Test Task: Building a Multimodal RAG System

Task Overview

Your task is to design and implement a **Multimodal Retrieval-Augmented Generation (RAG) System** that retrieves relevant news articles from The Batch, incorporating both textual and visual data. The system should allow users to input queries, answer questions, retrieve relevant articles, and display any associated media (such as images). The focus is on developing a practical solution that integrates different types of data efficiently and presents the results in a user-friendly format.

Requirements

You are required to build an end-to-end RAG system with the following features and outputs:

1. **Data Ingestion and Preprocessing**
    a. Retrieve news articles and their associated media from The Batch and prepare the data for use in your RAG system.
2. **Multimodal Database Creation**
    a. Store and index text and image data to enable efficient retrieval based on user queries.
3. **Query System for Retrieval**
    a. Implement a user interface where users can input queries and receive answers and relevant articles, including text and associated images.
4. **Multimodal Integration**
    a. Integrate textual and visual data to improve the quality and relevance of the retrieval results.
    b. We recommend using GPTs, Claude, or Gemini LLM model.
5. **System Evaluation**
    a. Evaluate the performance of your RAG system and explain how well it retrieves relevant content based on user queries.
6. **Simple UI for Testing and Demo**
    a. Create a simple user interface for testing and showcasing your RAG system. You can use Streamlit or a similar tool to build a clean, interactive UI that allows users to input queries and view the retrieved articles and images.
    b. **Requirement**: The UI should allow users to input a query, trigger the retrieval process, and display the retrieved news articles alongside any

relevant images. It should also be simple and intuitive to use.

7. **Optional Deployment**
    a. Optionally, the system can be deployed on a cloud platform for scalability and accessibility.

Final Submission

- **Documentation**: Provide a report detailing your approach, the tools and models you selected, and the reasoning behind your decisions.
- **Codebase for Submission:** You must submit the complete **codebase** for your RAG system, ensuring it can be easily set up and tested.
    - **Requirements**:
        - Ensure the codebase is organized and includes clear documentation (README file) with instructions for setting up the system, installing dependencies, and running the application.
        - If a specific environment or libraries are required, provide environment setup instructions or a requirements.txt file (or equivalent for your chosen platform).
        - The codebase should include all relevant scripts (data ingestion, processing, UI, etc.) and be structured for easy review and testing by others.
- **Demo**: Include a video with a demo showcasing how users interact with your system, using the simple UI to input queries and view the results.