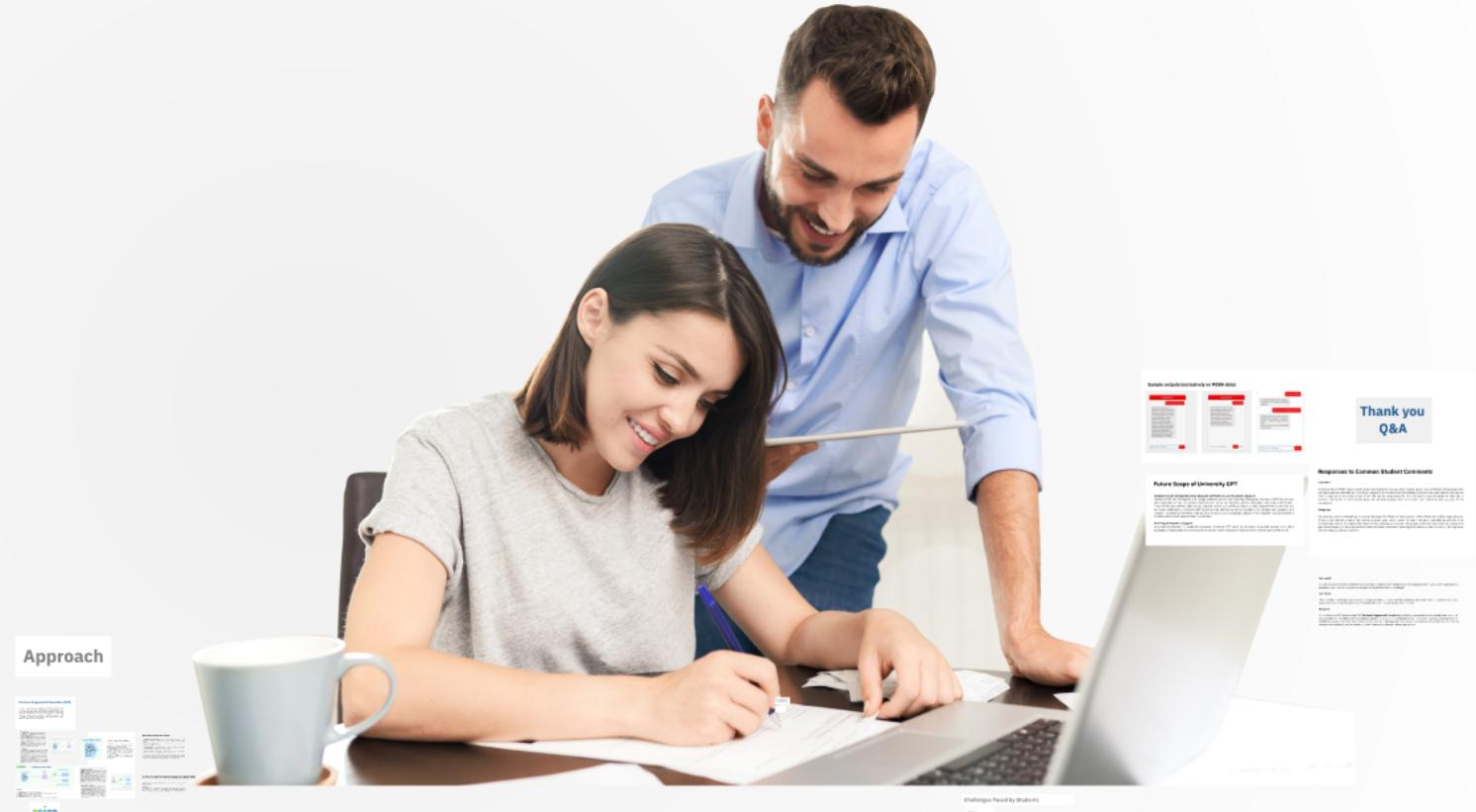


UNIVERSITY GPT

YOUR CAMPUS
ASSISTANT, ANYTIME,
ANYWHERE

Shravya Yalaka
Ramu Gara



INTRODUCTION

UNIVERSITY GPT serves as a virtual assistant, streamlining the process of accessing information about classes, professors, and university policies. It eliminates the frustration and inefficiency of navigating complex university websites, allowing students to focus on their academic and personal priorities.

