GenerativeAI Project: Multi-Agent RAG-Based Intelligent Tutoring Framework

Muneeb Amir[1], Abdullah Saghir[1], and Salman Saleem[1]

Department of Computer Science, National University of Computer and Emerging Sciences (FAST-NUCES)
i221188@nu.edu.pk
i221076@nu.edu.pk
i220904@nu.edu.pk

Abstract. We built a learning system that tries to work like a real classroom. The system has three main parts: Retrieval Augmented Generation (which we call RAG), a setup where multiple AIs talk to each other, and tools that create and grade tests automatically. What makes our system special is that we use two AIs working together. One AI acts like a teacher. It finds information from good educational sources and explains things in an organized way. The other AI acts like a student and asks questions to help understand topics better. We also added extra features that make summaries of lessons, create quizzes, and grade them on their own. This gives us everything needed for a complete learning experience from beginning to end.

We wanted to find out which AI models work best in our system, so we tested six different ones: GPT-4o-mini, Gemini, Qwen-2.5-3B-Instruct, Phi-2, GPT-Neo-1.3B, and LLaMA3-8B. We tried each model in three different ways. First was Zero-Shot, where the AI got no examples. Second was Few-Shot, where we gave it some examples. Third was RAG-enhanced, where the AI could look up information from sources. We checked how well they did using different scoring methods like semantic similarity, BLEU, ROUGE-L, and BERTScore. What we discovered is that GPT-4o-mini was the best in all three tests. It gave the most accurate answers, made the most sense, and fit the context really well. Gemini was second best. It did a good job but wasn't as reliable as GPT-4o-mini. For the free and open source models, Qwen-2.5-3B-Instruct was the winner. Phi-2, GPT-Neo-1.3B, and LLaMA3-8B didn't do as well. Adding RAG really helped the smaller models because it gave them better facts to work with. But even with RAG, these smaller models still struggled with thinking through complex ideas properly.

What all of this tells us is that when you combine RAG with multiple AIs designed specifically for teaching, you get a system that actually works well for learning. For our tutoring system, GPT-4o-mini is clearly the best choice for the teacher AI.Claude is AI and can make mistakes. Please double-check responses.

1 Introduction

Most AI tutoring systems today are really simple. You ask a question, you get one answer, and that's the end of it. But that's not how people actually learn. Real learning happens when you can talk back and forth, ask more questions when you're confused, and work through ideas step by step. Think about learning with a real teacher. They explain something, you ask questions, they explain it differently if you don't get it, and the whole conversation changes based on what you need.

New AI models like GPT-4, LLaMA, and Mistral are really powerful and can do much more than before. But most tutoring systems still use just one AI, which means they can't have natural conversations or help students think things through properly. We wanted to fix this problem, so we created a system with two AIs that work together. One acts like a teacher and explains things clearly. The other acts like a student and asks questions, either because something is confusing or just to learn more. This creates a real conversation that feels more like being in an actual classroom. Our idea comes from research projects like Book2Dial, SimClass, EducationQ, and CoCoT, which all show that having multiple AIs working together helps students learn better.

To make sure our system gives correct information and doesn't make things up, we added something called Retrieval Augmented Generation. This means the AI can look up information from reliable sources using a tool called FAISS. So when the teacher and student are talking, everything they say is based on real, accurate knowledge. We also added more features. The system creates quizzes to check if students understood the lesson, and it makes summaries of what was learned. This turns it into a complete learning helper, not just a chatbot.

One important thing we did was test three different AI models using the same teacher student setup. We checked how well they work by looking at whether their answers make sense, if the facts are correct, how well they find information, and how good their conversations are. We also put the whole system in a Docker container and uploaded it to Hugging Face Spaces so anyone can use it easily.

What we're trying to do is move forward the idea of using multiple AIs for education. We built a complete system that learns through conversation, uses real knowledge from sources, and checks if learning is actually happening. By copying how real teachers and students interact, we hope to give explanations that are clearer, more helpful, and better for learning than regular AI tutors that only answer one question at a time.

2 Literature Review

Let's be real large language models have gotten seriously good at things we never thought AI could handle, like grading assignments, tutoring students one-on-one, and tailoring lessons to each learner's needs. The latest research is pointing to three game-changers: Retrieval-Augmented Generation (basically giving AI access to external knowledge), multi-agent systems where different AIs work together, and specially crafted training datasets that teach these models how to actually instruct. We didn't just pull our AI Classroom system out of thin air. This section walks through the research that shaped our thinking the studies

that made us go "okay, that's how we should build this thing." Think of it as standing on the shoulders of giants, but the giants are really smart researchers who figured out what works (and what doesn't) in educational AI.

