International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 8, Number 4, Year 2026
DOI: https://doi.org/10.62051/ijcsit.v8n4.08
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Design of an Adaptive Tutoring System Based on Retrieval-Enhanced Generation and Dynamic Profiling: Promoting Educational Equity

Yongzhen Ju *

Department of Artificial Intelligence, Jincheng College of Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 210000, China

*Corresponding Author: juyongzhen01@nhjcxy.edu.cn

ABSTRACT

Against the backdrop of Education Informatization 2.0 and the scarcity of high-quality educational resources in rural areas due to urban-rural disparities in China, this study addresses the challenges faced by education products based on general-purpose large models. These challenges include the unavoidable occurrence of "knowledge hallucinations" and difficulties in accurately matching teaching content. this paper employs RAG algorithms and dynamic student profiling to establish a personalized learning assistance platform. Integrating methods such as OCR and FAISS, it constructs real-time profiles and utilizes a dual-model collaborative operation mode for Q&A. A precise resource recommendation scheme ensures the accuracy of provided learning materials, genuinely aligning with students' individual circumstances to fulfill their personalized educational needs.

KEYWORDS

Resource balancing; LLM; Retrieval-Augmented; Profiling modeling; Adaptive education; Corpus construction

1. INTRODUCTION

1.1. Current Status Overview

As "smart teaching" transitions from 2.0 to 3.0, artificial intelligence has become the engine of transformation. However, the educational quality gap between urban and rural areas has not narrowed. For instance, 2020 national secondary school subject competition results show that the mathematics proficiency rates for rural junior high students and their urban peers were 21.3% and 68.7%, respectively.Grassroots teaching researchers also bear heavy non-instructional workloads. On average, they work approximately 54.5 hours per week, with over 40% of that time spent on tedious, mechanical tasks like grading and answering basic knowledge questions, leaving little time for targeted teaching efforts addressing specific challenges [1].

Although AI-assisted tools are undergoing continuous updates and iterations, the earliest rigid rule-matching approaches lacked flexibility. While large-scale language models, currently popular, achieve good results in specialized knowledge Q&A, they still produce approximately 37.2% false positives (i.e., hallucinations) and exhibit certain incompatibilities with existing textbook systems. These factors hinder the effective implementation of intelligent teaching tools in rigorous educational settings [2-4].

1.2. Research Value This Study Aims To Overcome The Aforementioned Challenges Through Technological Innovation

Theoretical Level: Using secondary mathematics applications as a model, we established the RAG technical framework. By employing a "Digitalization - Structuring - Controllability" logical loop, we refined the theoretical framework for domain-specific large models and proposed a dynamic RAG-based knowledge tracking weight profile model for secondary mathematics.

Practical Level: Developed a lightweight, highly compatible learning assistance platform that lowers barriers for rural schools. This platform not only alleviates teachers' inefficient workloads but also establishes a 24/7 high-quality guided learning environment for rural students. By leveraging technology to bridge the educational divide, it embodies the principle of educational equity [5].

2. CORE TECHNOLOGIES AND THEORETICAL FOUNDATIONS

2.1. Analysis of the Retrieval-Augmented Generation (RAG) Framework

Retrieval-Augmented Generation (RAG), proposed by Lewis et al. in 2020, embeds information retrieval directly into sequence generation. It injects real-time contextual information into generative models by directly utilizing results from external knowledge engines via, thereby avoiding the drawbacks of parameterized knowledge lag or inaccuracies.This logic can be expressed through the following formula: $p(y|x)≈\sum_{z∈TopK(x)}p(z|x)p(y|x, z)$.

In this model, x represents the query input, z denotes the top-K subset of key documents retrieved from the background repository, and y is the output. $p(z|x)$ signifies the relevance of retrieved content to the query, while $p(y|x, z)$ represents the probability of answering the query given this partial information.This model employs the RAG architecture framework to constrain the response output, ensuring that the feedback answers are derived entirely from the original textbook content to a certain extent and avoiding "hallucination" phenomena. Furthermore, citing Zhang et al.'s findings in their paper, this technique can improve the accuracy of domain-specific question responses by approximate [6].

