

# Enhancing Q&A Systems with Multilingual Text Conversion and Speech Integration: Harnessing the Power of LangChain and Large Language Models

Priya K<sup>1</sup>, Akshatha Kamath<sup>1</sup>, Chandan K M<sup>1</sup>, Praveen C<sup>1</sup>, Omkar S N<sup>1</sup>, Aaditya S J<sup>1</sup>

<sup>1</sup>Dept of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, India.

Emails: [priyak@msrit.edu](mailto:priyak@msrit.edu), [akshathakamath@msrit.edu](mailto:akshathakamath@msrit.edu)

## Abstract

Searching through URLs and PDFs can be tedious and time-consuming due to the unstructured nature of these documents and the challenge of finding accurate, relevant information. LangChain addresses these issues by using advanced natural language processing algorithms to extract pertinent data from URLs and PDFs. With its user-friendly search interface, customizable filters, and efficient indexing and retrieval mechanisms, LangChain significantly enhances the search experience. Users can annotate important sections, store queries, and create bookmarks, making information retrieval from URLs and PDFs more efficient and improving overall productivity. Traditional text analysis systems often struggle with interactivity, flexibility, and data integration, making it difficult for users to gain meaningful insights from diverse data sources such as websites and PDFs. Our research combines state-of-the-art technologies, including Dash, LangChain, Google Generative AI, and FAISS, to provide a comprehensive solution for extracting, analyzing, and interacting with textual data from various sources. This includes handling both PDFs and data uploaded via URLs. Our research demonstrates significant improvements in the efficiency and accuracy of information retrieval, paving the way for more complex applications such as text summarization and question-answering. Our system is also capable enough to convert the text into speech and to translate text to 10 different languages.

**Keywords—LangChain, ChatGPT, OpenAI, Deep Learning, Google Generative AI, Vector Embeddings, FAISS, Semantic Search.**

## I. INTRODUCTION

In 2022, there was a notable surge in the use of generative AI algorithms. One of the most talked-about innovations was ChatGPT—short for Chat Generative Pre-Trained Transformers—introduced by OpenAI. This advanced AI system, powered by deep learning technology, amazed the world with its ability to produce human-like text and engage in conversations. But beyond chatting, have you ever thought about using such AI to query computer

documents instead of manually searching through them? Thanks to advancements in conversational and generative AI technologies, this is now possible. The infamous chatbot, ChatGPT, showcased by OpenAI, stands for Chat Generative Pre-Trained Transformers. This sophisticated AI system, driven by deep learning, has astonished the world with its impressive ability to generate text that mimics human writing and converse naturally with people. As digital documents, especially PDFs, become more prevalent, retrieving information from them has become increasingly challenging. However, LangChain, a revolutionary tool built on Natural Language Processing (NLP) and Large Language Models (LLMs), tackles these challenges head-on. LangChain simplifies the search and information extraction process from URLs and PDFs using advanced NLP algorithms. LangChain employs Dash, a user-friendly web application framework that doesn't require extensive knowledge of web development technologies like CSS and HTML, to create an intuitive interface. With Dash, deploying models becomes seamless and requires minimal coding effort. This integration allows users to effortlessly interact with URLs and PDFs, significantly enhancing the ease of document search and retrieval as in figure 1.