The primary users of UNIVERSITY GPT are students across diverse academic backgrounds. By addressing their needs for quick and reliable information, the tool is tailored to enhance the academic efficiency of undergraduate and graduate students alike.

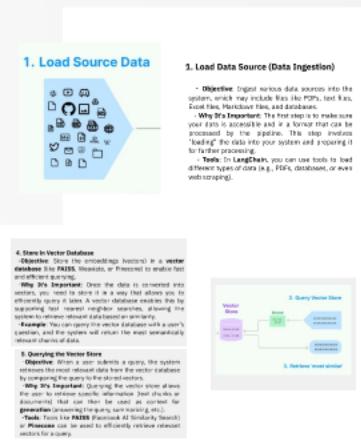
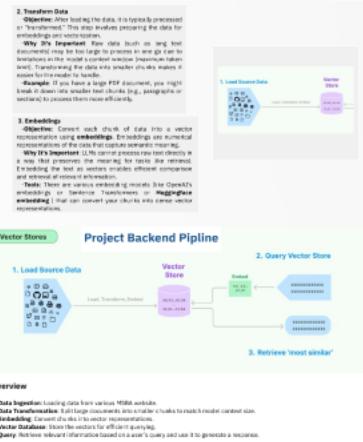
Challanges Faced By Students

- Navigating Complex Websites
- Time Consumption
- Information Overload
- Impact on Academic Priorities

Approach

Retrieval Augmented Generation (RAG)

LLMs can reason about wide-ranging topics, but their knowledge is limited to the public data up to a specific point in time that they were trained on. If you need AI applications to reason about new information or data introduced after the LLM's cutoff date, you need to augment the knowledge of the model with the specific information it needs. The process of bringing the appropriate information and inserting it into the model prompt is known as Retrieval-Augmented Generation (RAG).



Why Choose RAG for Your Project?

Handles Large Knowledge Bases: RAG can process vast datasets that wouldn't fit in the memory of a typical LLM, leveraging external sources to provide up-to-date, expansive knowledge.

Easy Knowledge Access: RAG retrieves live data from external sources, ensuring users always have access to the latest information.

Improves Accuracy: By combining generation and retrieval, RAG retrieves relevant documents and ensures that responses are factually accurate.

Flexible and Scalable: As new data sources are added, RAG can integrate them seamlessly into the pipeline without retraining, making it highly adaptable to evolving requirements.

In summary, RAG provides a reliable, scalable, and efficient system for generating responses, improved accuracy, and easy integration, making it a great choice for projects that require handling large and diverse data sources while ensuring reliable, up-to-date responses.