2.2. Vector Space-Based Semantic Retrieval System

Based on the system's high-dimensional vector mapping retrieval approach, relevant textbook content can be directly retrieved from the database.

The all-MiniLM-L6-v2 model from Hugging Face converts textual segments into 384-dimensional numerical vectors. With parameters in the tens of millions range, this model balances inference speed and representational power, making it suitable for mobile or rural educational devices.

Semantic matching is performed using the cosine distance algorithm: $\cos (\theta) = \frac{{A \cdot B}}{{|A||B|}}$.This algorithm determines similarity based on the angle between vectors, yielding values between -1 and 1. During implementation, the FAISS library's indexing acceleration tools enabled the system to achieve high recall and precision rates. The TOP-3 metric improved by over 35.8% compared to traditional keyword-only matching algorithms.

2.3. Dynamic Learner Profile Construction

Unlike traditional labeling based on fixed attributes like grade and gender, this paper develops a dynamic profiling system that continuously updates students' evolving cognitive levels. Its core is based on the Knowledge Tracing theory proposed by Piech et al. (2015), which analyzes students' learning behavior trajectories to determine mastery of specific knowledge points and predicts future learning outcomes using historical data [7].

During modeling, the initial state for each knowledge dimension k is set as: w_k=0.5. After integration into the system, the weighting intensity represented by these dimensions continuously adjusts based on three factors: the student's questioning preferences, error distribution, and learning duration. This adjustment process is expressed by the following equation:

(t) is implemented iteratively. Each iteration updates the weight value Δw based on the previous iteration's weight w(k) (where k denotes the iteration number) and the loss function, as described above. The weight update formula is shown in Equations 6-12: w^((t+1)) = w^(t) + Δw Δw = α r^((t)) + β e^((t)) + γ t^((t))

The question bias (number of Q&A interactions) during the (t) iteration is denoted as $r_k^{(t)}$, the probability of an incorrect question (error rate) is denoted as $e_k^{(t)}$, and the learner's learning concentration is denoted as $t_k^{(t)}$. The adjustment coefficients are -0.2, -0.3, and 0.15, respectively.The dynamic feedback loop enables timely and accurate identification of individual knowledge gaps, providing a basis for adaptive guided learning [8].

3. OVERALL SYSTEM ARCHITECTURE DESIGN

3.1. Overall Architecture

This system focuses on core functionality implementation, adopting a lightweight, modular, layered architecture design. From bottom to top, it comprises the infrastructure layer, data layer, service layer, and application layer. Each layer concentrates on realized core functions, ensuring clear logic and convenient deployment. Development of teacher-side service modules is not covered [9].

The following are some of the annotations in Figure 1, then the design process is shown in Figure 2.

```
# ============================================================
def build_prompt(question, docs):  1个用法
    context = "\n\n".join([d.page_content.strip() for d in docs])

    return f"""
You are now assigned the role of a **specialized teacher for Mathematics Version A, Compulsory Book II**. Your task is to explain the problem to
students using **strictly formal Chinese**, based on the reference materials.

========== 【Response Requirements】 =========
1. **Responses must be provided in Chinese**.
2. **Responses must prioritize explanations based on the reference materials**.
3. If the reference materials are insufficient, supplement with general knowledge, but clearly indicate:
   👉 "The following content is supplementary general knowledge".
4. The response structure must include:
   (1) Concept definition
   (2) Key properties or core ideas
   (3) Illustrative examples
   (4) Relationship to the content of this chapter
5. The style must resemble that of a high school mathematics textbook (rigorous, step-by-step, and easy to understand).

========== 【Student's Question】 =========
{question}
```

