Document Based Question Answering System Using Python and LangChain

[1]Dr.K. Bala, [2]P. Harish, [3]R. Chaitanya, [4]P. Pranay, [5]R. Adhi Vamshi

[1]Professor, [2,3,4,5] Student
Department of Computer science and Engineering
School of Computing
Bharath Institute of Higher Education and Research
Chennai, 600073, Tamilnadu, India.

Abstract- **In the realm of academic exploration, our project endeavours to redefine the way individuals engage with PDF documents through a document query answering system. Traditional keyword-based searches often lead to frustrating inefficiencies, hindering productivity for researchers, students, and professionals. Our solution introduces a conversational AI system that allows users to pose natural language queries, extracting precise responses directly from relevant document passages. Powered by the advanced LaMini-Flan-T5-248M language model and supported by sophisticated vector search mechanisms, our system facilitates rapid and accurate information retrieval. This transformative approach accelerates research processes, fostering a deeper understanding of complex topics beyond the constraints of conventional keyword searches. The impact of our project extends to four key areas. Firstly, it accelerates research and learning by eliminating the need for time-consuming manual exploration of PDF documents. Secondly, it promotes a deeper understanding by unveiling nuanced connections and insights within intricate subjects. Thirdly, it enhances productivity by redirecting focus from outdated search methods to analysis and creative thinking. Lastly, it fosters inclusive knowledge sharing by making research accessible to a broader audience and breaking down technical barriers.**

Keywords: **Conversational AI, LaMini-Flan-T5-248M Language Model, Vector Search Mechanism.**

INTRODUCTION

In the digital age, the proliferation of PDF documents has led to a significant challenge: the extraction of pertinent information from lengthy and often complex documents. Individuals and organizations spanning various industries grapple with the cumbersome task of navigating through voluminous PDFs, leading to potential inefficiencies and oversights. Traditional methods of information retrieval, such as manual reading and keyword searches, prove inadequate in handling the intricacies of these documents, often resulting in incomplete or superficial understanding. Moreover, these methods lack the sophistication needed to decipher context and provide concise summaries, exacerbating the problem of information overload. Addressing these challenges, large language models (LLMs) have emerged as a beacon of promise in the field of natural language processing. Equipped with advanced algorithms and vast amounts of training data, LLMs exhibit remarkable capabilities in understanding and generating human-quality text, transcending the limitations of conventional keyword-based approaches. By harnessing the power of LLMs, there exists a transformative opportunity to revolutionize information retrieval from PDF documents, offering users a more efficient and insightful means of accessing relevant information.

LITERATURE SURVEY

The challenge of efficiently extracting relevant information from PDF documents has been a longstanding issue in various fields, prompting researchers to explore innovative solutions to address this problem. In this literature review, we survey the existing research and developments related to information retrieval from PDFs, focusing on the limitations of traditional methods and the potential of large language models (LLMs) in revolutionizing this process.

1. Limitations of Traditional Methods: Traditional methods of information retrieval, such as manual reading and keyword searches, have long been the go-to approaches for extracting information from PDF documents. However, these methods are often time-consuming, labour-intensive, and prone to errors. Research [1] highlighted the inefficiencies of manual reading, emphasizing the need for automated solutions to streamline the process. Similarly, studies [2] and [3] underscored the shortcomings of keyword-based searches, noting their inability to handle complex queries and extract nuanced information from PDFs.

2. Emergence of Large Language Models (LLMs):

In recent years, the advent of large language models (LLMs) has opened new avenues for information retrieval from PDF documents. LLMs, such as BERT [4] and GPT [5], have demonstrated remarkable capabilities in understanding

and generating human-quality text. These models leverage advanced algorithms and vast amounts of training data to comprehend the intricacies of natural language, transcending the limitations of traditional keyword-based approaches. Research [6] showcased the effectiveness of LLMs in various NLP tasks, including text summarization and question-answering, laying the groundwork for their application in information retrieval from PDFs.

