Research on a Vertical-Domain Personalized Learning System Based on RAG and Multi-Agent Collaboration

Chao Geng[1], Xiaojun Fan[1], Yonghui Zheng[1], Ruiheng Zhang[1], Yawen Zhao[2], Mingming Gong[3,*]
[1]*College of Information Science and Engineering, Henan University of Technology, Zhengzhou, Henan, China*
[2]*College of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China*
[3]*IFLYTEK Co., Ltd., Hefei, Anhui, China*
**Corresponding Author*

Abstract: Specialized courses frequently encounter fragmented knowledge resources, insufficient adaptation to individual learning needs, and delayed feedback. This work presents a personalized learning system for vertical domains that combines Retrieval-Augmented Generation (RAG) with a multi-agent framework. Learner profiles incorporate prior knowledge, learning goals, weak knowledge points, and preferred difficulty levels, while a domain-specific knowledge base supplies external retrieval content. During resource generation, retrieved knowledge and learner profile data jointly inform the production of reading materials, quizzes, practical tasks, and animation scripts. A review agent evaluates generated resources prior to learner delivery. Learning records—including quiz results, coding performance, task completion, and learning time—update learner profiles and guide subsequent resource and path adjustments. Experimental results show that the proposed system achieves better resource adaptability than direct LLM generation and supports personalized learning in vertical domains.

Keywords: Vertical Domain; RAG; Multi-Agent Collaboration; Personalized Learning; Learner Profile; Intelligent Tutoring

1. Introduction

Personalized learning is widely studied in smart education, especially in courses that involve specialized domain knowledge [1-3]. Large language models have been introduced into educational applications, but general-purpose models may not provide sufficiently accurate information for some domain-specific tasks [4-5]. Existing learning platforms also have limitations in adjusting learning paths according to learner profiles and learning behavior [6-7]. In addition, traditional assessment methods often rely on test results and provide limited analysis of learning errors and weak knowledge points [8-9]. These problems indicate that domain knowledge, learner profiles, and learning process data should be considered together when designing personalized learning systems for vertical domains [10].

In this study, RAG is used to retrieve relevant information from a vertical-domain knowledge base and provide external knowledge for content generation [11-13]. A multi-agent framework is further introduced to support different learning tasks [14-16]. Based on these methods, we develop a personalized learning system that covers learner profile construction, resource generation, learning path planning, learning feedback, and performance evaluation. Figure 1 presents the overall system architecture.

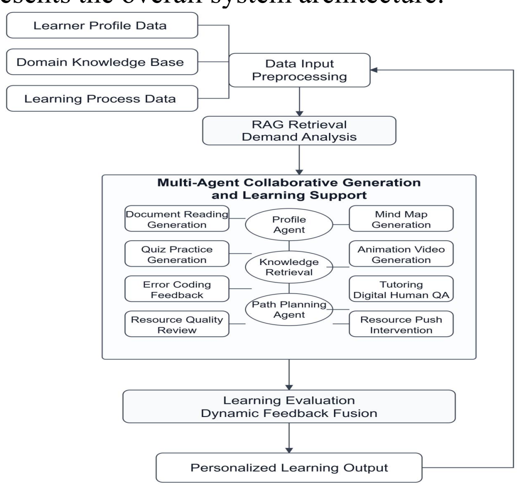


Figure 1. Overall Architecture

(1) Learner Profile Construction and Updating.

The initial learner profile is constructed using information such as academic background, prior knowledge, learning goals, and preferred question types. During the learning process, the system records test results, coding performance, learning time, and error information. These data are used to update the learner profile and provide information for subsequent resource generation and learning path adjustment [1,8].

(2) Resource Generation Based on RAG and Multi-Agent Collaboration. The RAG module retrieves relevant information from the vertical-domain knowledge base and provides it as contextual information for learning resource generation [11,12]. Different agents are responsible for different tasks, including knowledge retrieval, requirement analysis, document generation, quiz generation, and mind map generation [14,15]. The generated resources may also include animation scripts and reading materials. A review process is applied before the resources are provided to learners. In this way, resource generation is based on both domain knowledge and learner profile information.

(3) Learning Path Planning and Performance Evaluation. The system generates personalized learning paths according to learner profiles and the retrieved domain knowledge [6,10]. Learning performance is evaluated using data such as quiz results, coding scores, learning progress, and error records. The evaluation results are presented through capability analysis and learning reports. Based on the identified weak knowledge points, the system recommends corresponding learning resources and adjusts subsequent learning activities [8,9].