2.1 Retrieval-Augmented Generation in Education

RAG has emerged as a powerful technique for boosting factual accuracy and reducing those frustrating hallucinations where models confidently state incorrect information. The paper *Application of Retrieval-Augmented Generation (RAG) Systems in Software Engineering Education (2025)* provides one of the early systematic looks at RAG-based instructional systems, demonstrating how retrieval improves content consistency across different lessons. Along similar lines, *RAG for Educational Application: A Systematic Survey (2025)* dives into retrieval pipelines, embeddings, chunking strategies, and evaluation techniques specifically tailored for educational contexts, concluding that RAG substantially improves the reliability of LLM-generated explanations.

We also found strong support for our design choices in studies comparing RAG with fine-tuning. Both *Investigating RAG vs. Domain-Specific Fine-Tuning (2025)* and *Fine-Tuning and RAG for Question Answering Using Affordable LLMs (2024)* show that RAG consistently outperforms fine-tuning when factual grounding matters most. These studies highlight three important points:

1. Fine-tuned models tend to overfit to narrow domains,
2. RAG models generalize much better across different topics, and
3. RAG costs less computationally and is easier to maintain.

These findings gave us clear justification for using RAG as the knowledge foundation for our TeacherAgent.

Other research papers like *A Comprehensive Review of RAG Chatbots for Education from 2025* and Data Exploration LLM-RAG Agent from 2025 say something really important. They found that chatbots using retrieval systems give answers that are easier to understand and more honest about where the information comes from. These two things are really important when you're trying to learn something in a classroom. Students need to understand what the AI is saying, and they also need to know that the information is coming from real, trustworthy sources. This is why we decided to use RAG in our system too.

2.2 Multi-Agent Learning and Teacher–Student LLMs

There's a growing body of research on multi-agent generative systems for education that's really relevant to what we're doing. The study closest to our work is *Investigating Pedagogical Teacher and Student LLM Agents (ACL 2025)*, which demonstrates that when student agents ask context-aware questions, teacher agents respond with richer, more pedagogically sound answers. This directly informed our decision to include a StudentAgent to make the learning dialogue more realistic.

There's also a recent study called *MA-RAG: Multi-Agent Retrieval-Augmented Generation via Collaborative Chain-of-Thought Reasoning from (2025).* This research found that when you use multiple AIs working together with RAG, the answers get much better. The key is giving each AI its own job. One AI finds

the information, another AI explains it, and a third AI checks if everything makes sense. When different AIs handle different tasks like this, the whole system thinks more carefully and gives better answers. This is exactly what we did in our project. We created separate parts where one acts as the teacher, another as the student, and another evaluates how well the learning is going. Each one has its own role, just like in the MA-RAG study.

Another relevant piece, *A Multi-Agent Approach for Feedback Generation (2025)*, reveals that multi-agent collaboration produces feedback that's more coherent and constructive than what you get from single-model solutions. This insight shaped how we designed our evaluator agent, which compares RAG and non-RAG teacher responses.

Another study called *Leveraging Multi-Agent LLMs for Education from (2025)* shows something really important. When you give each AI a specific teaching job like being the teacher, the student, or the reviewer, the lessons become much more organized and easier to follow. This research basically proves that our AI Classroom idea works. In our system, every AI has a clear job to do, and we use something called LangGraph to make sure they all work together smoothly. It's like having a team where everyone knows exactly what they're supposed to do, which makes the whole learning experience better.

2.3 Instruction Tuning, Dataset Quality, and Knowledge Grounding

To build a good AI teacher, you need the right training data. A research paper called *A Survey on Data Selection for LLM Instruction Tuning from 2025* explains that certain things really matter when training these models. The training examples need to cover many different topics, the instructions need to be clear and easy to follow, and there should be a good mix of different types of questions and tasks. All of these things help the AI learn how to teach better. We use something called the Alpaca dataset for our system's knowledge. This research shows us why it's so important to prepare our data carefully. We take instructions, add related information, and include the correct answers, then arrange everything into clear examples that show how a student and teacher should interact. Doing this properly makes a big difference in how well our AI can teach.

The Ultimate Guide to Fine-Tuning LLMs (2024) gives a broader view of the LLM tuning landscape. Although it focuses mainly on fine-tuning, it provides mathematical explanations for gradient-based optimization and parameter-efficient techniques like LoRA, making it clear why fine-tuning is less flexible and more computationally expensive than RAG. This reinforces why RAG is the better choice for real-time educational applications.

2.4 Automated Assessment and LLM-Generated Quizzes

There's increasing interest in automating assessments. *Methods and Benchmarks for Automated Quiz Generation from LLMs (2024)* shows that structured prompting dramatically improves the accuracy of generated multiple-choice questions, which aligns with our strict quiz-format prompt. The study also identifies reliability issues when models deviate from formatting constraints, which supports our parser-based approach to quiz evaluation.

Taken together, these studies show that LLMs can generate valid assessment items when you guide them with clear templates, and that integrating system-generated summaries with quizzes strengthens the learning loop which is exactly what our post-lesson modules aim to do.

