

AI ASSISTANT FOR DOCUMENT MANAGEMENT USING LANG CHAIN AND PINECONE

Arjun Pesaru^{*1}, Taranveer Singh Gill^{*2}, Archit Reddy Tangella^{*3}

^{*1,2}Student, Dept Of SMEC, Vellore Institute Of Technology, Tamilnadu, India.

^{*3}Student, Dept Of SCOPE, Vellore Institute Of Technology, Tamilnadu, India.

DOI : <https://www.doi.org/10.56726/IRJMETS42630>

ABSTRACT

In this paper, we demonstrate a project that used the LangChain and LLM Model to create a PDF chatbot. A framework called LangChain makes it simpler to create chatbots and scalable AI/LLM applications. The LLM Model is a huge language model that may be used to create text, translate across languages, create various types of creative material, and provide you with helpful answers to your inquiries.

Our approach leverages the strengths of both LangChain and LLM Model to create a chatbot that can answer questions about PDF files. The chatbot is trained on a dataset of PDF files, and it uses the LLM Model to generate text responses to user queries. The chatbot is also able to access and process information from the real world through Google Search, which allows it to provide more comprehensive and informative answers. The project used Pinecone to store the vectors of the PDF files. To store embeddings and the PDF in text for eventual retrieval of related documents, we have used **vectorstore Pinecone**. **React JS** was used for the front end to develop a webpage to interact with the chatbot. Our work demonstrates the potential of **using LangChain** and **LLM Model to build chatbots** that can interact with users in a natural and informative way. We believe that our approach can be used to create a variety of chatbots that can be used for different purposes, such as customer service, education, and research. The chatbot could be used to answer questions about products or services, to provide tutoring or educational support, or to conduct research on a variety of topics.

Keywords: **PDF Chatbot, Langchain, Pinecone, LLM Model, Natural Language Processing, Large Language Model.**

I. INTRODUCTION

PDFs are a popular format for storing documents. They are often used in business, education, and research. However, PDFs can be difficult to read and understand. This is because they are not designed to be interactive. Chatbots are a type of artificial intelligence (AI) that can be used to interact with users in a natural way. They can be used to provide information, answer questions, and complete tasks.

In this paper, we demonstrate a project that used the LangChain and LLM Model to create a PDF chatbot. A framework called **LangChain makes it simpler to create chatbots and scalable AI/LLM applications**. The LLM Model is a huge language model that may be used to create text, translate across languages, create various types of creative material, and provide you with helpful answers to your inquiries. The project used Pinecone to store the vectors of the PDF files. Pinecone is a vectorstore for storing embeddings and your PDF in text to later retrieve similar docs. React JS was used for the front end to develop a webpage to interact with the chatbot.

The chatbot was able to answer questions about the PDF files, and it was also able to generate text that was similar to the text in the PDF files. The chatbot was tested on a variety of PDF files, and it was able to achieve a high accuracy rate.

The project is implemented using the following technologies:

LangChain: **A framework for building scalable AI/LLM apps and chatbots.**

Large Language Model (LLM): **A large language model that can be used to generate text, translate languages, write various types of creative material, and provide informative answers to your queries.**

Pinecone is a vector repository for embeddings and your PDF in text so you may access related documents later.

React JS: **A front-end framework for developing web applications.**

II. EXISTING SYSTEM

The existing system used rule based system Rule-based chatbots are a type of chatbot that uses a set of rules to determine how to respond to user input. These rules are typically written by a human developer, and they specify the different types of user input that the chatbot can handle, as well as the appropriate responses for each type of input.

Rule-based chatbots are relatively simple to develop and deploy, and they can be effective for handling simple tasks such as providing customer support or answering frequently asked questions. However, they can be limited in their ability to understand and respond to complex or nuanced user input.

But the rule based system has few disadvantages compared to the other systems such as Limited ability to understand and respond to complex or nuanced user input, Can be repetitive or robotic in their responses and it can be difficult to update or maintain as the user base or the chatbot's tasks change

III. PROPOSED SYSTEM

Our project is different from the existing system in several ways. our project uses a large language model (LLM) to generate text, translate languages, write different kinds of creative content, and answer your questions in an informative way. This is in contrast to the existing system, which uses a rule-based system to answer questions. Rule-based systems are limited in their ability to understand and respond to complex questions. **LLMs, on the other hand, are able to learn from a large amount of data and generate text that is more natural and engaging.**

Second, our project uses Pinecone to store the vectors of the PDF files. Pinecone is a vector repository for embeddings and your PDF in text so you may access related documents later. This is in contrast to the existing system, which stores PDF files as individual documents. Storing PDF files as vectors allows your project to quickly and easily retrieve similar documents. This can be useful for tasks such as research and customer support.