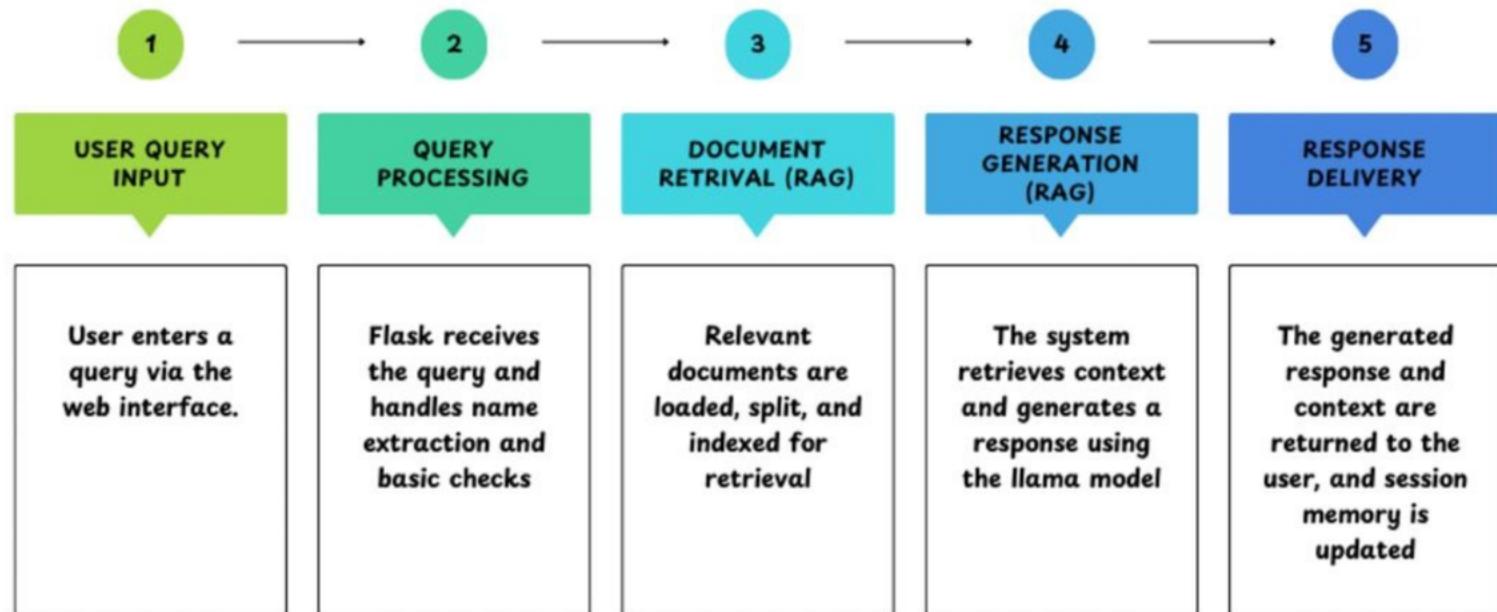
6. Pass to LLM for Natural Language Generation

Objective: Use the retrieved context to generate human-readable, cohesive, and conversational responses or outputs.

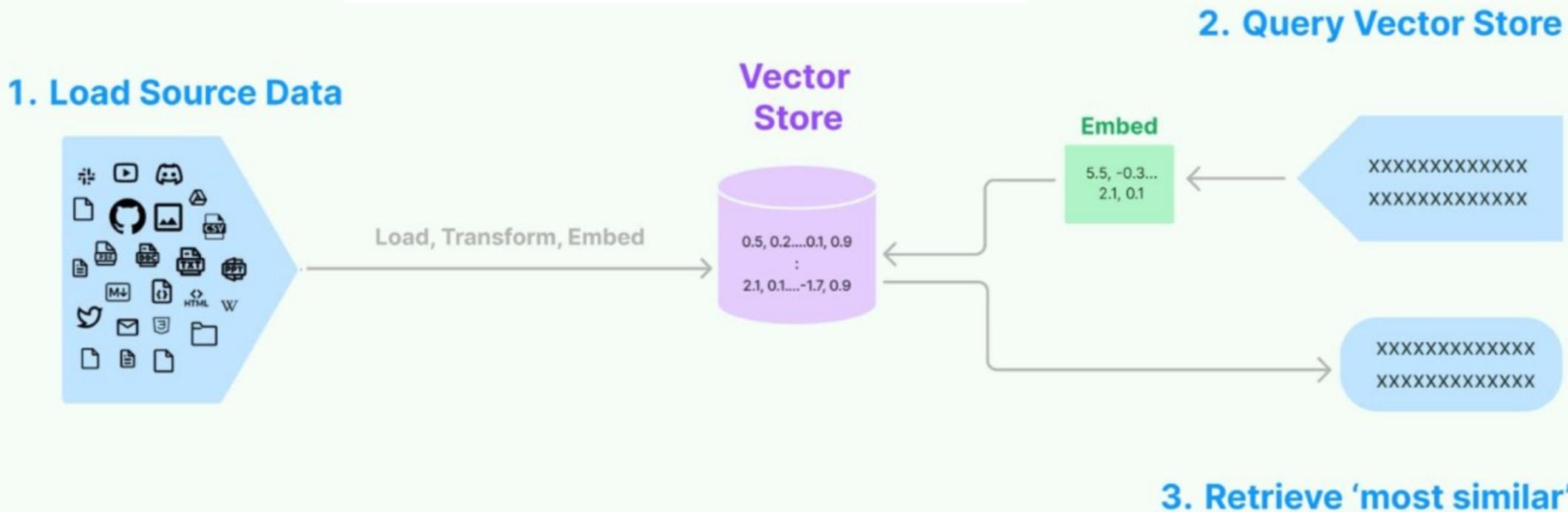
Description: After querying the vector database and retrieving the most relevant documents or information, the system sends the data to context to the Large Language Model (LLM). The LLM processes the context, integrates it with the query, and generates a natural language response or answer, suitable for the given task (e.g., question answering, summarization, etc.).