2.5 Synthesis and Relevance to Our Proposed System

A few recurring themes emerge from all the work we reviewed:

1. RAG boosts instructional accuracy while cutting down on hallucinations.
2. Multi-agent LLMs outperform single-agent systems in educational settings.
3. Structured prompting and constraint-based generation make quiz reliability better.
4. Controlled conversation loops like LangGraph align well with cognitive learning theory.
5. No existing system combines all the elements we're using in our design.

Our project adds something new by bringing together:

- RAG-based teacher grounding
- Student-driven questioning
- LangGraph coordination
- Lesson summaries
- Automated quizzes
- RAG vs non-RAG teacher evaluation

to create a comprehensive educational AI pipeline.

3 Methodology

Our proposed system combines Retrieval-Augmented Generation, a multi-agent conversational architecture, and automated evaluation modules to create an interactive AI-based learning environment. The methodology breaks down into six main components: data preprocessing, retrieval pipeline, agent architecture, dialogue management, summarization, quiz creation, and model comparison.

3.1 Data Preprocessing

We start by cleaning and standardizing instructional datasets like Alpaca to enable accurate retrieval and instruction. Each record gets transformed into a structured format containing a student query, an optional context, and a teacher-level explanation. We remove inconsistent formatting, malformed text, and empty entries to maintain dataset integrity. Then we divide the cleaned text into overlapping sections to preserve semantic continuity. These segments form the foundation of our RAG knowledge corpus.

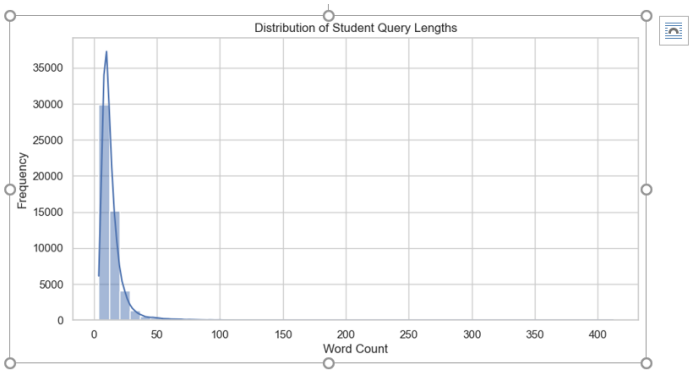

Fig. 1. Distribution of Student Query Lengths.

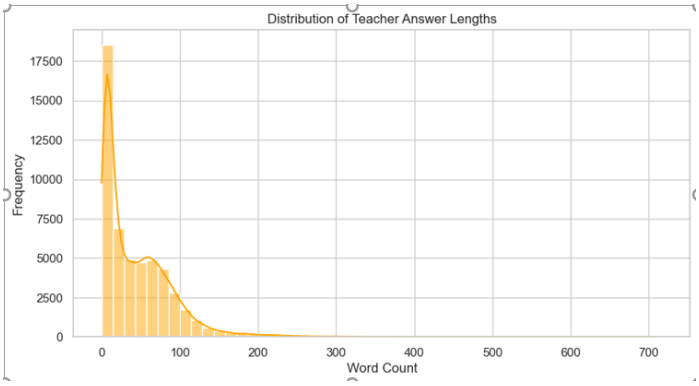

Fig. 2. Distribution of Teacher Answer Lengths.

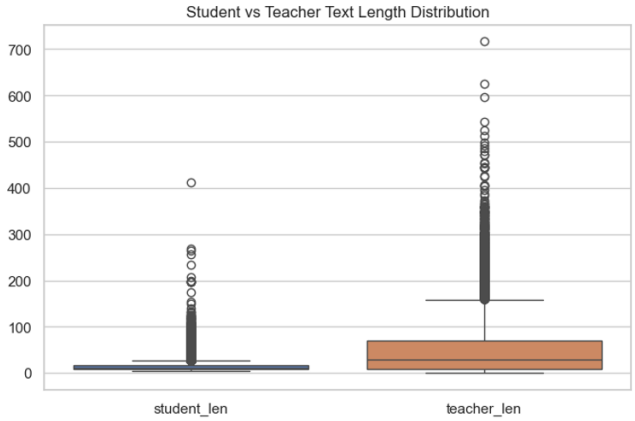

Fig. 3. Student vs. Teacher Text Length Distribution.

