

ABSTRACT

This report presents a venture success Large Language Model (LLM) designed to read, extract and process data from Portable Document Format (PDFs), Word Documents (DOCX), and Text (TXT) files utilizing Retrieval-Augmented Generation (RAG) to provide accurate answers to user queries, complete with transparent citations. It also provides a User Interface (UI) with the help of Stream-Lit application.

This project, which aims to improve decision-making processes in complex socio-technical contexts and optimize document management, is an excellent example of the synergy between organizational strategy and technology innovation. The model is a part of The Virtual Organizations as Socio-Technical Systems (VOSS) an intricate sociotechnical system that are interdependent and mutually influencing.

This idea emphasizes the integration of social and technological elements into organizational design and administration by fusing elements of socio-technical systems with virtual organizations.

1. INTRODUCTION

This report's objective is to give a summary of the work completed during the Machine Learning (ML) summer internship at Open Health Systems Laboratory (OHSL), with a particular emphasis on the creation of a large language model.

The main goal of the internship was developing a reliable and user-friendly large language model that could handle Word, Text, and portable document format files. The model can respond to queries depending on the material it has retrieved by reading, extracting, and analyzing data from various document types. To enable smooth interaction with the system, Stream-Lit was also used to create an intuitive user interface.

Large language models and machine learning are important technological developments that provide unmatched powers for data analysis and natural language comprehension. Even with the quick development of new technologies, machine learning and large language models are still essential for improving user experiences because of their advanced language processing powers. [1].

- **About the Company**

An organization dedicated to the public good, the Open Health Systems Laboratory (OHSL) forms, nurtures, and leads project teams to improve research and diagnostic results. In order to give their clients with solutions, they have technical expertise utilizing the most recent developments in information technology, including informatics, communication, natural language processing, and data collection and mining. Greater access to health care services and improved research outcomes for clients have been made possible by OHSL's understanding of the government's funding and contracting procedures for research in the United States, as well as their extensive global network and access to world-class researchers. They are informed about technological advancements and global health systems policies [2].



FIG 2.1 Company Logo

The International Network for Cancer Treatment and Research (INCTR), a nonprofit group dedicated to advancing cancer research and treatment, initiated OHSL as a programme in 2008. Established in 2011 as a stand-alone non-profit, OHSL reincorporated as a public benefit corporation (B-Corp) in 2017. OHSL's foundation is based on the work of its founders with other top research, education, and development institutions, including the World Bank, World Health Organisation, Institute of Medicine, National Cancer Institute, Public Health Foundation of India, Einstein School of Medicine/Yeshiva University, University of Maryland, Pune University, Ohio State University, Ohio State University, Indiana University, Boston University, Stanford University, and Institute of Medicine.

- **About the Project**

VOSS, or virtual organisations as sociotechnical systems, are intricate sociotechnical systems that interact and have an impact on one another. This idea emphasises the integration of social and technological elements into organisational design and administration by fusing elements of socio-technical systems with virtual organisations. A socio-technical system acknowledges the relationship between people, culture, organisational procedures, and technology.

The goal of our project was to create a system that could read, extract, and process data from several types of documents, such as Word documents, Text files, and Portable Document Format. With the usage of an open referencing system, the system concentrated on providing precise answers to queries based on content extraction. In order to do this, we investigated sophisticated large language models and made use of tools like LlamaIndex, Falcon 7B, LangChain, and Retrieval-Augmented Generation. These technological advancements made it possible to integrate and extract data from many document kinds efficiently.

The system also had web scraping features to retrieve pertinent information from specific websites. Data collection was streamlined during this procedure, and it was stored with other document formats for later processing. The system's capacity for thorough data analysis and response creation was improved by this integrated strategy, which made sure the system had access to a variety of current and varied information sources. [1].

- **Roles and responsibilities as a Machine Learning Intern**

1. **Data Collection and Preprocessing:**

To gather and carefully arrange data from different file formats, such as TXT, DOC, and PDF, in order to get it ready for model input. To make sure the data sources are prepared for further processing and analysis, the procedure entails obtaining them from a variety of formats and organising them methodically.

2. **Exploratory Data Analysis (EDA):**

To methodically employ EDA methods in order to obtain profound understanding of the linkages and underlying structure of the data. Through the use of sophisticated data visualisation techniques and tools, experts can find patterns that are essential for further feature selection and model improvement.

3. **Model Development and Implementation:**

To carefully adjust the hyperparameters of several machine learning algorithms in order to test and optimise them thoroughly. The goal of this iterative approach is to optimise performance for particular tasks, improving the model's capacity to generalise and generate precise predictions over a wide range of datasets.

4. **Question Answering System:**

To smoothly include natural language processing techniques intended to improve a question-answering system's accuracy and resilience. This entails modifying algorithms to manage various document formats efficiently, guaranteeing the system's flexibility and dependability in practical applications.

5. Web Scraping and Data Extraction:

To carefully follow moral standards when using online scraping methods to obtain additional information. To preserve integrity during the data collecting stage, this entails automating the data extraction process in an ethical manner while adhering to data usage guidelines and regulatory constraints.

6. Model Evaluation and Validation:

To evaluate the efficacy of the model using exacting performance metrics including accuracy, precision, and recall. By ensuring strong validation across several datasets, cross-validation approaches boost confidence in the generalizability and dependability of the model.

7. Documentation and Reporting:

To keep thorough and well-organized code repositories and documentation throughout the project lifecycle. Transparency, repeatability, and efficient version control are ensured by this procedure, which makes future improvements and collaboration easier.

8. Collaboration and Teamwork:

To encourage interdisciplinary teams to work together effectively in order to advance projects. Combining different viewpoints and areas of knowledge strengthens problem-solving skills and encourages creative approaches to challenging problems in data analysis and machine learning.

9. Continuous Learning and Development:

To participate in seminars and pursue self-directed learning in order to continuously update knowledge and abilities. This continuous learning process guarantees adherence to state-of-the-art developments in machine learning techniques, improving expertise and flexibility in R&D projects.

