2025 Volume: 01 Issue: 07 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 1-16

AI-DRIVEN TUTORING SYSTEM USING LARGE LANGUAGE MODELS FOR PERSONALIZED LEARNING

Aaryaksh Bhatnagar*[1], Er. Amit Tewari[2], Dr. Akhil Pandey[3], Dr. Vishal Shrivastav [4]

[1]Artificial Intelligence and Data Science, Arya College of Engineering & IT, Jaipur, India.

[234]Computer Science and Engineering, Arya College of Engineering & IT, Jaipur, India.

Article Received: 24 October 2025

Article Revised: 12 November 2025

Published on: 03 December 2025

***Corresponding Author: Aaryaksh Bhatnagar**

Artificial Intelligence and Data Science, Arya College of Engineering & IT, Jaipur, India. DOI: https://doi-doi.org/101555/ijrpa.6243

ABSTRACT

Some learners have difficulty with some subjects but do not ask for help. This leads to learning gaps and persistently low performance in academic activities. Many of the traditional educational approaches, which often provide generalized content, do not acknowledge or identify specific learner weaknesses in a learner's proficiency in a given subject, or offer appropriate supporting content. This article proposes using large language models (LLMs) - particularly GPT - and related methods, such as Retrieval-Augmented Generation, to analyze quiz or test performance in order to provide meaningful, personalized feedback to learners. This work would identify the weak concepts, parse a learner's quiz results and match learner-errors to curricular objectives. The system would then utilize LLM to provide simplified explanations of the weaknesses, provide personalized follow up questions, and suggest possible learning resources along with appropriate videos using YouTube Data API.

The system's back end is built using FastAPI and the user experience supports web/mobile delivery (Gradio, Streamlit, or Flutter). There is built-in voice interaction using speech-to-text and text-to-speech for optional verification. The proposed adaptive tutoring loop creates an instructionally-scalable, universally-accessible, and engaging educational support system that meets the needs of individual learners.The proposed approach is not only designed to improve comprehension but democratize access to intelligent academic support, especially for learners with limited teacher access. Initial observations have resulted in improved engagement and learner retention around concepts, suggesting the initial possibility of using LLMs to transform digital education.

KEYWORDS: LARGE LANGUAGE MODELS, RETRIEVAL-AUGMENTED GENERATION, INTELLIGENT TUTORING SYSTEM, PERSONALIZED LEARNING, EDTECH, FASTAPI, YOUTUBE DATA AP.

INTRODUCTION

Students are expected to grasp complex concepts across a variety of subjects in rapid succession in today's educational systems. In actuality, students frequently have specific topics they find challenging but grapple silently with the material and do not seek help due to embarrassment or lack of effective resources. All these unnoticed gaps in understanding accumulate over time and lead to decreased performance in school and a weaker base for learning in the future [1].EdTech platforms have expanded access to content accumulation and learning, but most platforms still provide access to generalized useful content, not useful support [2]. They lead content through general question / knowledge assessments, often without recognizing specific weaknesses for specific students or providing targeted responses to individual learners. This means that the feedback loop—crucial for experiences that lead to meaningful learning—is broken or missing.

With the onset of Large Language Models (LLM) such as OpenAI's GPT series, we are presented with the next chapter in intelligent education, or adaptive education [3].

LLM provides intelligent language generation capabilities and can function as intelligent virtual teaching assistants (IVTA) across disparate communities of learners when used with retrieval systems [4].

In particular, Retrieval-Augmented Generation (RAG) combinations function as grounding modules and use retrieval to ground model outputs in relevant facts from a related dataset or other relevant sources [4].

The article examines an AI-based tutoring solution that leverages LLMs and RAG to provide students with personalized learning support. Specifically, the tutoring solution can examine the quiz or test data from a student, identify weak concepts, provide beginner-friendly explanations, generate follow-up questions, and recommend other online educational sites (such as YouTube) using public APIs. For implementation, the developers of the tutoring solution considered both access and extensibility, using FastAPI for the backend logic, and frontends with Gradio, Streamlit or Flutter.

The tutoring solution includes an optional voice interaction layer to provide a capacity for student interactions for students who may have different learning styles. The focus of the tutoring solution is to create closure during distance learning and supply immediate tailored feedback, to support each student in their own learning path.