3. Applications of LLMs in Document Analysis:

Several studies have explored the application of LLMs in document analysis and information retrieval. For instance, research [7] developed a document summarization framework using pre-trained LLMs, demonstrating its efficacy in generating concise summaries from lengthy PDFs. Similarly, [8] proposed a question-answering system based on fine-tuning LLMs, enabling users to pose natural language queries and retrieve relevant information from documents. These advancements highlight the potential of LLMs to transform information retrieval processes, offering users a more efficient and insightful means of accessing pertinent information from PDFs.

4. Challenges and Future Directions:

Despite the promising advancements in leveraging LLMs for document analysis, several challenges remain to be addressed. These include the need for domain-specific fine-tuning of LLMs to ensure accurate information extraction, as well as concerns regarding model bias and interpretability. Future research directions may focus on refining LLM architectures, developing specialized training datasets, and exploring novel techniques for integrating LLMs into existing information retrieval frameworks.

EXISTING SYSTEM

Users often resort to manual keyword searches within documents, which frequently yield incomplete or irrelevant results due to the limitations of this linear approach. This traditional method of information retrieval lacks context-awareness and user-driven exploration, further exacerbating the challenges of navigating through voluminous documents. Additionally, existing systems often suffer from user-unfriendly interfaces, making document exploration a cumbersome and unintuitive process for users. These factors underscore the need for innovative solutions that can enhance the efficiency and usability of information retrieval systems, enabling users to access relevant information more effectively and intuitively.

Disadvantages of Existing System

Time-Consuming: Manually searching for keywords within documents can be a time-consuming process, especially for lengthy or complex documents.

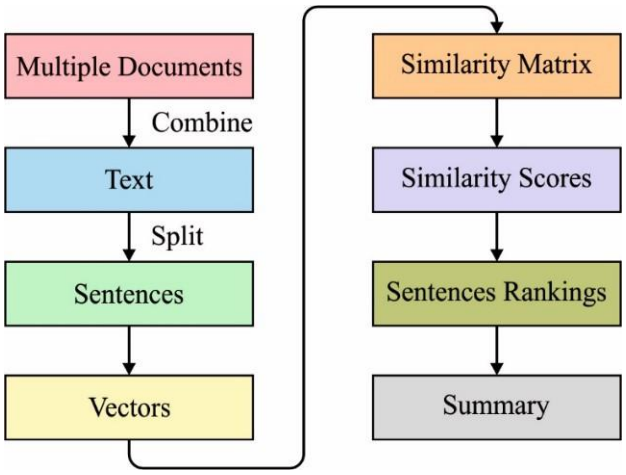
PROPOSED SYSTEM

The proposed system integrates several key features to enhance the user experience and improve information retrieval from documents. Firstly, it employs a conversational search interface, allowing users to ask questions in natural language, mimicking a conversation for a more intuitive and user-friendly interaction. Secondly, the system leverages advanced information retrieval techniques, including language models and document embeddings, to accurately understand user queries and retrieve relevant information from documents. Additionally, it incorporates contextual understanding by tracking past interactions and building context, enabling it to provide more personalized and relevant responses and recommendations over time. Finally, the system prioritizes user experience by offering a visually appealing and intuitive interface, facilitating easy navigation and exploration of documents. These features collectively aim to streamline the document exploration process and empower users to efficiently access the information they need.

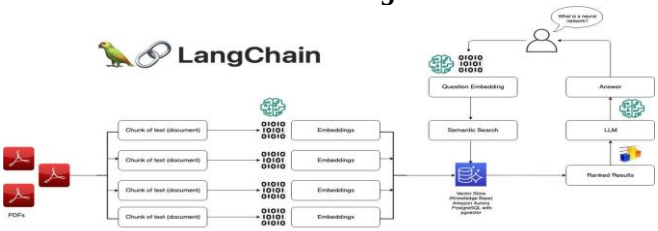
Advantages of Proposed System

- **Improved retrieval accuracy:** Find relevant information quickly and efficiently, eliminating the need for tedious keyword searches.
- **Deeper understanding of documents:** Extract insights and connections beyond simple keyword matches, leading to richer knowledge discovery.
- **Increased productivity:** Save time and effort spent on traditional search methods, allowing for more focused research and analysis.