LangChain Q&A Using Large Language Models

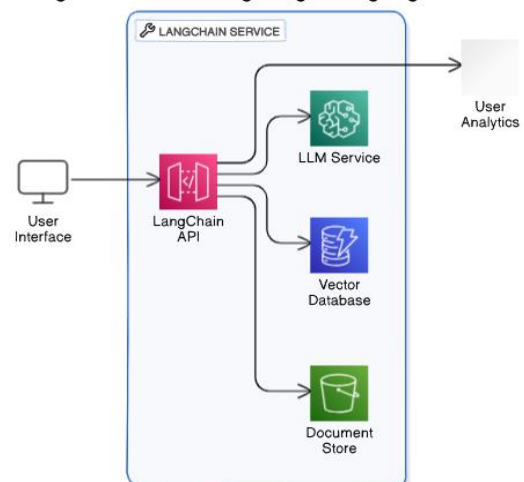


Fig.1: LLM Model for Q and A

## II. RELATEDWORKS

This study on “Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast” discusses the use of LangChain, an open-source library, for developing applications utilizing large language models (LLMs). It outlines the main components of LangChain, including Prompts, Memory, Chains, and Agents, and how they can be utilized in various use cases such as autonomous agents, chatbots, and code understanding. Featured works include the integration of LangChain with diverse data sources and its modular structure that facilitates the rapid development of custom AI applications. LangChain's ability to streamline the development process and its potential to spur further exploration in the field of LLMs are also highlighted.[1].

The study on “Agent for Recommending Information Relevant to Web-based Discussion by Generating Query Terms using GPT-3” addresses the exploration of an AI agent designed to enhance web-based discussions by generating and recommending relevant information using GPT-3. The agent intervenes in discussions, recommending information every 3 minutes, which helps participants understand the discussion better and stimulates more contributions. The experiment conducted with ten students showed that the agent's recommendations increased the overall discussion contributions by 46.2%. Key components include the use of GPT-3 for query generation and evaluation through participant questionnaires[2]. Welcome to the Era of ChatGPT et al. In this research the use of large language models (LLMs) and chatbots in radiology, highlighting their applications in literature reviews, data extraction, and preliminary data analysis. It emphasizes the potential of LLMs to enhance research efficiency by performing summarization, extraction, and classification tasks. Featured works include domain-specific models like RadBERT and Radiology-GPT, which are fine-tuned for radiology-specific tasks, and techniques such as prompt engineering, retrieval-augmented generation, and reinforcement learning from human feedback to improve model performance and reduce hallucinations. The paper also discusses data privacy concerns and mitigation strategies in healthcare applications [3].

ChatGPT and Large Language Models in Academia: Opportunities and Challenges research addresses the emergence of Large Language Models (LLMs) like ChatGPT, Bing Chat, and Google's Bard, highlighting their significant impact on various domains including business, education, and creative fields. It emphasizes the efficiency of LLMs in generating coherent text, thus shifting the focus to the content and ideas rather than the

mechanics of writing. The paper also addresses ethical considerations, potential for misuse, and the evolving role of prompt engineering. Featured works include creative applications, legal considerations of data use, and the development of domain-specific models like BioGPT. Further advancements are expected with better integration of real-time data and improved model architectures [4]. This research “A Complete Survey on LLM-based AI Chatbots” addresses the advent of Large Language Models (LLMs) and their potential impact on academic work, including writing, education, and programming. The authors highlight the ethical considerations and biases inherent in these models, while advocating for their effective use. The paper emphasizes the utility of LLMs in improving efficiency and the importance of addressing their limitations, such as accuracy issues and potential for plagiarism. Key methods discussed include iterative chat capabilities and the application of AI in various academic contexts [5]. A Comprehensive Comparative Analysis of Three Chatbot Approaches for Answering PDF-Based Questions study addresses the comprehensive survey of LLM-based AI chatbots, detailing their evolution, applications, and challenges. It examines chatbots like ChatGPT, BARD, Bing Chat, and Claude, highlighting their significant advancements over early conversational AI, which faced issues like limited contextual understanding and domain specificity. The paper categorizes applications across sectors such as education, research, and healthcare, and discusses challenges like scalability and ethical concerns. Featured works include a detailed taxonomy of applications and strategies to enhance ChatBot performance and reliability [6]. “What Can ChatGPT Do?” Analyzing Early Reactions to the Innovative AI ChatBot on Twitter this study addresses the evaluation of three AI-based ChatBot solutions—Doc ChatBot, Ask your PDF, and a Question and Answer System—using the OpenAI API. It focuses on cost-effectiveness, user experience, and response accuracy for interacting with vehicular manuals. The analysis finds "Ask your PDF" to be the most optimal solution, balancing user interface and precise responses, while "Doc ChatBot" provides less value for money, and the Question and Answer System needs further refinement for handling complex documents[7]. RAG-Based LLM ChatBot Using Llama-2, the research talks about the development and implementation of a chatbot leveraging large language models (LLMs), specifically Llama-2, to support victims of sexual harassment. Utilizing techniques such as retrieval-augmented generation (RAG), the chatbot delivers accurate, empathetic, and non-judgmental responses, achieving over 95% accuracy. Key enhancements proposed include live web-scraping for real-time counselor information, transitioning to cloud-based systems for scalability,

and incorporating multilingual support to improve accessibility and user experience [8].