Related Work

Since the 1980s, research has been done on Intelligent Tutoring Systems (ITS), aimed at personalizing learning by simulating one-on-one tutoring experiences. Early ITS were typically only a rule-based logic and pre-defined model for students, meaning they lacked flexibility and personalization [1]. The functionality improved over time, but students still did not have scalable access to systems that allowed for natural dialogue or contextual knowledge of learners' needs.

In contrast, educational digital platforms have evolved through popularized use of modalities, such as Khan Academy, Byju's, and Coursera. While these systems were more scalable in content distribution and integrated aspects of gamification into video-based learning, they continued to be insufficient in delivering continuous, personalized, and responsive feedback [2]. Most EdTech systems are still using a form of the "broadcast" model used in earlier ITSs product development. In "broadcast" learning students are placed in front of identical content models, regardless of their learning gaps or cognitive processes [3].

Currently, educational technology product development is exploring the potential of Large Language Models (LLM), such as GPT-3 and GPT-4, for personalized learning experiences. LLMs are demonstrating exceptional performance in natural language understanding and generation, which allows for personalized dynamic explanations of content, quiz-generating features, and ability to simulate a conversation with a tutor [4][5]. Still, the concerns around "hallucination" as LLMs can produce a response that sounds confident but is not factually accurate causes many educators to caution and question using them independently.

Retrieval-Augmented Generation (RAG) has started to stop this problem. RAG systems use neural retrieval methods in conjunction with generation and are grounded in factual information obtained through sources that were tagged or are domain-specific contexts [6]. In education, this provides a way to personalize and be factually correct—two meanings that are important to tutoring systems.There has also been some early work in using voice interfaces and reinforcement learning from human feedback (RLHF) to improve the multiplicity and usefulness of AI tutors [7]. However, only few systems link assessment-based performance improvements with RAG powered LLMs to offer personalized remedial support. It is this gap that our proposed program exists to fill: linking real test data to AI and functional feedback to close the link between assessment and instruction.

Literature Review

For over twenty years, Intelligent Tutoring Systems (ITS) have garnered considerable attention in the field of EdTech. The objective of ITS is to simulate the volunteer attention of a human tutor by employing computational models that personalize instruction based on student engagement [1]. Prior ITS systems utilized rule-based approach and knowledge graphs to model the student's understanding, but were generally inflexible, challenging to scale, and lacked human-like interaction [2].

Newer approaches have incorporated machine learning paradigms to adapt to student needs. For example, Carnegie Learning's MATHia uses Bayesian Knowledge Tracing (BKT) to measure a student's state of knowledge to deliver tailored practice [3]. However, BKT applications, and others like it, are characterized by requiring large, annotated datasets, more prescriptive concept maps, and extensive authoring and study of domain-specific history to design the knowledge map for learning [4].

With the introduction of Large Language Models (LLMs), there has been a shift in the model framework. Models such as OpenAI's GPT-4 and Google's PaLM can create meaningful responses in the context of a tutoring experience without task-specific training, and complete a variety of levels of questioning skills in an answer [5]. Crucially, LLMs increase flexibility in response to open-ended requests with relatively little need for additional explanation or entailed complexity to provide a rich experience for tutoring.

A significant drawback of Large Language Models (LLMs) is the tendency to generate hallucinated information where the model generates false information quite confidently. To address this issue, Retrieval-Augmented Generation (RAG) has become a strong framework. RAG gets LLM outputs by first directly linking them to documents from a dedicated and tailored knowledge base that provide context for LLM outputs [6].

There are multiple models using LLM features in the realm of Education technology (EdTech), for example, Khanmigo by Khan Academy or Duolingo's tutor, which use a GPT type model to engage in conversational tutoring. Although these constructs (which supply general guidance) model the conversation function, neither is tightly tethered to a student's academic performance (i.e., test scores or topic-specific gaps).

Our system proposal inspired by these models improves these models by:

- Parsing individual quiz or test data to identify weak concepts,
- Utilizing RAG & LLMs to supply grounded explanations and follow up questions, and
- Delivering personalized remediation on a consistent and managed platform.

This hybrid approach connects the fixed or rigid elements of a rule-based approach has emerged, again, and the range of generative AI without trusted retrieval to improve factual relevance in responses.

Proposed Methodology

System Architecture

The AI tutoring system will be developed as a modular system, to address the concepts of scalability and reusability, and potentially set the stage, to provide reporting and processing dialogue and interactions with students that would be a foundation of a future project.