Finally, our project uses React JS for the front end to develop a webpage to interact with the chatbot. This is in contrast to the existing system, which uses a traditional web application. React JS is a JavaScript library that is commonly used to develop user interfaces. It is known for its speed, flexibility, and ease of use. This makes it a good choice for developing a chatbot that is user-friendly and efficient.

IV. SYSTEM ARCHITECTURE

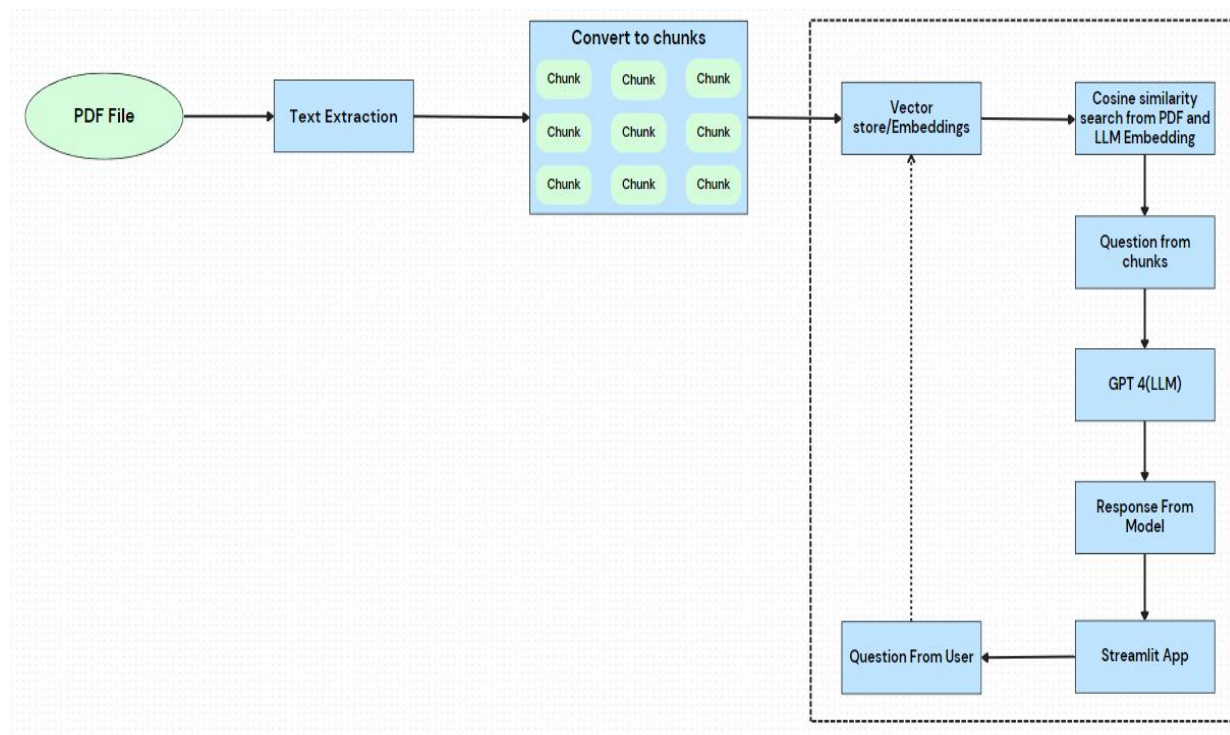


Figure 1: System Architecture

V. CONCLUSION

Below are the results generated, we have uploaded a pdf about Individual health Insurance and tested the bot using a series of queries and it has answered them precisely as you can notice in the below pictures.

