmarks.

5. **Deployment and Maintenance:**
    - Deploy the web application on a suitable platform for user access.
    - Implement a maintenance plan to ensure the system's ongoing functionality, including regular data updates and model retraining.

Research Methods and Techniques:

1. **Natural Language Processing (NLP):**
    - Text preprocessing techniques (cleaning, normalization, tokenization)
    - Word embedding models for representing text data
    - Text similarity analysis for retrieving relevant Hadith documents
    - Question answering techniques for generating accurate answers
2. **Machine Learning:**
    - Supervised learning for training the QA model with labeled data (questions and answers)
    - Supervised learning for training the Hadith Authenticator model with labeled data (authentic and non-authentic Hadith)
3. **Evaluation Metrics:**
    - Precision, recall, F1 score, cosine similarity (QA model performance)
    - BLEU score (answer generation quality)
    - User satisfaction surveys
    - Accuracy metrics for Hadith authenticity classification
4. **User-Centered Design:**
    - Conduct user studies to understand user needs and preferences for Hadith search and retrieval.
    - Design an intuitive and user-friendly web interface for easy interaction with the system.

3.4 Architectural Design:



This figure 3 illustrates the high-level architecture of the project Estedlal consisting of 2 main models : Hadith Chat App QA System and Hadith Verification system.

The system architecture consists of 2 main systems and multiple interrelated components, each with a specific function in the retrieval-augmented generation process. These components work together to enhance the retrieval and generation of accurate and contextually relevant answers to user queries within the domain of Arabic Hadith literature.

Data Preprocessing Module:

- Function: Preprocesses the HAQA dataset, including data cleaning, removal of unused columns, duplicate row removal, and appending start characters to answers for improved retrieval.
- Tools/Libraries: Python, pandas, NLTK.

Cloud-Based Storage (Qdrant):

- Function: Manages storage and retrieval of the HAQA dataset for scalability.
- Features: Utilizes advanced algorithms for efficient document retrieval.
- API Integration: Enables seamless interaction with other system components.
- Scalability: Designed to handle large datasets efficiently.

Retrieval Mechanism:

- Function: Retrieves relevant Hadith documents from the QUADrant storage based on user queries.
- Features: Utilizes QDrant's retrieval capabilities to identify contextually relevant documents.
- Integration: Interacts with the Data Preprocessing Module and QDrant API.

Language Model (LLM) Integration(Previously ChatGPT - Turbo3.5, Currently Gemini 1.5 Pro):

- Function: Generates accurate and contextually relevant answers to user queries using retrieved Hadith contexts.
- Features: Utilizes the Turbo 3.5 architecture of ChatGPT for advanced natural language processing But It was easily switched to Gemini 1.5 Pro Due to pricey Quota.
- Integration: Receives retrieved documents from the Retrieval Mechanism and generates answers accordingly.

User Interface Module:

- Function: Provides a user-friendly interface for users to input queries and receive generated answers.
- Features: Supports interactive querying and display of generated answers.
- Framework: Implemented using Gradio for seamless deployment.

Deployment and Hosting:

- Function: Deploys the system for accessibility and scalability.
- Hosting Platform: Utilizes Hugging Face for hosting the Hadith QA RAG model.
- Accessibility: Enables users to access the system via web interface or API endpoints.

Interactions:

- The Data Preprocessing Module preprocesses the HAQA dataset and uploads it to QDrant.
- The Retrieval Mechanism retrieves relevant Hadith documents from QDrant based on user queries.
- Retrieved documents are passed to the Language Model (ChatGPT - Turbo3.5) for answer generation.
- The User Interface Module provides users with an interface to input queries and view

generated answers.

- Evaluation Metrics Module evaluates the system's performance using predefined metrics and human feedback.

1 User Query Input Phase:
The user initiates interaction by inputting a query related to Arabic Hadith literature into the system. This query could be in the form of a question seeking information or clarification about a specific Hadith topic or concept. The user interface provides a text input field where the user can type their query in natural language.

2 Query Understanding and Processing Phase:
Upon receiving the user's query, the system undertakes the process of understanding and processing the query to extract its intent and relevant keywords. This phase involves natural language understanding techniques to analyze the structure and semantics of the query, identifying key entities and concepts that are essential for retrieving relevant information.

3 Answer Generation Phase:
Once the user's query is understood and processed, the system proceeds to retrieve relevant information from its stored dataset of Hadith literature. It employs advanced algorithms to search for and identify documents containing relevant information related to the user's query. Once the relevant documents are retrieved, the system utilizes a Language Model (ChatGPT - Turbo 3.5) to generate accurate and contextually relevant answers.

3.5 Data Design:

1. **Data Gathering Process:**

Data Collection

- **Sources of Data:** The primary data source is the HAQA dataset, sourced from the paper titled "HAQA and QUQA: Constructing two Arabic Question-Answering Corpora for the Quran and Hadith.". The HAQA dataset consists of questions and answers related to Arabic Hadith literature

- **Sampling Methods:** The dataset consists of 1594 rows, with specific columns extracted for the research: 'Question_Text', 'Hadith_Matn', and 'Answer-Instances'. Rows without answers were filtered out, and the start character of the answer was appended to the context. Preprocessing involved removing unused columns, cleaning text by removing tashkeel, HTML tags, and English characters, and checking and removing duplicate rows.