### III. PROPOSEDWORK

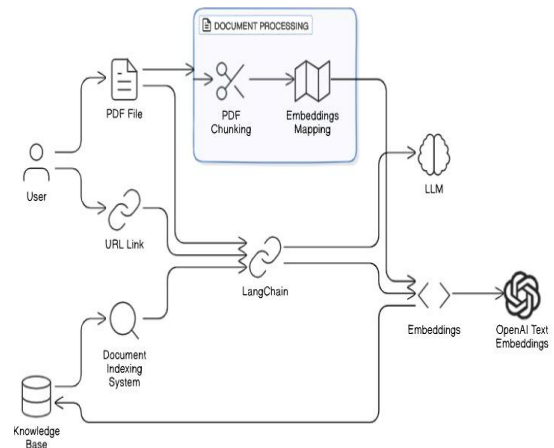
Our research is centred on leveraging LangChain, an advanced tool for NLP and LLM applications, to enhance question-answering capabilities using large language models (LLMs) specifically designed for processing URLs and PDF documents. At the core of our system is the OpenAI embeddings model, which creates the embeddings essential for this AI application. To boost the system's functionality, we also integrate prominent open-source models like Hugging Face models and Google's Universal Sentence Encoder.

For efficient management of large datasets, we use FAISS, an open-source vector store database that excels in storing vectors. In developing LangChain, we utilize the Conversational Retrieval Chain to improve conversational interactions by incorporating historical context, thereby enhancing response accuracy and relevance.

A key focus of our research is the seamless integration of these components, demonstrating their collective efficiency in building a robust and context-aware conversational AI application designed for real-world use. Figure 1 illustrates the architecture of the PDF Chat Model, showcasing how these components interconnect to deliver an enhanced user experience.

We implemented features in our project to allow querying and information retrieval using the OpenAI API key. However, during implementation, we faced challenges with the accuracy of the model's answers and encountered issues with the costs associated with using the OpenAI API key for embeddings and language model interactions.

Then we chose Dash for the graphical user interface (GUI) due to its capability to process backend data efficiently, create embeddings, and provide responses from the large language model via the OpenAI API. While Dash is our current choice, it's worth noting that it can easily be swapped with other open-source models to meet specific project needs and preferences.



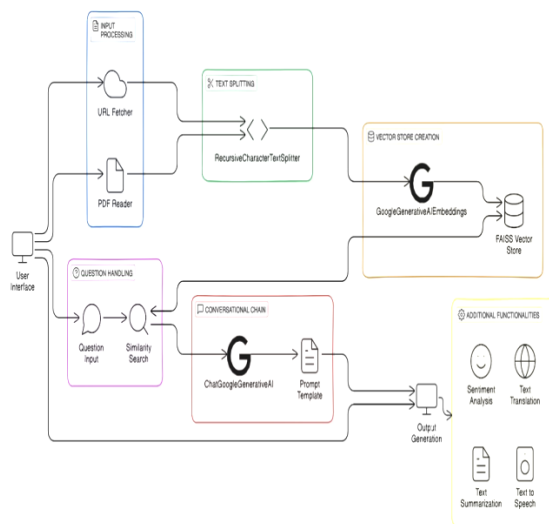
**Fig.2: System workflow for PDF and URL Processing**

In Figure 2, you can see the system's workflow, where users upload a PDF file or URL link. The system then processes user queries and generates responses based on the LLM and embeddings utilized. LangChain is crucial here, linking the language model API to external data sources, which allows for seamless communication and data extraction from PDF documents.

When dealing with extensive content in PDF documents that might exceed the token limit of the large language model, the system smartly divides the document into smaller, manageable chunks. Each chunk is mapped to its corresponding embeddings, effectively compressing the data without losing accuracy.

