AI Tutor Project: Intelligent Student Support through Gemma 1.5 Flash

Project Description

The AI Tutor project is designed to provide intelligent tutoring support using Gemma 1.5 Flash API for text generation, in addition to ChromaDB for student conversation storage and progress tracking. The system maintains contextual awareness, classifies questions into subtopics and topics, identifies question difficulty, and generates personalized learning recommendations.

Goals

The primary goals of the project are:

- Provide an interactive AI-driven tutoring experience that adapts to the learning progress of students.
- Track student learning history through a structured database.
- Topic and subtopic categorization of student questions for improved organization categorized question even if it is contextual i.e 1st question " What is thermodynamics", 2nd question " What are its laws" .
- Auto-classify question difficulty (Basic, Intermediate, Advanced) to refine learning suggestions.
- Generate personalized learning pathways based on past interactions and knowledge gaps.
- Store and recall prior conversations to keep context between interactions.

Technical Implementation

1. Student Progress Tracking

The StudentProgressTracker class manages student learning progress.

Loads prior interactions from ChromaDB and sets up student data in case there are no records.

Progress includes:

- Total questions attempted
-
- Subjects covered
-
- Last active date
-
- Performance feedback

-
- Difficulty distribution

Updates student progress after each interaction.

2. Query Classification (Topic & Subtopic Analysis)

- Classifies student questions using MODEL.generate_content().
- Considers prior interactions stored in ChromaDB to maintain context.
- Generates structured JSON output ({ "topic": "Mathematics", "subtopic": "Algebra"}).
- Maintains topic-specific classification even with a change in context.

3. Difficulty Analysis

- Uses AI model inference to classify the questions as Basic, Intermediate, or Advanced.
- Ensures valid categories are only returned.
- Defaults to "Basic" difficulty in the event of classification failure.

4. Context-Aware Response Generation

- The TutorAssistant class loads recent and relevant prior conversations.
- Combinations relevant interaction and recent conversation history.
- Constructs a formatted prompt for the AI tutor.
- Sends a response based on MODEL.generate_content().
- Stores the conversation to ChromaDB with duplicates omitted.

5. Learning Recommendations

- On the basis of monitored progress, the AI tutor suggests three new topics.
- Uses stored progress data to generate suggestions.
- Helps make students' weak points stronger and reinforce covered topics.

Main Features

- **Persistent Student Memory** – Remembers & stores history of previous questions & progress.
- **AI-Powered Topic Classification** – Classifies topics & subtopics based on dynamic identification.
- **Automated Difficulty Assessment** – Calculates question difficulty.
- **Context-Aware Response Generation** – Retains conversation history in responses.
- **Personalized Learning Recommendations** – Recommends lessons for students based on their history.

Future Improvements

1. Mastery Scores of Quizzes & Generation

- Provide support quizzes for chosen subtopics or topics at the learner's discretion.
- Apply mastery score on the basis of quiz outcome to allow measuring of mastery on a topic.
- Make use of upgraded progress monitoring featuring mastery scores and scales up the recommendations accordingly.

2. Meaningful Conversations in Depth

- Revamp the storage facility of past conversations to provide better context-based responses.
- Assign a semantic search engine facility for finding high relevance past conversation sessions rather than simple text similarity matches.