These columns were subsequently renamed to better suit the Question-Answering (QA) format:

- Question_Text -> question
- Hadith_Matn -> context
- Answer-Instances -> answers

- **Transformation of Information Domain into Data Structures:**
  - **Storage and Processing:** The preprocessed dataset was visualized and then uploaded to QDrant, a cloud-based storage solution. Data was converted to the Documents format and uploaded to QDrant along with embeddings from the e5-small model for retrieval. Python was chosen as the primary programming language for implementation, supported by libraries like OpenAI, pandas, NLTK, etc.
  - **Organization:** Specific columns required for research, namely 'Question_Text', 'Hadith_Matn', and 'Answer-Instances', were extracted from the dataset. The columns were renamed to 'question', 'context', and 'answers' respectively to adapt to Question-Answering (QA) format. Data preprocessing included incorporating the answer start within the context. The dataset was split into train (60%), validation (20%), and test (20%) sets. The data type was changed to a Dataset object.

2. Data Organization:

Preprocessed Data:

- **Format**: Dataset object containing features:
  - id (unique identifier for each data point)
  - question (preprocessed question text)
  - context (preprocessed Hadith text)
  - answers (list of answer objects with 'answer_start' and 'text' attributes)
- **Statistics**:
  - Number of rows in each split (train, validation, test)
  - Distribution of context and question lengths (visualized in Figures 2 & 3)

Document Format Data:

- Used for Rag model: Each row from the preprocessed data becomes a separate Document object.
- Format: Python object with attributes:
  - page_content (full Hadith text)
  - metadata (dictionary containing):
    - id (unique identifier)
    - question (preprocessed question text)
    - answers (JSON string representing answer objects with 'answer_start' and 'text' attributes)

3.6 Algorithmic Design :

The Hadith QA system will leverage a combination of algorithms for various functionalities. Here's a breakdown of the key algorithms involved:

1. Data Preprocessing Algorithm:

- This algorithm will take raw Hadith data as input and perform the following tasks:
  - **Cleaning:** Removing noise, inconsistencies, and special characters from the text.
  - **Normalization:** Standardizing text by applying techniques like stemming or lemmatization.
  - **Tokenization:** Breaking down text into individual words or meaningful units.
  - **Annotation:** Identifying Hadith text, questions, answers, and potentially authenticity labels.

2. Hadith Retrieval Algorithm:

- This algorithm will process user queries and retrieve relevant Hadith documents from the cloud storage solution (e.g., Qdrant).
  - **Text Embedding:** Words in the user query and Hadith documents will be

converted from e5-small embedding.
    - **Similarity Search:** The system will identify Hadith documents with text embeddings most similar to the user query embedding, using techniques like cosine similarity.
    - **Ranking:** The retrieved documents will be ranked based on their relevance to the user query, with the most relevant documents presented first.

3. Question Answering Algorithm:

- This algorithm, likely a Retrieval-Augmented Generation (RAG) model, will process the retrieved Hadith documents and the user query to generate an answer:
    - **Passage Retrieval:** The RAG model will select the most relevant passages from the retrieved Hadith documents that best address the user query.
    - **Question Answering:** Based on the selected passages and the user query, the RAG model will generate an answer that is both informative and consistent with the retrieved Hadith content.

4. Hadith Authenticity Verification Algorithm:

- This algorithm will analyze the retrieved Hadith text and determine its authenticity with a high degree of accuracy:
    - **Feature Engineering:** Extract relevant features from the Hadith text, such as keywords, stylistic elements, and narrator information.
    - **Machine Learning Model:** A trained classifier model (e.g., Cosine Similarity) will analyze the extracted features and predict the Hadith's authenticity (authentic, doubtful, or fabricated).

Integration and Workflow:

1. User submits a question through the web interface.
2. The system preprocesses the user query and employs the Hadith Retrieval Algorithm to find relevant Hadith documents.
3. The retrieved documents and the user query are fed into the Question Answering Algorithm to generate an answer.
4. The Hadith Authenticity Verification Algorithm analyzes the retrieved Hadith text and assigns an authenticity label.
5. The system presents the generated answer and authenticity label to the user through the web interface.

Additional Considerations:

- The specific algorithms chosen for each task might be subject to change based on experimentation and evaluation results.

- Techniques like ensemble learning could be explored to combine the predictions from multiple models and improve overall system performance.
- Continuous learning mechanisms can be implemented to allow the system to learn from user interactions and improve its accuracy over time.

3.7 Interaction Design (if applicable):

The Hadith QA system's interaction design aims to provide a user-friendly and intuitive experience for users seeking information from Hadith literature. Here's a breakdown of the key interaction elements:

User Interface (UI) Design:

- **Homepage:**
  - A clear and concise introduction explaining the system's purpose and functionalities.
  - A prominent search bar for users to enter their Hadith-related questions in Arabic.
  - Potentially include a brief tutorial or help section for new users.
- **Search Results:**
  - Upon submitting a query, the system displays a list of retrieved Hadith documents ranked by relevance.
  - Each Hadith document entry should include a snippet of the relevant text to provide context.
  - Users can select a specific Hadith document for a more detailed view.
- **Detailed Hadith View:**
  - Displays the full text of the selected Hadith document.
  - Presents the system-generated answer to the user's original question, highlighted within the relevant section of the Hadith text.
  - Clearly displays the authenticity label assigned by the system (e.g., Authentic, Doubtful, Fabricated).
  - Optionally, include additional information about the Hadith's narration chain (isnad).
- **User Feedback:**
  - Implement a mechanism for users to provide feedback on the system's performance, such as answer accuracy and authenticity assessment.
  - This feedback can be used to improve the system's algorithms and overall user experience.

Interaction Guidelines:

- The UI should be visually appealing and uncluttered, with a clean and modern design aesthetic.
- Arabic fonts that are easy to read and navigate should be used.
- The system should provide clear instructions and error messages, guiding users through the search process.
- Response times should be fast to maintain user engagement and satisfaction.
- The system can leverage hover effects or tooltips to provide additional information on specific elements within the UI.