The use of OpenAI's text embeddings enhances the system's capabilities by enabling efficient comparison of text similarities based on embeddings rather than direct text comparison. This method forms the foundation for creating a knowledge base, where document embeddings are used to retrieve relevant information based on the relatedness and closeness of queries.

Our document indexing system, combined with advanced document embedding techniques and semantic search methods, significantly improves the search experience within the LangChain Q&A model. These techniques work together to boost the system's efficiency in handling PDF files, providing users with accurate and context-aware responses when interacting with the AI-driven conversational interface as in figure3.



**Fig. 3: Architecture Diagram**

**Challenges with the current methodology,** Current text analysis methodologies face several challenges, including limited interactivity and integration capabilities, inadequate handling of diverse data sources like websites and PDFs, and a lack of flexibility and advanced analysis features. These limitations lead to inefficient and inaccurate text processing. Users often cannot interact dynamically with the system to ask questions based on specific content from URLs or PDFs. Moreover, traditional systems fail to effectively integrate multiple modern technologies to enhance text analysis capabilities.

We implemented a solution that integrates Dash, LangChain, Google Generative AI, and FAISS. This system allows users to provide links to websites and PDFs and ask questions based on the content within these sources.

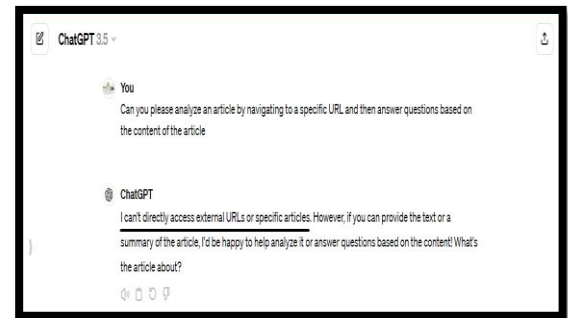
#### Benefits:

1. **Enhanced interactivity:** Users can interact dynamically with the system and ask questions based on the provided content.
2. **Comprehensive data processing:** The integration of advanced technologies improves the accuracy and efficiency of text analysis.
3. **User-friendly interfaces:** These interfaces enable seamless interaction and provide accurate answers based on the user-provided articles.

#### Why not other AI tools

- **1. Word Limit**
- **2. Can't provide URL as input**

ChatGPT has a hidden limit of around 500 words or 4,000 characters, but this depends on the request. However, OpenAI has not revealed a specific word or character limit for ChatGPT. If you reach the limit, you can ask the chatbot to continue its original response by typing "continue" in the message



**Fig.4: ChatGPT Limitations**

#### Proposed Pseudo code Algorithm

##### User Input Processing:

**Step 1:** The user inputs a URL or uploads a PDF file.

**Step 2:** The system fetches the text content from the provided URL using web scraping techniques with BeautifulSoup or reads the text from the uploaded PDF using PyPDF2.

##### Text Splitting:

**Step 3:** The fetched text content is split into manageable chunks using the RecursiveCharacterTextSplitter from LangChain. This ensures efficient processing and handling of large text data.

##### Vector Store Creation:

**Step 4:** The text chunks are embedded using GoogleGenerativeAIEmbeddings.

**Step 5:** The embedded text chunks are stored in a FAISS vector store, which enables efficient similarity searches and retrieval of relevant text segments.

##### Question Handling:

**Step 6:** The user inputs a question related to the content of the URL or PDF.

**Step 7:** The system retrieves relevant text chunks from the FAISS vector store using similarity search based on the user's question.

##### Conversational Chain:

**Step 8:** A conversational chain is created using the ChatGoogleGenerativeAI model and a prompt

template that includes the context (retrieved text chunks) and the user's question.

