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

Priya K[1], Akshatha Kamath[1], Chandan K M[1], Praveen C[1], Omkar S N[1], Aaditya S J[1]
[1]Dept of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, India.
Emails: priyak@msrit.edu, akshathakamath@msrit.edu

Abstract

Searching through URLs and PDFs can be tedious and time-consuming because of the unstructured nature of these documents and the challenge of finding accurate and, relevant information. LangChain addressed these issues 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 the 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 the 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 of converting text into speech and translating it into 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 artificial intelligence (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. However, beyond chatting, have you ever thought about using such AI to query computer documents instead of manually searching through them? This is now possible owing to advancements in conversational and generative AI technologies. 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, have become more prevalent, retrieving information from them has become increasingly challenging. LangChain, a revolutionary tool built on Natural Language Processing (NLP) and Large Language Models (LLMs), tackles these challenges. LangChain simplifies the search and information extraction process for URLs and PDFs using advanced NLP algorithms. LangChain employs Dash, a user-friendly web application framework that does not require extensive knowledge of web development technologies such as CSS and HTML, to create an intuitive interface. With Dash, deploying models becomes seamless and requires minimal coding. This integration allows users to interact effortlessly with URLs and PDFs, significantly enhancing the ease of document search and retrieval as in figure1.

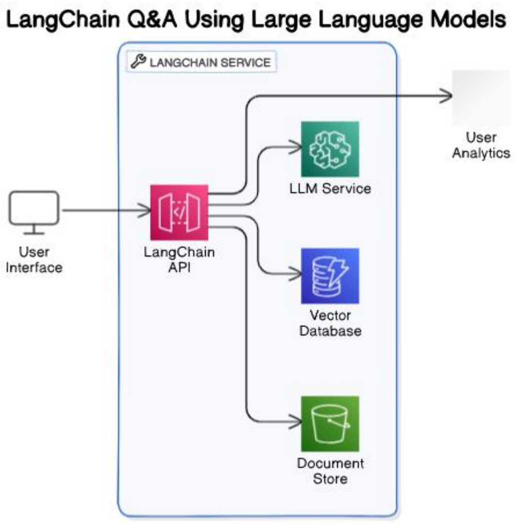


Fig.1: LLM Model for Q and A

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 cases such as autonomous agents, chatbots, and code understanding. Featured works include the integration of LangChain with diverse data sources and its modular structure, which 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.(Topsakal and Akinci,2023)

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 and, recommends information every three minutes, which helps participants better understand the discussion 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 (Kinoshita and Shiramatsu,2022). Welcome to the Era of ChatGPT, et al. In this research, large language models (LLMs) and chatbots are used in radiology, highlighting their applications in literature reviews, data extraction, and preliminary data analysis. This emphasizes the potential of LLMs to enhance research efficiency by performing summarization, extraction, and classification tasks. Featured works include domain-specific models such as 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. This paper also discusses data privacy concerns and mitigation strategies in healthcare applications (Teubner et al.,2023).

ChatGPT and Large Language Models in Academia: Opportunities and Challenges research addresses the emergence of Large Language Models (LLMs) such as ChatGPT, Bing Chat, and Google's Bard, highlighting their significant impact on various domains including business, education, and creative fields. This emphasizes the efficiency of LLMs in generating coherent text, thus shifting the focus to content and ideas rather than the mechanics of writing. This paper also addresses ethical considerations, the 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 such as BioGPT. Further advancements are expected with the better integration of real-time data and improved model architectures(Meyer et al.,2023). 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 their effective use. This paper emphasizes the utility of LLMs in improving efficiency and the importance of addressing their limitations, such as accuracy issues and the potential for plagiarism. The key methods discussed include iterative chat capabilities and the application of AI in various academic contexts(Dam S K et al.,2024). A Comprehensive Comparative Analysis of Three Chatbot Approaches for Answering PDF-Based Questions addresses the comprehensive survey of LLM-based AI chatbots, detailing their evolution, applications, and challenges. It examines chatbots such as ChatGPT, BARD, Bing Chat, and Claude, highlighting their significant advancements in early conversational AI, which faced issues such as limited contextual understanding and domain specificity. This paper categorizes applications across sectors such as education, research, and healthcare, and discusses challenges such as scalability and ethical concerns. Featured works include a detailed taxonomy of applications and strategies to enhance the ChatBot performance and reliability(Dam S K et al.,2024). “What Can ChatGPT Do?” Analyzing Early Reactions to the Innovative AI ChatBot on Twitter this study evaluate 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 when 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 to handle complex documents(Taecharungroj et al.,2023). 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. The 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(Vakayil et al.,2024).