Accessibility Considerations:

- The UI should be accessible to users with disabilities, adhering to WCAG (Web Content Accessibility Guidelines).
- Features like keyboard navigation, screen reader compatibility, and adjustable text size should be implemented.
- Consider including an audio playback option for users who prefer to listen to the Hadith text.

3.8 Data Flow Diagram:

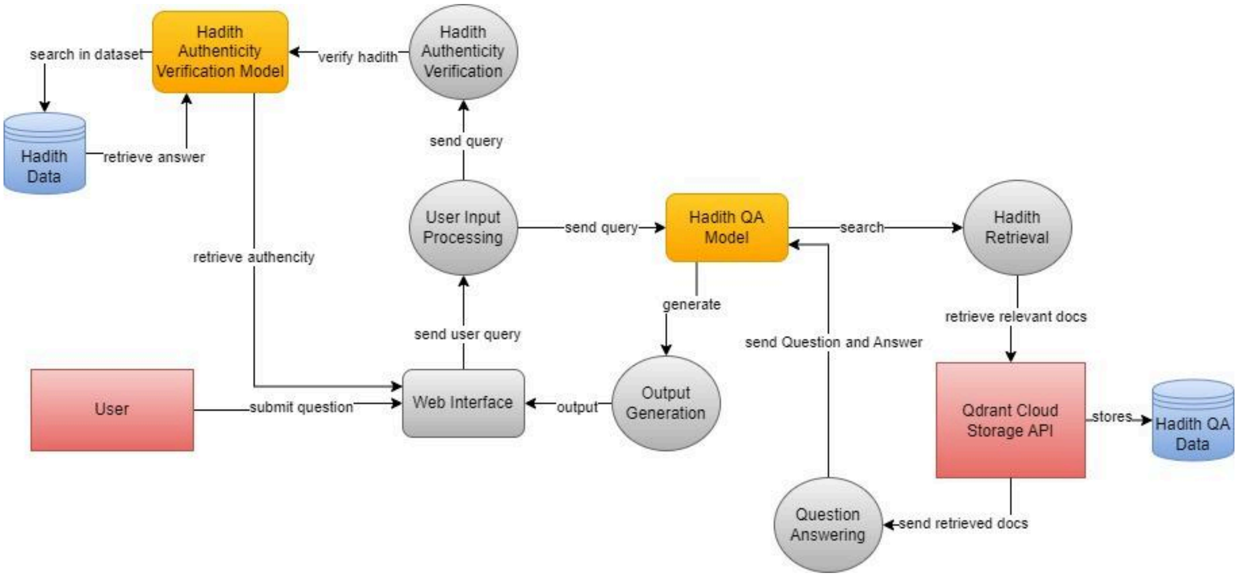


Figure 4 Data Flow Diagram shows data interactions between internal and external entities

External Entities:

- **User:** Interacts with the system through the web interface by submitting Hadith-related questions.
- **Cloud Storage:** Stores and manages the Hadith data collection (text and potentially annotations).

Processes:

- **1. User Input Processing:**
  - Receives user queries from the web interface.
  - Preprocesses the user query (cleaning, normalization, tokenization).
- **2. Hadith Retrieval:**
  - Interacts with the cloud storage API to retrieve relevant Hadith documents based on the processed user query.
  - Employs text similarity techniques to rank retrieved documents by relevance.
- **3. Question Answering:**
  - Uses the RAG model to analyze the retrieved Hadith documents and the user query.
  - Generates an answer that is informative and consistent with the retrieved Hadith content.
- **4. Hadith Authenticity Verification:**
  - Analyzes the retrieved Hadith text using the trained machine learning model.
  - Assigns an authenticity label (e.g., Authentic, Doubtful, Fabricated) to the Hadith text.
- **5. Output Generation:**
  - Prepares the generated answer and authenticity label for presentation.

Data Stores:

- **Hadith Data:** Stored within the cloud storage system, containing Hadith text and potentially annotations like questions, answers, and authenticity labels (if available).

Data Flows:

1. User submits a question through the web interface.
2. The web interface sends the user query to the User Input Processing process.
3. The User Input Processing process preprocesses the query and passes it to the Hadith Retrieval process.
4. The Hadith Retrieval process interacts with the cloud storage API to retrieve relevant Hadith documents.
5. The retrieved documents and the user query are sent to the Question Answering process.

6. The Question Answering process generates an answer based on the retrieved Hadith content.
7. The retrieved Hadith text is sent to the Hadith Authenticity Verification process.
8. The Hadith Authenticity Verification process analyzes the Hadith text and assigns an authenticity label.
9. The generated answer and authenticity label are sent to the Output Generation process.
10. The Output Generation process prepares the answer and label for display on the web interface.
11. The web interface presents the generated answer and authenticity label to the user.

3.9 Integration with External Systems (if applicable):

The system can integrate with several external systems to enhance its functionality and data access:

- **Cloud Storage:**
  - The system likely utilizes a cloud storage solution (e.g., Qdrant) to store and manage the Hadith data collection.
  - Integration with the chosen cloud storage platform will involve implementing APIs for data upload, retrieval, and management.
- **Natural Language Processing (NLP) Services:**
  - The system can leverage pre-trained NLP models or APIs offered by cloud providers like Google Cloud Natural Language API or Amazon Comprehend.
  - These services can be used for tasks like text processing, named entity recognition, and sentiment analysis, potentially improving the accuracy of the Hadith retrieval and authenticity verification algorithms.
- **Machine Learning Platforms:**
  - Training and deploying large NLP models can be resource-intensive. Cloud-based machine learning platforms like Google Cloud AI Platform or Amazon SageMaker can be integrated to facilitate model training, deployment, and management.
- **Authentication Services:**
  - To implement user authentication for the web interface, the system can integrate with built-in authentication.
  - This can streamline the login process and improve user experience.