These modules form the main workflow of the proposed personalized learning system for vertical-domain learning.

2. Methodology

2.1 Multi-Source Learning Data Collection and Feature Modeling

2.1.1 Data Description

The system relies on three data categories: domain knowledge, learner profiles, and learning process records. The vertical-domain knowledge base stores chapter content, supplementary readings, and instructional documents for RAG retrieval. Learner profiles capture academic background, grade level, prior knowledge, learning goals, preferred question types, weak knowledge points, and recommended difficulty. Learning process records track quiz scores, coding results, error records, learning path progress, and learning time accumulated during system use. These data support knowledge retrieval, learner modeling, resource generation, path planning, and performance evaluation. Figures 2, 3, and 4 display the knowledge base, learner profile data, and learning path/resource generation data, respectively.

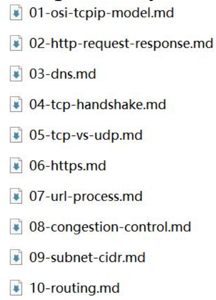
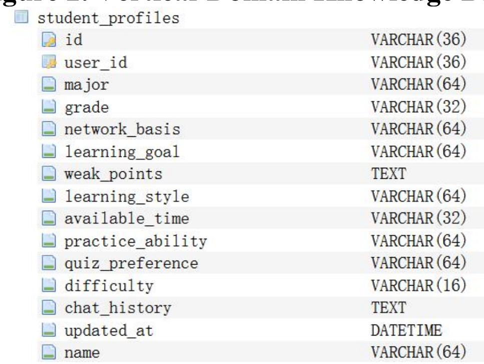


Figure 2. Vertical-Domain Knowledge Base



student_profiles	
id	VARCHAR(36)
user_id	VARCHAR(36)
major	VARCHAR(64)
grade	VARCHAR(32)
network_basis	VARCHAR(64)
learning_goal	VARCHAR(64)
weak_points	TEXT
learning_style	VARCHAR(64)
available_time	VARCHAR(32)
practice_ability	VARCHAR(64)
quiz_preference	VARCHAR(64)
difficulty	VARCHAR(16)
chat_history	TEXT
updated_at	DATETIME
name	VARCHAR(64)

Figure 3. Learner Profile Data

learning_paths	
id	VARCHAR(36)
profile_id	VARCHAR(36)
user_id	VARCHAR(36)
title	VARCHAR(256)
stages	JSON
recommend_reason	VARCHAR(1024)
created at	DATETIME

Figure 4. Learning Path and Resource Generation Data

2.1.2 Learning Data Processing

Domain documents are structured by chapter and knowledge point, then segmented into smaller chunks and converted into vector representations to support RAG retrieval. Figure 5 presents the processed knowledge base. Learner profile fields—such as academic background, current ability, and preferred difficulty—undergo standardization before use. Missing fields are collected through a brief dialogue, as illustrated in Figure 6. Learning resources, quiz records, and path data are stored with associated user IDs and profile IDs to enable subsequent evaluation and resource recommendation.

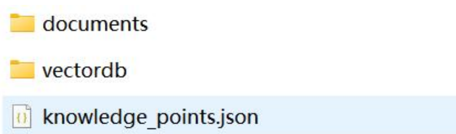
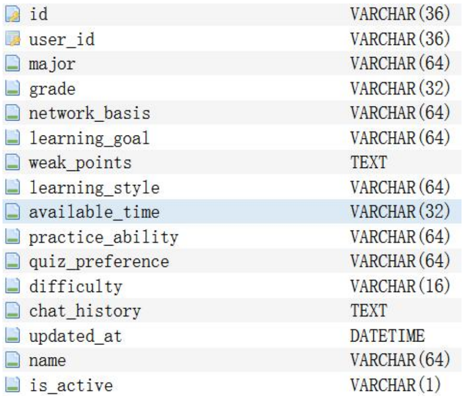


Figure 5. Processing Results of Vertical-Domain Knowledge Base



id	VARCHAR(36)
user_id	VARCHAR(36)
major	VARCHAR(64)
grade	VARCHAR(32)
network_basis	VARCHAR(64)
learning_goal	VARCHAR(64)
weak_points	TEXT
learning_style	VARCHAR(64)
available_time	VARCHAR(32)
practice_ability	VARCHAR(64)
quiz_preference	VARCHAR(64)
difficulty	VARCHAR(16)
chat_history	TEXT
updated_at	DATETIME
name	VARCHAR(64)
is_active	VARCHAR(1)

Figure 6. Data Processing Workflow of Learner Profiles

The system maintains domain documents, learner profiles, generated resources, learning paths, and feedback records. Domain documents undergo vectorization at initialization and are stored locally for later similarity retrieval. Learner profile data flow between the web module, mobile mini-program, and backend services. Generated resources and learning path records reside in the database, linked to their respective user and profile information. The data are organized using the structure "User ID — Profile ID — Resource Type — Knowledge Point," which is used for subsequent data retrieval, profile updating, and learning evaluation.