III. PROPOSEDWORK

Our research focuses 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. 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, such as Hugging Face models and Google's Universal Sentence Encoder.

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

A key focus of our research was 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, showing how these components interconnect to deliver an enhanced user experience.

We implemented the 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.

We chose Dash for the graphical user interface (GUI) because of 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 be easily swapped with other open-source models to meet specific project needs and preferences.

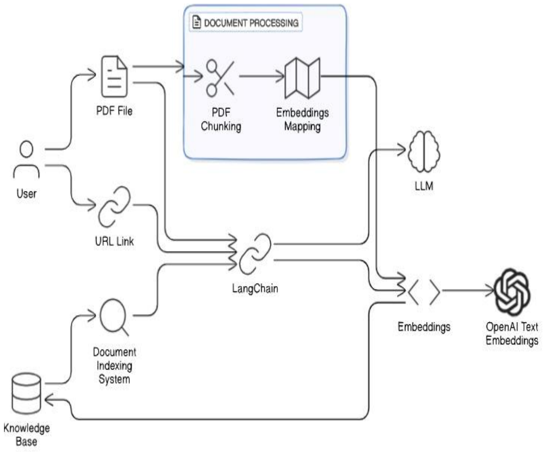


Fig.2: System workflow for PDF and URL Processing

In Figure 2, the system workflow can be seen, where users upload a PDF file or URL link. The system then processes user queries and generates responses based on the LLM and embeddings. LangChain is crucial here, linking the language model API to external data sources, which allows 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 was 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 an 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 of 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 an AI-driven conversational interface as in figure3.

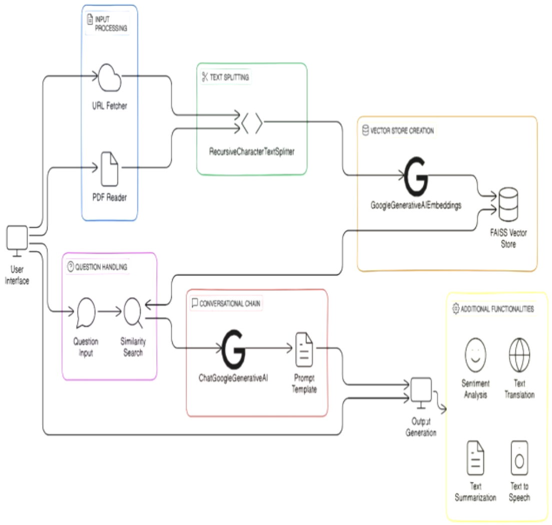


Fig. 3: Architecture Diagram

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 result in inefficient and inaccurate text processing. Users often cannot interact dynamically with the system to ask questions based on specific content from the URLs or PDFs. Moreover, traditional systems fail to effectively integrate multiple modern technologies to enhance text analysis capabilities.

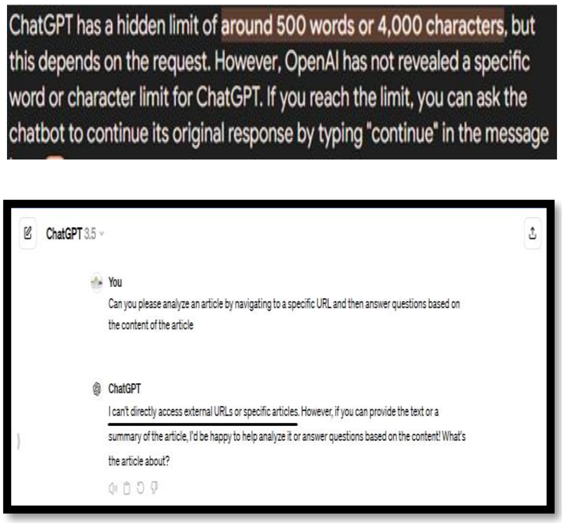
We implemented a solution that integrates Dash, LangChain, Google Generative AI, and FAISS. This system allows users to provide links to websites and PDFs and ask questions based on the content within these sources.