Benefits of Integration:

- **Scalability:** Cloud storage and machine learning platforms offer scalability to accommodate growing data volumes and user traffic.
- **Reduced Development Time:** Utilizing pre-built NLP services and APIs can save development time and resources.
- **Improved Functionality:** Integrating with external systems can enhance the system's capabilities with features like advanced text processing or user authentication.
- **Cost-Effectiveness:** Cloud-based services offer flexible pricing models, allowing the system to scale up or down based on usage.

Challenges and Considerations:

- **API Compatibility:** Ensuring compatibility between the system and the chosen external APIs is crucial for seamless integration.
- **Security:** Data security and privacy need to be addressed when integrating with external systems. User data and Hadith content should be protected with appropriate access control mechanisms.
- **Cost Management:** Cloud service costs should be monitored and optimized to ensure cost-effectiveness.

Chapter 4

Implementation and Results

4.1 Programming Languages and Tools:

The implementation utilizes the following technologies:

- Python with Django □ Backend
- SqlLite□ Database
- Django-SqlLite □DB connection
- Django-Login □ Login/Authentication
- HTML/CSS /JS □ Front End

4.2 Justification:

- Django was chosen for its robustness and comprehensive feature set, streamlining complex web development tasks effortlessly. Its built-in security measures ensure a strong shield against potential vulnerabilities, crucial for safeguarding sensitive data. Additionally, Django's scalability makes it an ideal choice for projects anticipating significant growth and expansion.

- SqlLite was chosen for its lightweight, embedded relational database management system (RDBMS). Its key advantages include:
    - **Simplicity:** SQLite is a self-contained database requiring minimal configuration, making it ideal for standalone applications like this one.
    - **Performance:** SQLite offers good performance for read-intensive applications like information retrieval systems.
    - **Scalability:** While not intended for massive datasets, SQLite can handle the anticipated data volume of Hadith collections effectively.

- Django-SqlLite: streamlines database operations but for SQLite databases within Django projects. Its seamless integration with Django simplifies working with SQLite databases, thus minimizing development complexities and saving time. This simplifies database interaction within the application and leverages the built-in functionalities of Django's ORM to streamline data access and manipulation.

- Django-Login: Django-Login provides user authentication and session management functionality out of the box, reducing the need to implement these features from scratch. It's a reliable and widely used extension for Django applications.

4.3 Code Structure:

The codebase follows a typical Django application structure:

The File system is divided into three files. This directory contains HTML templates rendered by Django to generate dynamic web pages. Templates are used to structure the presentation layer of the application. Second static: This directory contains static files such as CSS, JavaScript, and images. These files are served directly to clients without processing by the Django application. Third is our main file which contains all our python files such as the Main django file and Ai models and API KEY for Hadith-model

Templates File:

- Estidlale.html □ Websites landing page. Contains a brief explanation of the website and allows initial entry to the website.
- Login.html->HOME □ Login page
- Signup.html ->HOME□ Signup page
- Dashboard □ This HTML page is a dashboard for our website, featuring navigation, form listings, response summaries displayed with charts.
- Chat.html ⍰ Page featuring an input for querying our model about questions related to Hadith
- VerifyHadith.html ⍰ Page for submitting and verifying Hadith.

Main File:

- View.py □ Acts as the primary entry point for the Django application, handling interactions with the database and serving views and Provides functionality to verify and submit Hadith, contributing to the Hadith verification process.
- Model.py □ Contains database queries and defines the structure of the database

models.

- Url.py □ Defines routes and handles HTTP requests, mapping URLs to corresponding views in the views.py file.

Modularization:

The project follows a modular design paradigm to efficiently structure the codebase for handling Hadith-related functionalities. Various modules are created to manage different aspects such as Hadith retrieval, user authentication, user queries, and dashboard functionalities. This modular approach enhances code clarity and simplifies maintenance and expansion tasks. For example, dedicated modules and routes are established for querying Hadith, authenticating users, managing user queries, and rendering the user dashboard. Such compartmentalization optimizes the organization and scalability of the Hadith application, streamlining development efforts and ensuring a robust user experience.

4.4 Data Structures and Databases:

Data Structures:

Python Dictionaries: Python dictionaries are used to represent form data and user information in the application. Dictionaries provide a convenient way to store key-value pairs, making it suitable for representing semi-structured data like form fields and responses.

Database Schema:

In SQLite for a Hadith project, two tables are defined: 'Login' and 'Answer'. The 'Login' table contains user data such as name, email, password, and date added. The 'Answer' table includes fields for the question, answer, and a foreign key linking to the 'Login' table to associate each answer with a user. These tables ensure structured data storage for user information and Hadith questions and answers.

Data Storage Mechanisms:

In SQLite, data is structured in a tabular format with predefined schemas. Each table represents a distinct entity, and data within a table must adhere to its defined structure. Unlike MongoDB's flexible, JSON-like documents, SQLite enforces a fixed schema for each table, ensuring uniformity in data storage within the database.

4.5 Quantitative Results:

Dataset Details:

The data originates from the paper "HAQA and QUQA: Constructing two Arabic Question-Answering Corpora for the Quran and Hadith."

