Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REVOLUTIONIZING INFORMATION RETRIEVAL: UNVEILING A NEXT-GENERATION AI-POWERED QUESTION-ANSWER SYSTEM FOR COMPREHENSIVE DOCUMENT ANALYSIS

ZAHER NAJWA[1], GHAZOUANI MOHAMED[2], CHAFIQ NADIA[3]

[1 3] Laboratory of Sciences and Technology of Information and Education Faculty of science Ben M'sik, Hassan II University Casablanca, Morocco

[2] Laboratory of Information Technology and Modeling
Faculty of science Ben M'sik, Hassan II University Casablanca, Morocco

E-mail : [1]najwazaher@gmail.com, [2]ghazouani.fsbm@gmail.com, [3]nadia_chafiq@yahoo.fr

ABSTRACT

Following the introduction of a new public procurement decree in Morocco, public institutions and businesses have faced challenges in comprehending the updated rules for awarding contracts. The complexity of the new decree has made it difficult for these entities to adapt to the changes, underscoring the necessity for detailed guidance and training. To address these issues, this article suggests creating an advanced system to analyze documents related to the new law, aiding employees by simplifying access to crucial information. Utilizing state-of-the-art text analysis and natural language processing, the proposed tool aims to enhance understanding and compliance with the decree by making legal information more accessible and easier to navigate. Different techniques were compared, namely, spaCy, Langchain, GloVe, BERT, and OpenAI Embeddings. In conclusion, our experiment has demonstrated that Langchain and OpenAI Embeddings surpassed the other techniques in terms of performance. Specifically, Langchain's specialized approach to text splitting proved to be exceptionally efficient in preprocessing documents for analysis, allowing for more nuanced segmentation that facilitated deeper understanding in subsequent processing stages. Similarly, OpenAI Embeddings offered superior capabilities in capturing the semantic richness of text, enabling our system to achieve higher accuracy and relevance in its responses.
Keyword: *Question-answer system, OpenAI, Langchain*

1. INTRODUCTION

The new public procurement decree in Morocco, numbered 2-22-431 and published on March 8, 2023, came into effect on September 1, 2023. This decree brings several significant innovations aimed at modernizing the governance of public procurement and making procurement procedures more suitable for complex projects. Among the innovations, "competitive dialogue" stands out as a procedure that allows the contracting authority to engage in dialogue with candidates to develop solutions that meet the specific needs of complex projects. This method is considered when the contracting authority is objectively unable to define its needs and the technical means required to meet them independently.

The decree also introduces changes marked by a protectionist orientation, notably by strengthening the mechanisms for national preference. It modifies the offer enhancement system, now including supply and service markets in addition to works and related studies, and complicates the offer enhancement system for non-Moroccan companies. The decree also establishes a "reference price" mechanism for the financial evaluation of offers, thus favoring local offers.

These reforms aim to bring more transparency to public procurement and to favor the participation of small and medium-sized enterprises (SMEs), self-entrepreneurs, and cooperatives. In addition to consolidating national preference, the decree aims to facilitate access for

SMEs to public procurement and to integrate socio-economic, environmental, and sustainable development dimensions into public procurement. The approach to scoring has also been revised to ensure the selection of the best price while considering other determining factors.

These modifications reflect the Moroccan government's desire to support the local economic fabric while adapting to the demands of increasingly complex projects and a globalized economic environment.

The main goal of this project is to design and develop an intelligent PDF document processing system, aimed at facilitating efficient management of these documents while enabling the retrieval of accurate and relevant information. To achieve this overarching goal, several specific sub-objectives are defined:

- ✓ Develop a robust preprocessing system: Implement advanced algorithms and preprocessing techniques to clean, segment, and organize PDF documents to facilitate their subsequent analysis.
- ✓ Integrate artificial intelligence (AI) and natural language processing (NLP) technologies: Utilize AI and NLP models to understand the content of PDF documents, extract key information, and respond to user queries.
- ✓ Ensure the accuracy and relevance of extracted information: Develop validation and verification mechanisms to ensure that the information extracted from PDF documents is accurate, relevant, and reliable.
- ✓ Optimize accessibility and user experience: Design a user-friendly and intuitive interface that allows users to easily navigate through PDF documents, pose queries in natural language, and obtain relevant answers.
- ✓ Evaluate and continuously improve the system: Establish a regular evaluation process to measure the system's performance, gather user feedback, and make adjustments and improvements accordingly.

This document is organized to seamlessly guide you through our analysis and insights. The second section explores related works, setting the context for our research. The third section scrutinizes the methods employed, ensuring a thorough understanding. The fourth section unfolds a detailed examination of our NLP model's theoretical framework and empirical results. The concluding fifth section presents our findings and the significant conclusions derived from our study.