**Step 9:** The model processes the input and generates a detailed answer based on the provided context.

#### Additional Functionalities:

**Step 10:** If requested, the system can perform additional analyses, such as:

**Numeric Data Extraction:** Extracting numeric data from the text for further analysis or graph generation.

**Flowchart Generation:** Creating a flowchart of the text content for better visualization.

**Named Entity Recognition (NER):** Identifying and categorizing named entities within the text.

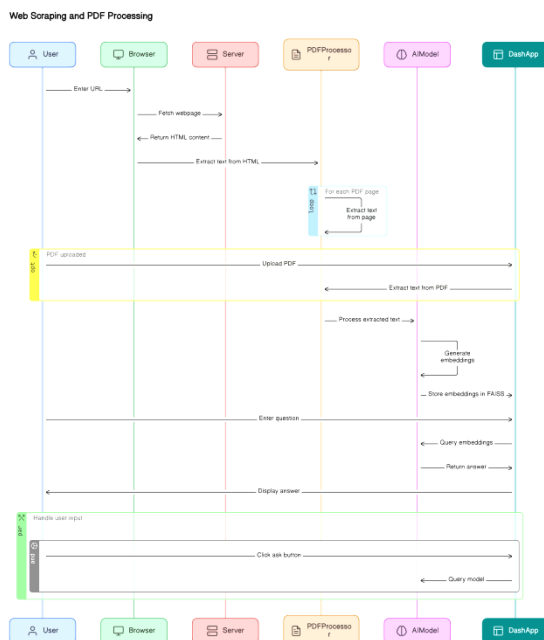
**Sentiment Analysis:** Analyzing the sentiment of the text content.

**Text Translation:** Translating the text to a specified language using Google Translate.

**Text Summarization:** Summarizing the text to provide a concise overview.

Output Generation:

**Step 11:** The system generates the final output based on the user's question and any additional requested analyses. This output is displayed to the user through the Dash web application interface. Entire Pseudo algorithm steps are shown in below sequence diagram in figure 5.



**Fig.5: Sequence Diagram**

## Implementation:

The proposed system includes a Dash web application framework, utilizing HTML, CSS, and JavaScript components to create a user-friendly interface for interacting with the text analysis system. For text processing, the LangChain Text Splitter module (`langchain.text_splitter`) is employed. This module is responsible for splitting text content into manageable chunks using the `RecursiveCharacterTextSplitter` class, which allows for configurable chunk size and overlap, facilitating efficient and accurate text analysis.

The system integrates Google Generative AI Embeddings through the `langchain_google_genai.GoogleGenerativeAIEmbeddings` module. This module embeds text chunks by leveraging the Google Generative AI API to generate embeddings. For storing and retrieving these text embeddings for similarity searches, the system uses the FAISS Vector Store, implemented via the `langchain.vectorstores.FAISS` module. The FAISS library is employed to ensure efficient vector storage and retrieval, enhancing the system's capability to perform rapid and accurate similarity searches.

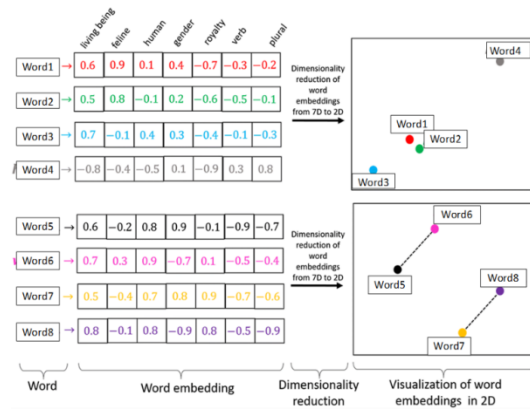
## IV .RESULTS AND DISCUSSION

In our Chat PDF and URL UI tool, users can effortlessly drag and drop PDF files or enter URLs to interact with the system. For instance, users can upload a PDF file containing information on various topics and ask specific questions about details like the number of states or relevant statistics. Similarly, users can input URLs of articles or documents to extract information and engage in a conversational interface.

### Word Embedding

When it comes to word embeddings, we employ dimensionality reduction techniques to retain essential semantic information while reducing the number of vector-based features. This reduction in complexity makes embeddings more efficient for processing large amounts of textual data from both URLs and PDFs. Figure 6 illustrates the process of transforming text into vector forms and using machine learning models to generate embeddings.