Flow



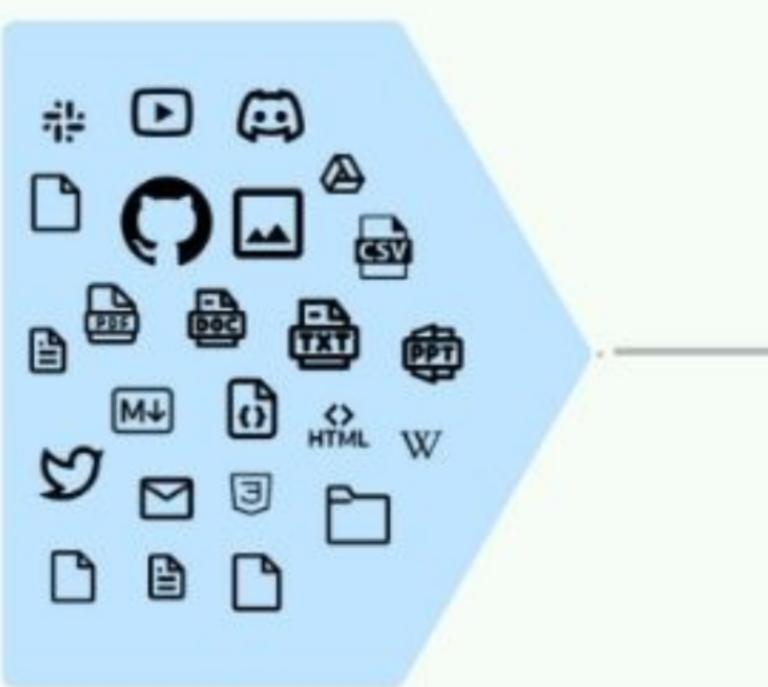
Project Backend Pipeline



Overview

- 1. Data Ingestion:** Loading data from various MSBA website.
- 2. Data Transformation:** Split large documents into smaller chunks to match model context size.
- 3. Embedding:** Convert chunks into vector representations.
- 4. Vector Database:** Store the vectors for efficient querying.
- 5. Query:** Retrieve relevant information based on a user's query and use it to generate a response.

1. Load Source Data



1. Load Data Source (Data Ingestion)

- **Objective:** Ingest various data sources into the system, which may include files like PDFs, text files, Excel files, Markdown files, and databases.
- **Why It's Important:** The first step is to make sure your data is accessible and in a format that can be processed by the pipeline. This step involves "loading" the data into your system and preparing it for further processing.
- **Tools:** In **LangChain**, you can use tools to load different types of data (e.g., PDFs, databases, or even web scraping).

2. Transform Data

-Objective: After loading the data, it is typically processed or "transformed." This step involves preparing the data for embeddings and vectorization.