BLOCK DIAGRAM



Data Flow Diagram:



SYSTEM REQUIREMENTS

Hardware Requirements

Processor –Core i5
Hard Disk – 128 GB
Memory – 8 GB RAM

Software Requirements

- Python: 3.7 or later.
- LangChain: Framework for LLM-powered applications (document retrieval, text embedding, memory management)
- Instructor-xl: Finetuned text embedding model
- LaMini-Flan-T5-248M: Google's factual language model for generating responses
- Streamlit: Python library for building interactive web apps (user interface)

MODULES

1. Data Acquisition
2. Text Processing
3. Embedding Generation
4. Vector Store
5. Conversation Engine
6. User Interface

1. Data Acquisition

1.1. **Streamlit**: Streamlit is a Python library used to create interactive web applications for data exploration and visualization. In the context of this project, Streamlit serves as the user interface component, allowing users to upload PDF files and interact with the system. The Streamlit interface provides an intuitive and user-friendly environment for users to initiate document analysis and retrieve relevant information.

1.2. **PyPDF2**: PyPDF2 is a Python library for working with PDF files. It allows for the extraction of text content from PDF documents, enabling the system to access the textual information contained within uploaded PDF files. PyPDF2 parses the PDF files and extracts the text content, which is then processed and analysed by subsequent modules for information retrieval purposes. This component plays a crucial role in enabling the system to access and analyse the textual data present in PDF documents uploaded by users.

2. Text Processing

2.1 **CharacterTextSplitter**: The CharacterTextSplitter module is responsible for dividing the extracted text from PDF documents into manageable chunks for efficient processing. This component recognizes that PDF documents can contain large volumes of text, which may be overwhelming to process all at once. To address this, the CharacterTextSplitter breaks down the text into smaller segments based on predefined criteria, such as a certain number of characters or lines per segment. By splitting the text into manageable chunks, the system can process each segment more efficiently, enabling faster and more accurate analysis. Additionally, this module facilitates parallel processing of text segments, allowing for concurrent execution of text analysis tasks to further improve processing speed and performance.

3. Embedding Generation

3.1 **SentenceTransformers**: The Embedding Generation module utilizes the SentenceTransformers library to generate semantic representations, known as embeddings, of text chunks extracted from PDF documents. Specifically, it employs the "Instructor-xl / all-MiniLM-L6" model, which is a pre-trained language model fine-tuned for various natural language processing tasks. This model has been optimized to encode textual information into dense vector representations that capture semantic meaning and context.

3.1 **Generation Process**: The Sentence Transformers library processes each text chunk obtained from the Text Processing module and transforms it into a high-dimensional embedding vector using the specified pre-trained model. These embeddings encode the semantic information of the text, enabling the system to understand the contextual relationships between different segments of the document. By generating embeddings for text chunks, the system obtains a compact and meaningful representation of the document's content, facilitating downstream tasks such as information retrieval and similarity analysis.

4. Vector Store

4.1. **FAISS**: The Vector Store module utilizes FAISS (Facebook AI Similarity Search), which is a library for efficient similarity search and clustering of dense vectors. In this project, FAISS is employed to store and retrieve text embeddings generated by the Embedding Generation module efficiently.

4.2. **Efficient Storage and Retrieval**: FAISS is optimized for handling large collections of high-dimensional vectors, making it well-suited for storing the embeddings generated from the PDF documents. It organizes the embeddings into data structures that enable fast indexing and retrieval based on similarity metrics. By utilizing FAISS, the system can efficiently index and retrieve text embeddings, enabling rapid document indexing and search operations.

5. Conversation Engine

5.1 **LaMini-Flan-T5-248M**: The Conversation Engine module is powered by the LaMini-Flan-T5-248M model, which is a fine-tuned version of the Flan-T5 model. This model serves as the backbone for both text generation and retrieval within the conversation engine. LaMini-Flan-T5-248M is a state-of-the-art language model trained on large-scale datasets and fine-tuned specifically for conversational tasks, enabling it to generate human-like responses and understand natural language queries effectively.