**Fig.6: Word Extraction and Feature mapping**

Converting text information into vector forms enhances efficiency significantly. Machine learning models transform texts into multi-dimensional vectors. Once converted, these vectors can be sorted, searched, and grouped, as depicted in Figure 5. This process goes beyond standard keyword-based database searches by capturing the semantic proximity of sentences, a capability significantly enhanced by machine learning.

### Develop a Chat Interface

The user interface (UI) of our application integrates functionalities for both PDF and URL processing. The chat window allows users to interact with the system using natural language queries, while a text box lets them input URLs or upload PDF files for analysis. By integrating the OpenAI API keys, the system can generate responses based on insights derived from the large language model's analysis of both URL content and PDF documents.

### Dash Framework

Dash is a powerful Python framework for building interactive web applications, and it plays a crucial role in our project. Here are some key points about Dash and its contributions:

**Pythonic Development:** Dash allows us to build web applications entirely in Python, from backend logic to frontend UI components. This is a perfect fit for our project, where we're already using Python extensively for machine learning and AI tasks.

**Interactive Data Visualization:** Dash provides a variety of interactive visualization components, such as graphs, charts, and widgets. These can be seamlessly integrated into our web application to visualize data, display insights, and enhance user interactions.

**Real-time Updates:** Dash supports real-time updates without needing to reload the entire page. This feature is crucial for our application, where users expect dynamic responses and continuous interaction with the AI models and data processing features.

**Customizable UI:** Dash offers full control over the UI design and layout. This allows us to create custom dashboards, arrange components as needed, and style the interface to match our project's branding and user experience goals.

**Integration with Data Science Libraries:** Dash integrates seamlessly with popular data science libraries like Pandas, NumPy, Plotly, and more. This makes the process of data manipulation, analysis, and visualization within our web application straightforward.

**Scalability and Deployment:** Dash applications can be easily deployed on various platforms, including cloud services, servers, and containers like Docker. This scalability ensures that our AI-driven web application can handle increasing user demands and data processing requirements.

### URL and PDF Processing

The system's ability to handle both URLs and PDFs demonstrates its versatility and applicability across various data sources. When users input URLs, the system fetches the content, processes it using natural language processing techniques, and generates relevant responses. Similarly, for PDF documents, the system extracts text, creates embeddings, and provides informative answers to user queries. This dual capability enhances the system's utility for information retrieval and knowledge discovery from diverse online sources.

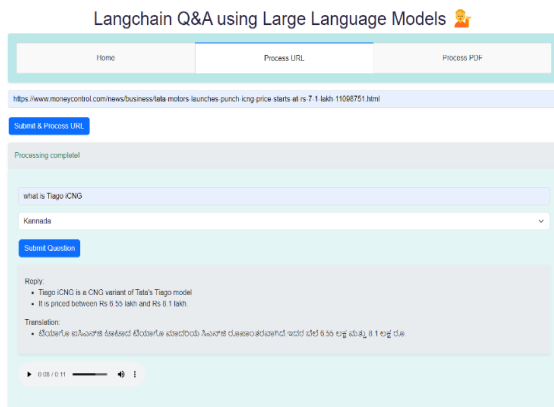
The seamless integration of URL and PDF processing functionalities within the chat interface powered by large language models and word embeddings showcases the system's effectiveness in handling different data formats. Users can benefit from a unified platform that efficiently extracts insights from both online documents and downloadable PDFs, making information retrieval and conversational interactions more accessible and productive. The utilization of Dash framework further enhances the user experience by offering a visually appealing and interactive environment for engaging with the AI-driven system.

## Screen Shots of the Results



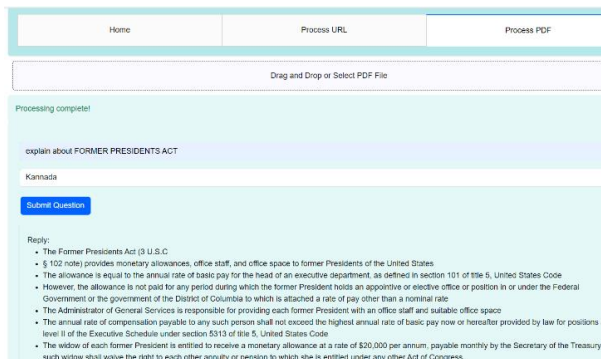
**Fig. 7: User Interface Home Page.**

This is the Web page looks when visiting the page, based on the User requirements User can choose to interact with the Application either he/she can choose a. Interact with URL's or with PDF's.