10. Ethical Considerations and Best Practices:

To respect strict privacy laws and ethical guidelines at every stage of data management and model creation. This commitment ensures ethical integrity and societal responsibility by removing biases, fostering fairness in model outcomes, and protecting sensitive information.

2. LITERATURE REVIEW

The goal of this literature study is to give a thorough analysis of the body of knowledge in the fields of developing Large Language Models and RAG System.

We did a lot of study on RAG, LangChain / LlamaIndex, and big language models like the Falcon 7B model in order to fully understand these new emerging AI technologies and be ready to develop our own model that is fine-tuned according to the needs.

- **Large Language Models**

Large Language Models are highly developed artificial intelligence systems that use vast amounts of data and complex algorithms to understand and produce text that is similar to that of an actual human being. These models use architectures like transformers to process billions of parameters in order to provide text predictions depending on the context. As demonstrated by models such as T5, which are capable of performing a wide range of language tasks such as summarization, translation, and question answering, LLMs exhibit a deep comprehension of intricate linguistic patterns [3]. Their capacity to comprehend and produce cohesive text makes them important in a wide range of fields and uses.

Large language models power chatbots and virtual assistants in customer service, which improve user experience and operational efficiency by providing prompt, accurate answers to consumer requests. Large language models are particularly beneficial to content creation since they speed up the creative process by helping writers and marketers produce high-quality language, like as articles and marketing copy. In the field of education, large language models drive intelligent tutoring systems that provide instantaneous feedback and personalized learning experiences, hence greatly enhancing the learning process [4]. Furthermore, large language models are essential to the healthcare industry because they help with medical documentation and information retrieval, which results in more accurate and efficient medical services.

The ethical issues of data protection, potential biases in generated material, and the proper application of AI must all be addressed in the process of deploying LLMs. Strong controls must be implemented by both developers and users to ensure ethical usage. If LLMs are

applied responsibly and ethically, they have the potential to spur innovation and increase productivity in a variety of industries as they develop further [5].

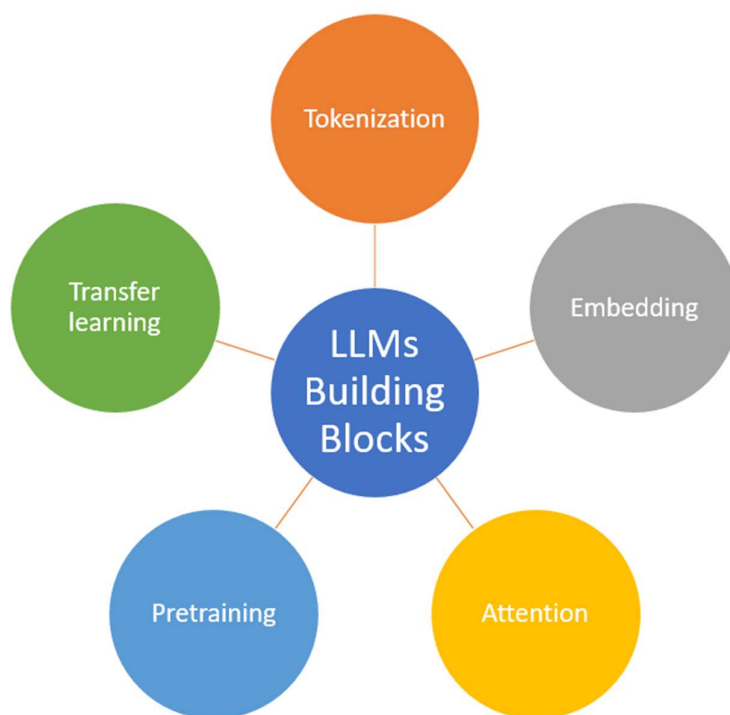


FIG 3.1 Building Blocks of LLM

- **Falcon 7B Model**

Built on the transformer architecture, the Falcon 7B model is an advanced language model with 7 billion parameters that can anticipate words in a sequence and produce text that resembles that of a person. It uses a large number of customizable weights to learn from training data, much like other sophisticated models like GPT-3. This allows it to generate text that is highly accurate and contextually appropriate [6]. Applications for this approach can be found in customer service, content production, education, healthcare, and research, among other fields. For example, it enables chatbots and virtual assistants in customer care to respond quickly and accurately, increasing service efficiency and decreasing human workload. It supports the creation of interactive tools in education that improve learning experiences by generating instructional content and providing individualized coaching [7].

Falcon 7B streamlines processes and guarantees correct medical reporting in the healthcare industry by helping with medical recording, patient engagement, and information retrieval. Its usefulness also extends to research, where it provides insightful analysis from a wealth of scientific literature to support data analysis, hypothesis formulation, and literature reviews [1]. On the other hand, the Falcon 7B deployment also highlights ethical issues like data protection, potential biases in generated content, and responsible AI use. To ensure the moral and responsible use of this potent technology, developers and users must put precautions in place to address these problems [7].

All things considered, Falcon 7B is a huge advancement in natural language processing technology, changing entire sectors with its powerful features. Its adaptability and promise to increase productivity and efficiency are demonstrated by its uses in customer service, education, healthcare, and research. However, in order to maximize its advantages and minimize any negative impacts, it is imperative to address the ethical implications. This will pave the way for future breakthroughs in AI that are more creative and responsible [1], [6], [7], [8].

- **Retrieval-Augmented Generation**

A sophisticated method in natural language processing called retrieval-augmented generation combines the best aspects of generation- and retrieval-based models to improve text production and comprehension. Using a generation model, retrieval-augmented generation generates a response that is dependent on the query and the information it has retrieved. Initially, it retrieves pertinent articles or passages from a vast corpus using a query. This technique leverages external knowledge sources to help the model produce more accurate and contextually appropriate replies; as a result, it is especially helpful for tasks requiring precise information or current knowledge [9].