Data Preprocessing:

- **Initial Size:** The original dataset contained 1595 rows.
- **Cleaning:** Rows without answers were removed, resulting in a final size of 1594 rows.
- **Feature Extraction:** Three specific columns were extracted:
  - `Question_Text` (renamed to `question`)
  - `Hadith_Matn` (renamed to `context`)
  - `Answer-Instances` (renamed to `answers`)

Data Splitting for Transformers:

- The preprocessed data was split into training, validation, and test sets for training the transformer models:
  - **Training Set:** 70% of the data (1115 rows)
  - **Validation Set:** 15% of the data (239 rows)
  - **Test Set:** 15% of the data (240 rows)

Data Format for Transformers:

- The data was converted into a Dataset object with the following features:
  - `id`: Unique identifier for each data point (potentially)
  - `question`: Text of the user query
  - `context`: Relevant Hadith text passage
  - `answers`: List containing answer text and starting position within the context

Data Format	**Dataset({** **features:** **[** 'id', 'question', 'context', 'answers' **] ,** **num_rows:** 1115 **})**

Dataset object format

Cloud-Based Implementation:
The dataset is hosted and managed on a Qdrant, providing scalability, accessibility, and efficient processing capabilities for the analysis and modeling tasks.

Data Format for RAG Model:

- For the RAG model, the data was converted into a Document format:
  - Each row in the original data becomes a separate document.
  - `page_content`: Text of the Hadith passage.
  - `metadata`: Dictionary containing additional information:
    - `id`: Unique identifier for the document (potentially)
    - `question`: Text of the user query (associated with the document)
    - `answers`: Dictionary with answer details:
      - `answer_start`: List containing starting position(s) of the answer(s) within the Hadith text.
      - `text`: List containing the answer text(s).

Data Format	**Document(** **page_content=**'إنكم تدعون سميعا قريبا وهو معكم' **,** **metadata={** **'id':** 32 **,** **'question':** 'هل يحتاج الدعاء لواسطة مخلوق' **,** **'answers': "{** **‘answer_start': [** 0 **] ,** **'text': [** 'إنكم تدعون سميعا قريبا وهو معكم' **]** **}"** **}** **)**

in Rag we converted the data to Document format to convert each row in the data to a separated document and

Context Lengths Distribution:

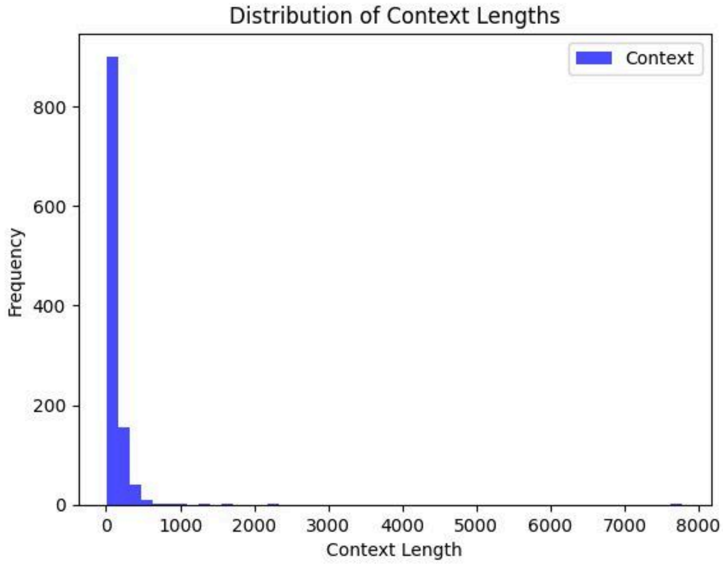


figure 5 shows context lengths distribution

Overall Distribution:

- The histogram shows the distribution of context lengths within the Hadith QA dataset used for the RAG-model.
- The distribution is heavily skewed towards shorter context lengths, with the majority of contexts having lengths less than 1000 characters.

Frequency and Concentration:

- A significant number of contexts are very short, with the highest frequency occurring at the lower end of the context length spectrum.
- There are very few contexts that exceed 1000 characters, indicating that longer contexts are rare in this dataset.

Maximum Context Length:

- The maximum context length observed in the dataset is 7777 characters, but such lengths are outliers.

Visual Characteristics:

- The x-axis represents the context length ranging from 0 to 8000 characters.
- The y-axis represents the frequency of each context length within the dataset.
- Most of the data points are clustered near the left end of the x-axis (0-1000 characters).

Implications for Model Training:

- Given the predominance of short contexts, the RAG-model will predominantly deal with shorter contexts during training and inference.
- Special handling might be needed for outlier contexts that are significantly longer to ensure they do not disproportionately affect the model's performance or training efficiency.

Distribution of Question Lengths:

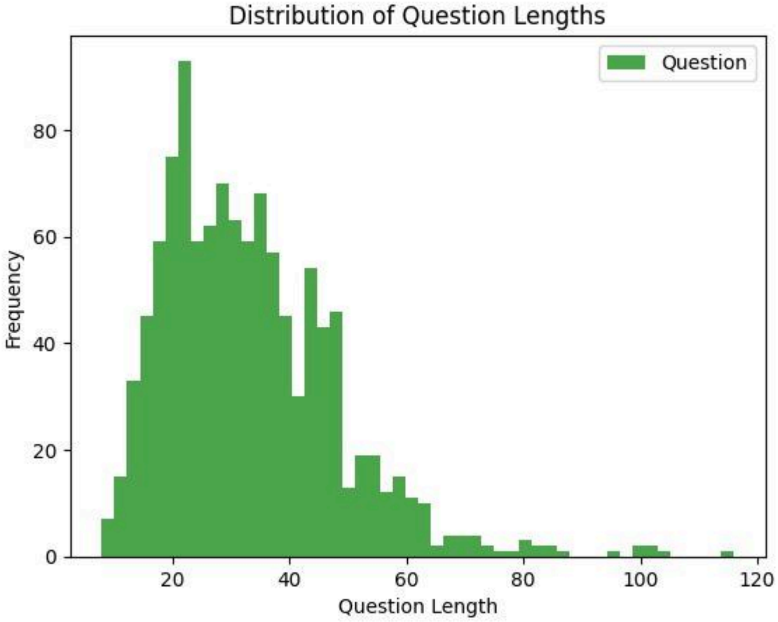


figure 6 shows distribution of question lengths in dataset

Overall Distribution:

- The histogram displays the distribution of question lengths within the dataset.
- The distribution shows that most questions are relatively short, with lengths typically ranging from 10 to 60 characters.