-Why It's Important: Raw data (such as long text documents) may be too large to process in one go due to limitations in the model's context window (maximum token limit). Transforming the data into smaller chunks makes it easier for the model to handle.

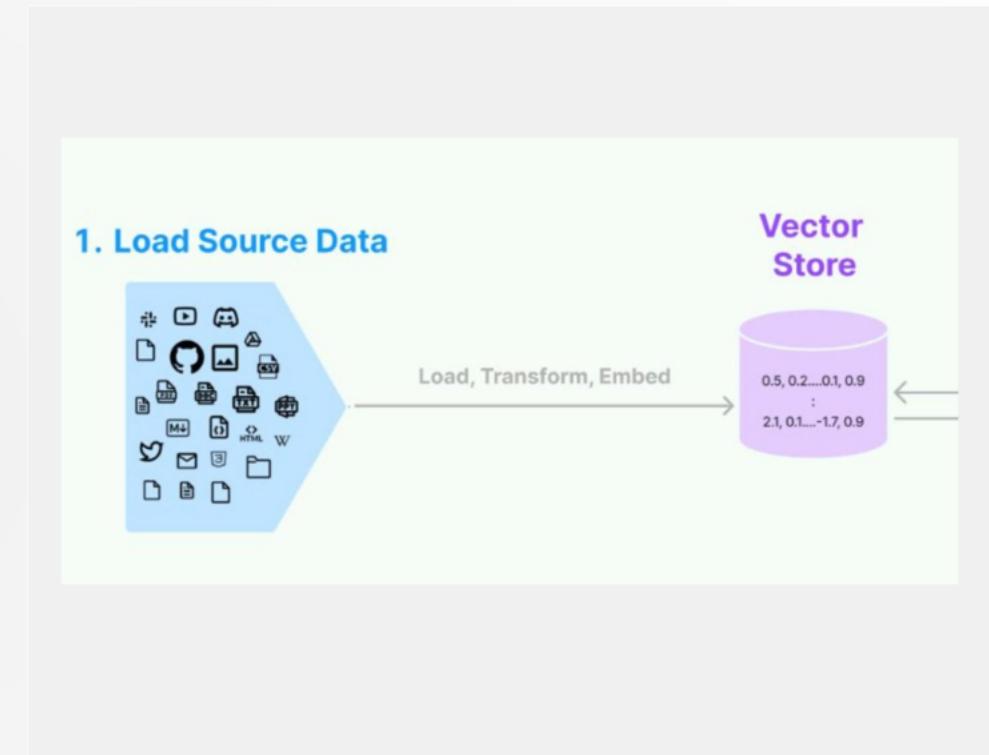
-Example: If you have a large PDF document, you might break it down into smaller text chunks (e.g., paragraphs or sections) to process them more efficiently.

3. Embeddings

-Objective: Convert each chunk of data into a vector representation using **embeddings**. Embeddings are numerical representations of the data that capture semantic meaning.

-Why It's Important: LLMs cannot process raw text directly in a way that preserves the meaning for tasks like retrieval. Embedding the text as vectors enables efficient comparison and retrieval of relevant information.

-Tools: There are various embedding models (like OpenAI's embeddings or Sentence Transformers or **Huggingface embedding**) that can convert your chunks into dense vector representations.



4. Store in Vector Database

-Objective: Store the embeddings (vectors) in a **vector database** (like **FAISS**, Weaviate, or Pinecone) to enable fast and efficient querying.

-Why It's Important: Once the data is converted into vectors, you need to store it in a way that allows you to efficiently query it later. A vector database enables this by supporting fast nearest neighbor searches, allowing the system to retrieve relevant data based on similarity.