2.1.3 Learning Feature Extraction

Feature extraction is performed on domain knowledge, learner profile data, and learning process data. In the domain knowledge module, course documents and knowledge points are represented as embedding vectors. For each learner query, the RAG module computes the semantic similarity between the query embedding and the embeddings of stored knowledge chunks. The retrieval results are ranked according to their similarity scores, and the higher-ranked chunks are used as contextual information for subsequent resource generation. The semantic similarity is calculated as follows:

$$\mathrm{Sim}(q, d_i) = \frac{q \cdot d_i}{\|q\| \|d_i\|} \tag{1}$$

Here, q denotes the vector representation of the learner query, and d_i denotes the vector representation of the i -th knowledge chunk. $\mathrm{Sim}(q, d_i)$ represents the semantic similarity between the query and the corresponding knowledge chunk. Knowledge chunks with higher similarity scores are ranked higher and are used as contextual information for subsequent resource generation or question answering.

The learner profile contains information on academic background, prior knowledge, learning goals, learning habits, practical ability, preferred question types, and recommended difficulty. To determine whether the profile contains the information required for subsequent processing, the completeness of the profile is evaluated before it is used by other system modules. The profile completeness score is calculated as follows:

$$C = \frac{N_f}{N} \times 100\% \tag{2}$$

Here, C is the profile completeness score. N_f is the number of completed profile fields, and N is the total number of profile fields. A larger value of C represents a more complete learner profile.

Mastery of each knowledge point is determined by combining learning progress, quiz scores, error records, and practice frequency, reflecting the learner's current level for that point. The mastery score is calculated as follows:

$$M_k = 0.6Q_k + 0.4P_k - \alpha W_k + \beta R_k + \gamma T_k \tag{3}$$

Here, Mk denotes the mastery score for knowledge point k. Qk, Pk, Wk, Rk, and Tk represent quiz score, learning progress, unresolved errors, corrected errors, and practice frequency, respectively. Coefficients α, β, and γ weigh each factor's contribution. Mk is recalculated whenever new learning data are recorded.

A comprehensive performance score is computed to assess the learner's overall learning state. The calculation incorporates goal achievement, knowledge mastery, learning activity, learning consistency, practical ability, and error correction, each weighted as shown in the following equation:

$$\begin{aligned} S &= 0.30M + 0.25A + 0.15L \\ &+ 0.10B + 0.10P + 0.10E \end{aligned} \tag{4}$$

In this equation, S denotes the overall performance score. M, A, L, B, P, and E correspond to knowledge mastery, goal achievement, learning activity, learning consistency, practical performance, and error correction, respectively. These indicators are combined to evaluate overall learning performance. The resulting score informs subsequent learning path adjustment and resource recommendation.

2.2 RAG-Driven and Multi-Agent Collaborative Personalized Learning System

A vertical-domain knowledge base supplies external information for resource generation. The RAG module retrieves relevant knowledge chunks and passes them, along with learner profile data, to the iFlytek Spark Ultra 4.0 model.

The model produces documents, quizzes, and practical tasks based on the retrieved content and learner characteristics. DeepSeek models review generated resources for content structure and domain accuracy; those failing review are revised before delivery.

The multi-agent framework operates across each learning cycle. Resource quality and learning path adaptability are assessed during system operation. Upon completion of a learning stage, test results and error correction records are stored for later analysis. These historical data update learner profiles, inform subsequent assessments, and refine resource recommendations.