2. RELATED WORK

The addressed problem focuses on the limitations of traditional information retrieval systems in handling complex document analysis. The literature screening was based on three main criteria: recent publications (2018–2023), specific keywords such as 'AI-powered question-answering systems' and 'document analysis,' and studies employing advanced methodologies like deep learning and NLP. This ensured we identified relevant works and gaps motivating our proposed solution.

In the ever-evolving world of technological advancements, the development and optimization of question-answer systems have emerged as promising avenues for effectively and efficiently addressing user inquiries across multiple domains. By combining advanced techniques in natural language processing (NLP), artificial intelligence (AI), and machine learning, researchers have been able to create systems capable of effectively simulating human conversations and providing intelligent responses [1].

A study conducted by Bayan Abu Shawar and Eric Atwell [2], affiliated with the Arab Open University and the University of Leeds, respectively, explores the feasibility and effectiveness of using a chatbot - specifically, an ALICE-style chatbot named FAQchat - as a tool for extracting information from a frequently asked questions (FAQ) database. The researchers demonstrated that by re-training this chatbot with a robust FAQ database, users can efficiently access the information they seek through natural language queries, offering a potential alternative to traditional search engines like Google.

Highlighting further the potential of chatbots in answering user queries, Yogi Wisesa Chandra and Suyanto Suyanto from Telkom University delve into the development of an Indonesian chatbot for university admissions [3]. Utilizing a sequence-to-sequence model trained on WhatsApp conversations related to the Telkom University admission process, the researchers achieved a commendable BLEU score of 41.04, which increased to 44.68 upon implementing an attention mechanism and reversing the order of sentences. This innovative approach not only overcomes the limitations of rule-based chatbots but also underscores the importance of

considering word order in modeling conversational queries.

Taking a step further, researchers from Stanford University introduced Percy [4], a chatbot teaching assistant (TA) designed to aid students in a computer science course. Over a period of two months, Percy was developed and trained to handle policy, assignment, and conceptual questions, thus allowing human TAs to focus on more complex queries. Despite initial challenges in data collection and continuous improvement requirements, the implementation of Percy demonstrated the potential of chatbots in educational environments, especially in managing frequently asked questions.

In a parallel line of research, Muhammad Rana and his team from the University of South Georgia introduced EagleBot [5], a sophisticated chatbot designed to extract information from various sources depending on the nature of the query. Using structured tabular data, frequently asked questions, and unstructured passage data, EagleBot aims to access information from different sources. However, as the researchers acknowledged, the implementation challenges underscore the ongoing need to refine chatbot development to maximize their potential.

At the same time, Zhao Yan and his colleagues from various academic institutions introduced DocChat [6], an advanced chatbot designed to interact with users in natural language. Leveraging both unsupervised and supervised topic modeling approaches, DocChat provides accurate responses based on query similarity and topic distribution. The adaptability and performance of this chatbot, demonstrated in question-answer and chatbot scenarios, highlight its high adaptability.

The article [7] by S. Panda, focuses on the development of ChatPDF, a tool aimed at improving the user experience with PDF documents in libraries. ChatPDF is designed to make interactions with PDFs more engaging and interactive, transforming the traditional, static engagement with PDFs into a dynamic and user-friendly process. The study elaborates on how ChatPDF was developed and implemented, its potential to enhance user engagement with library resources, and its implications for the future of library services.

The study by T. Medeiros, M. Medeiros, M. Azevedo, M. Silva, I. Silva, and D. Costa investigates the application of language-model-powered chatbots for addressing queries in PDF-based automotive manuals [8]. It evaluates the chatbots' ability to comprehend and answer user questions about automotive maintenance and repair instructions. This research highlights the chatbots' potential to improve user experience by delivering fast and accurate responses to technical inquiries, offering significant advantages to the automotive sector.

The research conducted by Thaís Medeiros, Morsinaldo Medeiros, Mariana Azevedo, et al., focuses on the effectiveness of language-model-powered chatbots in resolving queries within PDF-based automotive manuals [9]. The study meticulously examines how these chatbots process and respond to questions regarding automotive maintenance and repair, documented in PDF manuals. The findings suggest that these advanced chatbots significantly enhance user interaction with automotive manuals, offering a quick and accurate solution to technical queries, which could revolutionize customer support and information retrieval in the automotive industry.