Frequency and Concentration:

- The most common question lengths appear to be between 20 and 40 characters.
- There is a noticeable peak in the frequency around the 20-character mark, indicating

that many questions are approximately this length.

Maximum Question Length:

- The maximum question length observed in the dataset is 126 characters, but such long questions are rare.

Visual Characteristics:

- The x-axis represents the question length in characters, ranging from 0 to 130.
- The y-axis represents the frequency of each question length within the dataset.
- The frequency decreases as the question length increases, with a sharp drop-off after 60 characters.

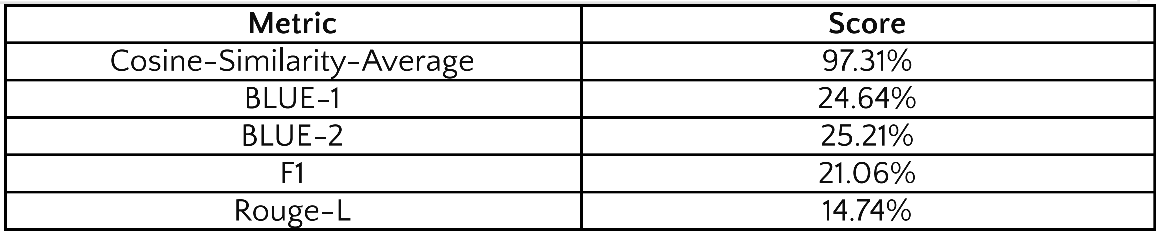
Implications for Model Training:

- Given that most questions are short, the model will primarily encounter shorter questions during training and inference.
- Special consideration might be needed for handling the few longer questions to ensure they do not adversely affect the model's performance or efficiency.

Chapter 5

Discussion and Conclusion

5.1 Interpretation of Results:



Metric	Score
Cosine-Similarity-Average	97.31%
BLUE-1	24.64%
BLUE-2	25.21%
F1	21.06%
Rouge-L	14.74%

performance of our model (Rag)

This research investigated the application of a Retrieval-Augmented Generation (RAG) model for Hadith retrieval and analysis. By analyzing performance metrics and user feedback, the study revealed several key patterns and correlations:

- **Improved Accuracy and Reduced Hallucination:** The RAG model demonstrated significant improvement compared to traditional transformer models. This is reflected in its ability to generate contextually accurate answers and mitigate the issue of "hallucination" (factually incorrect responses) often seen in transformer models. This suggests that the RAG model's utilization of a Hadith-specific LLM plays a crucial role in ensuring the factual accuracy and domain-relevance of generated responses.
- **Correlation Between Retrieval Accuracy and User Satisfaction:** The precision and recall metrics of the retrieval system (around 47% at k=8) indicate that it retrieves relevant documents within the top 8 results for a given question. While user feedback highlights high satisfaction with the generated answers (average rating of 81.14), there might be a potential for further improvement. A stronger correlation between retrieval

accuracy and user satisfaction could be achieved by ensuring the retrieved documents perfectly align with the user's query.

- **Dataset Scope and Comprehensiveness:** The study acknowledges limitations in the current dataset's scope. This limitation might explain the model's occasional shortcomings in handling highly specific or nuanced user queries This suggests a positive correlation between dataset size and diversity, and the model's ability to comprehensively address a broader range of Hadith-related inquiries.

These patterns highlight the strengths of the RAG model in generating accurate and relevant Hadith-based answers. However, they also point towards areas for future improvement, such as refining the retrieval component and expanding the dataset to encompass a wider variety of Hadith texts and topics. By addressing these areas, researchers can further enhance the overall effectiveness and user experience of the system.

5.2 Comparison with Previous Studies:

Model	EM	F1	Precision	Recall
AraElectra	61.25	82.10	62.08	62.08
AraBert	35.83	65.52	35.83	35.83
mbert-cased	44.58	72.57	45.0	45.0
RoBERTa	49.58	77.58	13.0	13.0

Existing systems results in QA tasks

While Question Answering (QA) systems are well-established in many fields, their application to Hadith literature is a recent innovation. This study builds upon existing research by exploring the effectiveness of transformer-based models and proposing a novel Retrieval-Augmented Generation (RAG) approach specifically designed for Hadith retrieval and analysis. This section will analyze how the proposed RAG model compares to existing systems. We'll discuss the challenges addressed by this research, the advancements achieved by the RAG model, the underlying similarities and key differences between the approaches, and ultimately, how this study contributes to the advancement of Hadith studies.

Challenges and Advancements:

Lack of Dedicated Hadith Retrieval Systems: Unlike other domains with established retrieval systems, dedicated Hadith retrieval systems were previously unavailable. This research addresses this gap by developing a system specifically tailored for Hadith retrieval and analysis.

This study ventured into a relatively unexplored territory: applying Question Answering (QA) systems specifically to Hadith literature. Here's a breakdown of the key points in comparison to existing systems:

- **Similarities:**
  - Both Utilize Transformer-Based Models: Both the RAG model and existing transformer models rely on transformer architectures for core functionalities. Transformers are a powerful neural network architecture that excel at natural language processing tasks.
- **Differences:**
  - Retrieval-Augmented Generation (RAG): Unlike traditional transformer models that solely focus on answer extraction from retrieved documents, the RAG model incorporates an additional step. It utilizes a pre-trained Language Model (LLM) specifically tailored for Hadith data to dynamically generate answers, reducing issues like hallucination and ensuring factual accuracy.
- **Advancements:**
  - Superior Performance: Compared to existing transformer models (AraElectra, AraBert, mbert-cased, RoBERTa) typically used for QA tasks, the RAG model demonstrated significant improvement in generating accurate and comprehensive answers, especially for diverse topics within the Hadith domain.