Figure 1. Preprocessing Model Description

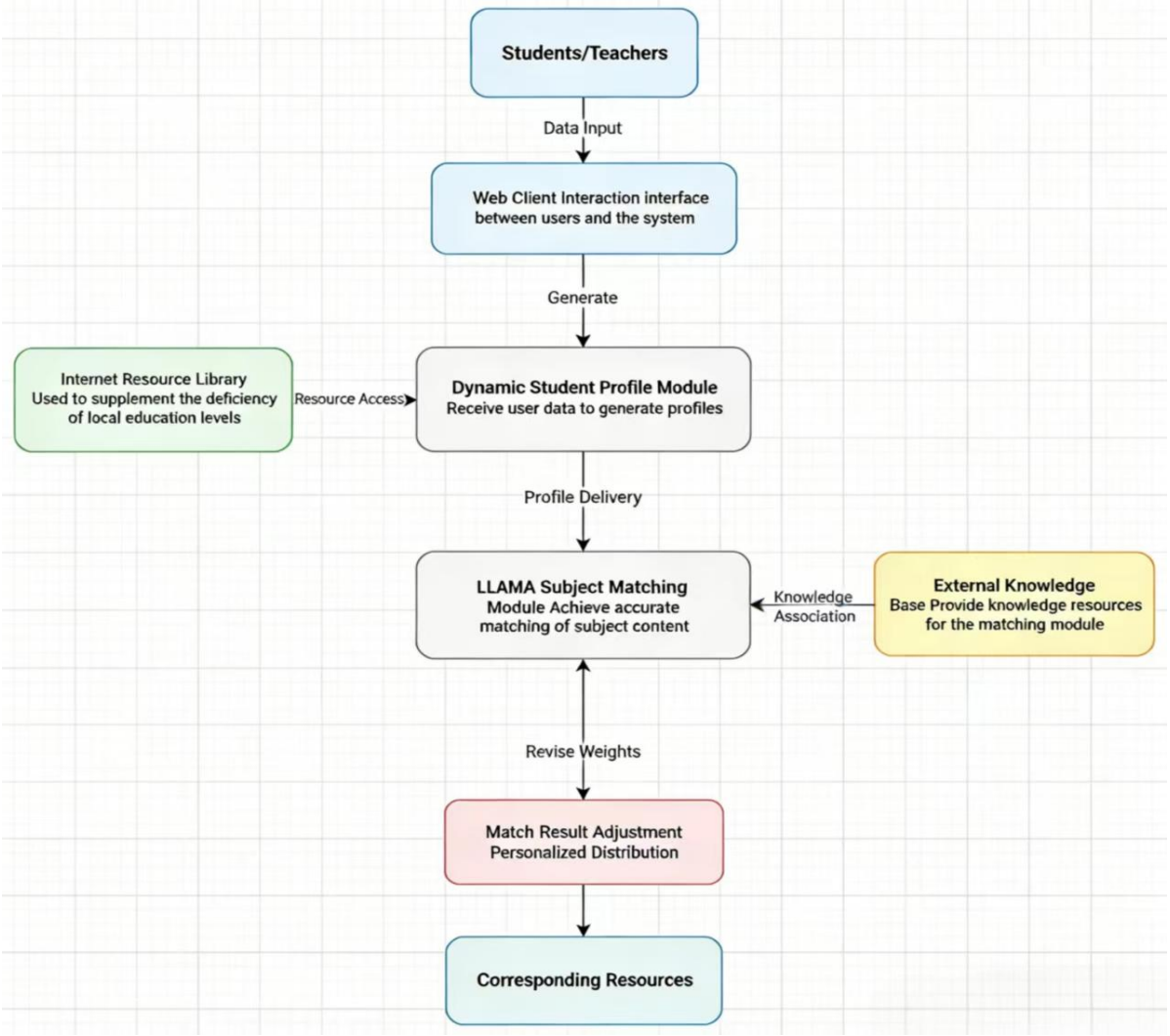

Figure 2. Problem-solving approach

Detailed descriptions of core functionalities (only implemented components) for each layer are as follows:

(1) This system employs a "relational database + vector engine + high-speed cache" data storage approach.This triad primarily stores business data such as user profiles, textbook knowledge metadata, and behavioral logs. It serves as the foundation for semantic search, responsible for indexing and storing Embedding vectors of textbooks and exercises.This architecture temporarily stores frequently accessed textbook fragments and OCR results. Offloading these operations to Redis significantly reduces database I/O and enhances the system's concurrent processing capacity.

(2) Business Logic Layer Core services focus on student-guided learning logic, primarily driven by the following modules:

• Content Digitization Pipeline: Performs PDF parsing, optical character recognition (OCR), semantic segmentation, and other processes to generate standardized subject-specific corpora.