2.2.1 Agent Architecture Design

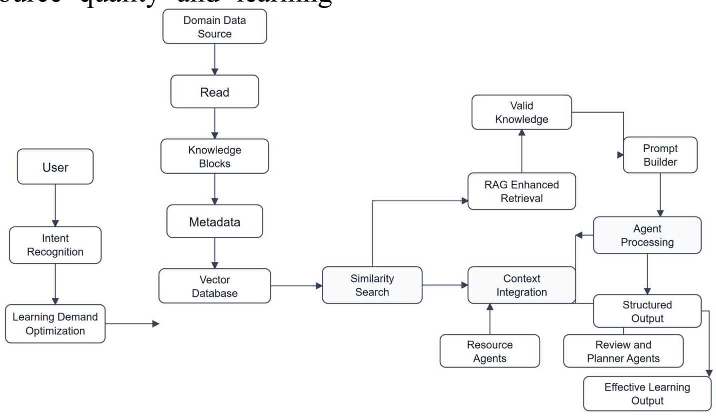


Figure 7. Agent Architecture Diagram

Figure 7 illustrates the agent architecture. A learner request is first parsed into a task description. Knowledge base documents are chunked, tagged with metadata, and stored as vectors. For each request, the RAG module computes similarity scores between the query and stored chunks, retrieving relevant content. The retrieved knowledge and dialogue context feed into different agents, which handle resource generation, path planning, and content review per their assigned roles. Outputs are returned to the system for subsequent learning activities.

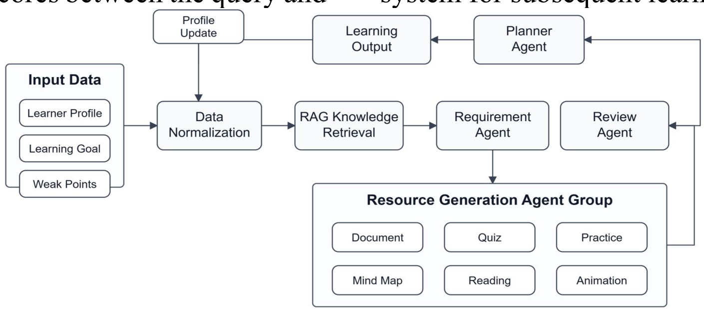


Figure 8. Personalized Learning Agent Workflow

2.2.2 Agent Workflow Construction

The workflow comprises five stages from data processing to output. First, learner profile information, learning goals, weak knowledge points, and historical progress are collected and standardized. Second, the retrieval agent obtains relevant domain knowledge via RAG, while the requirement analysis agent determines content needs from the learner profile. Third, generation agents produce resources in multiple formats—documents, quizzes, mind maps, and animation scripts. Fourth, these resources undergo review for content quality, domain accuracy, and difficulty suitability. Finally, the path planning agent combines learner goals with the domain knowledge structure to construct a staged learning path. Generated resources, the learning path, and evaluation results are returned to the front-end interface. Figure 8 shows the complete workflow.

2.2.3 Dynamic Learner Profile Updating Mechanism

Learner profiles are updated during the learning process based on predefined events, including test completion, knowledge point mastery, and learning session completion. These events trigger updates to the learner's prior knowledge level, recommended difficulty, and weak knowledge points. The system also stores the corresponding learning records and capability radar chart snapshots. The profile update is defined by the following function:

$$P_{t+1} = F(P_t, E_t, C_t) \quad (5)$$

Here, P_t and P_{t+1} represent the learner profiles before and after an update. E_t denotes the triggering event, and C_t denotes the corresponding contextual information. Quiz results are used to update the recommended difficulty and weak knowledge points. A knowledge point is added to or removed from the weak-point set according to its assessment accuracy and a predefined threshold. The corresponding update rule is given as follows:

$$W_{t+1} = \begin{cases} W_t \cup \{k\}, & r_k < 0.4 \\ W_t \setminus \{k\}, & r_k \geq 0.8 \\ W_t, & otherwise \end{cases} \quad (6)$$

Here, W_t represents the set of weak knowledge points before the update. k denotes the knowledge point being evaluated, and r_k represents the learner's accuracy for knowledge point k. The weak-point set is updated according to the learner's performance and the predefined accuracy threshold. The updated results are used to identify knowledge points that require further practice and to support subsequent resource recommendation.

The learner's knowledge level and recommended difficulty are updated after the completion of a learning stage. Stage completion is measured using the following completion ratio:

$$R_s = \frac{S_c}{S_t} \quad (7)$$

Here, R_s is the stage completion ratio, S_c is the number of completed stages, and S_t is the total number of stages. When R_s reaches the predefined threshold, the learner's knowledge level and recommended difficulty are updated. The difficulty level can shift from "Basic" to "Standard" or from "Standard" to "Advanced." Capability radar chart snapshots are stored periodically to track changes in learner characteristics. The corresponding feature vector is defined as:

$$V = [B, G, W, P, T, D, M, A] \quad (8)$$

In this vector, B, G, and W denote baseline level, goal clarity, and awareness of weak knowledge points. P represents practical ability, and T denotes learning time. D, M, and A correspond to current difficulty level, knowledge mastery, and learning activity. These feature vectors are stored at various learning stages to track changes over time. Historical records support subsequent path adjustment and performance evaluation.