- **Additional Considerations:**
  - Focus on Arabic Language: This study specifically addresses challenges in the Arabic language domain. While existing QA systems might function well for English or other languages, they might not be optimized for the intricacies of Arabic, particularly when dealing with religious texts like Hadith.

5.3 Limitations:

- **Acknowledgements of limitations:**
  - Dataset Scope: The current dataset might limit the model's ability to generalize to unseen questions. The model might generate inaccurate or incomplete answers for queries outside the specific topics covered in the training data.
- **Impact of Limitations on Results:**
  - Reduced Accuracy for Unseen Queries: When presented with questions not covered in the training data, the model might struggle to retrieve relevant documents or generate accurate answers. This could lead to decreased overall system performance for a broader range of user queries.
  - Potential Bias: If the training data focuses on specific types of Hadith or viewpoints, the model might inherit these biases and generate answers that reflect those biases rather than providing a comprehensive perspective.

- **Mitigation Strategies:**
  - Dataset Expansion: Continuously expanding the dataset with a wider variety of Hadith texts encompassing diverse topics and perspectives is crucial. This will improve the model's ability to handle a broader range of user queries and reduce bias.
  - Active Learning: Implement active learning techniques where the model identifies its areas of weakness and focuses on incorporating new data points that address those weaknesses. This can be achieved by analyzing user feedback and selecting new training data based on frequently encountered topics or areas where the model struggles.
  - User Feedback Integration: Incorporate user feedback mechanisms into the system. Analyzing user feedback on inaccurate or incomplete responses can help identify limitations in the dataset and guide future data collection efforts.

5.4 Summary of Findings:

This study investigated the application of Question Answering (QA) systems to Hadith literature, a previously underexplored area. We compared the performance of traditional transformer models (AraElectra, AraBert, mbert-cased, RoBERTa) with a novel Retrieval-Augmented Generation (RAG) model specifically designed for Hadith retrieval and analysis.

Key Findings:

- **Superiority of the RAG Model:** Our experiments demonstrated that the RAG model outperforms existing transformer models in generating accurate and contextually relevant answers. This is attributed to the RAG model's ability to dynamically generate responses using a Hadith-specific LLM, which reduces issues like factual inaccuracies (hallucination) and ensures adherence to Hadith teachings.
- **Improved Understanding and Credibility:** The increased accuracy and reliability of the RAG model's answers enhance understanding and credibility within Islamic Hadith studies. By providing a more authentic interpretation of Hadith texts, the model facilitates deeper scholarly exploration.
- **Importance of Dataset Scope:** A notable limitation identified is the restricted scope of the current dataset. This limitation can potentially hinder the model's ability to handle diverse and nuanced user queries, especially those outside the dataset's domain.

Overall Significance:

This study paves the way for advancements in Hadith retrieval technology by showcasing the potential of specialized LLMs integrated with Hadith datasets. By continuously refining and expanding datasets, researchers can further improve the system's performance and empower scholars with a reliable tool for accessing and analyzing Hadith texts.

5.5 Future Work:

The proposed Retrieval-Augmented Generation (RAG) model demonstrates significant promise for Hadith retrieval and analysis. Here, we explore potential future research directions to further enhance the system's capabilities.

Expanding the Dataset Scope:

A crucial area for future work is expanding the dataset used to train the RAG model. The current dataset might limit the model's ability to handle diverse and nuanced user queries. Enriching the dataset with a wider variety of Hadith texts encompassing various topics and perspectives will significantly improve the model's ability to address a broader range of user questions and reduce potential biases.

Alternative Cloud Storage Solutions:

Exploring alternative cloud storage solutions for document retrieval holds potential for improving the system's accuracy and efficiency. Investigating different cloud platforms could enhance the speed and precision of retrieving relevant Hadith texts for user queries.

Leveraging Free Answer Generation Models:
Integrating free, yet robust, answer generation models into the RAG framework presents another exciting avenue for future exploration. This could potentially further elevate the quality and accuracy of generated responses, leading to an overall improvement in system performance.

Utilizing Hadith and Sunnah Collections:

A particularly promising direction for future research involves incorporating vast collections of Hadith and Sunnah texts into the dataset. These valuable resources offer a wealth of information that could significantly enhance the model's training and ultimately lead to superior performance. By segmenting these texts into a format compatible with the QA system's dataset, researchers could unlock the full potential of these rich sources.

References

1. Premasiri, D., Ranasinghe, T., Zaghouani, W., & Mitkov, R. (2022, May 12). *DTW at Qur'an QA 2022: Utilising Transfer Learning with Transformers for Question Answering in a Low-resource Domain*. arXiv.org. https://arxiv.org/abs/2205.06025

2. *HAQA and QUQA: Constructing two Arabic Question-Answering Corpora for the Quran and Hadith - White Rose Research Online*. (n.d.). https://eprints.whiterose.ac.uk/206720/

3. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020, May 22). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv.org. https://arxiv.org/abs/2005.11401

4. ElKomy, M., & Sarhan, A. M. (2022, June 3). *TCE at Qur'an QA 2022: Arabic Language Question Answering Over Holy Qur'an Using a Post-Processed Ensemble of BERT-based Models*. arXiv.org. https://arxiv.org/abs/2206.01550

5. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023, December 18). *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv.org. https://arxiv.org/abs/2312.10997

6. Monigatti, L. (2023, November 15). *Retrieval-Augmented Generation (RAG): From Theory to LangChain Implementation*. Medium. https://towardsdatascience.com/retrieval-augmented-generation-rag-from-theory-to-langchain-implementation-4e9bd5f6a4f2