**Fig.8: Query with URL's using Langchain**

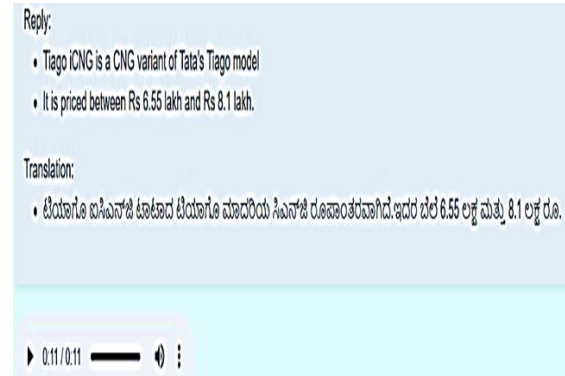
This is exactly what the web tool's UI would seem. The individual enter the URL's of anysite and click on Process. After processing for a few minutes, it will receive an additional input box where it can submit the query as shown in Fig.8.



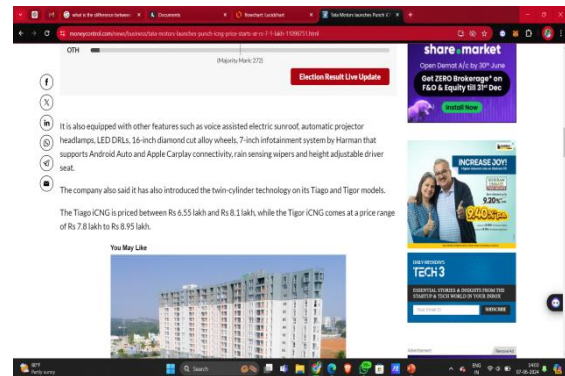
**Fig. 9: Processed Query related to the PDF file**

This is exactly what the web tool's UI would seem. The individual can upload a file on their device that is smaller than 200 megabytes by clicking on

Browse Files. After processing for a few minutes, it will receive an additional input box where it can submit the query as shown in Fig.9 After submitting the query it will display the result as in figure 10, if it match the rules.

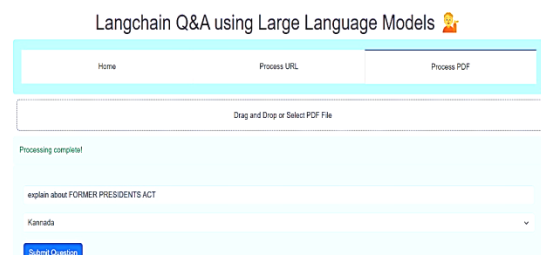


**Fig.10: Retrieved answer for the requested question**



**Fig.11: Proof of the answer retrieved and is existing in the content**

The interface allows users to upload a PDF file for text analysis. It retrieves the PDF's content, processes it, and stores it as vector embeddings using FAISS. Users can then ask questions based on the content, with the system providing detailed responses. Additional analysis features, such as translation, sentiment analysis, and summarization, are included, leveraging Google Generative AI for accurate results as in figure10, 11. Here it is shown for Kannada language in figure 12.



**Fig.12: Text to speech conversion in 10 different languages.**

## V. CONCLUSION

In our research on "Langchain Q&A using Large Language Models for URLs and PDFs," we delved into the capabilities of Large Language Models (LLMs), pinpointed their limitations, and showed how the LangChain framework can effectively address these challenges. By building a custom LangChain Q&A system for both URLs and PDFs, we've demonstrated a practical application that leverages the strengths of LLMs while mitigating their weaknesses.