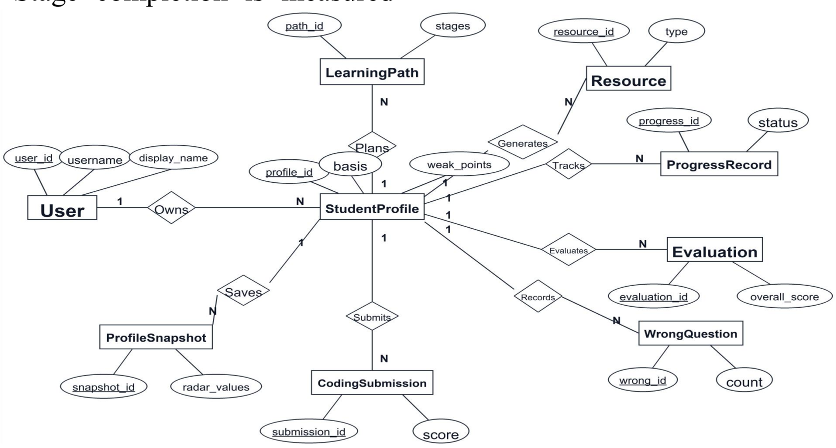


Figure 9. Database E-R Diagram

2.2.4 Database Technology

The backend employs SQLite for storage and SQLAlchemy for object-relational mapping between application models and database tables. The database holds records for users, learner profiles, learning paths, generated resources, learning progress, error records, coding exercises, performance evaluations, digital human sessions, and system announcements. Records are linked via User IDs and Profile IDs to support retrieval and management across learner profiling, resource generation, path planning, and feedback. Figure 9 shows the entity-relationship (E-R) diagram.

2.2.5 Implementation of RAG Retrieval and Resource Generation

After the learner profile, target knowledge points, and resource format are determined, the system performs retrieval, requirement analysis, resource generation, review, and learning path planning. ChromaDB is used to retrieve relevant knowledge chunks from the vertical-domain knowledge base through similarity search. A Top-K strategy is applied to select the higher-ranked chunks for the current knowledge point. The retrieved content is then organized and inserted into the prompt template as contextual information for subsequent resource generation.

Resource generation is performed through collaboration among several functional agents.

The retrieval agent obtains relevant domain knowledge, and the requirement analysis agent determines the resource requirements according to the learner profile. Different generation agents are then used to produce documents, mind maps, quizzes, practical exercises, and animation scripts. The generated resources are reviewed in terms of domain accuracy, content completeness, and difficulty suitability. After review, the path planning agent organizes the approved resources into a staged learning path. The generated resources, review results, and learning path are stored in the database for subsequent learner profile updating and learning evaluation.

2.2.6 RAG Retrieval Mechanism and Prompt Engineering

Course documents are converted into structured Markdown format and encoded using the all-MiniLM-L6-v2 embedding model. The generated vectors are stored in ChromaDB. Retrieval queries are constructed from the learner's current knowledge point, weak knowledge points, and learning goals. The RAG module ranks the stored knowledge chunks according to their similarity scores and selects the Top-K results. These retrieved chunks are incorporated into the prompt template to provide domain-specific context for the generation agents.

Prompt templates are designed separately for different agents. The requirement analysis agent uses learner profile information and knowledge-point information to determine resource requirements. Generation agents receive prompts that define the resource type, structure, difficulty, and domain constraints. The review agent evaluates the generated content for accuracy, difficulty, and learning relevance, and returns the evaluation results in JSON format. The prompts also include retrieved knowledge and output format requirements for subsequent processing.

3. Application, Validation, and Simulation

3.1 Datasets and Experimental Setup

The dataset contains four main types of data. Domain data include course structures, source materials, and vectorized documents for RAG retrieval. Learner profile data include academic background, prior knowledge, learning goals, and learning preferences. Learning path data record the stages and tasks generated for individual learners. The dataset also includes generated resources such as documents, quizzes, practical exercises, and mind maps, all used in subsequent system evaluation.