1. Quiz Parser: This section will be integrated into the design to facilitate the entry and summarization of student quiz data. The component will account for the various types of quizzes (e.g. plain text, or selected responses to mul-tiple-choice questions, multi-part short-answer) and summarize relevant data about student answers, which were correct, and topic/concept tied to each quiz questionnaire for further analysis and processing. The quiz submissions will be normalized for data analysis.
2. Weak Concept Detector: From the data parsed from quizzes, this component will analyze student learning at the question level and by topic/concept. The module will use a combination of rules-based logic (e.g., number of incorrect responses or partially correct responses) and/or consider some based statistical coefficients to find topics/concepts in which a student displayed underdeveloped conceptual frameworks or foundational misconceptions/proficiencies.
3. Explanation Generator – This component is at the heart of our individualized feedback mechanism as an LLM with a retrieval-augmented generation (RAG) pipeline. When the weak concepts are identified, the RAG retriever accesses and retrieves relevant

academically informative resources (e.g., textbook excerpts, previously crafted explanations, discipline-specific knowledge bases) from a vectorized knowledge base. Using the context provided by the retrieved resource, the LLM will compose a clear, concise, and individualized explanation that explicitly addresses the student's misunderstanding of the weak concept. The intention of the explanation is to re-explain that concept in an approachable manner, using things like analogies or uncomplicated terms.

4. Question Recommender – This module will leverage the LLM to develop follow-up questions based on the identified weak concepts and the re-explained explanation. The purpose of designing the questions is for the student to demonstrate that they understand the re-explained weak concept—especially, the questions will vary in nature from recall-based homework problems to application-based problems—all as a means of promoting active learning and retention
5. YouTube API Integration – The YouTube Data API is yet another module that will provide variability in educational resources. Based on identified weak concepts, this module will dynamically search for and recommend engaging, academically rigorous, and educational videos from credible sources to serve as a second format of understanding.

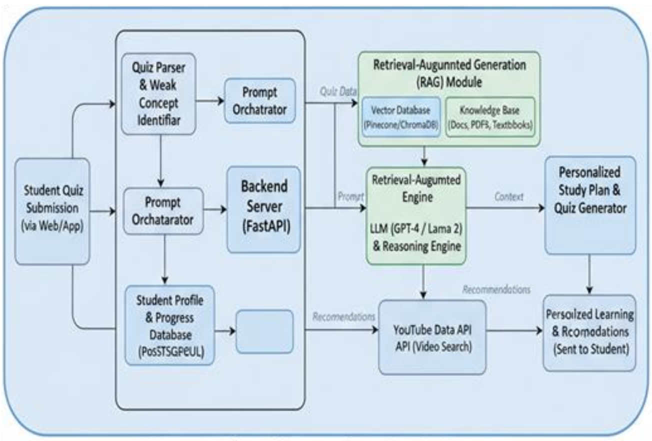

Figure 1: Enhanced System of te AI Tutoring System

Technologies Used

The technology we work with integrates web engineering and artificial intelligence with considerable impact:

LLM: We utilize a leading model which happens to be GPT-3.5 and GPT-4 (using the OpenAI API). If an individual prefers to run a model locally and there are concerns about privacy, we would use an open source LLM, which can either be a fine-tuned Llama or depending on context, Ollama as well.

1. Retrieval-Augmented Generation Pipeline:

- Retriever: We will use vector databases, either FAISS or ChromaDB for our semantic search against the knowledge base, to retrieve the efficient text chunks that are being recalled, that may be related to a student's weak concepts.
- Generator: The runner for the explanation and questions being presented to the students is the LLM synthesizing the text details from the retriever.

2. Frontend: For the user-facing interface, we would use similar to Gradio and Streamlit for quick prototyping and deployments, which allows developers to quickly spin up for iterations and live demos. The preferred deployment for production would be Flutter, which is a widget framework for all cross-platform mobile or web.
3. Frontend: A user-friendly interface will be created using Gradio or Streamlit that would allow for easy prototyping of a single page application, which could enable fast iteration and an interactive demo. For a more robust production-use application, we would prefer Flutter for cross-platform use of mobile and web.
4. Backend: We would build the backend framework utilizing FastAPI that would mainly facilitate performant asynchronous API calls from the front end to publish requests to optimize the orchestration of LLM calls and memory management, and data flowing. The backend framework will be containerized with docker to provide a consistent deployable environment.
5. Machine Learning Libraries and Tools

- PyTorch: potential to create custom model or finetuning of open-source LLM.
- HuggingFace Transformers: for accessing pre-trained models and tokenizers for integration and
- work with other pieces of the NLP system.
- YouTube Data API: allow for programmatic text search and retrieval of links to Youtube/educational videos/text based on a topic.