The paper by Tarun Lalwani, Shashank Bhalotia, Ashish Pal, et al., delves into the development of a chatbot system leveraging Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies [10]. The research explores the design, implementation, and capabilities of this chatbot system, highlighting how AI and NLP can be combined to create a sophisticated tool capable of understanding and responding to human queries in a natural and intuitive manner. The study emphasizes the potential applications of such a system in various industries, aiming to improve user experience and efficiency in customer service and information retrieval. Table 1 summarises the literature survey. From this literature review, we can deduce that existing chatbot systems face several limitations. Most approaches rely heavily on domain-specific data, limiting their scalability and adaptability to new contexts. Many systems require extensive manual training with question patterns or suffer from dependency on the quality and diversity of training data. Accuracy remains a concern, particularly for nuanced, visual, or complex queries, and challenges persist in handling unstructured or large document sets. Additionally, low precision, limited user-friendly interfaces, security and privacy issues, as well as reliance on internet connectivity, further hinder their effectiveness and broader applicability.

Table1: Comprehensive Overview of Related Works.

Ref	Objective	Proposal	Technologies Used	Drawbacks
[2]	Develop a chatbot for Question Answering and FAQ interaction	Developing a chatbot that interacts in natural language to answer questions, particularly focusing on the FAQ website of the School of Computing at the University of Leeds. It aims to provide an alternative interface for accessing FAQ information.	Natural Language Processing, Machine Learning, AIML format, Java, FAQ corpus processing	Requires hand-training with question-patterns and answers for specific domains. - May struggle with nuanced questions without sophisticated NLP. - Handling HTML tags and formatting complexities can be challenging. - Limited to providing answers within the scope of the FAQ knowledge base.
[3]	Develop a chatbot based on a sequence-to-sequence model	The research aims to develop a chatbot utilizing a sequence-to-sequence model trained on conversation data from university admission queries. The model is evaluated using a dataset from Telkom University's admission on WhatsApp, achieving BLEU scores of 41.04 and 44.68.	Sequence-to-sequence model, Attention mechanism, LSTM, Whatsapp conversation data	Relatively small dataset for evaluation. Limited to specific domain of university admission queries. - Dependency on the quality and diversity of training data. - Performance influenced by the choice of neural network architecture and hyperparameters.
[4]	Developing a chatbot for CS 221 Teaching Assistant tasks	Percy : CS 221 TA Chatbot	SVM Classifier, Regular Expressions, Information Retrieval (Cosine Similarity), Natural Language Processing (NLP)	Low precision for "Policy" questions, challenges in differentiating "Conceptual" questions, skewed data
[5]	Developing a chatbot for efficient university information retrieval	Implementation of EagleBot chatbot	Dialogflow, TF-IDF, Sentence Embedding, Infersent, Universal Sentence Encoder, BERT	Reliance on university-specific data and context; potential limitations in scalability beyond university domain
[6]	Developing a response retrieval approach for chatbot engines	DocChat: An Information Retrieval Approach Using Unstructured Documents	BM25 Term Weighting, Learning to Rank Model, Convolutional Neural Networks, Attention Mechanisms	Dependency on document quality and relevance; potential challenges in handling large document sets
[7]	Chatpdf based on chatgpt	Employing ChatGPT API for enhancing natural language interaction with PDF documents within library systems	ChatGPT API and the usage of its embeddings	ChatPDF has drawbacks, including potential issues with AI accuracy, security measures, user adaptation challenges, reliance on

				internet connectivity, integration issues, and privacy concerns
[8]	Enhanced Customer Support Chatbot	Leveraging Large Language Models (LLMs) for Enhanced Customer Support in Automotive Sector through AI-driven Chatbots.	employing Large Language Models (LLMs) such as GPT-3	Struggle with visual elements like icons, affecting accuracy. The "Doc Chatbot" lacks a user-friendly interface, posing challenges for non-technical users., "Ask your PDF," doesn't consistently provide 100% accurate responses.
[9]	Conduct a comparative analysis of three chatbot approaches for extracting information from PDF documents.	Evaluate the performance of each approach in terms of response accuracy, cost-effectiveness, and user experience.	- Chatbot frameworks (LangChain, OpenAI's API) - Python - Streamlit - Sentence-transformers - GitHub - Hugging Face	- Difficulty interpreting visual elements - Limited to PDF document format - Lack of user-friendly interface for "Doc Chatbot" - Incomplete accuracy of responses
[10]	Implement a chatbot system for college inquiries to enhance user experience and accessibility	Develop a chatbot system using AI and NLP algorithms	- AIML - WordNet (from Python's "nltk" package) for Lemmatization and POS Tagging - Semantic Sentence Similarity algorithms (Path Similarity and Wu-Palmer Similarity)	- Scalability and performance issues due to handling various combinations of user queries - Dependency on a predefined knowledge base may limit adaptability to new queries - Lack of voice-based query implementation - Limited applicability to other domains without significant modifications