Benefits:

1. **Enhanced interactivity**: Users can dynamically interact with the system and ask questions based on the content provided.
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 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

You

Can you please analyze an article by navigating to a specific URL and then answer questions based on the content of the article

ChatGPT

I can't directly access external URLs or specific articles. However, if you can provide the text or a summary of the article, I'd be happy to help analyze it or answer questions based on the content! What's the article about?

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 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 recursive character text splitter from LangChain. This ensures the 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 the relevant text segments.

Question Handling:

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

Step 7: The system retrieves the relevant text chunks from the FAISS vector store using a 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 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:
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 questions and additional requested analyses. This output is displayed to the user through a Dash web application interface. All pseudo-algorithm steps are shown in the sequence diagram in figure 5.

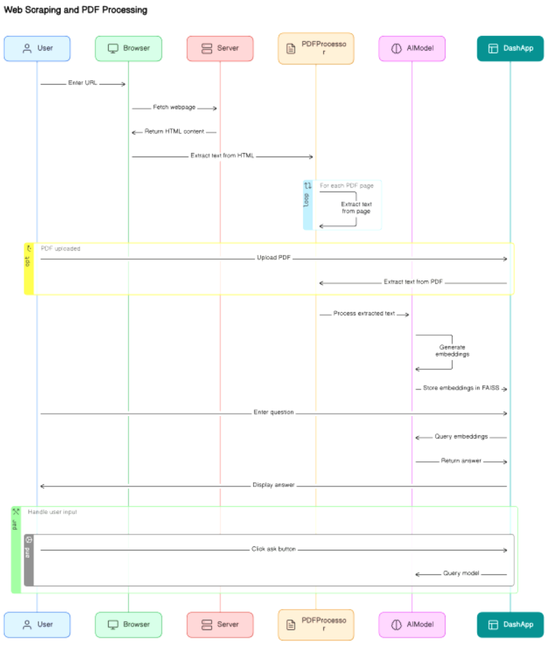


Fig.5: Sequence Diagram

Implementation:

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

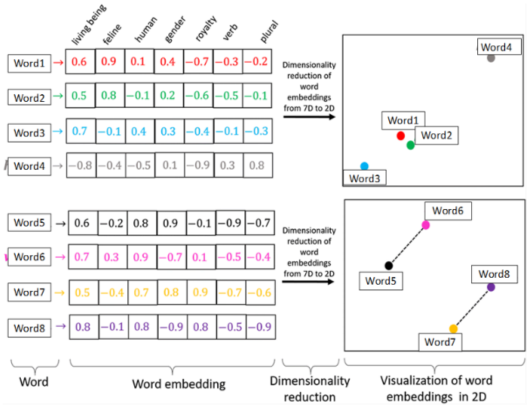
The system integrates Google Generative AI Embeddings through the langchain_google_genai.GoogleGenerativeAIEmbeddings module. This module embeds text chunks by leveraging the Google Generative AI API to generate embeddings. To store and retrieve text embeddings for similarity searches, the system uses the FAISS Vector Store, which is implemented via the Langchain_vectorstores_FAISS module. The FAISS library was employed to ensure efficient vector storage and retrieval, enhancing the capability of the system 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 a system. For instance, users can upload a PDF file containing information on various topics and ask specific questions about details such as 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

For word embeddings, we employed 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 the URLs and PDFs. Figure 6 illustrates the process of transforming text into vector forms and using machine learning models to generate embeddings.



	living being	feline	human	gender	royalty	verb	plural
Word1	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
Word2	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
Word3	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
Word4	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
Word5	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
Word6	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
Word7	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
Word8	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Dimensionality reduction of word embeddings from 7D to 2D
Word4
Word1
Word2
Word3
Dimensionality reduction of word embeddings from 7D to 2D
Word6
Word5
Word8
Word7
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

Fig.6: Word Extraction and Feature mapping

Converting text information into vector forms significantly enhances efficiency. Machine learning models transform texts into multi-dimensional vectors. Once converted, these vectors can be sorted, searched, and grouped, as shown in Figure 5. This process goes beyond standard keyword-based database searches by capturing the semantic proximity of sentences, a capability that is 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, whereas a text box allows them to 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 the URL content and PDF documents.

Dash Framework

Dash is a powerful Python framework for building interactive web applications that plays a crucial role in our project. Here are some key points regarding Dash and its contributions are as follows.