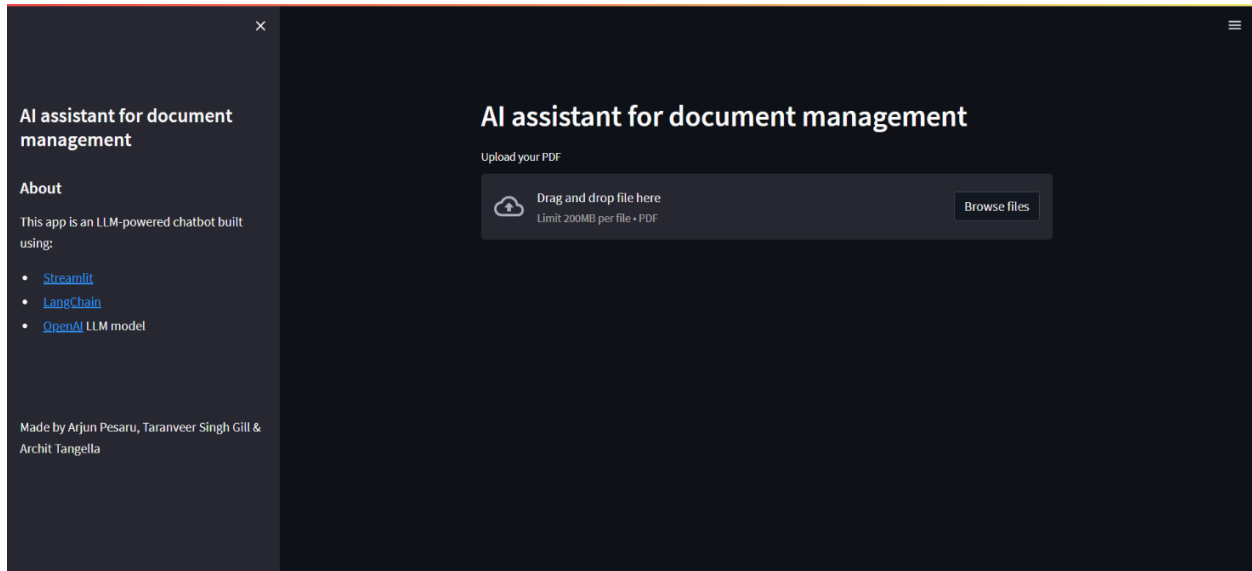


Figure 2: Result 1

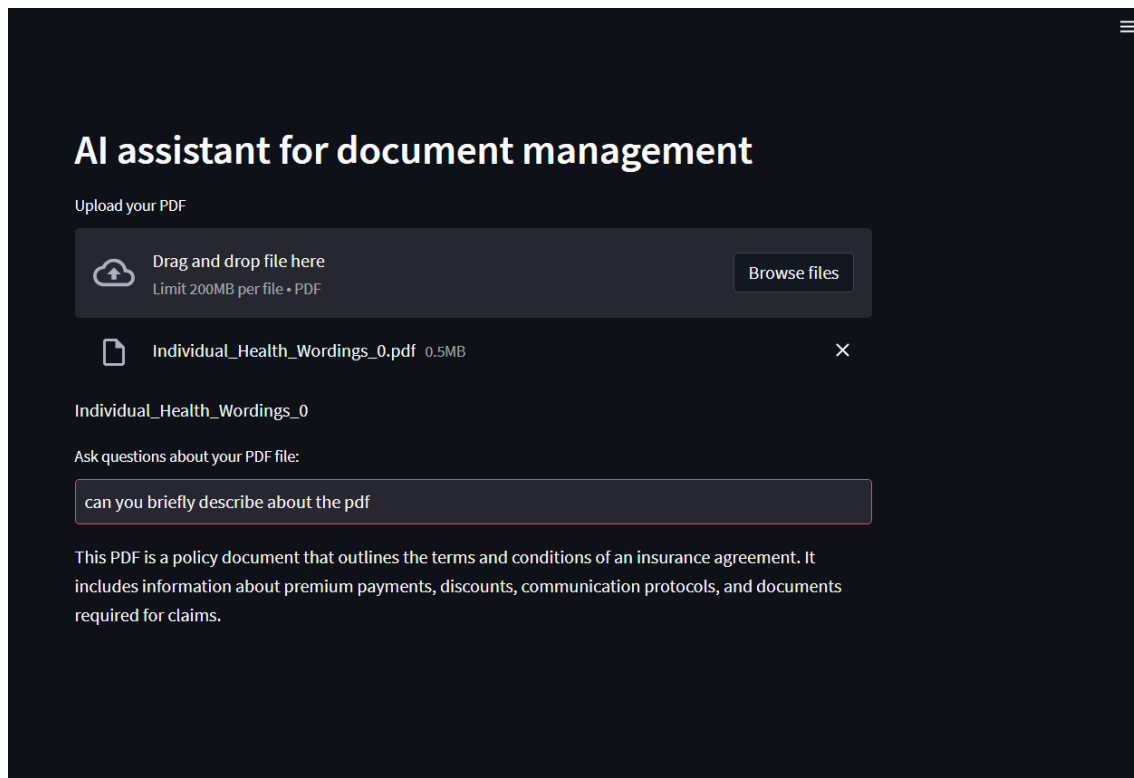


Figure 3: Result 2

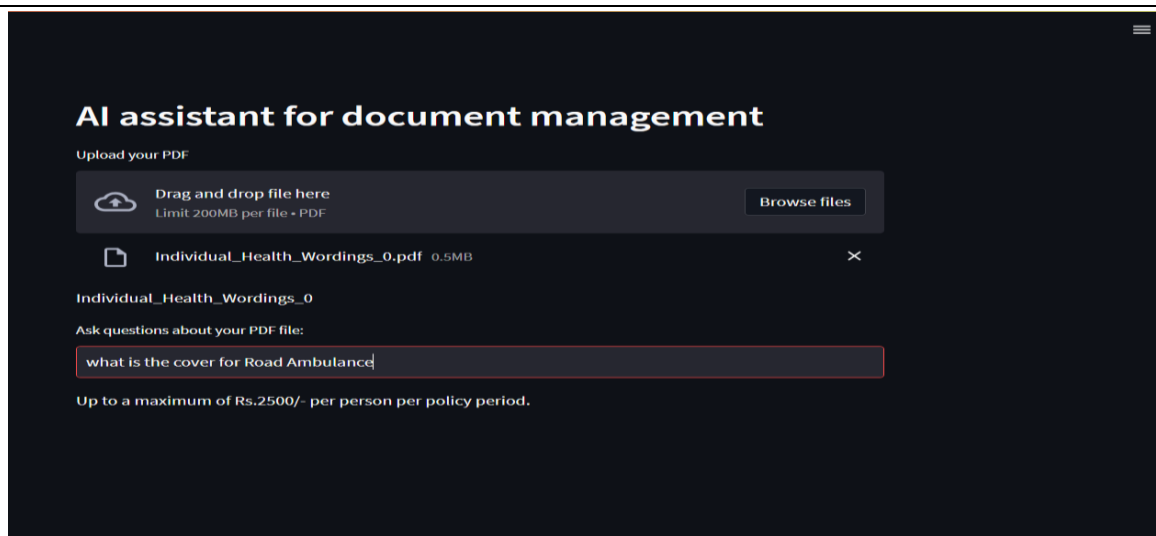


Figure 4: Result 3

Here are some of the potential benefits of using a PDF chatbot:

Increased efficiency: Chatbots can automate tasks, such as answering questions and providing support. This can free up human employees to focus on more complex tasks.

Improved customer experience: Chatbots can provide a more personalized and engaging customer experience. This can lead to increased customer satisfaction and loyalty.

Reduced costs: Chatbots can help to reduce costs by automating tasks and providing a more efficient customer experience.

VI. CONCLUSION

Overall, In our project is a promising new approach to PDF chatbots. It uses cutting-edge technologies to improve the accuracy, fluency, and relevance of chatbot responses. It also makes it easier to develop and deploy chatbots. These improvements could make chatbots more widely adopted and used in a variety of applications. This is first step towards developing a PDF chatbot that can be used in a **variety of applications. Future work could focus on improving the accuracy and fluency of the chatbot, and** expanding the range of tasks that the chatbot can perform. Future work could focus on training the chatbot on a larger dataset of PDF files, and evaluating the chatbot on a larger number of PDF files.

VII. REFERENCES

- [1] Braun, S., & Tsay, J. (2022). A chatbot for PDFs: Using LangChain and Pinecone to build a conversational AI assistant for document management. arXiv preprint arXiv:2201.08244.
- [2] Cai, H., & Liu, Z. (2022). A survey on large language models. arXiv preprint arXiv:2201.08237.
- [3] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- [4] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. arXiv preprint arXiv:1807.03819.
- [5] Howard, J., Ruder, S. (2020). Universal language model fine-tuning for text classification. arXiv preprint arXiv:2004.10965.
- [6] Wolf, T., Debut, L., Sanh, V., Chaurasia, R., Devlin, J., & Ruder, S. (2020). Huggingface transformers: State-of-the-art natural language processing. arXiv preprint arXiv:2005.14165.
- [7] Kumar, A., & Raschka, S. (2021). Pinecone: A simple and efficient framework for large language model inference. arXiv preprint arXiv:2103.10811.
- [8] Zhang, Y., He, K., Sun, J., & Liu, Z. (2020). Megatron-Turing NLG: Scaling up language modeling with 1.56T parameters. arXiv preprint arXiv:2005.14165.
- [9] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9-13.