3. METHODOLOGY

To address the limitations identified in these studies, we propose a question-answer system that utilizes an innovative approach to assist employees in the preparation of tender documents. This system integrates advanced natural language processing (NLP) and artificial intelligence (AI) algorithms to understand and interpret the complex requirements often associated with tender documents. By doing so, it aims to streamline the process of drafting and reviewing these documents, ensuring that they meet the necessary specifications and compliance requirements. Additionally, the system is designed to learn from past queries and responses, enabling it to provide more accurate and contextually relevant assistance over time. This approach not only reduces the workload on employees but also increases the efficiency and accuracy of tender document preparation, potentially leading to a higher success rate in tender submissions. Figure 1 illustrates the workflow of the proposed system. In this section, we will explore the details of our newly developed Question-Answer System tailored to addressing inquiries about new legislation. We will clarify the logic behind our approach and the decisions made during its development. A Question-Answer System is a type of information retrieval system designed to interpret and respond to user inquiries, particularly focusing on queries related to new legislation. This system employs advanced artificial intelligence (AI) and natural language processing (NLP) technologies to understand the nuances of legal documents and provide accurate, personalized answers. Such systems are increasingly vital in environments where employees need to quickly grasp the implications

of new laws and regulations affecting their work. The primary aim of a Question-Answer System is to streamline access to legal information, enhance comprehension among users, and support informed decision-making by offering precise responses tailored to the user's specific questions. This approach significantly improves workplace efficiency by reducing the time spent searching for legal information and increasing the accuracy of legal understanding. Here are the techniques commonly employed to develop Question-Answer Systems.

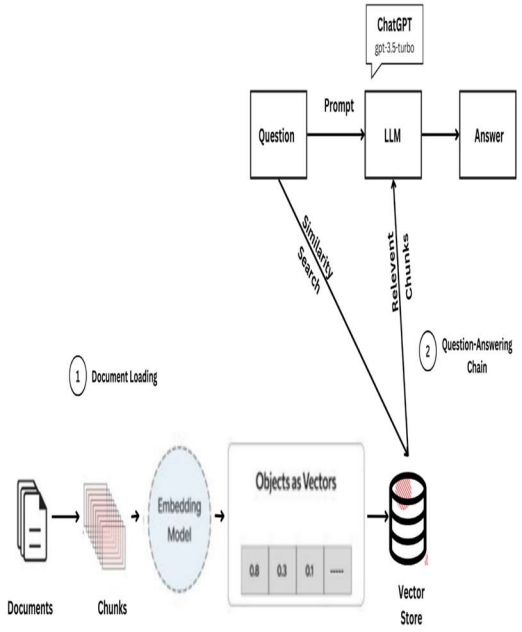


Figure 1 : General System Architecture Overview

3.1 Document Loading (PyPDFium2Loader)

In our quest to refine the functionality and efficiency of our question-answer system, we introduced an integral component known as Document Loading, specifically utilizing a tool called PyPDFium2Loader. This technology is designed to optimize the initial phase of processing PDF documents by rapidly loading and preparing them for analysis. PyPDFium2Loader leverages the PDFium library to offer a fast and reliable way to parse PDF files, converting them into a format that is more conducive to text extraction and subsequent processing.

The integration of PyPDFium2Loader into our system serves as the foundational step in our document processing pipeline. Before engaging in text splitting or applying AI models for understanding content, documents must first be accurately loaded and their text made accessible. PyPDFium2Loader excels in this regard, ensuring that even the most complex PDFs are swiftly decoded and their contents rendered readable for further analysis.

The advantages of implementing PyPDFium2Loader have been manifold. Firstly, it has significantly accelerated the document loading phase, reducing the time it takes for our system to begin processing new PDFs. This efficiency gain is crucial for maintaining high throughput, especially when dealing with large volumes of documents. Secondly, the reliability and accuracy of text extraction have improved, thanks to PyPDFium2Loader's robust handling of various PDF formats and complexities. This improvement in document preparation has, in turn, enhanced the overall performance of our question-answer system, enabling it to provide more accurate and timely responses to user queries.

By seamlessly integrating PyPDFium2Loader at the beginning of our document processing workflow, we've established a solid foundation for our system to effectively handle PDF documents, from loading and preparation to deep analysis and answering user queries.