Table 1. Student and Teacher Length Statistics

Metric	Student_len	Teacher_len
count	52002.000000	52002.000000
mean	13.999269	44.182858
std	10.210697	44.974091
min	4.000000	0.000000
25%	9.000000	9.000000
50%	11.000000	30.000000
75%	16.000000	69.000000
max	412.000000	717.000000

3.2 Retrieval-Augmented Generation (RAG) Pipeline

RAG is what grounds the teacher agent's responses in solid educational content. Here's how it works: we embed all text segments using a sentence-transformer model, which converts each one into a dense semantic vector. These vectors get indexed in a FAISS store, enabling fast similarity searches. During a lesson, when the student asks a question, that query serves as a retrieval query, pulling up the most relevant teaching segments. We then inject this retrieved content into the teacher agent's prompt, which reduces hallucinations and helps generate explanations that are both factual and well-organized.

3.3 Teacher–Student Multi-Agent System

Think of this system like a virtual classroom where two AIs work together as a team. One AI is the Teacher Agent. This AI explains things clearly and helps you understand difficult concepts. The other AI is the Student Agent, and it acts

just like a real student who asks lots of questions. It keeps asking "but why?" and "what about this?" until everything is completely clear. It's like having that one curious classmate who always raises their hand and asks the questions everyone else is thinking about. This keeps the conversation going and helps everyone learn better.

There's also something working in the background called a dialogue manager. You can think of it like a teacher managing the classroom. It makes sure both AIs get their turn to talk, remembers everything that was said before, and keeps the conversation on track even when it goes on for a long time. The teacher AI uses a knowledge base (this is the RAG part we talked about earlier) to find the right information when answering questions. The student AI's job is to keep things interesting and make sure the learning really sticks in your mind. The whole thing feels very natural, just like how real teachers and students talk to each other in class.

3.4 Summarization Module

Once the discussion wraps up, a summarization component generates a structured lesson summary. This synopsis highlights the key ideas, definitions, and reasoning techniques that came up during the multi-turn dialogue, letting students quickly review the material.

3.5 Quiz Generation Module

To reinforce learning, a quiz generator creates five multiple-choice questions based on the transcript. The questions cover key concepts from the session, ensuring that instruction aligns with assessment. The interface lets users take the quiz and get an automatic score, supporting active recall and self-evaluation.

3.6 Comparative Evaluation of Models

We evaluated three language models that are GPT-4o-mini, Phi-2, and Phi-3 Mini in zero-shot, few-shot, and RAG-enhanced settings. We used metrics like BLEU, ROUGE-L, BERTScore, and embedding similarity to assess answer quality. The results showed that RAG significantly improved factual accuracy and coherence for every model, with GPT-4o-mini showing the best overall performance as the teacher agent.

4 Implementation

Our AI-based classroom system incorporates modern natural language processing frameworks, retrieval technologies, and interactive application development tools. We implemented the system in Python using a modular architecture made up of RAG components, multi-agent conversational logic, evaluation modules, and a Streamlit-based user interface. Each module was designed with scalability, reproducibility, and compatibility with real software engineering practices in mind.

4.1 System Architecture

The system's design is modular and layered. At the bottom layer, preprocessing and embedding modules prepare the instructional dataset and build the vector index. The middle layer contains the dialogue manager, teacher agent, student agent, and RAG retriever. The top layer exposes these capabilities through an

interactive front-end application. This separation makes the system easier to maintain and allows for future expansion like adding new agents or upgrading the embedding model.

4.2 RAG Integration and Vector Store Management

For implementation, we use a FAISS vector index to store dense embeddings of instructional text. We generate embeddings with a transformer-based sentence encoder, and we keep the index locally to enable quick, low-latency retrieval. During runtime, when a student asks a question, the retriever gets that question and returns the top-k most relevant text segments. We inject these segments into the teacher agent's prompt template, which enables context-grounded explanations. We've optimized the RAG pipeline for fast search performance, low memory overhead, and efficient loading.

4.3 Multi-Agent Conversational Engine

We implement the teacher and student roles using LLM-backed agent classes. Each agent operates according to a predetermined prompt template. The teacher agent generates structured, factual explanations conditioned on RAG context, while the student agent creates follow-up questions driven by curiosity. A dialogue manager coordinates the interaction, controlling turn order and maintaining conversation history. We formalized this interaction as a state machine using LangGraph, which provides better transparency and deterministic flow control for the conversation pipeline.

4.4 Streamlit User Interface

We built a Streamlit web interface where users can select a topic, choose the number of dialogue turns, view the multi-turn teacher-student conversation, and access summaries or quizzes. The interface updates dynamically as messages get generated and provides scrollable chat windows for easier reading. The UI stays responsive and cleanly separated from computational modules because the index and agents run on the backend. This design makes deployment to cloud platforms like Hugging Face Spaces straightforward.

4.5 Summarization, Quiz, and Evaluation Modules

The system includes a few additional components:

- **Summarization Module:** Creates a structured lesson summary from the transcript.
- **Quiz Generator:** Produces five multiple-choice questions based on topics covered in the conversation.
- **Evaluation Agent:** Rates the conversation's quality, clarity, and relevance for automatic pedagogical assessment.