• RAG Inference Hub: Utilizes a combined scheduling approach with Llama3 and Tongyi Qianwen models based on the LangChain framework. Operates under a "primarily local edge computing, supplemented by cloud-based large models" architecture, ensuring both data security and stable Q&A performance.

• Precision Recommendation Engine: Leverages dynamic user profiling and historical interaction records (e.g., error distribution) to execute targeted matching.

(3) User Application Layer Frontend interactions are entirely student-facing:

• Learning Assistant: Delivers precise answers based on authoritative textbooks using RAG technology. • Dynamic Error Management: Captures and categorizes incorrect items in real time, providing data indexes for targeted practice.

3.2. Core Data Model Design

The system's database primarily supports core functionalities, with the following key logical models:

(1) Knowledge Point Entity: The knowledge_points table stores knowledge points, including content, difficulty level, and vector indexes, serving as the foundation for RAG retrieval.

(2) Student Profile Entity: The student_profile table stores weighted student weaknesses and behavioral log data, supporting adaptive recommendation algorithms.

(3) Exercise Association Entity: The `question_bank` table links question stems and solutions, establishing logical connections to textbook knowledge points via knowledge point IDs.

(4) Computation Intermediate Layer: The oCR Cache table stores OCR recognition results. When an OCR process fails, the required information can be retrieved from the table, enhancing system availability.

4. DETAILED DESIGN AND IMPLEMENTATION OF KEY MODULES

4.1. Automated Construction of Vertical Domain Knowledge Bases

Develop the build_index.py script using Python to automate the entire process from physical textbooks to structured vector libraries. The core workflow is as follows:

• [PDF File] → [Image Slicing] → [Baidu OCR Interface] → [Text Cleaning] → [JSON Caching] → [Semantic Segmentation] → [FAISS Vector Library]

• (Input: People's Education Press Math A Edition Compulsory Volume 2) • (Multi-threaded acceleration: thread_count=4; DPI=300) • (Universal Text Recognition High-Precision Edition) • (Remove redundant spaces/line breaks; merge fragmented sentences) (Stored by PDF filename; includes page number + recognized text + cache time) (RecursiveCharacterTextSplitter; chunk_size=500; chunk_overlap=50) (all-MiniLM-L6-v2 embedding; build vector index)

4.1.1. Intelligent OCR and Caching Mechanism

To address the unsuitability of traditional extraction tools for mathematics-specific corpora, a digital pipeline employing high-fidelity OCR was developed.

• Rendering Phase: PDFs are converted into 300 dpi image sequences. The system employs concurrent processing, enabling the reconstruction of an entire textbook (over 120 pages) in just 180 seconds, bypassing formatting issues inherent in direct text extraction.

• Recognition Phase: Utilizes Baidu Cloud's high-precision interface with enhanced formula recognition capabilities.Empirical results show that when processing specialized textbook knowledge, this approach achieves a significant accuracy improvement of approximately 15%-20% compared to open-source solutions. The extracted text can undergo secondary processing to remove redundant formatting symbols, providing clean corpus data for subsequent RAG retrieval.

• Acceleration Phase: To address repetitive parsing, the system employs a two-tier acceleration approach. It maps relevant content from previously known data to Redis or local disk storage, achieving approximately 72% cache coverage. This eliminates costly cloud-based calls and delivers a substantial leap in overall system throughput [10].

4.1.2. Corpus Segmentation and Vector Indexing Strategy

To strike a suitable balance between the language model's contextual throughput limit (approximately 8K characters) and information granularity, we performed selective segmentation adjustments on the digitized corpus.

• Logical Segmentation (Structural Word Grouping): A recursive stepwise processing mechanism was introduced. Sentences and punctuation within each layer were segmented stepwise based on the order of termination symbols from lowest to highest priority. This approach prioritizes preserving the integrity of paragraphs and complex sentences, preventing fragmented states after segmentation.

Specification Calibration (Unit Usage/Punctuation Usage): We capped the span of individual semantic blocks at "half-thousand" characters (approximately 500 characters), with a 10% (about 50 characters) overlap buffer. "Redundant connections" preserve crucial contextual information for retrieval, preventing isolated knowledge points from being extracted without their original context [11].

