Custom RAG-Based Tutoring Systems: A Case Study in Finance Education

John Zontos
Department of Electrical Engineering and Computer Science
Oregon State University
zontosj@oregonstate.edu

Abstract

Finance Tutor AI is an interactive educational chatbot designed to help students learn finance through personalized conversations. The system integrates a Retrieval-Augmented Generation (RAG) framework with OpenAI's large language models (LLMs) and a domain-constrained FAISS retrieval backend, drawing on high-quality sources such as the OpenStax Finance textbook and curated educational materials. This system provides context-aware, adjustable responses and integrates optional knowledge checks to reinforce learning. Evaluations suggest the system delivers relevant, high-quality explanations and shows promise for adaptation across diverse educational domains.

1 Introduction

Recent advancements in large language models (LLMs) have demonstrated their potential to support learning through natural language interactions. However, their tendency to hallucinate facts, lack of transparency in reasoning, and limited domain grounding pose significant challenges when used as autonomous tutors.

Despite the growing popularity of AI-based educational tools, students frequently lack access to tailored academic support that is both trustworthy and interactive. Existing tools either rely on static content or offer limited personalization, and purely generative models can hallucinate facts, undermining their educational value.

To address these challenges, we present *Finance Tutor AI*[1][2], a domain-constrained Retrieval-Augmented Generation (RAG) system that combines the strengths of retrieval-based grounding with the generative capabilities of LLMs. Our system retrieves high-quality, curated content from the OpenStax Finance textbook[4] and injects it into custom prompt templates to guide the LLM's response. This enables the tutor to produce accurate, context-aware explanations that adapt to different student needs and levels of expertise.

While the current implementation focuses on finance, the architecture is modular and extensible to other academic domains. We provide a case study in finance to demonstrate how custom RAG pipelines can be adapted for educational applications. The system includes a responsive frontend with sliders to adjust explanation depth and toggles for enabling knowledge checks. Preliminary evaluation suggests the approach is effective at generating relevant and pedagogically sound responses, paving the way for future expansion into broader classroom use.

[1] https://finance-tutor-ai.vercel.app/

[2] https://github.com/johnczontos/finance-tutor-ai

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Background and Related Work

2.1 Technical Background

Retrieval-Augmented Generation (RAG) is a hybrid architecture that enhances the capabilities of large language models (LLMs) by incorporating external context retrieved from a knowledge base. Instead of relying solely on the model's internal parameters, RAG pipelines retrieve relevant documents and inject them into the prompt to guide generation, enabling more grounded and domain-specific responses.

At the core of the retrieval process is vector similarity search. Given a user query, both the query and a corpus of documents are embedded into a high-dimensional space using an embedding model (in our case, OpenAI's `text-embedding-ada-002`). The relevant documents are then identified using cosine similarity.

We use FAISS (Facebook AI Similarity Search) [2] as the vector index to support scalable and efficient nearest-neighbor search. We empirically tested multiple retrieval strategies, each with distinct trade-offs.

- **Top-k Nearest Neighbors:** Retrieves the k most similar documents by cosine similarity, regardless of their absolute score. While effective in ensuring relevant content, this method occasionally included irrelevant documents due to lack of filtering.
- **Similarity Score Thresholding:** Returns documents whose similarity exceeds a fixed threshold, which helps filter out loosely related or irrelevant passages. This method consistently produced the most faithful and contextually grounded responses in our experiments.
- **Maximal Marginal Relevance (MMR):** Balances relevance and diversity among retrieved documents to reduce redundancy. Although useful for covering varied aspects of a topic, MMR underperformed in our setting due to occasional inclusion of less relevant content.

Once the relevant documents are retrieved, they are inserted into the appropriate prompt template and passed to the LLM for answer generation. Prompt engineering plays a critical role in guiding the model's behavior, especially in educational settings where tone, depth, and clarity must align with student needs. We design prompts with adjustable verbosity and explanation complexity, enabling the model to adapt to varying levels of user expertise.

2.2 Related Work

Recent advances in LLM-based tutoring systems have led to the emergence of tools like Khanmigo (developed by Khan Academy in collaboration with OpenAI) and Socratic by Google. These systems aim to deliver personalized academic support, though they often operate in closed domains or rely heavily on proprietary content. Moreover, most lack transparency in source attribution and retrieval mechanisms.

In academic research, RAG architectures have been explored for a range of applications including question answering, summarization, and code generation. For example, Lewis et al. [3] introduced RAG models combining dense retrieval with sequence-to-sequence generation, showing improved factuality in QA tasks. More recent work has focused on retrieval strategies such as MMR and dynamic re-ranking to optimize the relevance-diversity tradeoff [5].

Compared to these systems, our work focuses on education-specific tuning and adaptability. It demonstrates how a RAG-based tutoring system can be customized for a specific subject (finance) using public resources. Additionally, the system incorporates features such as explanation depth controls and knowledge check toggles, which are designed to enhance pedagogical effectiveness.