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, in which users expect dynamic responses and continuous interaction with AI models and data-processing features.

Customizable UI: Dash offers full control over UI design and layout. This allowed 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 seamlessly integrates with popular data science libraries such as Pandas, NumPy, and Plotly. This creates the process of data manipulation, analysis, and visualization within our web application.

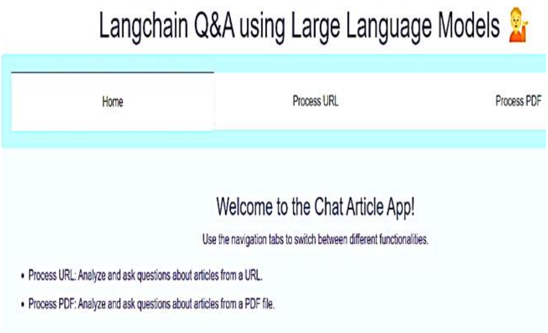
Scalability and Deployment: Dash applications can be easily deployed on various platforms, including cloud services, servers, and containers such as Docker. This scalability ensures that our AI-driven web application can handle increasing user demands and data-processing requirements.

URL and PDF Processing

The ability of the system 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 the user queries. This dual capability enhances the utility of the system in 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 show 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, thus making information retrieval and conversational interactions more accessible and productive. The utilization of the Dash framework further enhances user experience by offering a visually appealing and interactive environment for engaging with the AI-driven system.

Screen Shots of the Results


Langchain Q&A using Large Language Models 💁

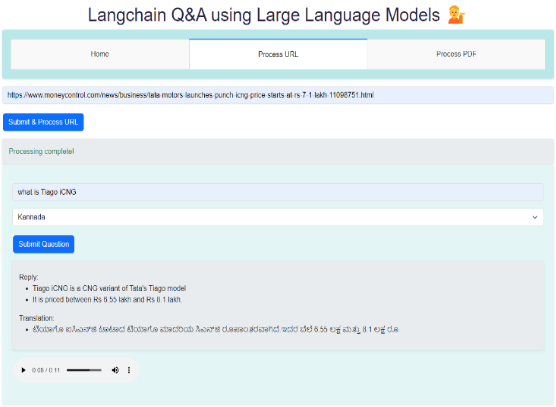
Welcome to the Chat Article App!

Use the navigation tabs to switch between different functionalities.

- Process URL: Analyze and ask questions about articles from a URL.
- Process PDF: Analyze and ask questions about articles from a PDF file.

Fig. 7: User Interface Home Page.

Based on the user requirements, the user can choose to interact with the application either by interacting with URL's or with PDF's.


Langchain Q&A using Large Language Models 💁

https://www.moneycontrol.com/news/business/tata-motors-launches-punch-icng-price-starts-at-rs-7-1-lakh-11098751.html

Processing complete!

what is Tiago iCNG

Kannada

Reply:

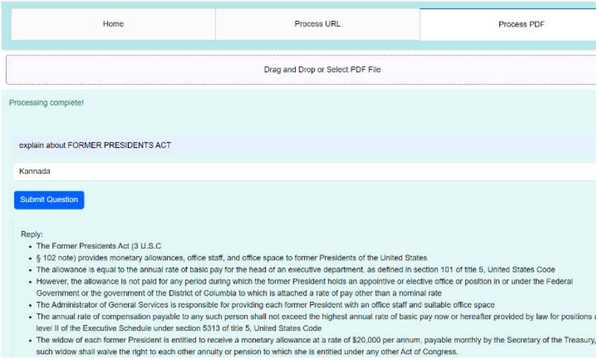
- Tiago iCNG is a CNG variant of Tata's Tiago model
- It is priced between Rs 6.55 lakh and Rs 8.1 lakh.

Translation:

- ಟಿಯಾಗೊ ಐಸಿಎನ್‌ಜಿ ಟಾಟಾದ ಟಿಯಾಗೊ ಮಾದರಿಯ ಸಿಎನ್‌ಜಿ ರೂಪಾಂತರವಾಗಿದೆ.ಇದರ ಬೆಲೆ 6.55 ಲಕ್ಷ ಮತ್ತು 8.1 ಲಕ್ಷ ರೂ.

Fig.8: Query with URL's using Langchain

This is exactly what the Web tool's UI would seem. The individual enters the URL's of any site and clicks on the process. After processing for a few seconds, it receives an additional input box where it can submit the query as shown in Fig.8.