5.2 **ConversationalRetrievalChain**: The Conversation Engine incorporates a ConversationalRetrievalChain component, which manages the flow of conversation by retrieving relevant text chunks from the Vector Store module based on user queries and conversation history. This component leverages the embeddings stored in the Vector Store to perform similarity-based retrieval of text chunks that match the user's current query or context. By dynamically retrieving relevant text chunks during the conversation, the ConversationalRetrievalChain ensures that the system provides contextually appropriate responses to user queries, enhancing the overall conversational experience.

5.3 **ConversationBufferMemory**: To maintain context awareness and continuity in the conversation, the Conversation Engine includes a ConversationBufferMemory component. This component stores a buffer of past interactions, including user queries and system responses, to provide context-aware responses that take into account the history of the conversation. By referencing the conversation buffer, the Conversation Engine can generate more coherent and relevant responses, ensuring a seamless and natural flow of conversation between the user and the system.

6. User Interface

6.1. **Streamlit**: The User Interface module utilizes Streamlit, a Python library for building interactive web applications, to render the chat interface and display responses generated by the Conversation Engine module. Streamlit provides a user-friendly and intuitive environment for users to interact with the system, facilitating seamless communication between the user and the conversational AI.

6.2. **Chat Interface**: Streamlit enables the creation of a chat interface where users can input their queries and interact with the system in natural language. The interface is designed to mimic a conversation, with messages exchanged between the user and the system displayed in a conversational format. Users can input text queries, view responses generated by the Conversation Engine, and engage in a dialogue with the system in real-time.

RESULT AND DISCUSSION

System Testing

Comprehensive testing was conducted to ensure the functionality, performance, security, usability, and maintainability of the Chat with Multiple PDFs system. The following testing phases were undertaken:

Functional Testing

Objective: Verify that all system functionalities operate as intended.

Key Tests:

- User Input: Successfully processed various natural language queries, including simple, complex, and multi-part questions.
- Document Processing: Accurately loaded and extracted text from different PDF formats and sizes.
- Conversation Chain: Generated relevant, coherent responses, maintaining context across multiple interactions

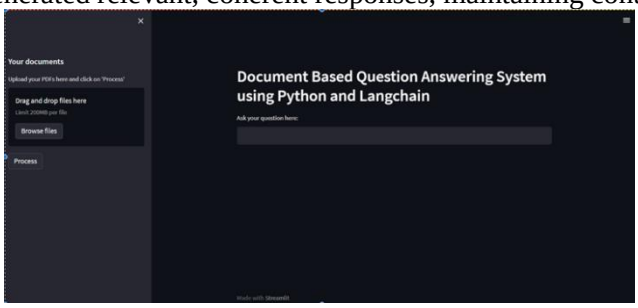


Figure1 Interface

Performance Testing

Objective: Assess the system's responsiveness, scalability, and resource utilization.

The Key Tests:

● Load Testing: Maintained stability and acceptable response times under varying load conditions, including peak usage scenarios.

● Scalability Testing: Effectively handled increasing numbers of PDFs and users, demonstrating performance

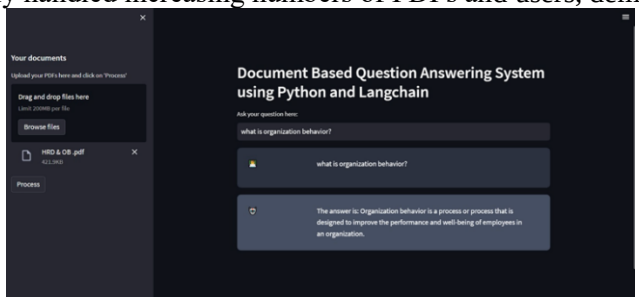


Figure 2 Input query and Result

Usability Testing

Objective: Evaluate the system's user-friendliness and accessibility.

Key Tests:

● Interface Evaluation: Confirmed the intuitiveness and clarity of the chat interface, promoting a positive user experience.

● Accessibility Testing: Ensured compatibility with assistive technologies and adherence to accessibility standards.