3 Technical Approach and Methods

3.1 Data Collection and Preprocessing

To ensure grounded and accurate responses, we curated high-quality educational materials focused on finance. These included the OpenStax Principles of Finance textbook (structured as DOCX)

and manually vetted YouTube video transcripts relevant to personal finance, investing, and financial literacy.

Each document was split into semantically coherent chunks using a recursive character-based strategy. Chunks were enriched with metadata such as section headings, URLs, and source identifiers. All chunks were embedded into a high-dimensional vector space using OpenAI's `text-embedding-ada-002` model to enable efficient semantic search.

3.2 Backend Architecture

The backend follows a Retrieval-Augmented Generation (RAG) pipeline that integrates retrieval and language generation to produce context-aware answers. Figure 1 provides a visual overview of the system architecture.

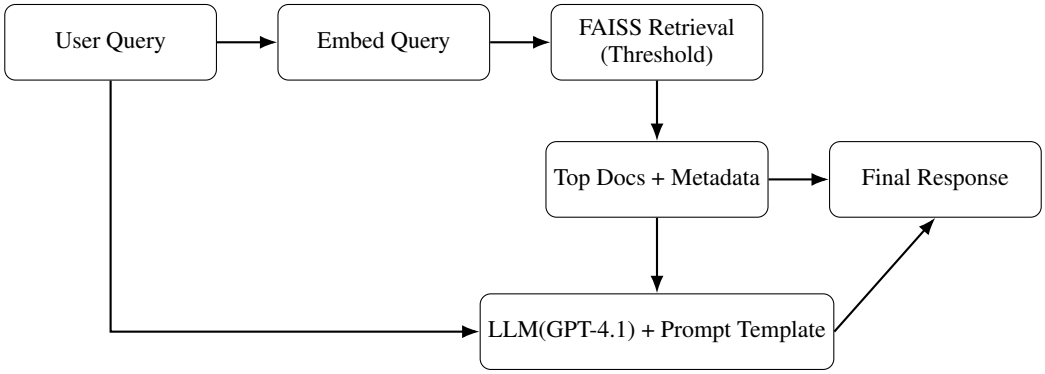

Figure 1: Overview of the RAG tutoring pipeline. The user query is embedded, relevant documents are retrieved via FAISS, and the results are injected into a prompt template for the LLM to generate a response.

We experimented with three retrieval strategies—top-k similarity, similarity score thresholding, and Maximal Marginal Relevance —to optimize the relevance and diversity of retrieved context. After evaluating their impact on response quality, we selected similarity score thresholding as the final method due to its superior performance in producing accurate and well-grounded answers. The deployed system retrieves up to $k = 4$ chunks with similarity scores above 0.6. Retrieved context is then inserted into a structured prompt template. The prompt includes controls for explanation style, enabling adaptive answers tailored to different learner preferences. This is then passed to the OpenAI GPT model to generate a targeted response.

3.3 Frontend Design

The frontend is built with React and styled using TailwindCSS for responsiveness and clarity. It provides a conversational interface with interactive user controls:

- A **technicality slider** adjusts the explanation depth across "simple," "regular," and "in-depth" modes.
- A **knowledge check toggle** prompts the model to generate self-assessment questions after each answer.

To enhance transparency and learning depth, each response includes clickable citations that link directly to the relevant section of the OpenStax Finance textbook. Additionally, when applicable, the system recommends lecture-style YouTube videos, displayed alongside the response and timestamped to the most relevant section. These multimodal references allow learners to explore concepts through both text and audiovisual explanations.

We also implemented Server-Sent Events (SSE) to support real-time streaming of model responses, improving perceived responsiveness and interactivity.

4 Results and Evaluation

To evaluate the effectiveness of our custom RAG-based tutoring system, we conducted both quantitative and qualitative assessments. Our goals were to assess (1) the quality and reliability of the generated responses using RAGAS[2] metrics, and (2) the latency and responsiveness of the system under real-world usage.

4.1 RAGAS Evaluation Metrics

We evaluated 213 sampled question–answer pairs using the RAGAS framework, which measures the alignment between a generated answer, the input question, and the supporting retrieved context. RAGAS produces several distinct metrics:

- **Faithfulness**: The degree to which the generated answer is supported by the retrieved context, i.e., how well the model "sticks to" the retrieved evidence.
- **Answer Relevancy**: The semantic relevance of the generated answer with respect to the input question.
- **Context Precision**: The proportion of retrieved context that is relevant to answering the question.
- **Context Recall**: The proportion of all relevant information (present in the corpus) that was actually retrieved for the question.