Retrieval-augmented generation models have demonstrated a great deal of potential in numerous applications. In the area of question answering, for example, retrieval-augmented generation can yield more accurate replies by locating pertinent documents and producing answers that take into account the most important data. Because of its dual ability, retrieval-augmented generation performs better than conventional models that just use pre-encoded knowledge, which can become antiquated or insufficient for particular inquiries [10]. Furthermore, retrieval-augmented generation models are used in customer support systems to draw in pertinent papers or FAQs and provide responses that go deeper into answering consumer questions. As a result, client satisfaction is increased and inquiry processing is handled more quickly.

Retrieval-augmented generation models help in literature review and data analysis in research and development by finding pertinent research papers or datasets and producing summaries or insights from them. When it comes to helping researchers swiftly synthesize material from a wide range of sources, this tool is especially helpful. Retrieval-augmented generation can also be used in content creation, where it gathers pertinent data and creates cohesive narratives from it to assist produce articles or reports. Producing high-quality content with a solid factual foundation is made possible by the combination of retrieval and generation [11].

Retrieval-augmented generation model application, however, also requires tackling issues like guaranteeing the accuracy of information retrieved and minimizing biases in the retrieval and creation processes. To keep the model credible and usable, developers need to put strong processes in place to check the accuracy of the sources and the generated

content. If these ethical and technical issues are sufficiently resolved, Retrieval-augmented generation technology's potential to improve a range of natural language processing applications is expected to increase as it develops [12].

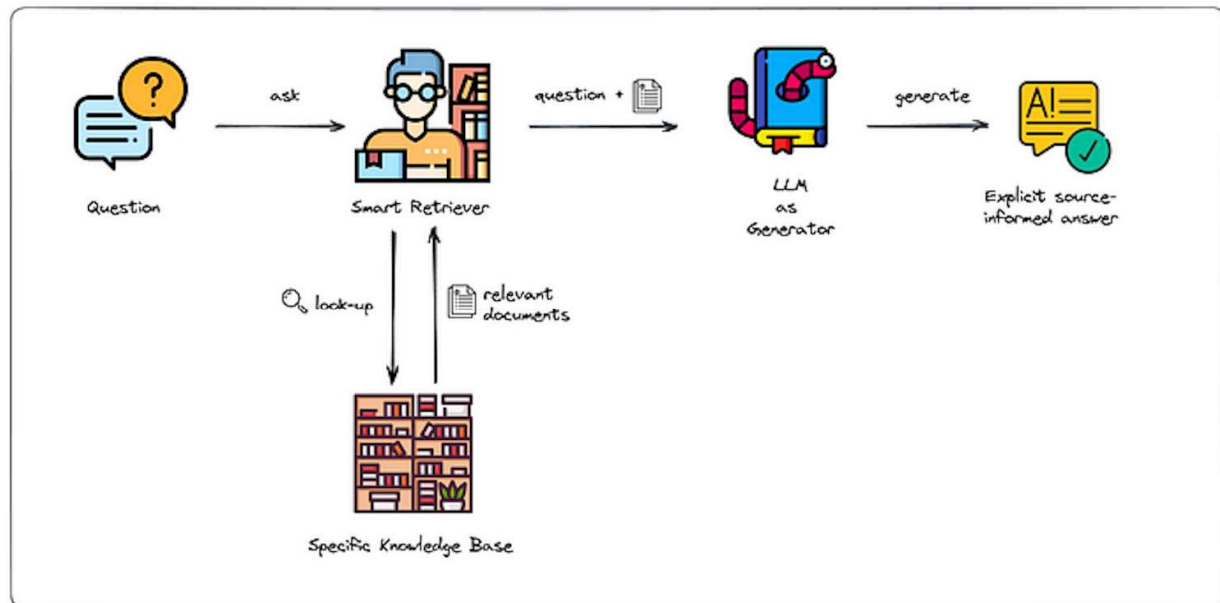


FIG 3.2 Working of RAG Model

RAG Architecture Model

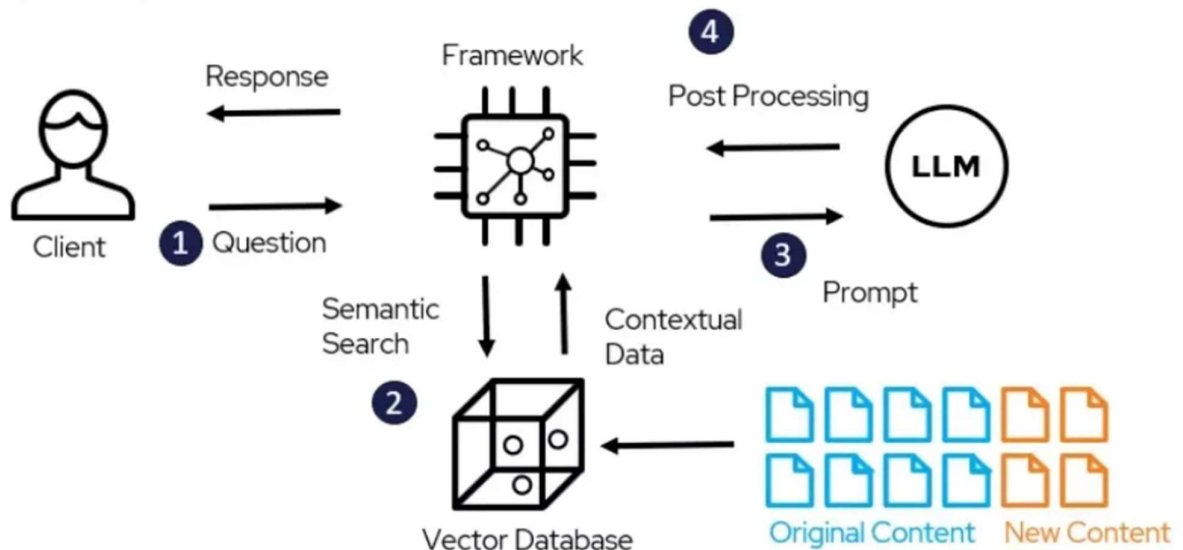


FIG 3.3 RAG Architecture Model

- **LangChain and LlamaIndex**

The novel frameworks LangChain and LlamaIndex are intended to augment language models' capabilities by facilitating more complex and effective information retrieval and text processing.