Processing complete!

explain about FORMER PRESIDENTS ACT

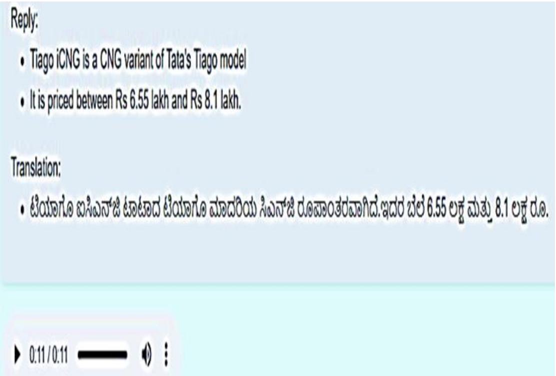
Kannada

Reply:

- The Former Presidents Act (3 U.S.C
- § 102 note) provides monetary allowances, office staff, and office space to former Presidents of the United States
- The allowance is equal to the annual rate of basic pay for the head of an executive department, as defined in section 101 of title 5, United States Code
- However, the allowance is not paid for any period during which the former President holds an appointive or elective office or position in or under the Federal Government or the government of the District of Columbia to which is attached a rate of pay other than a nominal rate
- The Administrator of General Services is responsible for providing each former President with an office staff and suitable office space
- The annual rate of compensation payable to any such person shall not exceed the highest annual rate of basic pay now or hereafter provided by law for positions a level II of the Executive Schedule under section 5313 of title 5, United States Code
- The widow of each former President is entitled to receive a monetary allowance at a rate of $20,000 per annum, payable monthly by the Secretary of the Treasury, such widow shall waive the right to each other annuity or pension to which she is entitled under any other Act of Congress.

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 smaller than 200 megabytes by clicking on Browse Files. After processing for a few minutes, it receives an additional input box where it can submit the query, as shown in Fig.9, which displays the result as shown in figure 10, if it matches the rules.

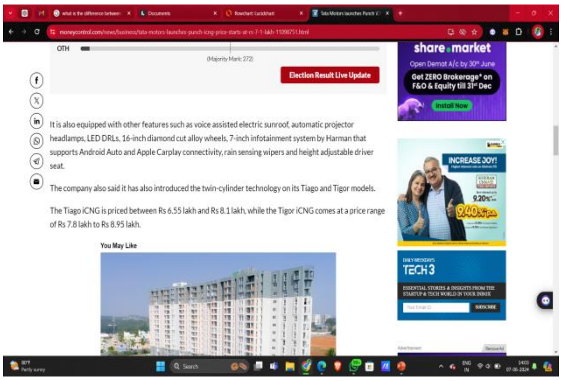

Reply:

- Tiago iCNG is a CNG variant of Tata's Tiago model
- It is priced between Rs 6.55 lakh and Rs 8.1 lakh.

Translation:

- ಟಿಯಾಗೊ ಐಸಿಎನ್‌ಜಿ ಟಾಟಾದ ಟಿಯಾಗೊ ಮಾದರಿಯ ಸಿಎನ್‌ಜಿ ರೂಪಾಂತರವಾಗಿದೆ.ಇದರ ಬೆಲೆ 6.55 ಲಕ್ಷ ಮತ್ತು 8.1 ಲಕ್ಷ ರೂ.

Fig.10: Retrieved answer for the requested question


moneycontrol.com/news/business/tata-motors-launches-punch-icng-price-starts-at-rs-7-1-lakh-11098751.html
OTH
(Majority Mark: 272)
Election Result Live Update
share.market
Open Demat A/c by 30th June
Get ZERO Brokerage* on F&O & Equity till 31st Dec
Install Now
INCREASE JOY!
9.40%
TECH 3
SUBSCRIBE
You May Like

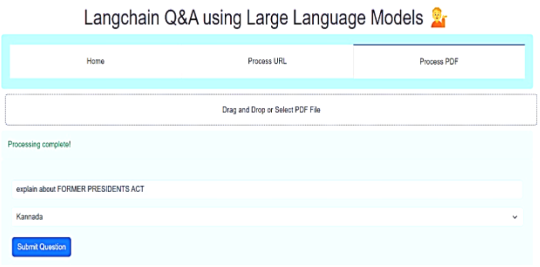
It is also equipped with other features such as voice assisted electric sunroof, automatic projector headlamps, LED DRLs, 16-inch diamond cut alloy wheels, 7-inch infotainment system by Harman that supports Android Auto and Apple Carplay connectivity, rain sensing wipers and height adjustable driver seat.

