DeepDive AI - PaperInsight

1. Project Overview

- **Objective:** To develop a Retrieval-Augmented Generation (RAG) application that allows users to input queries related to AI research papers and receive concise, context-aware answers.

2. Problem Statement

- The complexity of AI research papers makes it challenging for researchers, students, and enthusiasts to extract meaningful insights.
- There is a need for an application that can facilitate understanding and provide quick answers based on the content of these papers.

3. Project Description

- The application enables users to upload AI research papers in PDF format, ask questions, and receive precise answers based on the paper's content.
- Key features include:
    - Paper Upload
    - Query-Based Retrieval
    - Summary Generation
    - Interactive Q&A
    - Citation Assistance
    - Multi-Paper Support (optional)

4. Key Features

- **Paper Upload:** Users can upload AI research papers in PDF format.
- **Query-Based Retrieval:** Implement semantic search to retrieve specific sections related to user queries.

- **Summary Generation:** Generate concise summaries of key sections (e.g., abstract, methodology, results).
- **Interactive Q&A:** Provide answers to natural language questions based on the paper's content.
- **Citation Assistance:** Offer citation suggestions for specific sections or ideas in the paper.
- **Multi-Paper Support:** Allow users to query across multiple research papers (optional feature).

5. 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 or Google GeminiAi,
- **Frontend:** Streamlit, Flask, or FastAPI.
- **Deployment:** Streamlit Cloud, Hugging Face Spaces, or AWS.

6. Steps to Build

1. **Data Handling:**
    - Extract text from research papers, ensuring proper handling of sections like Abstract, Introduction, Methodology, Results, and 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 for uploading research papers, inputting queries, and viewing retrieved sections and AI-generated responses.
6. **Evaluation:**
    - Test the application with various research papers to ensure accurate retrieval and generation.

7. 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."

8. Project Deliverables

- **RAG Application:** Developed using Streamlit, Flask, or any platform of choice based on convenience and compatibility.

- **Documentation:** Steps for data preparation, model training, and deployment.
- **Demo:** Hosted 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.

9. Conclusion

The DeepDive AI - PaperInsight application aims to bridge the gap between complex AI research papers and users seeking insights. By leveraging advanced text extraction and natural language processing techniques, the application provides a valuable tool for researchers and students to navigate scientific literature efficiently. Future enhancements may include multi-paper support and improved citation assistance features.