The LangChain framework facilitates the integration of structured data sources, logic chains, and big language models, enabling developers to create more intricate and dynamic applications. Applications that can carry out complex tasks like multi-step reasoning, data analysis, and dynamic content production can be created thanks to LangChain's tools for connecting LLMs with databases, APIs, and custom logic. The capacity to run specified logic chains and interact fluidly with structured data is one way that this integration improves the versatility and application of LLMs across a range of areas, from financial research to customer support [13].

LlamaIndex (formerly known as GPT-Index) aims to improve the efficiency and accuracy of information retrieval for LLMs. It creates a hybrid index that facilitates quick and accurate document retrieval by fusing conventional indexing methods with contemporary NLP techniques. LlamaIndex retrieves important documents from huge corpora fast, allowing LLMs to construct responses based on this information, which improves LLM performance in tasks like document summarization and open-domain question answering. By utilizing the advantages of both retrieval and generation, this method guarantees that the responses are rich in content and contextually accurate [14].

Providing strong frameworks that expand the capabilities of LLMs, LangChain and LlamaIndex both mark important advances in the realm of natural language processing. They increase the effect of language models across multiple industries by resolving integration and information retrieval issues and opening up more advanced and useful applications.

- **Web Scrapping**

Web scraping is a potent method for obtaining data from websites that enables users to gather and handle vast volumes of data from the internet. Through programmatic access to web pages, content parsing, and data extraction, pertinent information can be obtained for a range of uses, including content aggregation, data analysis, and market research.

Web scraping uses a variety of tools and libraries to make the extraction process automated. With the help of well-known frameworks like BeautifulSoup, Scrapy, and Selenium, users may manipulate dynamic content, traverse webpages, and extract structured data from HTML. For parsing HTML and XML documents, BeautifulSoup is a popular tool that makes navigating the parse tree and extracting data simple. A more complete framework called Scrapy offers capabilities for data processing, data extraction, and storage. Conversely, Selenium is used to automatically render JavaScript-heavy material in web browsers, which is utilized for scraping dynamic websites [15].

Web scraping has a wide range of uses. Web scraping is a tool used by businesses to collect product information, track competitor prices, and do market research. In order to examine trends and obtain information, academic academics scrape data from social media, online journals, and other sources. Web scraping is also used in content aggregation, which is the process of gathering data from several sources and presenting it in a single format on websites like news aggregators and review sites [16].

Web scraping, however, also presents moral and legal issues. Important factors for ethical web scraping are observing terms of service on websites, avoiding flooding servers with queries, and protecting user privacy. To reduce these hazards and safely utilise the capability of web scraping, developers must abide by regulatory requirements and best practices [17].

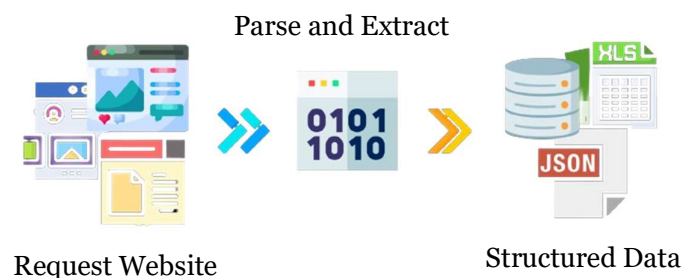


FIG 3.4 Web Scrapping Flow

3. METHODOLOGY

The project aims to improve the Falcon 7B model's ability to process a variety of document types, including PDF, DOCX, and TXT, with a focus on efficiency and accuracy. The Falcon 7B model, which has extensive natural language processing capabilities, was linked with the LangChain and LlamaIndex libraries to provide robust document processing. LangChain enabled seamless connectivity between the model and structured data sources, whereas LlamaIndex optimized information retrieval by combining classical indexing with current NLP techniques [13], [14].

After extracting data from documents, the system analyzed it to guarantee it was searchable and capable of producing contextually correct responses. retrieval-augmented generation techniques, which combine retrieval and generation models, were used to improve text interpretation and answer generation [9]. The system was refined to not only provide responses but also generate citations, assuring transparency and trustworthiness by specifying the source of information within documents [13].

To supplement its knowledge base, the system used web scraping technologies such as BeautifulSoup and Requests, evaluating their performance to optimize data extraction and storage [15]. A Stream-Lit-based user interface was created to make user interaction easier, allowing for smooth document uploads, query submissions, and browsing of chat history. Throughout the project, intense experimentation optimised the system's configurations and methodologies, resulting in optimal performance and usability. The resulting approach dramatically improved the Falcon 7B model's capacity to analyse and interpret texts, delivering precise, context-aware responses swiftly and consistentl

4. PROJECT OVERVIEW

The current project focuses on creating a mobile application that makes it simple to post tests and notes. The goal is to provide a platform that is simple to use, allows users to post their educational resources, makes it easy to access them, and encourages cooperation amongst students, educators, and professionals. This section gives a summary of the project's goals, intended users, and salient characteristics while highlighting its contribution to the field of education.

I. Workflow of the proposed Model

The workflow for constructing the enhanced Falcon 7B model system was comprehensive and varied, with steps ranging from document processing to user interaction. The procedure is described in detail below, with specific phases broken down.

1. Document Processing

The initial phase focused on using the Falcon 7B model to read and extract data from several document formats, including PDF, DOCX, and TXT. This stage had numerous sub-processes:

1.1. Document Loaders

Custom loaders were created for each document format to make the reading process easier. PDFs were created using libraries such as PyMuPDF and PyPDF2. DOCX files were processed with the python-docx module, whereas TXT files were handled using conventional Python file operations [13].

