AI Agent Architecture Document

Ankit Yadav

1. Overview

The designed AI agent automates the process of academic revision by summarizing PDF documents and generating theoretical questions. It follows a modular **Planner–Executor architecture**, ensuring clear separation between reasoning (task planning) and execution (task performance). The agent leverages a fine-tuned large language model (`google/gemma-2-2b`) using QLoRA for domain adaptation and performance efficiency.

2. Core Components

2.1 Planner

The **Planner** is responsible for interpreting user goals and structuring the overall workflow. When a user uploads a PDF, the planner:

- Identifies the objective (e.g., summarize, generate questions).
- Decomposes the objective into sub-tasks: text extraction, segmentation, summarization, question generation, and formatting.
- Defines the execution sequence and data flow between components.

2.2 Executor

The **Executor** carries out the steps determined by the Planner. It:

- Extracts text from PDFs using `PyPDF2` and `pdfminer.six`.
- Calls the fine-tuned Gemma model hosted on Hugging Face for summarization and question generation.
- Converts the AI-generated content into LaTeX for clean academic visualization.
- Generates a final PDF output combining summaries and questions.

2.3 Model

Model Used: `google/gemma-2-2b` (fine-tuned using QLoRA).

- **Why Gemma-2-2b:** Small yet capable model optimized for reasoning, summarization, and educational content generation.
- **Why QLoRA:** Allows parameter-efficient fine-tuning using limited GPU memory while retaining model quality.
- **Training Data:** Custom dataset created from scraped academic PDFs using Gemini API-based summarization and question generation pairs.

2.4 Interface Layer

The user interacts through a simple upload-and-generate interface. Once a file is uploaded:

- The planner triggers the processing pipeline.
- The executor handles text extraction, summarization, and LaTeX rendering.
- The system outputs a final downloadable revision-ready PDF.

3. Interaction Flow

1. **User Input:** User uploads a PDF.
2. **Planning:** The planner module defines the processing sequence.
3. **Execution:**
    - Text extraction and chunking.
    - Model inference for summary and question generation.
    - Formatting and LaTeX-based PDF rendering.
4. **Output:** User receives a well-formatted summarized PDF containing key takeaways and questions.

4. Reasons for Design Choices

- **Planner–Executor Pattern:** Offers modularity, easier debugging, and future scalability (e.g., adding RAG or evaluation modules).

- **QLoRA Fine-tuning:** Efficiently adapts Gemma-2-2b to the academic summarization domain with minimal compute cost.
- **LaTeX Output:** Ensures structured, readable, and professional-quality revision notes.
- **Hugging Face Deployment:** Provides reliable API-based access, version control, and portability.

5. Conclusion

The AI agent integrates reasoning, execution, and presentation into a coherent workflow for academic assistance. The architecture emphasizes modularity, interpretability, and deployment scalability. Future extensions may include retrieval-augmented generation, voice-based inputs, or UI-based planner controls for improved user experience.