Project Group - 12

Intelligent Document Assistant: A RAG Approach for Conversational Knowledge Access

1. Project Objective/Idea

In today's information-rich environment, users often struggle to extract relevant knowledge from large texts, such as textbooks or research articles. This project aims to develop an intelligent chatbot interface that allows users to upload lengthy documents and engage with the content through natural language queries. The model will parse the uploaded documents, create a knowledge base using a vector database, and respond to user questions by fetching relevant information from this knowledge base. This will enhance learning and provide instant access to information, ultimately improving user engagement and knowledge acquisition.

2. Approach to Solution

To achieve the project's objectives, we will implement the following approach:

- **Document Parsing**: We will use text parsing libraries (e.g., `spaCy`, `PyPDF2`) to preprocess the uploaded documents, extracting relevant text while removing unnecessary metadata.
- **Knowledge Base Creation**:
    - We will create embeddings of the parsed text using pre-trained language models (Hugging face transformers).
    - A vector database (e.g., FAISS, Pinecone, ChromaDB) will be employed to store these embeddings, enabling efficient retrieval based on semantic similarity.
- **Chatbot Interface**:
    - A user-friendly chatbot interface will be built using the Django Framework.
    - For response generation, we will leverage a pre-trained LLM via APIs (Hugging Face Transformers) that uses the relevant chunks of text retrieved from the vector database and generates coherent responses.

3. Data Overview

1. Text Documents/Textbooks

- **Types of Documents**: The primary data source will be various types of educational materials, including:
    - **Textbooks**: Comprehensive books covering specific subjects or topics, typically structured in chapters and sections.

- **Research Papers**: Academic articles presenting original research findings, often with abstracts, introductions, methods, results, and discussions.
- **Lecture Notes**: Summaries of academic lectures that cover essential concepts and topics discussed in class.
- **Online Articles**: Relevant articles from reputable sources that provide additional context or supplementary information on the subject matter.

.

3. Algorithms

- **Neural Networks for Vector Embeddings**:
  - We will employ transformer-based models such as **BERT** or **Sentence-BERT** to convert text documents into dense vector embeddings. These embeddings capture the semantic meaning of the text, facilitating effective retrieval of relevant content.
- **K-Nearest Neighbors for Retrieval**:
  - We will implement the **k-nearest neighbors (KNN)** algorithm to identify the most relevant document sections based on user queries. The retrieved sections will be determined by calculating cosine similarity or Euclidean distance between the query embedding and document embeddings. Attention will be given to collinearity in the vector space to ensure diverse and meaningful results.
- **Pre-Trained Models for Generation**:
  - We will leverage **Hugging Face** pre-trained models to generate responses. After retrieving the relevant text segments, this model will generate coherent, contextually relevant answers to user queries.

3. Related Work

To ensure our approach builds on existing research, we will consider the following recent studies:

1. **Paper 1**:
   - Title: *"BUILDING A RETRIEVAL-AUGMENTED GENERATION SYSTEM FOR ENHANCED STUDENT LEARNING: CASE STUDY AT PRIVATE UNIVERSITY"*
   - http://www.jatit.org/volumes/Vol101No22/31Vol101No22.pdf

2. **Paper 2**:
    - Title: *"Vector Search on Billion-Scale Data Collections"*
    - *https://vldb.org/2024/files/phd-workshop-papers/vldb_phd_workshop_paper_id_13.pdf*

4. Assessment Methodology

The project's effectiveness will be evaluated using the following methods:

- **Performance Evaluation Measures**:
    - **Precision@k**: To assess the relevance of the top *k* retrieved chunks.
    - **Recall@k**: To evaluate the proportion of relevant information retrieved.
    - **Mean Reciprocal Rank (MRR)**: To measure the average rank of the first relevant chunk.
- **Cross-Validation Strategy**:
    - We will implement K-Fold Cross-Validation on both the retrieval and response generation components to ensure robustness. Each fold will be tested on different document subsets, providing insights into model performance under various data conditions.
- **Ablation Settings**:
    - We will perform ablation studies by varying input dimensions (chunk sizes), altering preprocessing methods (different parsing techniques), and adjusting algorithm complexity (varying the number of retrieved chunks). By analyzing the impact of these changes on performance metrics, we can determine which aspects most significantly contribute to the system's effectiveness.

5. Workload Distribution Among Team Members

Team Member	Responsibilities
Ayaazuddin Mohammad	Document Parsing and Preprocessing, Chunking and Vectorization
Mukesh Kumar Javvaji	Knowledge Base Creation and retrieval. Chatbot Interface Development.
Satvika Eda	Model Evaluation, Performance Assessment, and Related Work Research

- Each member will collaborate closely, with regular meetings to discuss progress, share findings, and ensure alignment on project goals.