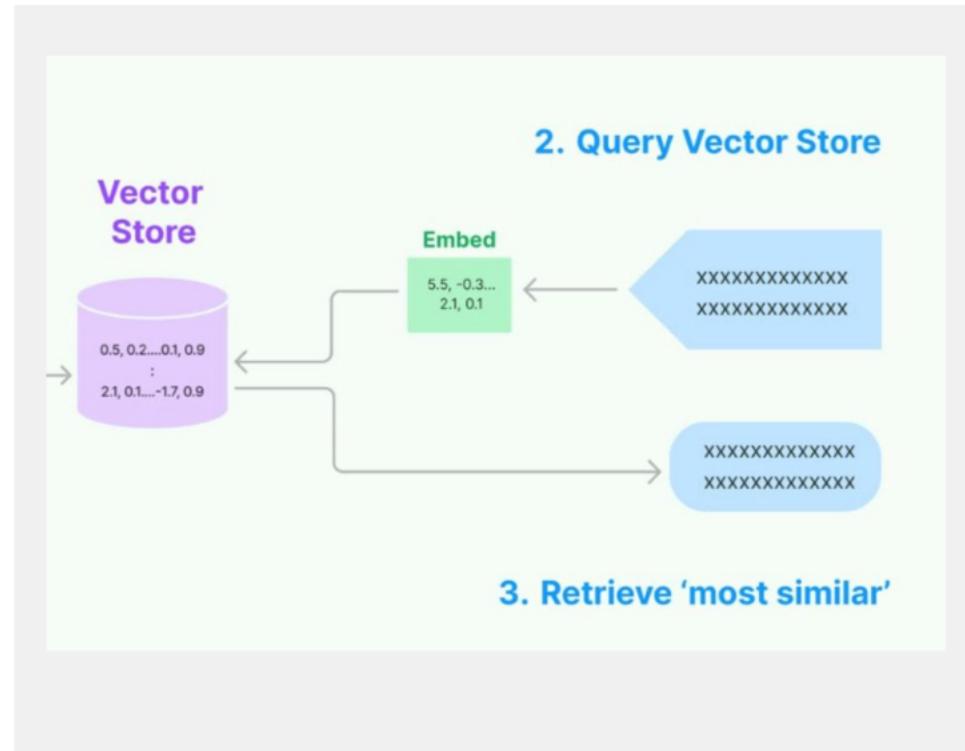
-Example: You can query the vector database with a user's question, and the system will return the most semantically relevant chunks of data.

5. Querying the Vector Store

-Objective: When a user submits a query, the system retrieves the most relevant data from the vector database by comparing the query to the stored vectors.

-Why It's Important: Querying the vector store allows the user to retrieve specific information (text chunks or documents) that can then be used as context for **generation** (answering the query, summarizing, etc.).

-Tools: Tools like **FAISS** (Facebook AI Similarity Search) or **Pinecone** can be used to efficiently retrieve relevant vectors for a query.



6. Pass to LLM for Natural Language Generation

Objective: Use the retrieved context to generate human-readable, coherent, and contextually relevant responses or outputs.

Description:

After querying the vector database and retrieving the most relevant documents or information, the system passes this data as context to the Large Language Model (LLM **Example - llama**). The LLM processes the context, integrates it with the query, and generates a natural language response or answer, suitable for the given task (e.g., question answering, summarization, etc.).

Retrieval Augmented Generation (RAG)

LLMs can reason about wide-ranging topics, but their knowledge is limited to the public data up to a specific point in time that they were trained on. If you want to build AI applications that can reason about private data or data introduced after a model's cutoff date, you need to augment the knowledge of the model with the specific information it needs. The process of bringing the appropriate information and inserting it into the model prompt is known as Retrieval Augmented Generation (RAG)

Why Choose RAG for Your Project?

- **Handles Large Knowledge Bases:** RAG can process vast datasets that wouldn't fit in the memory of a typical LLM, leveraging external sources to provide up-to-date, expansive knowledge.
- **Dynamic Knowledge Access:** RAG retrieves live data from external sources, ensuring responses are both current and contextually relevant.
- **Improves Accuracy:** By grounding generation in retrieved information, RAG minimizes hallucinations and ensures that responses are factually accurate.
- **Flexible and Scalable:** As new data sources are added, RAG can integrate them seamlessly into the pipeline without retraining, making it highly adaptable to evolving datasets.