We compared our system's results against pre-defined quality thresholds based on prior work:

Table 1: RAGAS Evaluation Metrics vs. Targets

Metric	Score	Target	Met Target?
Faithfulness	0.757	≥ 0.80	No
Answer Relevancy	0.930	≥ 0.90	Yes
Context Precision	0.953	≥ 0.90	Yes
Context Recall	0.673	≥ 0.75	No

While the system performed well in terms of **answer relevancy** and **context precision**, it underperformed slightly in **faithfulness** and **context recall**. The high precision suggests that retrieved chunks were generally relevant, but the lower recall and faithfulness scores indicate that some correct answers lacked sufficient grounding in the retrieved evidence, potentially due to the LLM incorporating prior knowledge beyond the retrieved context.

4.2 Latency Analysis

We recorded response generation times to assess the system's responsiveness. Two metrics were measured: (1) total latency, defined as the time from query submission to complete response generation, and (2) time to first token (TTF), defined as the time from query submission to the model's first streamed output.

Table 2: Latency and Time to First Token Summary (in seconds, $n = 213$)

Statistic	Total Latency	Time to First Token (TTF)
Mean	13.81	0.76
Std. Dev.	5.32	0.24
Min.	4.83	0.49
25%	9.87	0.62
Median	13.07	0.68
75%	16.90	0.77
Max.	29.87	2.40

The system maintained a reasonable median total latency of 13.07 seconds, with most responses delivered under 17 seconds. Time to first token was significantly faster, with a median of 0.68 seconds, contributing to a responsive and conversational user experience. While RAG architectures can introduce additional overhead compared to purely generative systems—due to embedding, vector search, and context injection—we are satisfied that our implementation achieves low latency without compromising retrieval quality or user experience.

4.3 Qualitative Observations

Qualitative review revealed that the system consistently produced structured, step-by-step explanations that were both clear and contextually grounded. Responses aligned well with the instructional tone we aimed to model and demonstrated an ability to guide learners through underlying reasoning—not just deliver answers.

The system was explicitly designed with an instructional philosophy in mind: to emulate the behavior of a skilled human tutor. Rather than merely providing solutions, the model was prompted to walk students through the logic behind each answer, simulating a tutoring experience focused on conceptual understanding and confidence building.

The knowledge check feature also performed reliably, generating relevant follow-up questions that encouraged students to reflect and engage more deeply with the content. These formative assessments supported active recall and self-directed review.

In addition, the system now integrates links to relevant YouTube videos when appropriate. These videos are timestamped to align with the queried topic and are presented alongside the text-based response. Early user feedback suggests this multimodal strategy enhances comprehension, particularly for learners who benefit from audiovisual explanations. Both students and instructors have responded positively. According to one report, the tool has already helped students gain confidence and clarity in challenging material [1].

5 Future Work

Several opportunities exist to further develop and evaluate Finance Tutor AI. A key direction involves conducting structured human studies to assess the educational effectiveness of specific system features. For example, future experiments could measure learning gains, retention, and user engagement across different interface settings (e.g., explanation depth levels or presence of knowledge checks) to better understand how these controls impact student outcomes.

Another area for development is the refinement of the knowledge check mechanism. While the current implementation reliably generates basic multiple-choice questions, it does not yet respond to incorrect answers with tailored feedback. Introducing corrective explanations conditioned on the learner's answer choice would increase the instructional value and effectiveness of the knowledge checks. In addition, expanding the variety of question types (e.g., fill-in-the-blank, matching, or numeric input) would support a broader range of cognitive skills and learning preferences.

Finally, as usage grows, we aim to productionize the system to serve multiple classes and instructors across institutions. This aligns with a broader vision for AI-enhanced self-directed learning—where personalized educational tools complement in-person and hybrid instruction across different platforms.

6 Summary

Finance Tutor AI illustrates the potential of domain-specific Retrieval-Augmented Generation systems to deliver accurate, explainable, and adaptable educational support. The system performs strongly on relevance, context precision, and usability—particularly in guiding students step by step through financial reasoning. While some challenges remain, the system already mirrors key traits of a skilled tutor: clarity, accuracy, and responsiveness.

This work underscore the importance of careful data curation, retrieval strategies, and prompt engineering in building reliable RAG systems. As we continue development, our goal is to evolve

Finance Tutor AI into a production-ready tool that complements existing instruction while empowering learners through personalized, on-demand tutoring.

References

[1] Nick Houtman. Online finance tutor boosts student confidence, 2024. Oregon State University Business Matters Blog, Accessed May 2025.

[2] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kulkarni, Yash Sun, Siyan Pang, Henry Pitt, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[4] OpenStax. *Principles of Finance*. OpenStax, Rice University, 2022. Accessed May 2025.

[5] Yuning Xu, Wayne Xin Zhao, Yang Song, Jian-Yun Nie Zhou, and Ji-Rong Wen. Retrieval-augmented generation with diverse sources improves robustness of qa models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5672–5686, 2022.