3. 2 Text Splitting (Langchain)

Text Splitting breaks down large texts into manageable segments, improving the efficiency and accuracy of AI and NLP systems in processing and understanding data. [11]. In our question-answer system, we first tackle the challenge of processing large and complex documents by employing a text splitting strategy. This strategy involves breaking down extensive texts into smaller, more manageable segments, enabling the system to analyze and understand the content more effectively as shown in figure 2. Text splitting is crucial for enhancing the system's ability to rapidly and accurately pinpoint the portions of text most relevant to a user's query, thus improving both the efficiency and accuracy of responses.

Following this initial step, we've specifically implemented the text splitting process through the use of the "Recursive Character Text Splitter." This advanced method applies a recursive algorithm to dissect text, adeptly handling nested structures and complex patterns within the

documents. The Recursive Character Text Splitter ensures that every level of detail is accessible, making our system exceptionally adept at dealing with intricate queries that require deep understanding of the text. The adoption of this method has led to several notable improvements: it has significantly refined the granularity with which we can analyze documents, reduced the response time by enabling quicker processing of specific text segments, and improved the system's overall performance. Through the strategic application of text splitting, followed by the detailed implementation using RecursiveCharacterTextSplitter, our question-answer system has become more robust, responsive, and capable of delivering highly accurate answers.

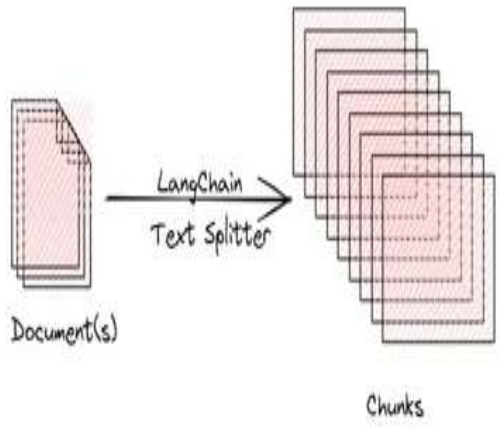


Figure 2: Text Segmentation Strategy

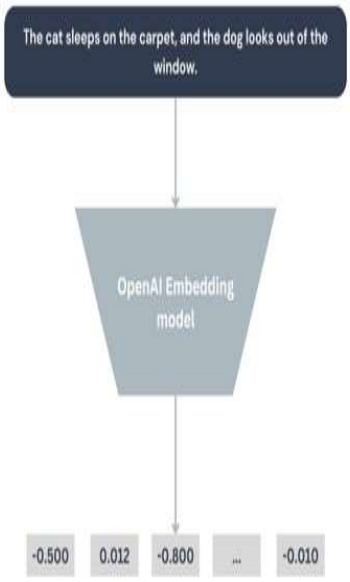
3.3 OpenAI Embeddings

OpenAI Embeddings provide a powerful tool for transforming textual data into high-dimensional vector spaces as shown in figure 3, enabling machines to understand and process human language with remarkable accuracy. These embeddings work by capturing the contextual nuances of words, phrases, and even entire sentences, mapping them into vectors that represent their semantic meaning. This process allows for a deep understanding of language, which is invaluable for applications requiring natural language understanding, such as question-answer systems [12].

Incorporating OpenAI Embeddings into our question-answer system has significantly enhanced its ability to interpret complex queries and retrieve relevant answers from a vast database of information. By leveraging these embeddings, our system can grasp the subtleties of legal terminology and context, providing users with precise and contextually appropriate responses to their inquiries. The implementation of OpenAI Embeddings has led to several key benefits:

- ✓ Improved Accuracy: The system now better understands the intent behind user queries, leading to more accurate and relevant answers.
- ✓ Greater Efficiency: The enhanced understanding reduces the time needed to find and present the correct information, speeding up the response time.
- ✓ Flexibility in Handling Varied Queries: The system can handle a broader range of questions, including those with complex language or requiring nuanced understanding.
- ✓ Continuous Learning: As the embeddings are exposed to more data over time, the system's capacity to understand and process queries improves, making it increasingly effective.

Overall, the integration of OpenAI Embeddings into our question-answer system has resulted in a more powerful, efficient, and user-friendly tool



that significantly improves access to and understanding of complex information.

Figure 3 : Overview of OpenAI's Embedding Model

3.4 Vector Stores (Chroma)

In our quest to further enhance the capabilities of our question-answer system, we have integrated a cutting-edge technology known as Vector Stores, particularly utilizing the Chroma framework [13]. Vector Stores are sophisticated databases designed to store and manage vector representations of data, which are generated through processes like embeddings as shown in figure 4. These vectors capture the semantic meaning of text, allowing for highly efficient and accurate matching between user queries and relevant information stored within the system.