In summary, RAG provides a unique combination of dynamic knowledge access, improved accuracy, scalability, and reduced hallucination, making it an ideal choice for projects that require handling large and diverse data sources while ensuring reliable, up-to-date responses

Sample outputs (exclusively on MSBA data)

University GPT

what are program objectives

The program objectives are: 1. Students who graduate will be able to effectively use advanced big data and analytics technologies for extracting and managing large amounts of data and building analytical models. 2. Students who graduate will be effective in applying quantitative methods and tools of business analytics. 3. Students who graduate will be able to propose and evaluate alternatives for making data-driven, effective business decisions. 4. Students who graduate will be able to write effectively to convey project ideas, methodology, and findings.

Type your message... Send

University GPT

hi I'm shopping

It seems like you may have reached the wrong page. This appears to be a page for the Master of Science in Business Analytics program at California State University, East Bay, with information on helpful links for current students and job positions. If you're looking for shopping information, you might want to try a different website or search engine.

Type your message... Send Microphone

who is Dr Sarnikar

Dr. Surendra Sarnikar is the Program Coordinator for the Master of Business Analytics.

Office hours of Professor Inkyu Kim

Professor Inkyu Kim's office hours for Fall 2024 are Tuesdays or Thursdays 11:00am - 12:00pm by appointment on Zoom (<https://csueb.zoom.us/j/7744085903>, passcode: aa).

Type your message... Send

Future Scope of University GPT

Integration with College Websites, Educational Platforms, and Academic Support:

University GPT could integrate with college websites, portals, and Learning Management Systems (LMS) like Canvas, allowing students to quickly access resources such as course materials, grades, schedules, and campus information. It could offer personalized logins to stay updated on events, book library books, schedule appointments, send emails, and more. Additionally, University GPT would provide tailored content and guidance for assignments, projects, and research, suggesting references, step-by-step solutions, and conceptual support. This integration would streamline student tasks and enhance academic workflows.

Multilingual Academic Support:

As global collaboration in academia increases, University GPT could be enhanced to provide support in multiple languages, helping students and researchers communicate and access resources from different parts of the world.

Responses to Common Student Comments

Comment

University AI CHATGPT sounds exciting and impressive! I have a question regarding your use of FAISS or Elasticsearch for storing document embeddings. How do you ensure that the vector search efficiently retrieves the most relevant documents from a large set of university-related data? Will you be using keywords from the user's query alongside embeddings to retrieve information, or how exactly does the retrieval process work to maintain both relevancy and accuracy in the responses?

Response

We store document embeddings in a vector database like FAISS or Elasticsearch, which efficiently handles large datasets. When a user submits a query, the retrieval process uses vector search to match the query embeddings with the most relevant documents. This allows the model to find contextually relevant information, even from vast amounts of data. The generated response is then grounded in these retrieved documents, ensuring both relevancy and accuracy in the response, without relying solely on keywords.

Comment1

How do you plan to handle potential inconsistencies or outdated information when retrieving data for University GPT, especially in a dynamic environment where policies, courses, and deadlines frequently change?

Comment2

This is a really interesting study. I was wondering, how will you ensure that the chatbot provides real-time and accurate responses, especially considering that university information like deadlines or policies might change?

Response

In the University GPT project using RAG (**Retrieval-Augmented Generation**), handling inconsistencies or outdated information is a key consideration. To address this, we regularly update the document embeddings stored in the vector database, ensuring that the model has access to the most recent and accurate data. By integrating real-time data sources (such as APIs or live feeds for policies, courses, and deadlines), we can ensure that the information retrieved is always up-to-date

Thank you
Q&A

UNIVERSITY GPT

YOUR CAMPUS
ASSISTANT, ANYTIME,
ANYWHERE

Shravya Yalaka
Ramu Gara