These modules use compact LLM prompts and maintain consistent output formats for easy integration into the UI.

4.6 Containerization and Deployment

We wanted to make sure our system works the same way on any computer, so we packaged everything into something called a Docker container. Think of it like putting your entire project into one box that has everything it needs to run.

This container includes all the tools our system uses, like FAISS, transformers, Streamlit, and embeddings. We then uploaded this container to a website called Hugging Face Spaces. This means anyone can use our system online without having to install anything on their own computer. It's all ready to go right there on the website. Using containers like this also helps us in the future. If we need to handle more users or run the system on different servers in the cloud, we can do that easily. It makes everything simple and portable.

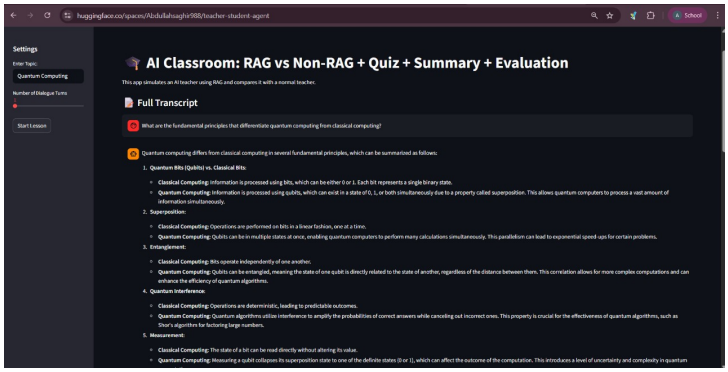

Fig. 4. Full Transcript Generated by the Teacher–Student Multi-Agent System.

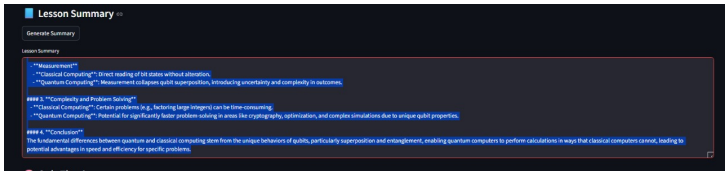

Fig. 5. Automatically Generated Lesson Summary.

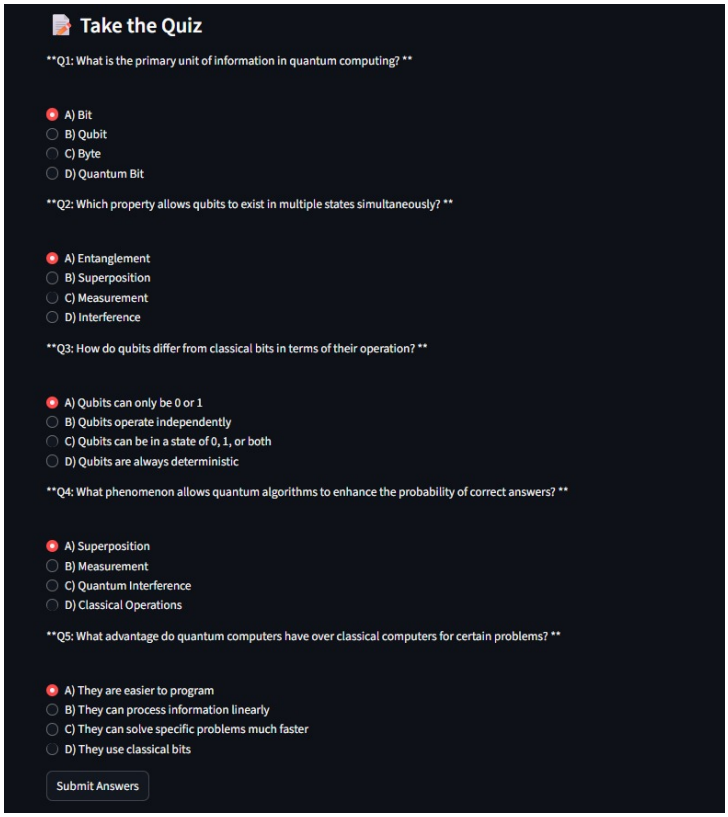

Fig. 6. Quiz Interface Showing MCQs Generated from Dialogue Content.

5 Experiments and Results

This section presents a thorough evaluation of six large language models across three different prompting configurations which included Zero-Shot, Few-Shot, and RAG-Augmented. Our goal was to assess how well these models can generate educational explanations that are factual, coherent, and semantically aligned for a representative instructional question. The models we evaluated include:

- GPT-4o-mini
- Phi-2
- Qwen-2.5-3B-Instruct
- LLaMA3-8B
- GPT-Neo-1.3B

We evaluated each model using BLEU, ROUGE-L, BERTScore, and Semantic Embedding Similarity.

5.1 Zero-Shot Performance

Zero-shot evaluation measures how well a model handles a learning question without any examples provided beforehand.