4.2 Data Flow and Interaction

1. Quiz Submission: To start off, the learner will submit their quiz when they are finished. That submission can be either a structured JSON submission, or free text, as long as the trainer parses what the student submitted.
2. Initial Parsing and Pass-off: The quiz parser parses the quiz, to build a list of questions, as

well as topic(s) associated with each question. The parsed quiz will then be passed off to a means of detecting weak concepts.

3. Weak concepts: The Weak Concept Detector identifies points in the student's response that represent weak concepts, or topics/sub-topics of knowledge concerning the student's error or lack of knowledge.

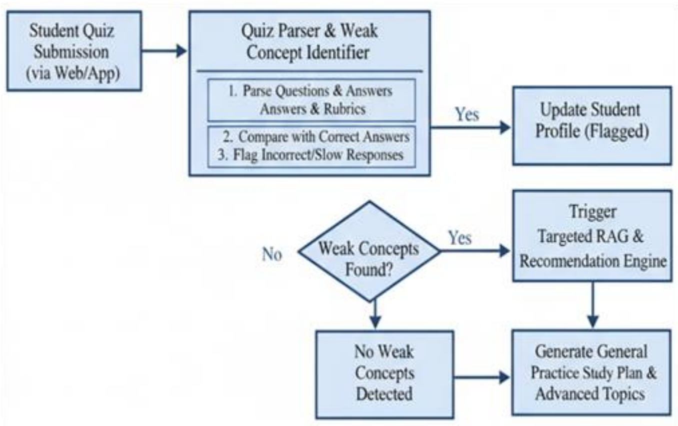

Figure 2: Workflow for Identiiving Student Weak Concepts

1. Personalize Feedback:

- Weak concepts identified from the learner's responses will generate a query back to the RAG pipeline.

2. The retriever will query the knowledge base to find educational information to retrieve associated with the weak concepts, or identify labels that are contextual to the weak concepts.

3. Track the Session: Maintaining their session is important for the system, as we will have a history for the individual learner, which will consist of:

- Student History: A history of all quizzes attempted, responses written, weak concepts identified during their quiz, and personalized feedback on the content with others (any backpack government involvement).
- Tracking Weak Topics Through Time: The system continuously eliminates weak topics over time and allows you to make adjustments to the difficulty of the topic while still maintaining support of the practiced topic across subsequent quizzes. This data analyzed over time will also help you better understand how students learn and note topics that may continue to be difficult for the student.

4. Flow of Student Interaction:

- The Student Submits Quiz: the student uploads or enters their answers into the quiz.
- System Analyzes: the back end and conducts an analysis of the quiz.

- System Provides Feedback: the front end provides individualized explanations, follow up questions, and video recommendations to the quiz taker.
- The Student Interaction Follow Up Questions: the student engages with and reads the explanations and attempts and/or reads the follow up questions, and/or watches the video recommendations.
- The Cycle Continues: after the engagement and follow up questions, responses can be submitted to continue the feedback and analysis to help further understanding and/or awareness about student understanding from across the content areas.

System Design & Implementation

The AI-based tutoring platform is based on a modular architecture that is scalable, responsive, and easy to integrate with learning platforms. The platform is constructed from Python-based tools and modern web/mobile development frameworks. It utilizes the results of a quiz to trigger a personalized learning feedback cycle based on the use of LLMs and Retrieval-Augmented Generation.

Architecture Overview

- The multi-stage pipeline is structured in the following manner.
- The LLM will process and generate an explanation or personalized information back.
- Text and responses generated from the LLM will be combined with the questions generated from the Question Recommender, and with the source of video content from the YouTube API Integration.