As Generative AI Technologies continue to evolve, we see a persistent trend toward more advanced and innovative solutions. LangChain's flexible architecture allows us to overcome many of the common drawbacks of traditional LLMs, enabling us to create bespoke solutions for specific needs. By integrating Q&A functionalities for both URL links and PDF documents using the OpenAI API, and employing techniques like text splitting, embeddings, and question-answering mechanisms, we've built an interactive platform. This platform empowers users to engage in meaningful conversations and extract valuable insights from various sources, significantly enhancing information retrieval and knowledge extraction.

The adaptability and customizability of this approach mean it can be easily tailored to different requirements and scenarios. This makes it a versatile and invaluable tool for harnessing the power of LLMs in handling queries based on URLs and PDFs. As we continue to refine and expand this framework, it promises to be a transformative asset for researchers, educators, and professionals, facilitating efficient and accurate information retrieval from a broad range of digital content sources.

## References

- [1] Topsakal, Oguzhan, and Tahir Cetin Akinci. "Creating large language model applications utilizing langchain: A primer on developing llm apps fast." International Conference on Applied Engineering and Natural Sciences. Vol. 1. No. 1. 2023.
- [2] Kinoshita, Ryosuke, and Shun Shiramatsu. "Agent for recommending information relevant to web-based discussion by generating query terms using GPT-3." 2022 IEEE International Conference on Agents (ICA). IEEE, 2022.
- [3] Teubner, Timm, et al. "Welcome to the era of chatgpt et al. the prospects of large language models." Business & Information Systems Engineering 65.2 (2023): 95-101.
- [4] Meyer, Jesse G., et al. "ChatGPT and large language models in academia: opportunities and challenges." BioData Mining 16.1 (2023)
- [5] Dam, Sumit Kumar, et al. "A Complete Survey on LLM-based AI Chatbots." arXiv preprint arXiv:2406.16937 (2024).
- [6] Dam, Sumit Kumar, et al. "A Complete Survey on LLM-based AI Chatbots." arXiv preprint arXiv:2406.16937 (2024).
- [7] Taecharungroj, Viriya. "'What can ChatGPT do?'" Analyzing early reactions to the innovative AI chatbot on Twitter." Big Data and Cognitive Computing 7.1 (2023): 35.
- [8] Vakayil, Sonia, D. Sujitha Juliet, and Sunil Vakayil. "RAG-Based LLM Chatbot Using Llama-2." 2024 7th International Conference on Devices, Circuits and Systems (ICDCS). IEEE, 2024.
- [9] Pokhrel, Sangita, et al. "Building Customized Chatbots for Document Summarization and Question Answering using Large Language Models using a Framework with OpenAI, Lang chain, and Streamlit." Journal of Information Technology and Digital World 6.1 (2024): 70-86.
- [10] Chen, Jin, et al. "When large language models meet personalization: Perspectives of challenges and opportunities." World Wide Web 27.4 (2024): 42.
- [11] Kumar, V., Srivastava, P., Dwivedi, A., Budhiraja, I., Ghosh, D., Goyal, V., & Arora, R. (2023, December). Large-Language-Models (LLM)-Based AI Chatbots: Architecture, In-Depth Analysis and Their Performance Evaluation. In International Conference on Recent Trends in Image Processing and Pattern Recognition (pp. 237-249). Cham: Springer Nature Switzerland.
- [12] Medeiros, Thaís, Morsinaldo Medeiros, Mariana Azevedo, Marianne Silva, Ivanovitch Silva, and Daniel G. Costa. "Analysis of language-model-powered Chatbots for query resolution in



PDF-based automotive manuals." *Vehicles* 5, no. 4 (2023): 1384-1399.

[13] Alto, Valentina. *Building LLM Powered Applications: Create intelligent apps and agents with large language models*. Packt Publishing Ltd, 2024.

[14] Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. "Large language models: A survey." *arXiv preprint arXiv:2402.06196* (2024).

[15] Zhang, Chengyi, Xingyu Wang, and Ziyun Wang. "Large language model in electrocatalysis." *Chinese Journal of Catalysis* 59 (2024): 7-14.

[16] Topsakal, Oguzhan, and Tahir Cetin Akinci. "Creating large language model applications utilizing langchain: A primer on developing llm apps fast." In *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, pp. 1050-1056. 2023.