The company also said it has also introduced the twin-cylinder technology on its Tiago and Tigor models.

The Tiago iCNG is priced between Rs 6.55 lakh and Rs 8.1 lakh, while the Tigor iCNG comes at a price range of Rs 7.8 lakh to Rs 8.95 lakh.

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

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


Langchain Q&A using Large Language Models 💁

Processing complete!

explain about FORMER PRESIDENTS ACT

Kannada

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 demonstrate a practical application that leverages the strengths of LLMs while mitigating their weaknesses.

As Generative AI Technologies continue to evolve, there is a persistent trend toward more advanced and innovative solutions. LangChain's flexible architecture overcomes 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 such as text splitting, embeddings, and question-answering mechanisms, we built an interactive platform. This platform empowers users to engage in meaningful conversations and extract valuable insights from various sources, thereby significantly enhancing information retrieval and knowledge extraction.

Looking ahead, we plan to expand the capabilities of this system by implementing additional features, such as **numeric data extraction** for further analysis and graph generation, and **flowchart generation** to provide visual representations of text content. These enhancements will further enrich the user experience by offering more comprehensive and accessible ways to interpret and utilize the information retrieved from digital content.

The adaptability and customizability of this approach mean that it can be easily tailored to different requirements and scenarios. This makes it a versatile and invaluable tool for harnessing the power of LLMs to handle 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 retrieval of information from a broad range of digital content sources.

Data Availability Statement:

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

[1] Topsakal, O. and Akinci, T.C., 2023, July. 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).

[2] Kinoshita, R. and Shiramatsu, S., 2022, November. Agent for recommending information relevant to web-based discussion by generating query terms using GPT-3. In *2022 IEEE International Conference on Agents (ICA)* (pp. 24-29). IEEE.

[3] Teubner, T., Flath, C.M., Weinhardt, C., van der Aalst, W. and Hinz, O., 2023. Welcome to the era of chatgpt et al. the prospects of large language models. *Business & Information Systems Engineering*, *65*(2), pp.95-101.

[4] Meyer, J.G., Urbanowicz, R.J., Martin, P.C., O'Connor, K., Li, R., Peng, P.C., Bright, T.J., Tatonetti, N., Won, K.J., Gonzalez-Hernandez, G. and Moore, J.H., 2023. ChatGPT and large language models in academia: opportunities and challenges. *BioData Mining*, *16*(1), p.20.

[5] Dam, S.K., Hong, C.S., Qiao, Y. and Zhang, C., 2024. A Complete Survey on LLM-based AI Chatbots. *arXiv preprint arXiv:2406.16937*.

[6] Taecharungroj, V., 2023. "What can ChatGPT do?" Analyzing early reactions to the innovative AI chatbot on Twitter. *Big Data and Cognitive Computing*, *7*(1), p.35.

[7] Vakayil, S., Juliet, D.S. and Vakayil, S., 2024, April. RAG-Based LLM Chatbot Using Llama-2. In *2024 7th International Conference on Devices, Circuits and Systems (ICDCS)* (pp. 1-5). IEEE.

[8] Pokhrel, S., Ganesan, S., Akther, T. and Karunarathne, L., 2024. 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), pp.70-86.

[9] Chen, J., Liu, Z., Huang, X., Wu, C., Liu, Q., Jiang, G., Pu, Y., Lei, Y., Chen, X., Wang, X. and Zheng, K., 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, *27*(4), p.42.

[10] Kumar, V., Srivastava, P., Dwivedi, A., Budhiraja, I., Ghosh, D., Goyal, V. and 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.

[11] Medeiros, T., Medeiros, M., Azevedo, M., Silva, M., Silva, I. and Costa, D.G., 2023. Analysis of language-model-powered Chatbots for query resolution in PDF-based automotive manuals. *Vehicles*, *5*(4), pp.1384-1399.

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

[13] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z. and Du, Y., 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

[14] Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

[15] Zhang, C., Wang, X. and Wang, Z., 2024. Large language model in electrocatalysis. *Chinese Journal of Catalysis*, *59*, pp.7-14.