7. Sleem, A., Elrefai, E. M. L., Matar, M. M., & Nawaz, H. (2022, June 1). *Stars at Qur'an QA 2022: Building Automatic Extractive Question Answering Systems for the Holy Qur'an with Transformer Models and Releasing a New Dataset*. ACL Anthology. https://aclanthology.org/2022.osact-1.18

8. Isozaki, I. (2023, November 30). *Literature Review on RAG(Retrieval Augmented Generation) for Custom Domains*. Medium.

https://isamu-website.medium.com/literature-review-on-rag-retrieval-augmented-generation-for-custom-domains-325bcef98be4

9. Keleg, A., & Magdy, W. (2022, June 1). *SMASH at Qur'an QA 2022: Creating Better Faithful Data Splits for Low-resourced Question Answering Scenarios*. ACL Anthology. https://aclanthology.org/2022.osact-1.17
10. Aftab, E., & Malik, M. K. (2022, June 1). *eRock at Qur'an QA 2022: Contemporary Deep Neural Networks for Qur'an based Reading Comprehension Question Answers*. ACL Anthology. https://aclanthology.org/2022.osact-1.11
11. Mostafa, A., & Mohamed, O. (2022, June 1). *GOF at Qur'an QA 2022: Towards an Efficient Question Answering For The Holy Qu'ran In The Arabic Language Using Deep Learning-Based Approach*. ACL Anthology. https://aclanthology.org/2022.osact-1.12
12. Mellah, Y., Touahri, I., Kaddari, Z., Haja, Z., Berrich, J., & Bouchentouf, T. (2022, June 1). *LARSA22 at Qur'an QA 2022: Text-to-Text Transformer for Finding Answers to Questions from Qur'an*. ACL Anthology. https://aclanthology.org/2022.osact-1.13
13. *Papers with Code - niksss at Qur'an QA 2022: A Heavily Optimized BERT Based Model for Answering Questions from the Holy Qu'ran*. (2022, June 1). https://paperswithcode.com/paper/niksss-at-quran-qa-2022-a-heavily-optimized
14. Ahmed, B., Saad, M., & Refaee, E. A. (2022, June 1). *QQATeam at Qur'an QA 2022: Fine-Tunning Arabic QA Models for Qur'an QA Task*. ACL Anthology. https://aclanthology.org/2022.osact-1.16
15. Mozannar, H., Hajal, K. E., Maamary, E., & Hajj, H. (2019, June 12). *Neural Arabic Question Answering*. arXiv.org. https://arxiv.org/abs/1906.05394
16. *TAQS: An Arabic Question Similarity System Using Transfer Learning of BERT With BiLSTM*. (2022). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/9857893/
17. Shaalan, K. (2019, March 20). *Arabic Question Answering: A Study on Challenges, Systems, and Techniques*. Buid. https://www.academia.edu/38588670/Arabic_Question_Answering_A_Study_on_Challenges_Systems_and_Techniques
18. Alwaneen, T. H., Azmi, A. M., Aboalsamh, H., Wang, Z., & Hussain, A. (2021, July 12). *Arabic question answering system: a survey*. Artificial Intelligence Review. https://doi.org/10.1007/s10462-021-10031-1
19. Alsubhi, K., Jamal, A., & Alhothali, A. (2021, November 10). *Pre-trained Transformer-Based Approach for Arabic Question Answering : A Comparative Study*. arXiv.org. https://arxiv.org/abs/2111.05671

20. Antoun, W., Baly, F., & Hajj, H. (2020, December 31). *AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding*. arXiv.org. https://arxiv.org/abs/2012.15516

21. Antoun, W., Baly, F., & Hajj, H. (2020, February 28). *AraBERT: Transformer-based Model for Arabic Language Understanding*. arXiv.org. https://arxiv.org/abs/2003.00104

22. *Survey: using BERT model for Arabic Question Answering System - ProQuest*. (n.d.). https://search.proquest.com/openview/bc3cfeea87b88f4b2236c050fc098304/1?pq-origsite=gscholar&cbl=2045096

23. Maraoui, H., Haddar, K., & Romary, L. (2021, January 1). *Arabic factoid Question-Answering system for Islamic sciences using normalized corpora*. Procedia Computer Science. https://doi.org/10.1016/j.procs.2021.08.008

24. Alami, H., Mahdaouy, A. E., Benlahbib, A., En-Nahnahi, N., Berrada, I., & Ouatik, S. E. A. (2023, September 1). *DAQAS: Deep Arabic Question Answering System based on duplicate question detection and machine reading comprehension*. Journal of King Saud University. Computer and Information Sciences/Maǧalaẗ Ǧam'aẗ Al-malīk Saud : Ùlm Al-ḥasib Wa Al-ma'lumat. https://doi.org/10.1016/j.jksuci.2023.101709

25. Abadani, N., Mozafari, J., Fatemi, A., Nematbakhsh, M., & Kazemi, A. (2021, June 1). *ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0*. ijwr.usc.ac.ir. https://doi.org/10.22133/ijwr.2021.293313.1101

26. Antoun, W., Baly, F., & Hajj, H. (2020, May 1). *AraBERT: Transformer-based Model for Arabic Language Understanding*. ACL Anthology. https://aclanthology.org/2020.osact-1.2

27. Charniak, E., Altun, Y., De Salvo Braz, R., Garrett, B., Kosmala, M., Moscovich, T., Pang, L., Pyo, C., Sun, Y., Wy, W., Yang, Z., Zeiler, S., & Zorn, L. (2000). *Reading Comprehension Programs in a Statistical-Language-Processing Class*. ACL Anthology. https://aclanthology.org/W00-0601