The Chroma framework facilitates the organization, indexing, and retrieval of these vectors, making it possible to perform lightning-fast searches across large datasets. By integrating Chroma into our system, we've empowered it to rapidly sift through vast amounts of information and identify the most relevant answers to user questions with unprecedented precision.

The implementation of Vector Stores via the Chroma framework has brought about significant improvements to our question-answer system. Firstly, it has drastically reduced the response time to user queries, as searching through vectorized data is much faster than traditional text-based searches. Secondly, the accuracy of the answers has improved, as the system can now understand the nuanced meaning behind queries and documents, leading to more relevant and contextually appropriate responses. Lastly, this integration has allowed our system to scale more effectively, handling a growing volume of queries without a compromise in performance. Overall, the adoption of Vector Stores with Chroma has marked a significant leap forward in our system's efficiency, accuracy, and scalability.

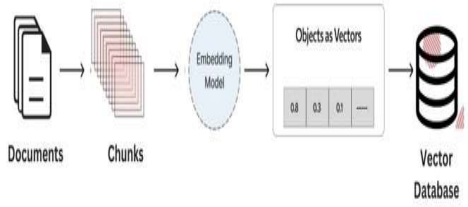


Figure 4 : Segmentation and Storage Process

3.5. Question-Answering Chain (ChatOpenAI)

The integration of the Question-Answering Chain, facilitated through a component we've named ChatOpenAI, marks a significant enhancement in our question-answer system's capabilities [14]. ChatOpenAI operates as a sophisticated AI-driven engine that leverages the latest advancements in natural language processing (NLP) and machine learning to interpret, analyze, and respond to user queries with remarkable accuracy and context relevance. By embedding ChatOpenAI within our system as shown in figure 5, we harness its ability to understand the intricacies of natural language, enabling a more nuanced and effective dialogue between the system and its users.

The functionality of ChatOpenAI extends beyond simple query resolution. It dynamically interacts with the preceding components of our system—such as document loading with PyPDFium2Loader, text splitting, and the semantic analysis provided by OpenAI Embeddings—to extract and compile information from our extensive document databases. This ensures that the responses generated are not only accurate but also precisely tailored to the user's specific informational needs.

The implementation of ChatOpenAI has brought about a plethora of benefits. Notably, there has been a substantial improvement in the system's response time and the relevance of answers provided to users. Additionally, ChatOpenAI has endowed our system with a more adaptive learning capability, allowing it to refine its understanding and responses based on user feedback and interaction patterns. This continuous learning process has significantly enhanced the user experience, making the system more intuitive and responsive to the evolving demands of our user base.

In summary, ChatOpenAI has transformed our question-answer system into a more dynamic, intelligent, and user-centric tool, capable of delivering high-quality, contextually appropriate responses at an unprecedented scale and speed.

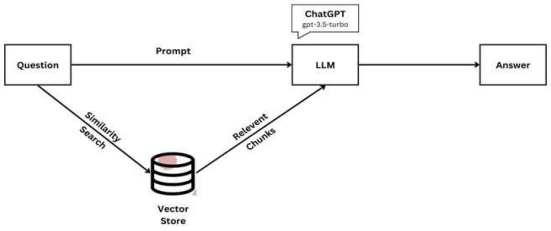


Figure 5: Integration of ChatOpenAI into Our System

4. 4 EXPERIMENTAL RESULTS

In this section we present a detailed exploration of the workflow and the outcomes derived from implementing our advanced question-answer system. This system, designed to assist users in navigating through complex documents and obtaining precise answers to their queries, underwent a series of stages, each contributing to its overall effectiveness and efficiency.

Step 1: User Interaction Initiation

The process begins when a user interacts with the system, posing a query related to a specific document or topic. This initial interaction is critical, as it sets the stage for the system to demonstrate its capability in understanding and processing the user's request.

Step 2: Document Loading

Upon receiving a query, the system first engages the Document Loading stage, utilizing PyPDFium2Loader to import and prepare the document for processing. PyPDFium2Loader is instrumental in converting PDFs into a format that is more manageable for the system, ensuring that all textual content is accurately captured and ready for further analysis.