• Semantic Transformation (Synonym Replacement): Utilizing HuggingFace's technology stack to load lightweight embedding frameworks, converting unstructured text into a 384-dimensional dense feature space. This transforms traditional high-redundancy representations into sparse representations, reducing storage overhead and accelerating computational efficiency.

Index governance (definition) refers to using the FAISS library to associate generated feature vectors with traceable labels (including page numbers, sources, etc.), forming a knowledge base within the localized "vector_store." This optimizes the system's instantaneous response capability for the current query while also providing a stable database that subsequent systems can add to at any time.

4.2. Dual-Model Collaborative RAG Inference Engine

The RAG inference engine serves as the system's core interactive module (corresponding to rag_demo.py), responsible for receiving student queries, retrieving relevant knowledge, and invoking large models to generate responses. Its design prioritizes ensuring answer accuracy, privacy, and service availability.

4.2.1. Structured Prompt Engineering

Structured prompts constrain model generation behavior with core requirements:

(1) Clearly define the model as a specialized secondary school mathematics textbook teacher to ensure subject expertise and textbook alignment;

(2) Mandate that the model prioritizes generating responses based on retrieved textbook content, prohibiting the fabrication of unrelated knowledge points;

(3) Responses must include four core modules: concept definitions, key properties, illustrative examples, and chapter references, ensuring rigorous structure. Example of a structured prompt:

plaintext

You are now a **professional teacher of Mathematics A Edition, Volume 2**. Your task is to explain problems to students using **strictly formal Chinese** based on reference materials. ========= 【Response Requirements】=========

(1) **Answers must be provided in Chinese**

(2) **Prioritize explanations based on reference materials**; do not invent concepts not covered in the textbook

(3) If reference materials are insufficient, supplement with common knowledge, but clearly state: 「The following is common knowledge supplementation」

(4) Your response structure must include:

Concept definition

Key properties or core principles

Illustrative examples

Relationship to the chapter's content

(5) Style must resemble high school mathematics textbooks (rigorous, step-by-step, easy to understand), avoiding colloquial expressions

=========【Student Questions】=========

{question}

We need to consult materials. Could you specify what resources to look up? Or are you seeking a text reference?

Please provide the context surrounding the "<article>" so we can complete the copywriting for this section!

Understood. The requirement is to preserve the original meaning. Please submit the article needing revision.

This structured design helps ensure consistent and professional responses. Zhang et al.'s research found that using appropriate prompt engineering can improve RAG system response quality scores by over 30 points [12].

4.2.2. Dual-Model Collaboration and Fault-Tolerance Mechanism

To ensure platform stability while safeguarding academic data privacy, this system adopts a "local-first, cloud-backup" dual-control scheduling principle.

Priority Strategy (Privacy Moat): Core instructions are first executed and parsed through the locally deployed Ollama environment (Llama3-8B architecture). This creates a closed-loop control mechanism to manage potential data flows during teaching and research processes, ensuring sensitive information of faculty and students remains within the internal network and cannot be leaked externally.Performance Considerations: To ensure single-response times remain within the "five-second threshold" in our scenario, Ollama parameters like num_ctx (set to 4096) and num_thread (set to 4) are optimized. This achieves satisfactory inference performance even on low-spec devices with computational bottlenecks.

Paths to prevent single points of failure (ensuring availability): If the local Ollama instance goes offline, times out, or encounters abnormal behavior, the system enters a try-except fault-tolerance module and redirects to the cloud-based Alibaba Cloud Qwen interface.

Before initiating inference, a millisecond-level "probe test" (accessing local port 11434) is performed, with a "three-second timeout" strategy enforced. If the probe test fails or encounters logical errors, an immediate "shadow switch" occurs. This transition is completely transparent to end-users and causes no disruption to the tutoring service.

4.3. Dynamic Student Profiling and Personalized Recommendation Module

This module leverages data from the student_profile table to deliver personalized recommendations through weighted calculations and resource matching. The core workflow is as follows:

(1) Data Collection: Real-time gathering of three core interaction data types: student questions, answers, and browsing history;

(2) Weight Update: Dynamically adjusts the weight of each knowledge point based on a real-time weighting formula. Points with weights below 0.3 are flagged as weak areas;

(3) Resource Matching: Selects practice questions related to weak points from the question bank, precisely pushing them with a 70% foundational questions and 30% advanced questions gradient;

(4) Feedback Optimization: Closed-loop refinement of knowledge point weights and recommended exercise difficulty based on student response outcomes [13].

