DeepDive AI

In-depth insights from AI Research papers

Problem Statement

Develop a **RAG (Retrieval-Augmented Generation)** application that allows users to input queries related to a given AI research paper. The application should retrieve relevant sections from the paper and generate concise, context-aware answers.

Project Description

The aim is to build an application that facilitates understanding and extracting insights from complex AI research papers. Users can upload a research paper, ask questions, and get precise answers based on the content of the paper. This project focuses on enabling researchers, students, and enthusiasts to navigate and comprehend scientific literature efficiently.

Key Features

1. **Paper Upload**:
    - Allow users to upload AI research papers in PDF format.
2. **Query-Based Retrieval**:
    - Implement a semantic search to retrieve specific sections or paragraphs related to user queries.
3. **Summary Generation**:
    - Generate concise summaries of key sections (e.g., abstract, methodology, results).
4. **Interactive Q&A**:
    - Provide users with answers to natural language questions based on the paper's content.
5. **Citation Assistance**:
    - Offer citation suggestions for specific sections or ideas in the paper.
6. **Multi-Paper Support** (optional):
    - Allow users to query across multiple research papers.

Tech Stack

- **Text Extraction**:
  - PyPDF2, PDFMiner, or Tesseract (if OCR is required).
- **Vectorization**:
  - Sentence Transformers (`all-MiniLM-L6-v2` or similar).
- **Vector Database**:
  - Pinecone, FAISS, or Weaviate.
- **LLM Integration**:
  - OpenAI GPT models or Hugging Face models.
- **Frontend**:
  - Streamlit, Flask, or FastAPI.
- **Deployment**:
  - Streamlit Cloud, Hugging Face Spaces, or AWS.

Steps to Build

1. Data Handling

- Extract text from research papers, ensuring proper handling of sections like:
  - Abstract
  - Introduction
  - Methodology
  - Results and Discussion
  - Conclusion
- Handle multi-column layouts and citations.

2. Data Preprocessing

- Clean and tokenize the extracted text.
- Chunk the content into manageable sections (200-500 words).
- Add metadata such as section headings and page numbers.

3. Embedding Creation

- Use sentence embeddings to create vector representations of the chunks.
- Store embeddings in a vector database for efficient retrieval.

4. Query Processing

- Accept user queries in natural language.

- Perform semantic search in the vector database to retrieve relevant chunks.
- Pass the retrieved chunks to the LLM for generating detailed responses.

5. Frontend Development

- Build a user-friendly interface where users can:
    - Upload research papers.
    - Input queries.
    - View retrieved sections and AI-generated responses.

6. Evaluation

- Test the application with various research papers to ensure accurate retrieval and generation.

Example Use Case

Input:

> *"What is the main contribution of the paper?"*

Process:

- Retrieve relevant sections from the paper's abstract and conclusion.
- Use an LLM to generate a response combining retrieved information.

Output:

> *"The main contribution of the paper is the introduction of a novel transformer-based architecture that improves model efficiency by 25% while maintaining state-of-the-art performance on benchmark datasets."*

Few Research Paper Links

1) Attention Is All You Need - **Link**
2) DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning - **Link**
3) GPT-4 Technical Report - **Link**
4) The Llama 3 Herd of Models - **Link**
5) Gemini: A Family of Highly Capable Multimodal Models - **Link**

Project Deliverables

- **RAG Application**:
    - Develop the solution using Streamlit, Flask, or any platform of your choice based on convenience and compatibility.
- **Documentation**:
    - Include steps for data preparation, model training, and deployment.
- **Demo**:
    - Host the application on a platform like Streamlit Cloud, Hugging Face Spaces, or Heroku for live testing.
- **Evaluation**:
    - Test accuracy by comparing responses to the Research Paper.