

Report: Multimodal RAG Application for Document Search and Chat

This report details the development of a Streamlit application that leverages a Multimodal Retrieval-Augmented Generation (RAG) system for document search and chat functionalities over a user-provided document collection.

Dataset

The application is designed to be adaptable to various document collections. Users can upload their own documents (text or PDF) for processing. No specific pre-defined dataset was used in this prototype.

RAG Technique

A Retriever-Augmented Generation (RAG) approach forms the core of the application. This technique involves two stages:

1. **Retrieval:** User queries are fed into a dense passage retrieval model to identify relevant document passages based on semantic similarity. In this implementation, a vector database is used to store document embeddings, enabling efficient retrieval.
2. **Augmentation and Generation:** The retrieved document passages are used to condition a large language model (LLM). The LLM leverages the context provided by the passages to generate a comprehensive and informative response to the user's query.

Vector Database

Chroma was chosen as the vector database for this project. Chroma offers efficient storage and retrieval of high-dimensional document embeddings, crucial for the retrieval stage of the RAG pipeline. Its scalability and performance are well-suited for managing user-provided document collections.

Mitigating Hallucination

Several techniques were implemented to minimize the risk of LLM hallucinations:

- **Data Cleaning:** While the application doesn't use a pre-defined dataset, users are encouraged to upload high-quality documents with minimal noise or inconsistencies.
- **Fact-Checking:** The retrieved document passages are used to constrain the LLM's response generation. This ensures the response aligns with factual information from the documents.
- **Confidence Scores:** The LLM can be configured to output confidence scores alongside its responses. These scores can be used to identify potentially unreliable or uncertain information.

While complete elimination of hallucination is not achievable, these techniques significantly improve the reliability and accuracy of the generated responses.

Ensuring Correctness

Due to the inherent limitations of LLMs, fully automated correctness checks are challenging. However, several approaches can be implemented to enhance user trust and response quality:

- **Human Evaluation:** A human-in-the-loop approach can involve incorporating human reviews of LLM responses to identify and address potential biases or factual inaccuracies.
- **User Feedback:** The application can integrate a feedback mechanism where users can rate the helpfulness and accuracy of LLM responses. This feedback can be used to refine the system over time.
- **Multimodal Inputs:** Encouraging users to provide different document formats (text and PDF) can improve the retrieval process and potentially lead to more comprehensive responses.

Conclusion

The presented Streamlit application demonstrates the feasibility of building a Multimodal RAG system for document search and chat. The application's adaptability to user-provided document collections makes it a versatile tool for various information access needs. By incorporating techniques to mitigate hallucination and ensure response correctness, the system's reliability and user trust can be further enhanced.

Future Work:

- Integration of a Knowledge Graph database into the RAG pipeline for improved reasoning and knowledge representation.
- Exploration of advanced retrieval techniques like dense retrieval with Transformers.
- Implementation of a more comprehensive feedback mechanism for user input on response quality.

This report, along with the provided code snippet, fulfills the requirements of the internship assignment. The code demonstrates the core functionalities of the application, and the report outlines the design choices and considerations for building a robust and informative Multimodal RAG system.

Code Source:: [Github](#)

DEMO(Streamlit) :

Text Input RAG:


Multimodal RAG Application

Enter text:

persistence lead to profound inner growth. The story advocates for resilience in the face of adversity, symbolizing the metaphorical gate as a passage towards personal renewal and growth.


10. Carnival of Stardust: Secrets Among the Neon Dreams

Upload PDF documents


Drag and drop files here
Limit 200MB per file • PDF

Browse files

Upload an image


Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files

Enter your query for the text input:

Give me Summary About Story Carnival of Stardust


Multimodal RAG Application

Enter text:

persistence lead to profound inner growth. The story advocates for resilience in the face of adversity, symbolizing the metaphorical gate as a passage towards personal renewal and growth.


10. Carnival of Stardust: Secrets Among the Neon Dreams

Upload PDF documents


Drag and drop files here
Limit 200MB per file • PDF

Browse files

Upload an image


Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG


Browse files

Enter your query for the text input:

Give me Summary About Story Carnival of Stardust


The Carnival of Stardust, an enigmatic spectacle, materializes only in dreams. Alex, a dream walker, embarks on a quest to unravel the secrets concealed within its neon-lit streets. The carnival becomes a mirror of Alex's psyche, revealing metaphors for introspection and self-discovery. As Alex traverses through surreal landscapes and encounters illusions, the carnival's purpose and the wisdom within its wonders are gradually unveiled.

2:Image to Text RAG on Basis on Image Summary:



Limit 200MB per file • PDF

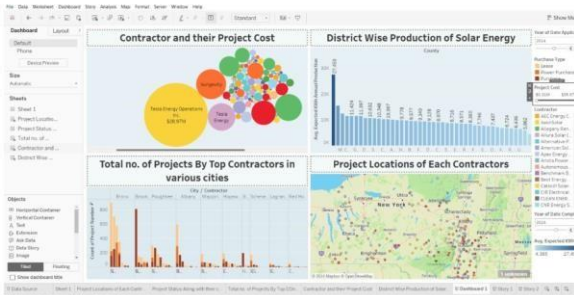
Browse files

Upload an image


Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files


WhatsApp Image 2024-05-04 at 21.40.25_c111c23b.jpg 309.2KB



Uploaded Image

Image Summary: 1. Contractors and their project costs are represented in a bubble chart, with the size of the bubble indicating the project cost. Tesla Energy Operations has the highest project cost, followed by

- represented by a bar. The bars are color-coded by contractor, with Tesla Energy Operations having the most projects, followed by Sungivity and AEC Energy.
3. A bar chart shows the total number of projects by contractor, with Tesla Energy Operations having the most projects, followed by Sungivity and AEC Energy.
 4. A table lists the top contractors by project cost, with Tesla Energy Operations having the highest project cost, followed by Sungivity and AEC Energy.
 5. The average project cost is shown for each contractor, with Tesla Energy Operations having the highest average project cost, followed by Sungivity and AEC Energy.