1.2. Data Extraction

The Falcon 7B model was used to extract text from the papers. The paradigm, when coupled with LangChain, enabled efficient text processing while preserving the document's structural integrity. LangChain's capacity to connect LLMs to structured data sources was critical here, ensuring that the retrieved data was clean and well-organized [13].

1.3. Data Indexing

To improve retrieval efficiency LangChain was used to index the extracted data. LangChain combines traditional indexing methods with new natural language processing methodologies to create a hybrid index that enables rapid and exact document retrieval. This stage guaranteed

that the data was not only available, but also formatted in a way that allowed for rapid and accurate retrieval during the query process [13].

2. Data Processing

Once the data had been retrieved and indexed, it needed to be processed so that it could be searched and answered. This phase involved:

2.1. Structuring Data

The retrieved data was organized in a searchable fashion. This entailed organizing the data into a database or data frame that could be accessed efficiently. Techniques from information retrieval and database management were used to ensure that the data was optimally formatted for search operations.

2.2. Implementing RAG

Retrieval-Augmented Generation approaches were used to improve question-answering performance. Retrieval-augmented generation works by first retrieving relevant documents or passages from the indexed material using a query. The Falcon 7B model then created replies based on both the question and the information it had retrieved. This dual strategy ensured that the answers were contextually correct and appropriate [9].

3. Question Answering and Citations

The following phase concentrated on improving the question-and-answer mechanism and ensuring that responses were accompanied by citations.

3.1. Query Processing

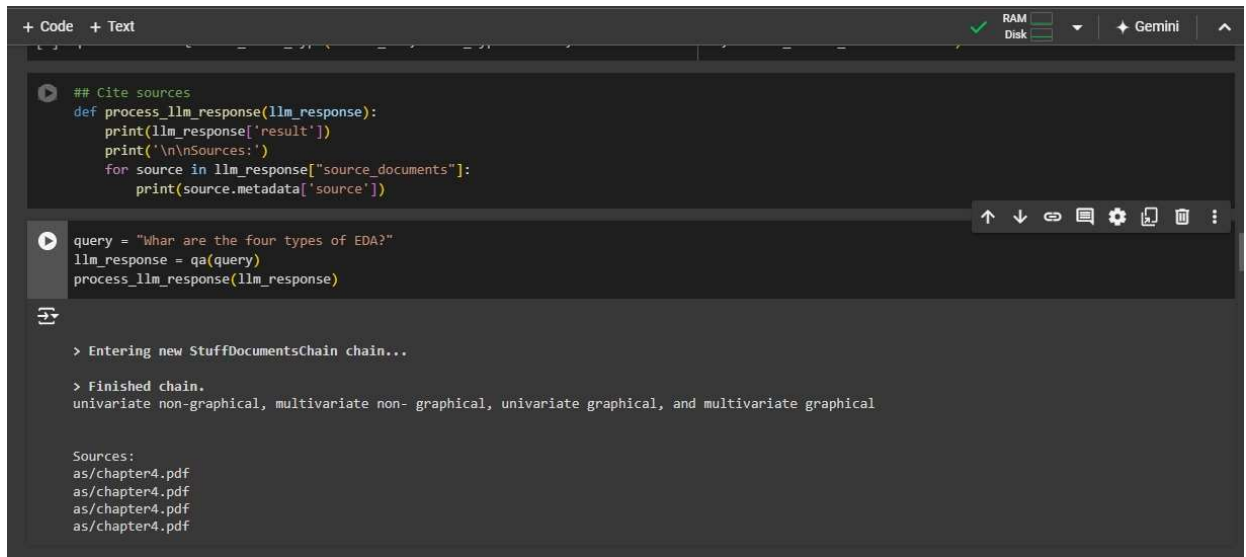
User queries were analyzed to extract the most relevant information from the indexed data. The retrieval system utilized powerful search algorithms to determine the precise passages or documents that provided the answer to the inquiry.

3.2. Answer Generation

The Falcon 7B model created responses depending on the information retrieved. The model was refined to produce precise, context-aware replies. It was also trained to provide citations, which indicate the source of the material. This aspect was critical to maintaining the trustworthiness and traceability of the responses [13].

3.3. Citation Mechanism

A rigorous citation process was put in place to trace the source of each answer. This entailed identifying each piece of received information with metadata indicating its origin. The system then incorporated this metadata into the generated responses, allowing users to readily trace the answers back to their originating papers.



```
+ Code + Text
RAM
Disk
Gemini

## Cite sources
def process_llm_response(llm_response):
    print(llm_response['result'])
    print('\n\nSources:')
    for source in llm_response["source_documents"]:
        print(source.metadata['source'])

query = "What are the four types of EDA?"
llm_response = qa(query)
process_llm_response(llm_response)

> Entering new StuffDocumentsChain chain...

> Finished chain.
univariate non-graphical, multivariate non- graphical, univariate graphical, and multivariate graphical

Sources:
as/chapter4.pdf
as/chapter4.pdf
as/chapter4.pdf
as/chapter4.pdf
```

FIG 4.1 Prompt and Citation Code Output

The four types of EDA are univariate non-graphical, multivariate non-graphical, univariate graphical, and multivariate graphical.

FIG 4.2 Data Snippet from the Fetched File

4. **Web Scraping**

Web scraping was used to add essential information and keep the dataset up to date. The web scraping implementation included:

4.1. Performance Comparisons

Various web scraping tools were investigated to find the best fit for the project's requirements. The performance of each tool was evaluated using factors such as speed, accuracy, and convenience of use.

4.2. Using Beautiful Soup and Requests

Beautiful Soup and Requests were chosen for their efficiency in data extraction and retrieval. Beautiful Soup's extensive parsing capabilities, along with Requests' ease of handling HTTP requests, made it suitable for web scraping activities [15].

4.3. Streamlining Data Collection

The data gathering procedure was simplified to guarantee that the extracted data was stored consistently for further processing. This entailed creating an automated program that scraped and updated the dataset on a regular basis with the most recent information.