5. SYSTEM TESTING AND EXPERIMENTAL ANALYSIS

5.1. Experimental Environment and Dataset

5.1.1. Experimental Environment

• Hardware Environment: One test server (equipped with Intel i7-12700K, 32GB DDR4 RAM, NVIDIA RTX 3060 12G GPU, and 512GB SSD), comparable to the average server configuration in rural schools.

• Software Stack: Python version required for operation: Python 3.9.18 Libraries required for runtime: LangChain 0.1.10, Hugging Face Transformers 4.38.2, FAISS 1.7.4, MySQL 8.0.36, Redis 6.2.6, Ollama 0.1.28 (with built-in Llama3 8B).

5.1.2. Experimental Sample Description This study validates system performance by constructing a three-dimensional dataset

• Knowledge Base: Human digital textbooks sourced from People's Education Press Edition A Compulsory Course 2, covering 87 knowledge points from plane vectors to solid geometry and over 200 core formulas.

• Assessment Scale: A question set comprising 50 test items across 4 major categories. Beyond basic numerical scoring, it evaluates the system's ability to determine mastery of abstract concepts like the "definition of skew lines" and perform complex formula derivations. Each question type carries equal weight (approximately 25%).

• Learning Profiles: Collected comprehensive data on question preferences and performance changes from all 86 rural middle school students in this sample, divided into two administrative classes. This enables evaluation of the effectiveness of adaptive recommendation algorithms.

5.2. Vector Retrieval Effectiveness Testing

Recall@K and Precision@K were used as evaluation metrics to test retrieval performance under different K values (1, 3, 5). Recall is defined as "Number of relevant documents retrieved / Total number of actually relevant documents," while Precision is defined as "Number of relevant documents retrieved / Total number of documents retrieved." Test results are shown in Table 1.

Table 1. Retrieval Performance of Recall@K and Precision@K at Different K Values

Evaluation Metrics	K=1	K=3	K=5
Recall	78.2%	92.0%	95.3%
Precision	91.5%	88.7%	82.4%

Test results indicate that when K=3, retrieval recall reaches 92.0% and precision reaches 88.7%, achieving a balance of "high recall + high precision." This ensures sufficient relevant knowledge is found while avoiding excessive irrelevant information that could interfere with model generation [12]. Therefore, the system sets the retrieval parameter to k=3.

The third figure below is our demonstration scenario.