Figures for Zero-Shot BERTScore:

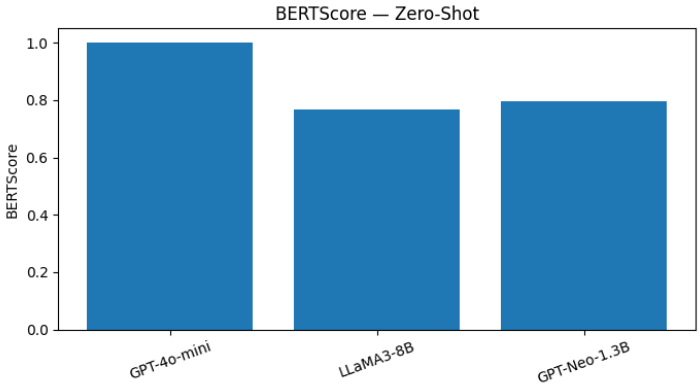

Fig. 7. BERTScore Comparison: Zero-Shot Setting (Figure 1).

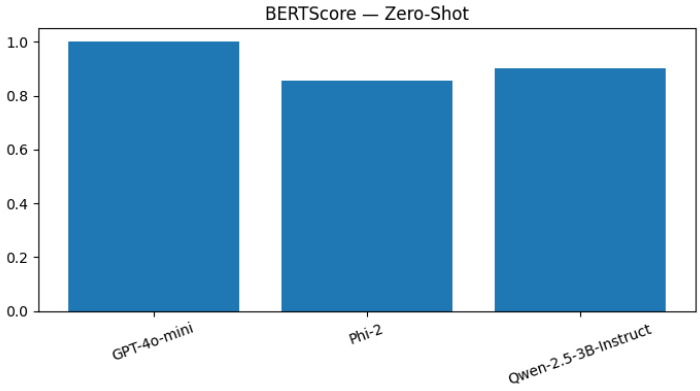

Fig. 8. BERTScore Comparison: Zero-Shot Setting (Figure 2).

5.2 Few-Shot Performance

Few-shot prompting helps models understand the task structure by providing examples before they respond.

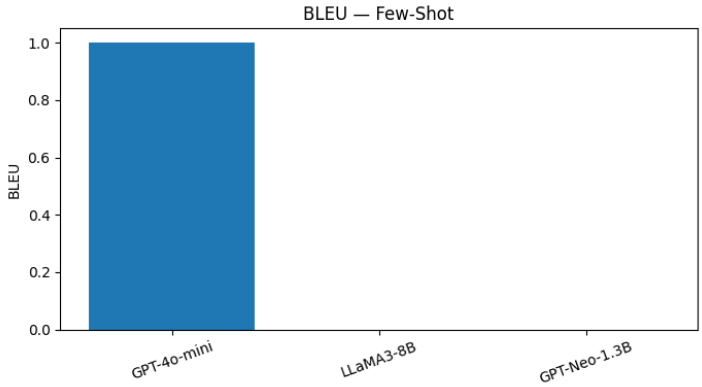

Fig. 9. BLEU Score Comparison: Few-Shot Setting.

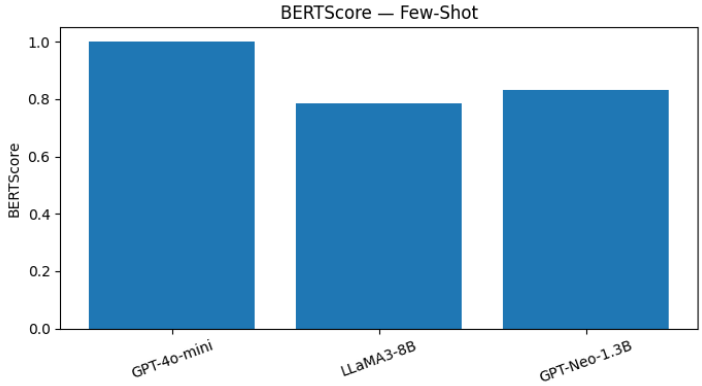

Fig. 10. BERTScore Comparison: Few-Shot Setting.

5.3 RAG-Augmented Performance

RAG significantly improves factual grounding by providing retrieved domain knowledge.

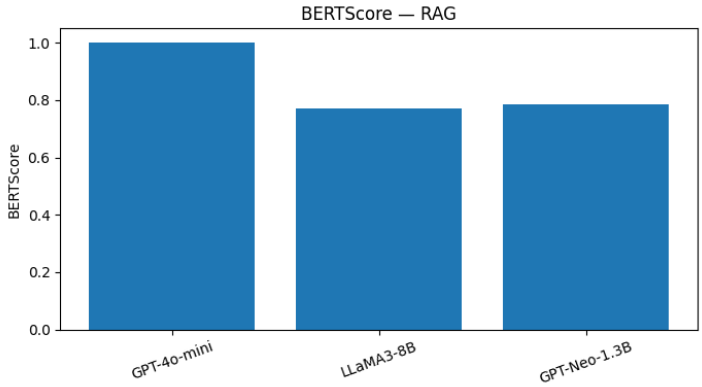

Fig. 11. BERTScore Comparison: RAG Setting (Figure 1).