4.4. URL Conversion and Data Storage

The data retrieved from various websites was subsequently saved into text documents for further processing by the Falcon 7B Model. This ensures that when users query the Large Language Model (LLM), they can access the information efficiently and effectively. To avoid fetching data from the same websites repeatedly, we implemented a URL converter function. This function uniquely identifies each website, ensuring that the extracted data is saved without duplication.

```

def scrape_text_from_url(url):
    response = requests.get(url)
    if response.status_code != 200:
        raise Exception(f"Failed to fetch the URL: {url}")

    soup = BeautifulSoup(response.text, 'html.parser')
    paragraphs = soup.find_all('p')
    text = "\n".join([para.get_text() for para in paragraphs])
    return text

# Converts a URL to a valid document name
def url_to_document_name(url):
    parsed_url = urllib.parse.urlparse(url)
    filename = parsed_url.netloc + parsed_url.path
    filename = filename.replace('/', '_')
    filename = filename.replace('.', '_')
    return filename

```

FIG 4.3 Web Scrapping Code

url.txt X

FIG 4.4 Web Scrapping Code Output

```

1
2
3 The Indian Independence Movement was a series of historic events in South Asia with the ultimate aim of ending British rule in India.
4
5 The first nationalistic movement for Indian independence emerged in the Province of Bengal. It later took root in the rest of India.
6
7 The stages of the independence struggle in the 1920s were characterised by the leadership of Mahatma Gandhi and Congress.
8
9 Few leaders followed a more violent approach, which became especially popular after the Rowlatt Act, which permitted the British to arrest
10
11 The Indian independence movement was in constant ideological evolution. Essentially anti-colonial, it was supplemented by Hindu nationalism.
12
13 India remained a Crown Dominion until 26 January 1950, when the Constitution of India established the Republic of India.
14
15 The first European to reach India via the Atlantic Ocean was the Portuguese explorer Vasco da Gama, who reached Calicut in 1498.
16
17 Over the next two centuries, the British[note 1] defeated the Portuguese and Dutch but remained in conflict with the Marathas and the
18
19
20 After the defeat of Tipu Sultan, most of southern India came either under the company's direct rule, or under its influence.
21 Maveeran Alagumuthu Kone was an early rebel against the British presence in Tamil Nadu. He became a military leader.
22
23 In Eastern India and across the country, Indigenous communities organised numerous rebellions against the British administration.
24
25 The Santhal Hul was a movement of over 60,000 Santhals that happened from 1855 to 1857 (but started as early as 1784).

```

5. **Building the UI**

The final phase involved developing a user-friendly interface using Stream-Lit.

5.1. Stream-Lit Integration

Stream-Lit was chosen because it is simple and effective for developing interactive web apps. The user interface allowed users to easily upload files, ask questions, and examine their chat history. The process for the Stream-Lit application contained:

1. File Upload Feature: Users can easily upload their PDFs, DOCX, and TXT files.
2. Process Button: Once the documents are uploaded, users press the "Process" button to initiate backend processing.
3. Chat Button: After processing is complete, users press the "Chat" button to start querying the model.
4. Query Interface: Users input their questions, and the system retrieves and displays the required information from the uploaded documents.

5.2. User Interaction

The user interface was created to be intuitive, allowing users to easily move through the document upload, processing, and querying stages. The chat history function allowed users to keep track of their interactions with the system, offering a detailed overview of their queries and responses.

The process of working on the Falcon 7B model had several stages, each with its own focus and set of duties. From document processing and data extraction to question answering and user interface creation, each aspect was methodically carried out to assure the system's efficiency and efficacy. The combination of advanced NLP algorithms, sophisticated indexing methods, web scraping, and a user-friendly interface produced a powerful tool that gives detailed, context-aware responses with citations, resulting in a smooth and efficient user experience.

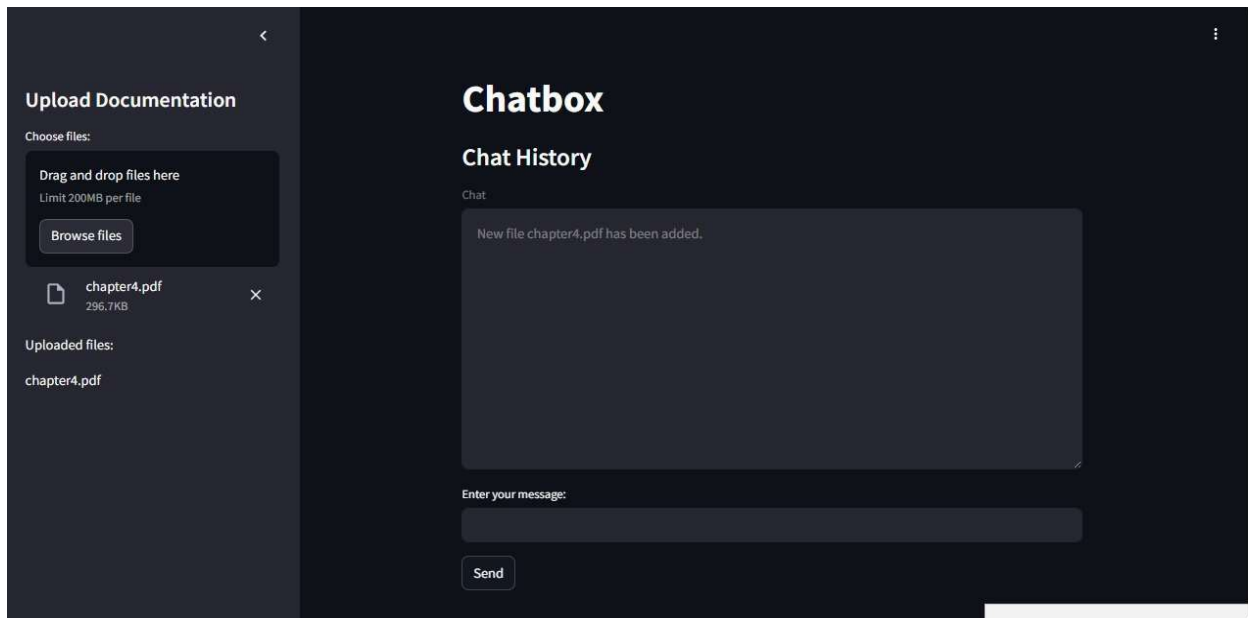


FIG 4.5 UI Interface Sample Preview

II. Result

The developed system is capable of reading, extracting, and processing data from various document formats such as PDFs, DOCs, and TXT files. This capability ensures that information may be efficiently retrieved and utilized across multiple document forms, hence improving data accessibility and usability in a variety of applications [1].