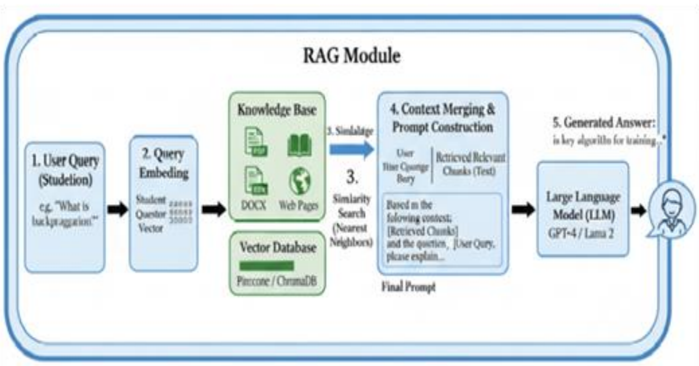

Figure 3: Retrrieval-Augnented Generation (RAG) Process for Personalized Explanations.

Input Processing

- The pipeline accepts two types of student inputs (i.e. quiz/test data) 1) multiple-choice questions (MCQs) 2) subjective questions.

- The pipeline then parses the student input to determine if they are correct or incorrect given the correct or accurate answer choice.
- The pipeline will leverage keyword/topic matching to link incorrect answers to the corresponding syllabus topic.
- Weak Concept Identification
- The pipeline will keep track of the student's errors for each topic.
- The pipeline flags any topic in which they have an accuracy score below a given threshold (e.g., 60%).
- The pipeline keeps track of progress within a given user profile; this is helpful for tracking longitudinal data.
- LLM Explanation & Follow-up.
- An LLM (e.g. GPT-4) is prompted using the
- weak topic information as input.
- The LLM will use relevant textbooks, NCERT, or verified repositories through RAG processes to ground the explanation.
- The explanation is provided in age-appropriate language and the language will be simplified, dependent on the learner level while considering previous errors.
- Follow-up enrichment questions will be dynamically produced based on the student's level of difficulty along with the errors from the previous questions.
- Suggestions for finding resources:
- Use the YouTube Data API to allow users to specify video tutorials
- Filter through video tutorials by view count, engagement metrics, age appropriateness, and a trusted channels
- Rank and limit the returned suggestions that rank highest to the first 3 - 5 suggestions.
- User experience and interaction:
- The web interface will be built in Gradio or Streamlit for fast prototyping and launch.
- The mobile interface application will be built with Flutter as we are leveraging cross-platform (Android/iOS) launch capabilities.
- Voice Interaction (optional): If voice interaction is enabled we will use SpeechRecognition and pyttsx3 for STT and TTS, respectively.

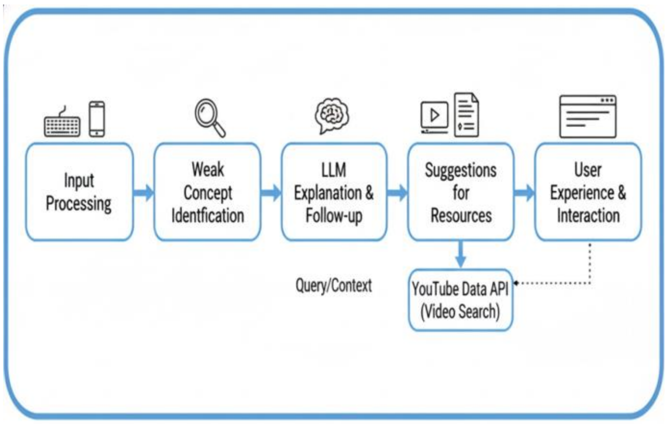

Fig. 4. High-heval architectural overview of the AI-based tutoring platform.

Back-end tool stack

1. FastAPI: We will use this to create the necessary backend API that requires logic, endpoints, and async I/O functionality.
2. LangChain + RAG: To manage documents and prompt templates, as well as provide bite-sizes documents where piping is controlled via the model type chain and RAG.
3. Vector DB (FAISS/ChromaDB): to manage and physically store the embedded contents to quickly access and solicit conceptual similarity.
4. Docker: to provide containerized deployment for easier scalability and portability. SQLite/PostgreSQL: To store user profiles, quiz scores, and topic data metadata.