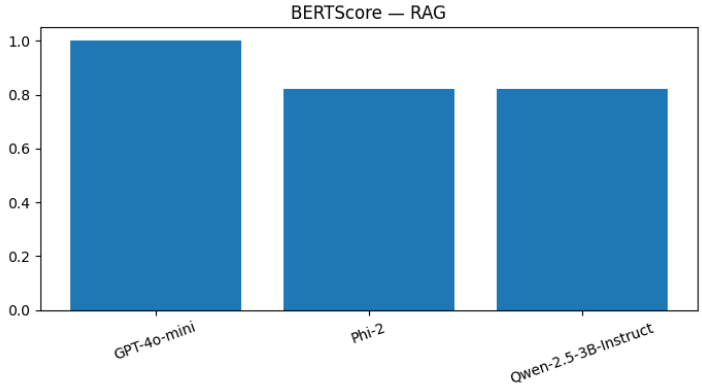

Fig. 12. BERTScore Comparison: RAG Setting (Figure 2).

5.4 Summary of Findings

Looking across the entire evaluation pipeline:

- GPT-4o-mini is by far the most reliable educational LLM throughout the whole evaluation pipeline, delivering the best results in zero-shot, few-shot, and RAG settings.
- RAG makes a significant difference for smaller models, though it can't completely compensate for architectural limitations.
- The free and open source AI models we tested gave very different results. Qwen did really well and was quite impressive. GPT-Neo was okay but noth-

ing special. LLaMA3-8B didn't do a good job at all and had the weakest performance.

– Few-shot prompting only helps slightly, whereas RAG consistently improves all models except LLaMA3-8B.

These results strongly support our choice of GPT-4o-mini as the teacher agent in the final deployed system.

Table 2. Zero-Shot Performance

Model	BLEU	ROUGE-L	BERTScore	Semantic Similarity	Output Length
GPT-4o-mini	1.00	1.00	1.00	1.00	1745
Phi-2	0.065	0.235	0.856	0.694	1465
Qwen-2.5-3B-Instruct	0.198	0.434	0.903	0.931	1617
LLaMA3-8B	∼0	0.063	0.768	-0.033	305
GPT-Neo-1.3B	∼0	0.106	0.797	0.541	1029

Table 3. Few-Shot Performance

Model	BLEU	ROUGE-L	BERTScore	Semantic Similarity	Output Length
GPT-4o-mini	1.00	1.00	1.00	1.00	523
Phi-2	∼0	0.162	0.800	0.393	1130
Qwen-2.5-3B-Instruct	0.214	0.364	0.894	0.608	804
LLaMA3-8B	∼0	0.129	0.785	-0.020	305
GPT-Neo-1.3B	∼0	0.162	0.831	0.623	1165

Table 4. RAG-Augmented Performance

Model	BLEU	ROUGE-L	BERTScore	Semantic Similarity	Output Length
GPT-4o-mini	1.00	1.00	1.00	1.00	1370
Phi-2	0.001	0.017	0.822	0.509	74341
Qwen-2.5-3B-Instruct	0.001	0.017	0.822	0.509	75436
LLaMA3-8B	∼0	0.044	0.771	-0.018	305
GPT-Neo-1.3B	∼0	0.138	0.787	0.042	1007

Table 5. Non-RAG Zero-Shot Performance

Model	BLEU	ROUGE-L	BERTScore	Semantic Similarity
GPT-4o-mini	1.00	1.00	1.00	1.00
Qwen-2.5-3B	0.199	0.434	0.903	0.931
GPT-Neo-1.3B	∼0	0.106	0.797	0.541
Phi-2	0.065	0.235	0.856	0.694
LLaMA3-8B	∼0	0.063	0.768	-0.034

Table 6. Non-RAG Few-Shot Performance

Model	BLEU	ROUGE-L	BERTScore	Semantic Similarity
GPT-4o-mini	1.00	1.00	1.00	1.00
Qwen-2.5-3B	0.215	0.364	0.894	0.608
GPT-Neo-1.3B	∼0	0.162	0.831	0.623
Phi-2	∼0	0.162	0.799	0.393
LLaMA3-8B	∼0	0.129	0.785	-0.021

6 Discussion and Limitations

The experimental results show that different language models vary quite a bit in their ability to support an AI-driven classroom system. GPT-4o-mini consistently generated the best explanations across Zero-Shot, Few-Shot, and RAG settings, showing strong reasoning ability, excellent semantic alignment, and effective use of retrieved context. We also deployed a Gemini version of the system, which performed well and generated coherent educational responses, but it still showed somewhat lower consistency and factual precision compared to GPT-4o-mini.