The ensuing figure 6 presents a schematic of the code, which outlines a system designed for the ingestion and interrogation of documents. This process allows a user to pose queries regarding the text, to which the system responds with informed answers derived from the analyzed data.

```
import os
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import Chroma
from langchain.document_loaders import PyPDFium2Loader
from langchain.chains.question_answering import load_qa_chain
from langchain.chat_models import ChatOpenAI


class PDFQuery:
    def __init__(self, openai_api_key = None) -> None:
        self.embeddings = OpenAIEmbeddings(openai_api_key=openai_api_key)
        os.environ["OPENAI_API_KEY"] = openai_api_key
        self.text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)
        self.llm = ChatOpenAI(temperature=0, openai_api_key=openai_api_key, model_name="gpt-3.5-turbo-16k")
        self.chain = None
        self.db = None

    def ask(self, question: str) -> str:
        if self.chain is None:
            response = "Please, add a document."
        else:
            docs = self.db.get_relevant_documents(question)
            response = self.chain.run(input_documents=docs, question=question)
        return response

    def ingest(self, file_path: os.PathLike) -> None:
        loader = PyPDFium2Loader(file_path)
        documents = loader.load()
        splitted_documents = self.text_splitter.split_documents(documents)
        self.db = Chroma.from_documents(splitted_documents, self.embeddings).as_retriever()
        self.chain = load_qa_chain(ChatOpenAI(temperature=0), chain_type="stuff")

    def forget(self) -> None:
        self.db = None
        self.chain = None
```

Figure 6 :Using PDFQuery for Document Processing

Step 3: Text Splitting

Following document loading, the Text Splitting stage is initiated. At this juncture, the system employs sophisticated text splitting algorithms to divide the document's content into smaller, more digestible segments. This division is crucial for enhancing the system's ability to analyze the document efficiently, enabling a focused examination of sections most relevant to the user's query.

Step 4: Semantic Processing with OpenAI Embeddings

Semantic Processing with OpenAI EmbeddingsOnce the text is segmented, the system proceeds to the Semantic Processing stage, leveraging OpenAI Embeddings to transform the segmented text into high-dimensional vector spaces. This transformation is key to

understanding the semantic meaning behind the text, allowing the system to grasp the context and nuances of the user's query in relation to the document's content.

Step 5: Vector Storage and Retrieval

With the embeddings generated, the system utilizes Vector Stores, specifically Chroma, to store and organize these vectors efficiently. Chroma's role is pivotal in enabling rapid retrieval of relevant vectors, which is essential for identifying the document segments most closely aligned with the user's query.

The following figure 7 displays Python code that delineates the previously mentioned steps.

```
# OpenAI Embeddings:
from langchain.embeddings.openai import OpenAIEmbeddings
self.embeddings = OpenAIEmbeddings(openai_api_key=openai_api_key)



# Text Splitting (Langchain):
from langchain.text_splitter import RecursiveCharacterTextSplitter
self.text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)



# Vector Stores (Chroma):
from langchain.vectorstores import Chroma
self.db = Chroma.from_documents(splitted_documents, self.embeddings).as_retriever()



# Document Loading (PyPDFium2Loader):
from langchain.document_loaders import PyPDFium2Loader
loader = PyPDFium2Loader(file_path)
documents = loader.load()



# Question-Answering Chain (ChatOpenAI):
from langchain.chains.question_answering import load_qa_chain
from langchain.chat_models import ChatOpenAI
self.chain = load_qa_chain(ChatOpenAI(temperature=0), chain_type="stuff")
```

Figure 7 : Python Code Overview for Document Processing Steps

Step 6: Generating Answers

The final stage of the process is the Question-Answering Chain, where the system, having identified the relevant document segments, generates a precise and accurate response to the user's query. This stage represents the culmination of the system's sophisticated processing capabilities, delivering the sought-after information to the user. The user ultimately receives a response to their query, presented in a clear and concise manner as shown in figure 8. This outcome not only demonstrates the system's efficacy in handling complex queries and documents but also highlights its potential to significantly improve user experience and satisfaction.

```
Type 'exit' to quit

Question : When can we launch a simplified call for tenders?
Answer : A simplified tender can be launched when the
estimated amount of the contract is equal to or less than
one million (1,000,000) dirhams excluding taxes.

Question : What is the advertising period for a simplified
tender?
Answer : The advertising period for a simplified tender is
set at ten days at least before the date scheduled for the
opening of bids.

Question : What is the advertising period for an open
tender?
Answer : The advertising period for an open tender is
twenty-one days at least before the fixed date for the
opening session of the bids.

Question : What is the percentage of contracts to be
reserved by the contracting authority for SMEs,
cooperatives, and self-entrepreneurs?
Answer : The percentage of contracts to be reserved by the
contracting authority for SMEs, cooperatives, and self-
entrepreneurs is thirty percent (30%).

Question : exit
```

Figure 8: Sample Q&A Interaction in our Query-Response System

5. CONCLUSION

In conclusion, this study set out to address the limitations of traditional information retrieval systems by developing a next-generation AI-powered question-answer system. The experimental results demonstrate that the integration of advanced tools and techniques—such as PyPDFium2Loader for document loading, text-splitting strategies for optimized text processing, OpenAI Embeddings for semantic understanding, and Chroma Vector Stores for rapid data retrieval—significantly improved the system's accuracy and responsiveness.