The system's processing pipeline employs rigorous data pretreatment techniques to ensure that the extracted data is correct and ready for analysis. Data cleaning, transformation, and augmentation techniques were used to increase data quality and the system's ability to deliver accurate responses to user queries [18].

A critical feature of the system is its capacity to deliver accurate replies to user questions based on processed data. The system uses advanced natural language processing techniques, such as retrieval-augmented generation, to ensure that responses are not only precise but also contextually appropriate. This strategy improves user interaction by providing precise, context-aware responses straight from documents, with transparent referencing to show the source of information.

The Stream-Lit-based user interface improves the system's usability by offering a fluid and intuitive platform for users to upload files, submit queries, and interact with the system with ease. This UI design simplifies the user experience, allowing for quick access to information and effective navigation through the system's features.

The Virtual Organizations as Socio-Technical Systems project is a framework for integrating social and technology components to create cohesive systems capable of meeting complex organizational needs. By recognizing the interconnection of these factors, VOSS hopes to optimize organizational processes through the effective use of technologies such as the Falcon 7B model and LangChain. These technologies were critical in improving the system's analytical capabilities, providing deeper insights into document content, and facilitating informed decision-making processes.

The experimental phase of the project included rigorous testing and development of these technologies, with a focus on improving the Falcon 7B model's performance in analyzing and understanding PDFs, Word documents, and text files. This iterative procedure improved the system's ability to handle a wide range of document types efficiently and accurately, providing the groundwork for scalable and flexible document management and analysis systems [15].

The VOSS project represents the intersection of technology innovation and organizational planning, with the goal of improving document processing and information retrieval capabilities. By harnessing

cutting-edge AI technologies and integrating them into a socio-technical framework, the project not only improves operational efficiency but also fosters a seamless user experience via intuitive interface design and transparent data handling procedures.

