COVID-19 FAQ Retrieval API from PDF Dataset

Problem Statement: 1

Objective:

Develop a FastAPI application that enables users to query FAQs from a COVID-19 PDF dataset. The system should embed FAQ content into a vector store and use RAG to retrieve and answer questions with contextual explanations.

Requirements:

- Extract questions and answers from the COVID-19 FAQ PDF.
- Convert content into vector embeddings.
- Store and retrieve based on semantic similarity.
- Generate answers using LLM with context.

API Endpoint:

- `/ask` (POST): Accepts a user query and returns an answer from the FAQ.

Sample Questions:

1. What are the symptoms of COVID-19?
2. How effective are face masks in preventing infection?

Expected Deliverables:

- PDF parsing module
- Embedding + vector database integration
- FastAPI interface
- RAG-based answering logic

News Headline Query Assistant Using PDF Dataset

Problem Statement: 2

Objective:

Create a FastAPI application to answer questions about historical events by extracting headlines and summaries from a News Category PDF dataset. Implement RAG to return relevant news snippets.

Requirements:

- Parse categorized headlines from PDF.
- Store summaries in a searchable vector format.
- Query retrieval and context-based answering using LLM.

API Endpoint:

- `/news_query` (POST): Accepts user input and returns headline and summary.

Sample Questions:

1. What happened during the US elections in 2020?
2. Show me news about NASA discoveries in 2022.

Expected Deliverables:

- Vectorized summaries from headlines
- FastAPI query interface
- LLM-driven response summarization

Duplicate Question Detection API Using Quora PDF Dataset

Problem Statement: 3

Objective:

Develop a FastAPI API to detect if a user-submitted question has existing duplicates in the Quora question pairs PDF dataset. Integrate semantic similarity search.

Requirements:

- Parse and embed Quora Q&A from PDF.
- Use cosine similarity or FAISS to retrieve duplicates.
- Provide ranked suggestions via API.

API Endpoint:

- `/similar_questions` (POST): Input a question, receive similar ones.

Sample Questions:

1. How can I improve my English vocabulary?
2. What is the best way to prepare for data science interviews?

Expected Deliverables:

- Cleaned Q&A corpus
- Embedded and indexed vector DB
- FastAPI interface for retrieval

Recipe Recommendation API Using RecipeNLG PDF Dataset

Problem Statement: 4

Objective:

Develop a FastAPI app that allows users to input ingredients and get recipe recommendations by parsing and embedding cooking instructions from a PDF.

Requirements:

- Parse recipe titles, ingredients, and instructions.
- Store them in a vector index.
- Retrieve contextually matched recipes using RAG.

API Endpoint:

- `/find_recipe` (POST): Input ingredients or type, return matching recipes.

Sample Questions:

1. What can I cook with tomatoes, pasta, and garlic?
2. Suggest a vegetarian Indian dinner recipe.

Expected Deliverables:

- Vectorized recipe database
- FastAPI interface with filters
- RAG-based cooking assistant

Twitter Customer Support Assistant from PDF Logs

Problem Statement: 5

Objective:

Create a support chatbot that uses a PDF of Twitter customer support interactions to retrieve similar past responses and generate context-aware replies.

Requirements:

- Parse customer queries and replies from PDF.
- Create embeddings for historical issues.
- Retrieve closest past examples and generate suggested answers.

API Endpoint:

- `/support_query` (POST): User describes an issue.

Sample Questions:

1. My phone doesn't charge - what can I do?
2. How to update my Twitter app settings?

Expected Deliverables:

- Embedded customer support PDF
- FastAPI chatbot with RAG
- Suggested replies using historical match + LLM

Dataset Documentation Navigator using Datasheets PDF

Problem Statement: 6

Objective:

Enable developers to query best practices and documentation techniques from a PDF dataset of datasheets for datasets.

Requirements:

- Extract and structure sections from PDF.
- Create chunk-wise embeddings.
- Use RAG to answer data documentation queries.

API Endpoint:

- `/ask_doc` (POST): Ask a question on dataset documentation.

Sample Questions:

1. What sections should be included in a dataset datasheet?
2. How should bias in datasets be addressed?

Expected Deliverables:

- Indexed and chunked datasheet PDFs
- FastAPI API
- Documentation knowledge assistant

Knowledge Graph QA with Persian Dataset (PeCoQ)

Problem Statement: 7

Objective:

Develop a question-answering interface for Persian queries using the PeCoQ PDF dataset and structured knowledge graph links.

Requirements:

- Parse SPARQL queries and responses from PDF.
- Convert QA into embeddings.
- Translate user input and retrieve structured responses.

API Endpoint:

- `/ask_persian_qa` (POST): Ask a complex Persian question.

Sample Questions:

1. Who is the president of Iran in 2020?
2. When was the last space launch from Iran?

Expected Deliverables:

- Bilingual query handling
- Graph-aware semantic retrieval
- Persian NLP pipeline

Factoid QA with ComQA PDF Dataset

Problem Statement: 8

Objective:

Create a question-answering system that handles complex, multi-step, and comparison queries using the ComQA PDF dataset.

Requirements:

- Parse grouped paraphrased questions.
- Build embedding index with clusters.
- Enable comparison and reasoning using RAG.

API Endpoint:

- `/ask_factoid` (POST): Accept complex natural language questions.

Sample Questions:

1. Which US presidents served in both WW1 and WW2?
2. Who was the CEO of Google before Sundar Pichai?

Expected Deliverables:

- RAG with paraphrase understanding
- Multi-hop QA support
- PDF-based cluster embedding

Technical Debt Q&A from Apache Project PDFs

Problem Statement: 9

Objective:

Design a tool for querying technical debt data extracted from Apache Foundation project reports in PDF. Questions will focus on code smells, fault types, and refactoring.

Requirements:

- Extract and clean metric tables from PDF.
- Create searchable embedding for code analysis terms.
- Enable QA for code quality insights.

API Endpoint:

- `/ask_tech_debt` (POST): Query about technical issues or trends.

Sample Questions:

1. Which project had the most refactorings in 2021?
2. What type of code smells are most common?

Expected Deliverables:

- Metrics-based document index
- RAG assistant for software quality

Scientific Reading Comprehension Assistant

Problem Statement: 10

Objective:

Build an assistant to help users read and understand academic PDFs using QA pairs and scientific content from a research paper comprehension dataset.

Requirements:

- Extract QA and paper sections.
- Build vector index with context windows.
- Generate reasoning-aware answers.

API Endpoint:

- `/ask_science` (POST): Submit a science question.

Sample Questions:

1. What are the limitations of current LLMs for QA?
2. How does the proposed model outperform the baseline?

Expected Deliverables:

- Academic QA vector database
- RAG-powered LLM assistant
- Structured scientific responses