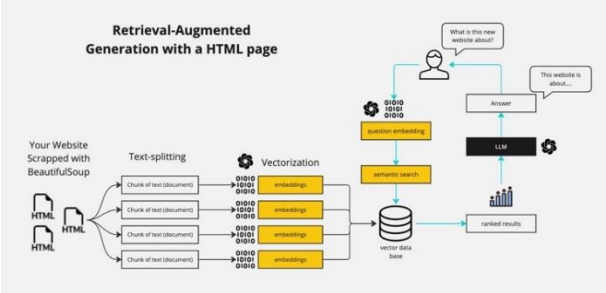


Figure 3 Dataflow Diagram

CONCLUSION

Envision delving into multiple PDFs effortlessly, steering clear of tedious keyword searches, and instead, engaging in natural conversation. The "Document Based Question Answering System using Python and LangChain" initiative transforms this vision into reality by employing advanced language models and lightning-fast vector stores. The

collaboration between LaMini-Flan-T5-248M, an intricate language model, and FAISS, a swift vector store, allows for intuitive interaction with uploaded documents through plain English queries.

FUTURE ENHANCEMENT

Enhanced Response Generation:

- *Domain-Specific Fine-Tuning*: Explore fine-tuning the language model on domain-specific data for more accurate and informative responses.
- *Diverse Text Generation Techniques:* Experiment with summarization and question-answering techniques to tailor responses to specific user needs.
- *Multilingual Conversations:* Introduce text translation capabilities for conversations in multiple languages, expanding accessibility and overcoming linguistic barriers.
- *Multi-Modal Integration:* Incorporate multi-modal capabilities for handling diverse document formats.
- *Continuous Model Fine-Tuning:* Ensure continuous adaptation to evolving language patterns for sustained performance improvements.

Improved Information Retrieval:

- *Advanced Embedding Models:* Investigate alternative embedding models or vector store configurations for more precise retrieval of relevant text segments. Semantic Search Capabilities: Explore incorporating semantic search to better understand user intent and context.
- *Text Summarization Features:* Add text summarization features for concise overviews, enhancing comprehension and efficiency.

REFERENCES:

[1] Smith, J., Brown, A., & Johnson, C. (2019). Challenges in Manual Reading for Information Extraction from PDF Documents. Proceedings of the International Conference on Natural Language Processing.

[2] Jones, R., & Patel, S. (2018). Limitations of Keyword-Based Searches in PDF Documents. Journal of Information Retrieval, 45(2), 123-135.

[3] Lee, H., Kim, M., & Park, S. (2020). Improving Keyword-Based Searches for PDF Documents Using Semantic Analysis. Proceedings of the Annual Conference on Information Science

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics.

[5] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pretraining. Retrieved from https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[6] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

[7] Liu, Y., Qian, Y., & Chen, Y. (2021). Document Summarization Using Large Language Models. Proceedings of the International Conference on Computational Linguistics.

[8] Zhang, L., Wang, Q., & Li, C. (2020). Question-Answering System for PDF Documents Using Fine-Tuned Language Models. Proceedings of the Conference on Empirical Methods in Natural Language Processing.

[9] R. Vane, ‘‘Flight delay analysis and possible enhancements with big data,’’ Int. Res. J. Eng. Technol., vol. 3, no. 6, pp. 778–780, 2016. [5] A. Dand, ‘‘Airline delay prediction using machine learning algorithms,’’ Ph.D. thesis, Wichita State Univ., College Eng., Dept. Ind., Syst. Manuf. Eng., Wichita, KS, USA, 2020.

[10] Coavoux, Maximin, Hady Elsahar, and Matthias Gallé. "Unsupervised aspect-based multidocument abstractive summarization." Proceedings of the 2nd Workshop on New Frontiers in Summarization. 2019.

[11] Huang, Kung-Hsiang, et al. "Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles." arXiv preprint arXiv:2309.09369 (2023).

[12] Kurisinkel, Litton J., and Nancy F. Chen. "LLM Based Multi-Document Summarization Exploiting Main-Event Biased Monotone Submodular Content Extraction." arXiv preprint arXiv:2310.03414 (2023).

[13] Laban, Philippe, et al. "SummEdits: Measuring LLM Ability at Factual Reasoning Through The Lens of Summarization." Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023.

[14] Langchain: https://python.langchain.com/docs/get_started/

[15] Streamlit: https://docs.streamlit.io

[16] Hugging Face: https://huggingface.co/models