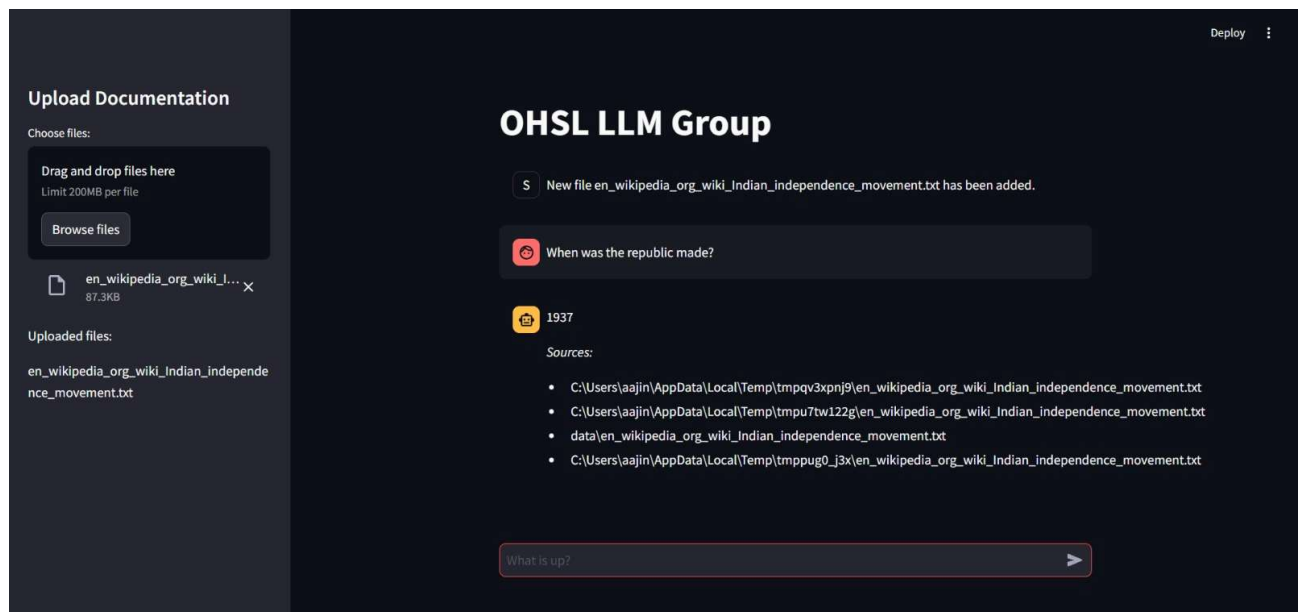


FIG 4 UI Interface

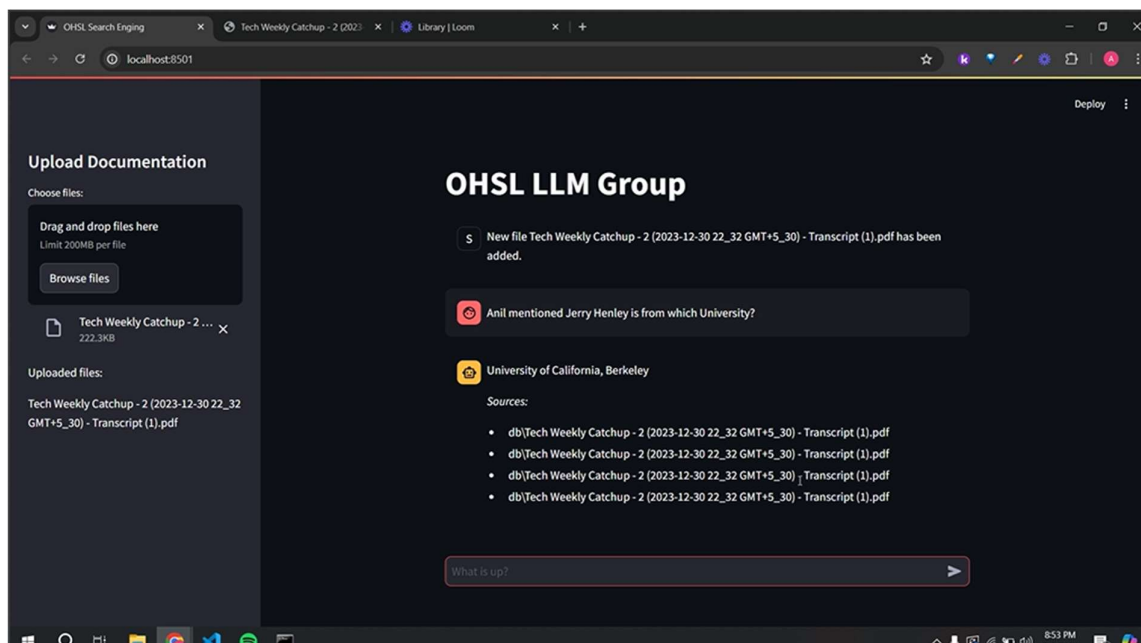


FIG 4 UI Interface In Production

5. FUTURE WORK

1. Enhancements:

Moving forward, there are several key areas where the system can be enhanced to further improve its functionality and performance. Firstly, continuous efforts will focus on enhancing the accuracy and speed of both data extraction and question-answering processes. This will involve employing advanced optimization techniques to fine-tune the system's algorithms and parameters, thereby achieving quicker and more precise data handling capabilities [18].

2. Feature Expansion:

In terms of feature expansion, the system aims to broaden its capabilities to support additional document types beyond PDFs, DOCs, and TXT files. This expansion will enable the system to accommodate a wider range of inputs, enhancing its versatility and applicability in various domains [1]. Furthermore, improvements to the user interface are planned to enrich the user experience by adding more intuitive functionalities and enhancing accessibility. These enhancements will facilitate easier navigation, seamless interaction, and improved user satisfaction. Moreover, the integration of more advanced natural language processing techniques is crucial for enhancing the quality and depth of the system's responses. Techniques such as semantic understanding, sentiment analysis, and entity recognition will be explored to enrich the contextual understanding and accuracy of the answers provided.

3. Scalability:

Ensuring scalability is another pivotal aspect of future developments. Efforts will concentrate on optimizing the system to handle larger datasets effectively, maintaining robust performance even with substantial increases in data volume. Scalability will also encompass supporting multiple simultaneous users, ensuring high efficiency and responsiveness under increased load conditions. To achieve this, scalable architecture solutions will be implemented to accommodate future growth and usage spikes. This includes leveraging cloud computing resources and distributed computing frameworks to ensure consistent performance and reliability as demand for the system grows [15].

6. CONCLUSION

The Virtual Organizations as Socio-Technical Systems project has successfully used advanced AI technologies such as the Falcon 7B model, LangChain, and LlamaIndex to create a robust system capable of processing a wide range of document formats—PDFs, DOCX, and TXT files—with accuracy and efficiency. The system uses Retrieval-Augmented Generation techniques to not only retrieve and analyses data, but also generate contextually accurate replies, which are supported by transparent citations that trace material back to its original source. The Stream-Lit-based user interface improves usability by offering a unified platform for users to engage with the system intuitively. This project highlights the interaction between technological innovation and organizational strategy, with the goal of optimizing document management and improving decision-making processes in complex sociotechnical environments.

REFERENCES

- [1] Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” [Online]. Available: <https://github.com/tensorflow/tensor2tensor>
- [2] Open Health Systems Laboratory, "Open Health Systems Laboratory,". Available: <https://ohsl.us/>. [Accessed: July 1, 2024].
- [3] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” 2020. [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>.
- [4] M. Lewis *et al.*, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” Oct. 2019, [Online]. Available: <http://arxiv.org/abs/1910.13461>
- [5] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” Jun. 2019, [Online]. Available: <http://arxiv.org/abs/1906.08237>
- [6] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017, [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [7] T. B. Brown *et al.*, “Language Models are Few-Shot Learners,” May 2020, [Online]. Available: <http://arxiv.org/abs/2005.14165>
- [8] A. R. Openai, K. N. Openai, T. S. Openai, and I. S. Openai, “Improving Language Understanding by Generative Pre-Training.” [Online]. Available: <https://gluebenchmark.com/leaderboard>
- [9] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” May 2020, [Online]. Available: <http://arxiv.org/abs/2005.11401>
- [10] G. Izacard and E. Grave, “Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering,” Jul. 2020, [Online]. Available: <http://arxiv.org/abs/2007.01282>
- [11] V. Karpukhin *et al.*, “Dense Passage Retrieval for Open-Domain Question Answering,” Apr. 2020, [Online]. Available: <http://arxiv.org/abs/2004.04906>
- [12] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, “REALM: Retrieval-Augmented Language Model Pre-Training,” Feb. 2020, [Online]. Available: <http://arxiv.org/abs/2002.08909>
- [13] Z. Tan *et al.*, “Large Language Models for Data Annotation: A Survey,” Feb. 2024, [Online]. Available: <http://arxiv.org/abs/2402.13446>
- [14] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey,” Dec. 2023, [Online]. Available: <http://arxiv.org/abs/2312.10997>
- [15] R. (Ryan E.) Mitchell, *Web scraping with Python : collecting more data from the modern web*.

