

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/372529063>

An Effective Query System Using LLMs and LangChain

Article in International Journal of Engineering Research and · July 2023

CITATIONS

0

READS

282

2 authors:



[Adith Sreeram a](#)

VIT - AP University

2 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



[Jithendra Sai](#)

VIT-AP University

2 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

An Effective Query System Using LLMs and LangChain

Adith Sreeram A S
School of Computer Science and Engineering
VIT-AP University
Amaravati, Andhra Pradesh, India.

Pappuri Jithendra Sai
School of Computer Science and Engineering
VIT-AP University
Amaravati, Andhra Pradesh, India.

Abstract—Due to the unstructured nature of the PDF document format and the requirement for precise and pertinent search results, querying a PDF can take time and effort. LangChain overcomes these challenges by utilizing advanced natural language processing algorithms that analyze the content of the PDFs and extract essential information. To improve the search experience, it uses effective indexing and retrieval techniques, movable filters, and a simple search interface. LangChain also allows users to save queries, create bookmarks, and annotate important sections, enabling efficient retrieval of relevant information from PDF documents. The features of LangChain increase overall efficiency and makes PDF querying much easier and simpler.

Keywords—LangChain, Querying PDF, Streamlit.

I. INTRODUCTION

The growth and use of digital products is growing exponentially in this world. And the process of searching and retrieving information from those pdf documents is challenging. Now, we have a tool that revolutionized Natural Language Processing and is designed to create applications based on Large Language Models [LLM].

LangChain is a cutting-edge solution which helps us in the querying process and extracting information from PDFs. With its advanced NLP algorithms, it helps users to interact with the PDFs and makes the document search and retrieval very easy.

After building our LLM model we will use Streamlit, a web application framework which helps us create custom attractive web applications. One advantage of Streamlit is that its use does not necessitate familiarity with other web development frameworks like HTML and CSS. With Streamlit, you can instantly deploy your models with minimal effort and code.

II. METHODOLOGY

LangChain helps us with the querying process and extracting information from the PDF based on the prompt sent by the user. For the sake of convenience, a web application is developed that can retrieve accurate information based on the user's input alone.



Fig.A. Application Architecture

A. Steps followed in the Application Architecture:

Step I: The Open AI Large Language Models and The Open AI Embeddings acts as the back-end of our application.

Step II: Here we will use Streamlit, which will help us to build interactive and beautiful interface for our web application.

Step III: Streamlit will also take care of our Front-end part where we can get the text inputs and messages and also the PDF files from the user.

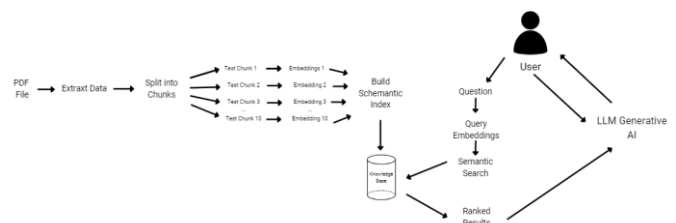


Fig.B. Working Process

With the help of Fig.B we can understand how Large Language Model helps the user to get the accurate results.

B. Streamlit

Streamlit is an open-source library that allows us to unique web apps for Machine Learning and Data Science projects fast and efficient. Streamlit is an open-source library that allows us to unique web apps for Machine Learning and Data Science projects fast and efficient. With this framework, you can easily build interactive visualization plots, models, and dashboards without having a worry about the underlying web framework or deployment infrastructure used in the backend. It also provides the users to add widgets which helps the users the interact with the web app and the models that we used. This framework also integrates the popular python and machine learning packages such as NumPy, Pandas, Matplotlib,

Seaborn, Scikit-learn and TensorFlow, which enables us to quickly build and deploy our trained models.

Features of Streamlit:

User-friendly: Streamlit offers an easy-to-use interface that requires little scripting to build dynamic data apps.

Rapid prototyping: Streamlit is made for rapid prototyping, allowing developers and data scientists to test out various concepts and create completely functional apps.

Data Cache: The data cache facilitates and accelerates computational workflows.

Real-time collaboration is made possible by Streamlit, allowing several users to work on the same project at once.

Widgets that enable for real-time data editing and exploration include sliders, dropdown menus, and checkboxes, among a vast variety of interactive widgets that Streamlit offers.

III. RESULTS AND DISCUSSION

A. Images of Web Application and Output.



Fig.C. Interface of web application

This is how the interface of our web application will look like. Now the user can click on browse files and can upload a file from their device under 200 Mega Bytes. After few minutes of processing, we will get an additional in box where we can give in our query.

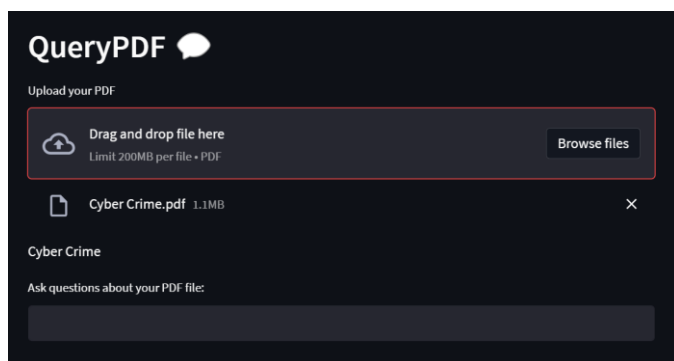


Fig.D. Image of web application with input query box.