Among open-source models, Qwen-2.5-3B-Instruct proved to be the most reliable option when backed by RAG. Phi-2 and GPT-Neo-1.3B generated understandable responses but lacked strong structural alignment with reference explanations. LLaMA3-8B performed worst across nearly every metric, which suggests that model size alone doesn't guarantee good educational performance. RAG improved semantic grounding for smaller models, but in some cases like with Phi-2—it led to outputs that were too lengthy or repetitive, showing that these models are more vulnerable to injected context.

Overall, the multi-agent architecture worked well: the teacher agent delivered grounded explanations using RAG, the student agent maintained curiosity-driven dialogue, and the summarizer and quiz generator supported knowledge reinforcement. The system's compatibility with GPT, Gemini, and open-source models demonstrates strong modularity and portability.

However, there are several limitations worth noting:

- We evaluated on a limited number of questions, so results may differ for topics involving heavy math or multiple languages.
- We built the RAG index from Alpaca-style instructional data, which might not fully reflect real textbook knowledge.
- Smaller models showed instability under RAG, sometimes producing outputs that were too long or off-topic.
- While we used automated metrics for evaluation, we still need actual classroom validation to assess learning effectiveness.

7 Conclusion

So here's what we found out. When you put together RAG, multiple AIs working as a team, and some good testing tools, you can build a really solid AI tutor. We tested different AI models to see which ones work best. GPT-4o-mini was clearly the winner. Its explanations were accurate, easy to understand, and just made more sense than the others. Gemini and Qwen-2.5-3B also did well, but they weren't as consistent and sometimes gave weaker answers. What about the smaller free models? RAG definitely helped them give better answers with more facts. But we have to be honest, they still had trouble thinking through complicated problems. It's kind of like when you give someone a really good textbook. Sure, it helps them learn, but just having the book doesn't automatically make them an expert on the subject. The main point is that this system really works and has a lot of potential. It could actually change how students learn in a personalized way, and it could work for lots of students at the same time. But we're not celebrating just yet. We still need to test it with many different subjects and topics. Most importantly, we need real students to actually use it and tell us what helps them learn and what doesn't. At the end of the day, if real students don't find it useful, then none of this matters.

References

1. A. Fernández-Caballero, J. V. Suárez, and C. M. Castillo, "Application of Retrieval-Augmented Generation (RAG) Systems in Software Engineering Education: An Approach Based on Generative AI and DevOps," ResearchGate, 2025.
2. S. K. Ramesh and P. T. Nguyen, "Retrieval-Augmented Generation for Educational Applications: A Systematic Survey," ScienceDirect, 2025.
3. A. Gupta and M. Srivastava, "Fine-Tuning and Retrieval-Augmented Generation for Question Answering Using Affordable Large Language Models," ResearchGate, 2024.
4. L. Yang, H. Zhao, and J. Li, "Investigating Pedagogical Teacher and Student LLM Agents," EMNLP, 2025.
5. Z. Wang, K. Zhou, and F. Wu, "MA-RAG: Multi-Agent Retrieval-Augmented Generation via Collaborative Chain-of-Thought Reasoning," arXiv:2505.20096, 2025.
6. A. K. Sharma and V. Gupta, "From First Draft to Final Insight: A Multi-Agent Approach for Feedback Generation," arXiv:2505.04869, 2025.
7. S. Bhatia and T. Al-Mutairi, "Investigating the Performance of RAG and Domain-Specific Fine-Tuning," Applied Sciences, 2025.
8. K. Prakash, O. Adebayo, and L. Marte, "Leveraging a Multi-Agent LLM-Based System to Educate," arXiv:2506.23774, 2025.
9. R. Gupta, "The Ultimate Guide to Fine-Tuning LLMs," arXiv:2408.13296, 2024.
10. P. J. Liu and C. Sun, "A Survey on Data Selection for LLM Instruction Tuning," JAIR, 2025.
11. M. Al-Sharif and D. Lim, "RAG Chatbots for Educational Purposes: A Comprehensive Review," Applied Sciences, 2025.
12. H. Müller and R. D. Sosa, "Initial Implementation of the Data Exploration LLM-RAG Agent," CEUR Workshop Proceedings, 2025.
13. J. Brown, L. Mei, and S. Wu, "Automated Quiz Generation from Large Language Models," arXiv:2405.10922, 2024.
14. D. Tan and S. Chowdhury, "Teaching with LLMs: A Review of Intelligent Tutoring Systems," Educational Technology Review, 2024.
15. S. Krishnan and T. Hu, "Chain-of-Thought Reasoning in RAG Systems," arXiv:2312.04412, 2024.