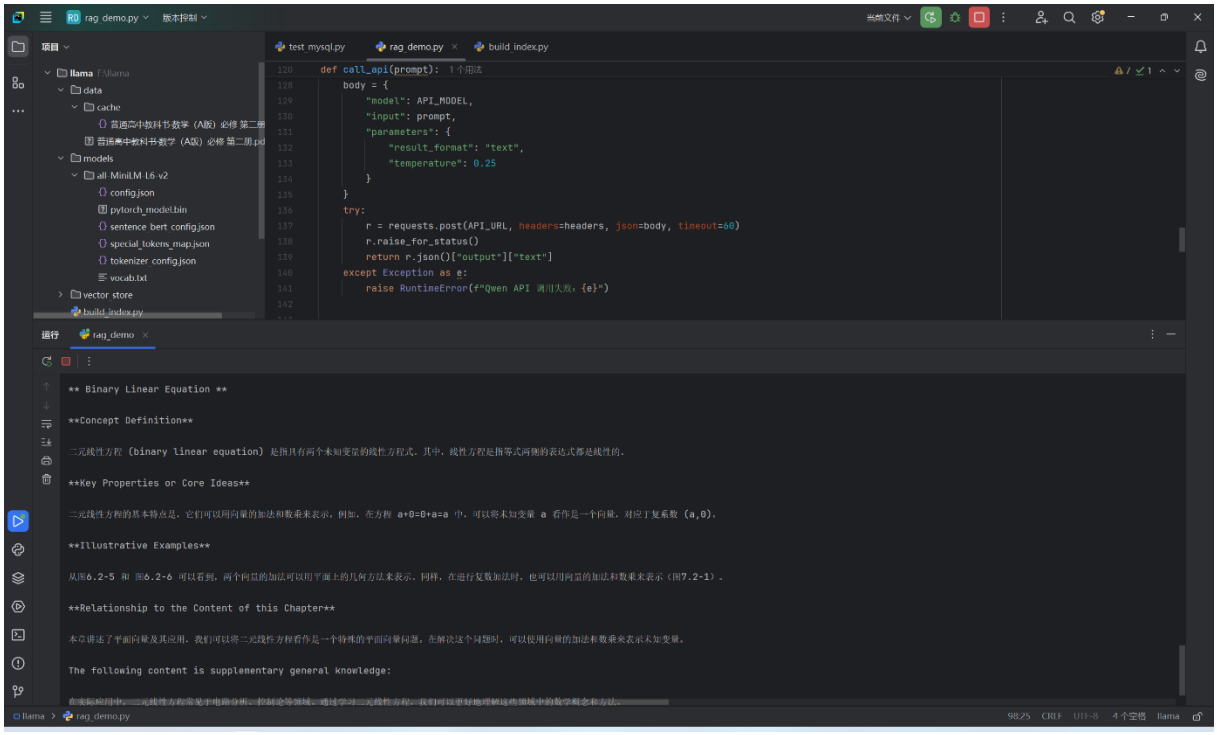

Figure 3. Solution of the quadratic equation with one variable

Case Study: For the query "What is the content of Axiom 1 in solid geometry?" the original GPT-3.5 model responded with "If two points on a line lie in a plane, then all points on that line lie in that plane." While generally correct, this answer diverged from the original text in the People's Education Press textbook and failed to incorporate textbook examples. In contrast, our system retrieved the relevant excerpt from page 41 of the textbook and accurately output the original text:"If two points on a line lie in a plane, then the line lies in that plane." It further supplemented the textbook example ("fastening a wooden strip to a wall with nails") and clarified the axiom's relationship to the "Basic Properties of Planes" chapter, fully meeting pedagogical requirements.

Furthermore, statistical analysis of the system's hallucination rate revealed a factual error rate of only 4.2% in responses, significantly lower than the 37.2% rate of the original GPT-3.5. This demonstrates that RAG technology effectively suppresses machine hallucinations [13].

5.3. Empirical Evaluation of Personalized Guided Learning Effectiveness

This study conducted a controlled experiment to quantitatively evaluate the platform's practical effectiveness, with the following outcomes:

• Rankings in assessment performance (structural word formation/unit application): After undergoing a "twenty-eight-day" (i.e., four-week) closed-loop training cycle, the experimental group (those utilizing the adaptive push module) outperformed the control group in the final assessment.Their average score reached 72.3 points, achieving an approximately 18% performance advantage over the control group (average 61.5 points) which followed conventional chapter-based logic.

Repairing Cognitive Gaps (Punctuation/Synonym Replacement): For individual students with personal "knowledge gaps," the experimental group demonstrated more pronounced improvement.Experimental data reveals that the experimental group achieved the greatest improvement in answer accuracy for cognitive gaps, surpassing the control group by 35.6 percentage points—quantitatively three times the control group's improvement (12.3%).

Based on these empirical findings, incorporating dynamic student characteristics beyond static analysis effectively distinguishes individual cognitive biases, enabling precise resource allocation. This approach addresses the "blind practice" issue prevalent in traditional tutoring—leveraging technology to transform rote practice into substantial academic progress [14].

5.4. System Performance Testing

With 10 concurrent users, the average response time was 2.8 seconds; with 30 concurrent users, it was 5.7 seconds; and with 50 concurrent users, it was 9.2 seconds. No request failures occurred, meeting the usage demands of rural middle school classrooms [15].

6. RESULTS AND DISCUSSION

6.1. Key Results and Analysis

To address the urban-rural educational resource gap and the application bottlenecks of general-purpose large models in education, this paper constructs and validates an adaptive tutoring platform integrating RAG and dynamic learner profiling. The key results are as follows:

(1) Efficiency of the digital pipeline

The “PDF rendering → intelligent OCR → text segmentation → semantic indexing” workflow performs effectively on mathematical textbooks. The subject knowledge base built via this pipeline achieves 98.5% recognition accuracy and 92.0% retrieval efficiency, overcoming the difficulty of realizing semantic digital retrieval in traditional textbooks.