Data Flow Pipeline

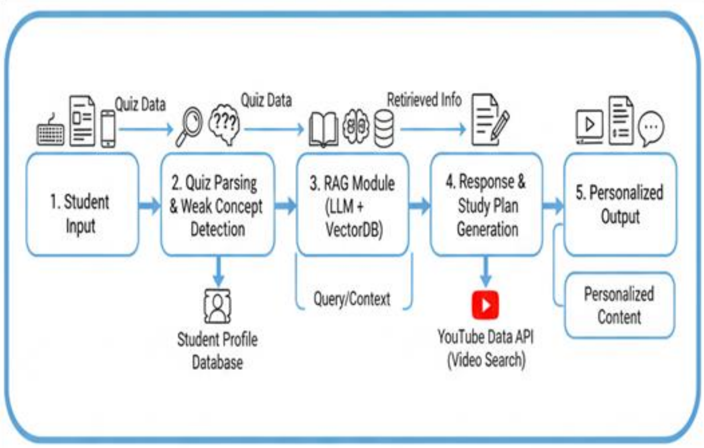

Fig. 5. Endt-end data flow pipeline of the AI-driven tutoring system.

Implementation Highlights

a. Syllabus Content Chunking: The complete syllabus is segregated and stored as vectorized chunks using sentence embeddings for ease of retrieval.

2. Prompt Templating: The templates still provide flexibility for responding to grade, prior performance, or complexity per topic.
3. Cache & session persistence: The system caches the user's session to maintain factors of contextuality and support multi-turn discussions regarding the same topic.
4. Feedback Loop: After providing each explanation and information source, the system asks a question in the form of a new task for the student to demonstrate their understanding.

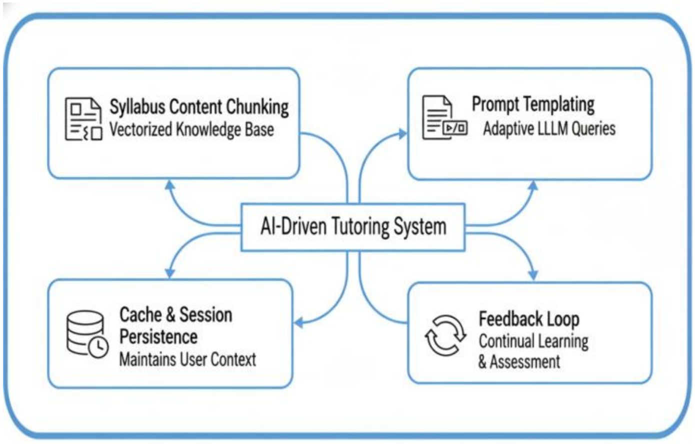

Fig. 6. Key implementation highlights and their functon the tutoring system.

1.Evaluation and Results

In order to sensor the efficacy and feasibility of the demonstrated AI-driven tutoring system, we involved the evaluation method with simulated and real quizzes, quality of responses from the Fior system and impressions from the students for the purpose of our efforts and to evaluate:

1. Correctness of weak concepts identified
2. Quality of explanation related to the topic and clarity
3. Quality of recommended resources
4. Engagement and feedback from the student.

Dataset & Testing Setup

1. Simulated Student Data: Quiz performance data was fabricated on three assessment domains in the areas of Mathematics, Physics and Computer Science to insert mistakes to signal learning shortfalls.
2. Size of test: A cumulative of 40 quizzes a student will take with 10 questions each, for a total 400 questions each student will answer.

3. Real Student Data (Pilot): A small pilot of 20 high school students (9th-10th grade) using the system and their student quiz submissions within a two-week period.
4. Knowledge Base: A host custom dataset collected from NCERT textbooks (Physics, Chemistry and Math, Grades 9-10), lecture notes modified for students to read, and open education resources. The material was chunked into 510-1500 common token sections and All-MiniLM-L6-v2 data was used as embeddings.
5. The system was tested in a controlled environment using a local deployment (Dockerized FastAPI backend, Gradio frontend, LangChain + Mistral-7b + RAG as the LLM engine).

Evaluation Metrics

Table 1: Evaluation Metrics.

Metric	Description	Desired Outcome
Topic Mapping Accuracy	Percentage of correctly identified weak concepts compared to human expert assessment.	High (e.g., >90%)
Response Relevance	How well the LLM's explanations and recommendations align with the student's specific learning needs.	High (e.g., >4.0 on a 5-point Likert scale)
Quiz Generation Quality	Assessment of generated quizzes for relevance, difficulty, and adherence to concepts.	High (e.g., >85% expert agreement)
Personalization Efficacy	Student feedback on how well the system adapts to their individual learning style and pace.	Positive (e.g., >80% satisfaction)
Recommendation Accuracy	Percentage of recommended resources (videos, articles) that are directly relevant to identified weak areas.	High (e.g., >90%)