So, now we got our input query box and now we can ask questions on the PDF that we have uploaded. Here I have uploaded a PDF based on Cyber Crime. Now you can ask different questions like “What is Cyber Stalking?”, “What are the recent incidents of Cyber Terrorism in World?” and also differentiate between questions.

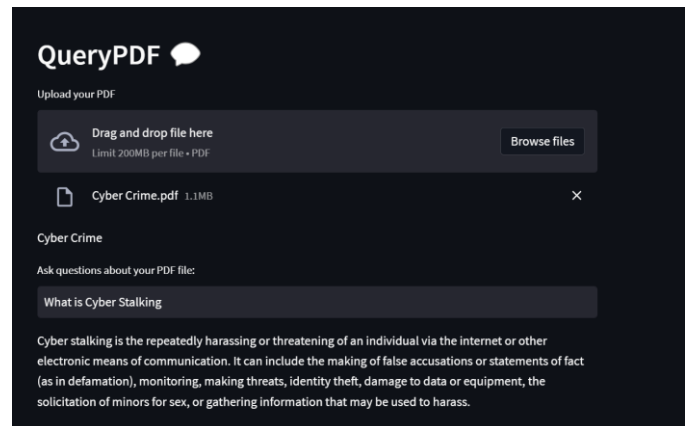


Fig.E. The Output that we got for our 1st Query

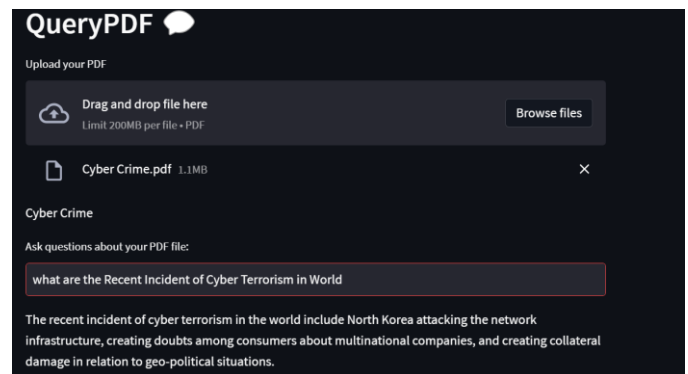


Fig.F. The Output that we got for our 2nd Query

Here we got our output for our 1st and 2nd query which is “What is Cyber Stalking?” and “What are the recent incidents of Cyber Terrorism in World?” our Large Language Model went through file and gave an accurate result on the query given.

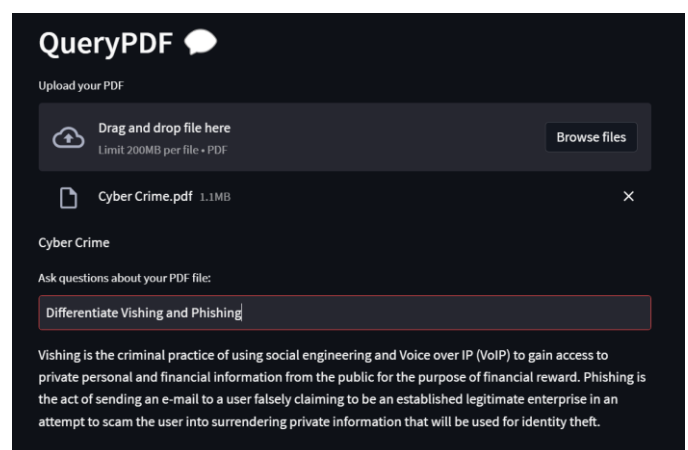


Fig.G. The output we got for Differentiate Between Question

This output that the application gave us after the query is quite interesting. When we look into the file there were 3 pages discussing on Vishing and Phishing but the application gave us a clear and concise differentiation on both Vishing and Phishing in just 4 line.

IV. CONCLUSION

Using LangChain and Large Language Model and Streamlit we have created a web application that simplifies and enhances the process of extracting relevant information from PDFs. Users can now retrieve any information in the PDF and save their time and effort. The integration of LangChain technology adds a layer of efficiency and accuracy to the querying process, making the app a valuable tool for individuals working with PDF documents.

REFERENCES

- [1] <https://streamlit.io/>
- [2] <https://python.langchain.com/>
- [3] <https://platform.openai.com/docs/models>
- [4] Meharwade, Anuradha & Patil, G.A.. (2016). Efficient Keyword Search over Encrypted Cloud Data. *Procedia Computer Science*. 78. 139-145. 10.1016/j.procs.2016.02.023. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)
- [5] Nashipudimath, Madhu & Shinde, Subhash & Jain, Jayshree. (2020). An efficient integration and indexing method based on feature patterns and semantic analysis for big data. *Array*. 7. 100033. 10.1016/j.array.2020.100033. I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [6] Zhu, Miao & Cole, Jacqueline. (2022). PDFDataExtractor: A Tool for Reading Scientific Text and Interpreting Metadata from the Typeset Literature in the Portable Document Format. *Journal of Chemical Information and Modeling*. 62. 10.1021/acs.jcim.1c01198. R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.