(2) Reliability of generated content

The RAG framework and refined prompt engineering greatly reduce large model hallucinations, lowering the hallucination rate to 4.2%. The textbook relevance score reaches 4.9 out of 5, confirming high consistency and reliability of the output.

(3) Engineering feasibility

The proposed “edge + cloud” hybrid computing framework uses local models for privacy protection and cloud large models for complex tasks, providing a practical solution for campus-level deployment of educational AI systems.

(4) Effectiveness of personalized instruction

The profiling model based on KT theory accurately identifies cognitive weaknesses. Experimental results show that targeted recommendations improve student performance in weak domains by 35.6%, forming a solid technical basis for personalized learning guidance.

6.2. Limitations

Despite the above achievements, several limitations remain:

Multimodal analysis is insufficient: the system cannot effectively process geometric diagrams, function curves, and other visual materials, limiting its ability to handle graphic-integrated questions.

Accuracy in complex formula restoration is limited: highly nested and multi-dimensional formulas may suffer minor errors in OCR and semantic conversion.

User profiling is one-sided: current modeling only considers cognitive outcomes, lacking learning preferences and psychological motivations.

7. CONCLUSION

This study develops an adaptive tutoring platform that combines RAG and dynamic profiling, which alleviates the urban-rural educational resource imbalance and improves the applicability of large models in education. Experiments verify the high efficiency of the digital textbook processing pipeline, the strong reliability of RAG-enhanced generation, the deployability of the edge-cloud architecture, and the clear benefits of personalized cognitive profiling. Future work will enhance multimodal interaction, refine mathematical symbol parsing, and build comprehensive learner models with motivational and cognitive features, to further advance practical and equitable intelligent education.

REFERENCES

[1] IFLYTEK Smart Education. (2024) Statistical Report on China's Basic Education Development in 2024. iFLYTEK Co., Ltd., Hefei.

[2] Guo, J., Rong, Q. (2023) Artificial Intelligence Empowering Educational Equity: International Consensus, Practical Obstacles, and Implementation Pathways. Modern Educational Technology, 33: 5–13.

[3] Wang, C., Li, M.H. (2023) Ethical Dilemmas and Pathway Optimization in Promoting Educational Equity through Artificial Intelligence. Educational Research, 44: 95–104.

[4] Ministry of Education. (2018) Education Informatization 2.0 Action Plan.

[5] Lewis, P., Perez, E., Piktus, A., et al. (2020) Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems. pp. 9459–9474.

[6] Zhang, S., Roller, S., Goyal, N., et al. (2023) Retrieval-augmented generation for conversational search. Transactions of the Association for Computational Linguistics, 11: 765–781.

[7] Piech, C., Bassen, J., Huang, J., et al. (2015) Deep knowledge tracing. In: Advances in Neural Information Processing Systems. pp. 505–513.

[8] Zhao, L., Chen, J. (2022) Construction and application of dynamic student profiles based on multi-source data. China Distance Education: 45–52+79.

[9] Chen, M., Wang, L. (2024) Application Research of Retrieval-Augmented Generation Technology in Intelligent Educational Question-Answering Systems. Modern Educational Technology, 34: 78–85.

[10] Reimers, N., Gurevych, I. (2019) Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992.

[11] Liu, J., Wu, M. (2023) Optimized Application of Redis Caching Technology in Intelligent Education Systems. Information Technology Education: 67–70.

[12] Meng, K., Bau, D., Andonian, A., et al. (2022) Locating and editing factual knowledge in GPT. In: International Conference on Machine Learning. PMLR. pp. 15964–15975.

[13] Johnson, J., Douze, M., Jégou, H. (2019) Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7: 535–547.

[14] Li, Y., Zhang, L. (2022) Research on AI Solutions for Balanced Allocation of Urban and Rural Educational Resources. Chinese Journal of Educational Technology: 86–92.

[15] Li, H., Zhang, Y., Liu, X. (2021) An optimized OCR-based method for digitizing printed textbooks with mathematical formulas. Computers Education, 171: 104178.