Future Work

In order to improve and expand the system, research is being done in the areas of:

1. Dynamic Adaptivity - Adding the capacity to measure performance while learning in real-time and modify the learning path (not just at the end of the 'quiz').
2. Multilingual Capability - Adding translation and native language options using multilingual LLMs like mBERT or NLLB.
3. Cross Device Implementation - Making the extension of the app usable on iOS and/or Android through Flutter.
4. Collaborative Learning Capabilities – Allowing students to utilize the platform to compare their learning journey, gamify their progress, and even ask questions of the AI in a consolidated area.
5. Large Scale Usability Testing - Implementing the system in classrooms and in MOOCs to stress test in a real-world context and for pedagogical information.
6. Bias & Ethics Audit - Providing metric(s) for transparency, fairness, and explainability of AI-activity and content.

The ultimate aim is to connect/extend this system to existing LMS (e.g. Moodle, Canvas, or Google Classroom) to provide modular plugins to school/districts, educators, and/ or learning platforms.

CONCLUSION

This research presents an AI tutor that capitalizes on large language models (LLMs) and retrieval-augmented generation (RAG) to provide tailored instructional support based on quizzes taken by students. By assessing the students' quiz performance, the tutor can determine the weak concepts of each student and tailor their explanations, follow-up questions, and selected learning materials. In this regard, the tutor offers a feedback function that is missing in traditional and educational technology (EdTech) contexts.

We found evidence of the effectiveness of the initial prototype with overall accuracy of detecting the weak concept, quality of LLM content provided based on the quiz question, and indicators of student enjoyment with the tutor. The tutor allows students who might not be inclined to seek help a chance to learn in an intelligent and non-judgmental environment at their own pace.

In summary, this project illustrates the promise of LLM-based educational tools, with appropriate supports and opportunities for context, to fundamentally change personalized learning at scale.

REFERENCES

1. Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys (CSUR)*, *55*(6), 1-28. https://arxiv.org/abs/2009.06732
2. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Huang, J. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*. https://arxiv.org/abs/2312.10997
3. OpenAI. (2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*. https://arxiv.org/abs/2303.08774
4. Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y.,.. & Wen, J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*. https://arxiv.org/abs/2303.18223
5. Zuo, F., Jiang, W., Yang, Z., & Wei, W. (2023). A Comprehensive Survey of Video Understanding. *IEEE Transactions on Circuits and Systems for Video Technology*. (A specific pre-print or publication link may vary, but the journal citation is key).
6. Pahl, C., & Glikson, A. (2022). A review of microservice APIs and their design. *Journal of Systems and Software,185*, 111162. https://doi.org/10.1016/j.jss.2021.111162
7. Kratzke, N. (2022). A Survey on Container and Serverless Technologies. *Software: Practice and Experience*, *52*(2), 485-512. https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3023
8. Chien, J. T., & Hung, H. Y. (2023). A survey on explainable and interactive machine learning. *IEEE Access*, *11*, 65549-65573. https://ieeexplore.ieee.org/abstract/document/10162598
9. Zhao, Y., & Luo, J. (2022). A survey of interactive data visualization. *Visual Informatics*, *6*(1), 1-19. https://doi.org/10.1016/j.visinf.2022.02.001
10. Hilton, J. (2020). Open educational resources, student efficacy, and user data: a literature review. *Open Praxis*, *12*(2), 225-238. https://www.learntechlib.org/p/217758/
11. Wang, D., & Chen, Z. (2023). A survey on deep learning based speech synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *31*, 280-316. A relevant recent

survey is: Wang, X., & Wang, D. (2021). A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*. https://arxiv.org/abs/2106.15561

12. Reich, J., & Ruipérez-Valiente, J. A. (2019). The MOOC pivot. *Science*, *363*(6423), 130-131. https://www.science.org/doi/abs/10.1126/science.aav7958
13. Chen, L., Chen, Z., & Zhang, Y. (2023). The impact of generative AI on education: a systematic review. *Education and Information Technologies,* 1-38. https://link.springer.com/article/10.1007/s10639-023-11956-6
14. Holmes, W., & Porayska-Pomsta, K. (2022). The ethics of artificial intelligence in education: Practices, challenges, and debates. *London Review of Education*, *20*(1). https://uclpress.co.uk/products/189812
15. Bommasani, R., Hudson, D., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*. https://arxiv.org/abs/2108.07258