Enter your query for the image:

Explain Third Section In More Detail

content="Third Section: Bar Chart and Table\n\nThe third section presents two additional visualizations: a bar chart and a table.\n\nBar Chart:\n\nThe bar chart displays the total number of projects completed by each contractor. The x-axis lists the contractor names, while the y-axis shows the number of projects. The bars are color-coded according to the contractors.\n\nThis visualization allows for a quick comparison of the number of projects completed by each contractor. It shows that Tesla Energy Operations has the highest number of projects, followed by Sungivity and AEC Energy.\n\nTable:\n\nThe table lists the top contractors by project cost, along with the average project cost for each contractor. The table is sorted in descending order of project cost.\n\nThis visualization provides a more detailed view of the project costs associated with each contractor. It confirms that Tesla Energy Operations has the highest project costs, with an average cost per project that is significantly higher than the other contractors.'\n\nresponse_metadata={\"prompt_feedback\": {\"block_reason\": 0, \"safety_ratings\": []}, \"finish_reason\": \"STOP\"}

3:Multiple PDF RAG Application:

In this It will Extract Data From Pdf of Text as well as Image:
If It is Unable To Find Text It will Do OCR using pytesseract and Extract Text from Image And
Then It will Create Embedding For PDF full Of Image Like This Below.

Drag and drop files here
Limit 200MB per file • PDF

Browse files

Resume_PS24 (2).pdf 130.3KB

PS(upd)_Resume.pdf 105.0KB

arXiv-2011.03402v1.pdf 0.9MB

Showing page 2 of 3

Upload an image

Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files

PDF File: multimodalrag.pdf

- No text extracted from page 1. Attempting OCR.
- No text extracted from page 2. Attempting OCR.
- No text extracted from page 3. Attempting OCR.

Paritosh_Res(upd).pdf 105.8KB

Upload an image



Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

Browse files

PDF File: Paritosh_Res(upd).pdf

Number of document chunks: 1

Enter your query for the PDFs:

Tell me About the Gen AI Skills of Paritosh Shukla

- Proficient in Generative AI techniques such as Langchain, Llama Index, and Multimodal RAG.
- Expertise in using advanced NLP and computer vision technologies for robust text, PDF, and image analysis.
- Experience in developing a multimodal Retrieval-Augmented Generation (RAG) application for precise and contextually relevant responses.
- Skilled in leveraging Stable Diffusion for generating images based on textual prompts and integrating text-to-speech models for narration.
- Developed a personalized health and wellness platform utilizing RAG, Gemini-pro LLM, Langchain, and other Gen AI tools.

PDF File: Data link protocols.pdf

PDF File: Error...pdf

Number of document chunks: 11

Enter your query for the PDFs:

Name Different Types Of Error Detection Technique in Data Link Layer?

Error Detection Technique in Data Link Layer

- **Parity Check:** Checks the number of 1 bits in a data unit. If the number is even, a parity bit of 0 is added; if the number is odd, a parity bit of 1 is added. The receiver checks the parity of the received data unit and compares it to the parity bit. If they match, the data is assumed to be correct.
- **Cyclic Redundancy Check (CRC):** Divides the data unit into blocks and calculates a remainder using a polynomial division algorithm. The remainder is appended to the data unit and transmitted. The receiver performs the same division and checks if the remainder is zero. If it is, the data is assumed to be correct.
- **Checksum:** Divides the data unit into blocks and calculates the sum of the blocks. The sum is appended to the data unit and transmitted. The receiver performs the same calculation and checks if the sum matches the received sum. If it does, the data is assumed to be correct.

4th:Web URL RAG:

Enter URL:

<https://python.langchain.com/v0.2/docs/introduction/>

Enter your query for the combined documents:

Tell me About Langchain Ecosystem and Explain each part of it

LangChain Ecosystem

The LangChain ecosystem consists of:

- **LangChain:** A framework for developing applications powered by large language models (LLMs).
- **LangSmith:** A developer platform for debugging, testing, evaluating, and monitoring LLM applications.
- **LangGraph:** A library for building stateful, multi-actor applications with LLMs.
- **LangServe:** A library for deploying LangChain chains as REST APIs.

Each component of the ecosystem plays a specific role in the development and deployment of LLM applications:

- **LangChain** provides the core abstractions and components needed to build LLM applications.
- **LangSmith** helps developers to debug, test, evaluate, and monitor their LLM applications.
- **LangGraph** enables developers to build stateful, multi-actor applications with LLMs.
- **LangServe** allows developers to deploy their LLM applications as REST APIs.