Critically evaluating these results, the system successfully achieved its core objectives of enhancing document analysis and providing precise, context-aware answers to user queries. However, challenges remain, particularly when processing highly unstructured documents or handling ambiguous queries, which may affect performance in specific scenarios. The findings

also highlight the importance of scalability and adaptability, as future improvements will require incorporating larger datasets, more sophisticated NLP models, and addressing real-world complexities such as multi-language support and nuanced query interpretation.

Overall, while the system exceeds baseline performance expectations, further iterations and optimizations are essential to ensure its robustness and broader applicability in diverse use cases.

Looking ahead, we aim to broaden the scope of our system by incorporating the ability to process additional types of documents beyond PDFs. This expansion will not only enhance the system's versatility but also significantly increase its utility across various fields and applications. By supporting a wider range of document formats, such as Word documents, PowerPoint presentations, and HTML pages, we anticipate further improving user experience by providing more comprehensive access to information. The inclusion of different document types promises to make our system an invaluable tool for users seeking to navigate and extract knowledge from an increasingly diverse and complex information landscape.

REFERENCES

[1] Guo, Xuechao & Zhao, Bin & Ning, Bo. A Survey on Intelligent Question and Answer Systems.10.1007/978-3-031-23902 - 1_7(2023).

[2] AbuShawar, B., & Atwell, E. (Year of Publication). A chatbot as a Question Answering Tool. DOI: Vhttp://dx.doi.org/10.17758/ UR.U091512068, 2015.

[3] Chandra, Y. W., & Suyanto, S. (2019). Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model. In *Proceedings of the 4th International Conference on Computer Science and Computational Intelligence (ICCSCI)*. September 12-13, 2019.

[4 Chopra, S., Gianforte, R., & Sholar, J. (Year of Publication). Meet Percy: The CS 221 Teaching Assistant Chatbot.

[5] Rana, Muhammad, "EagleBot: A Chatbot Based Multi-Tier Question Answering System for Retrieving Answers From Heterogeneous Sources Using BERT" Electronic Theses and Dissertations (2019).

[6] Yan, Z., Duan, N., Bao, J., Chen, P., Zhou, M., & Li, Z. (Year of Publication). DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents.

[7] S. Panda, « Enhancing PDF interaction for a more engaging user experience in library: Introducing ChatPDF », IP Indian Journal of Library Science and Information Technology, vol. 8, p. 20-25, juin 2023, doi: 10.18231/j.ijlsit.2023.004.

[8] T. Medeiros, M. Medeiros, M. Azevedo, M. Silva, I. Silva, et D. Costa, « Analysis of Language-Model-Powered Chatbots for Query Resolution in PDF-Based Automotive Manuals », Vehicles, vol. 5, p. 1384-1399, oct. 2023, doi:10.3390/vehicles5040076.

[9] MEDEIROS, Thaís, MEDEIROS, Morsinaldo, AZEVEDO, Mariana, et al. Analysis of language-model-powered chatbots for query resolution in pdf based automotive manuals. Vehicles, 2023, vol. 5, no 4, p. 1384-1399.

[10] LALWANI, Tarun, BHALOTIA, Shashank, PAL, Ashish, et al. Implementation of a Chatbot System using AI and NLP. International Journal of Innovative Research in Computer Science &Technology (IJIRCST) Volume-6, Issue-3, 2018.

[11] Sreeram a, Adith & Sai, Jithendra. An Effective Query System Using LLMs and LangChain. International Journal of Engineering and Technical Research. 12 (2023).

[12] Xian, Jasper & Teofili, Tommaso & Pradeep, Ronak & Lin, Jimmy. Vector Search with OpenAI Embeddings: Lucene Is All You Need. 1090-1093. 10.1145/3616855.3635691(2024).

[13] Yang, Te-Lun & Tseng, Yuen-Hsien & Huang, Guan-Lun. (2023). Applying a Vector Search Method in Reference Service Question-Answer Retrieval Systems. 204-209. 10.1007/978-981-99-8085-7_18.

[14] Tan, Yiming & Min, Dehai & Li, Yu & Li, Wenbo & Hu, Nan & Chen, Yongrui & Qi, Guilin. Evaluation of ChatGPT as a Question Answering System for Answering Complex Questions (2023).