The proposed system was compared against two baselines: direct LLM generation and a conventional RAG approach without learner profile constraints. Four aspects were assessed: Knowledge Retrieval Accuracy (relevance of retrieved knowledge to the target concept), Resource Generation Adaptability (consistency between generated resources and learner profiles), Learning Path Rationality (organization of generated learning stages), and Learning Feedback Effectiveness (whether recommendations adjust based on learner performance and process data).

3.2 Experimental Results and Analysis

The proposed system was assessed on knowledge retrieval accuracy, resource generation adaptability, learning path rationality, and learning feedback effectiveness. Results were compared with a direct-generation LLM baseline, as shown in Figure 10.

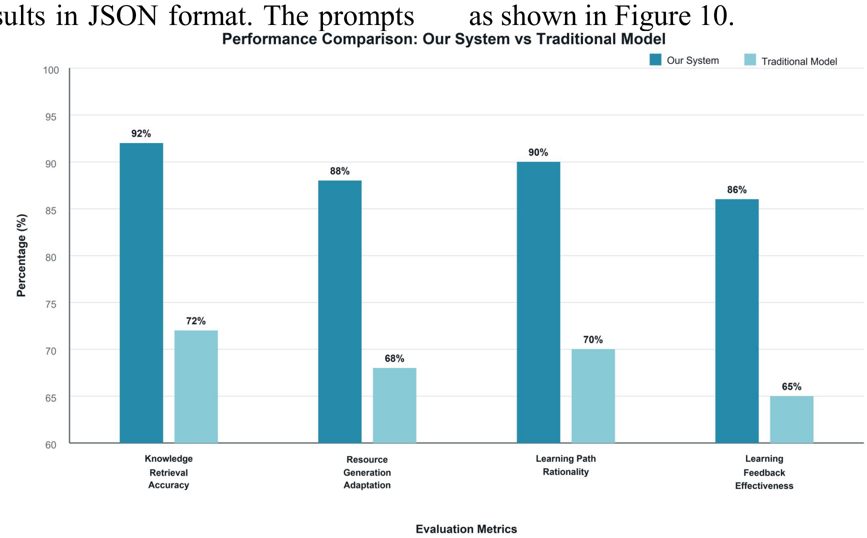


Figure 10. Comparison of Evaluation Metrics

The proposed system outperformed the direct-generation LLM baseline on all four metrics. As Figure 10 shows, the proposed architecture consistently exceeds the baseline across all evaluated dimensions. Domain-specific retrieval raised knowledge retrieval accuracy from 72% to 92%, reducing hallucination risks. With dynamic learner profiling, resource generation adaptability reached 88% versus 68% for the baseline, indicating that personalized difficulty alignment and weakness-targeted prompting improve content relevance. The collaborative agent framework further raised path rationality to 90% and feedback effectiveness to 86%, demonstrating that multi-agent orchestration yields more coherent learning sequences than static, one-shot LLM generation.

3.3 Case Study

A representative learner case was selected to demonstrate system operation. The learner exhibited a relatively strong academic background but had weaknesses in core concept understanding, knowledge application, and practical tasks. Using the learner profile—covering learning goals, prior knowledge, identified weak points, and preferred difficulty—the system generated retrieval queries against the vertical-domain knowledge base. Relevant chunks retrieved via the RAG module were passed to agents for resource generation and path planning. Generated resources included explanatory materials, quizzes, practical tasks, and mind maps. The path planning agent organized these into a staged learning sequence based on the learner's knowledge state and objectives. Figure 11 shows the generated resources, and Figure 12 presents the personalized learning path.

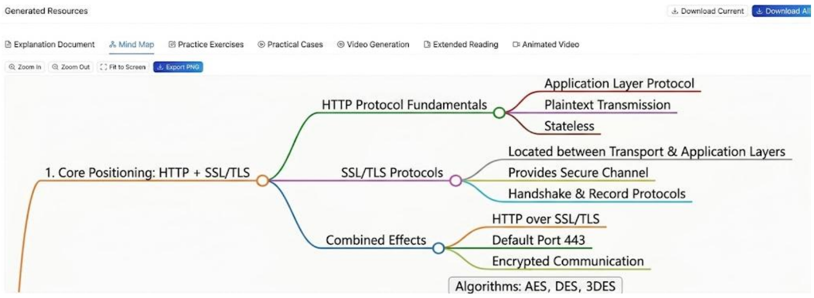


Figure 11. RAG-Enhanced Personalized Learning Resource Generation Results

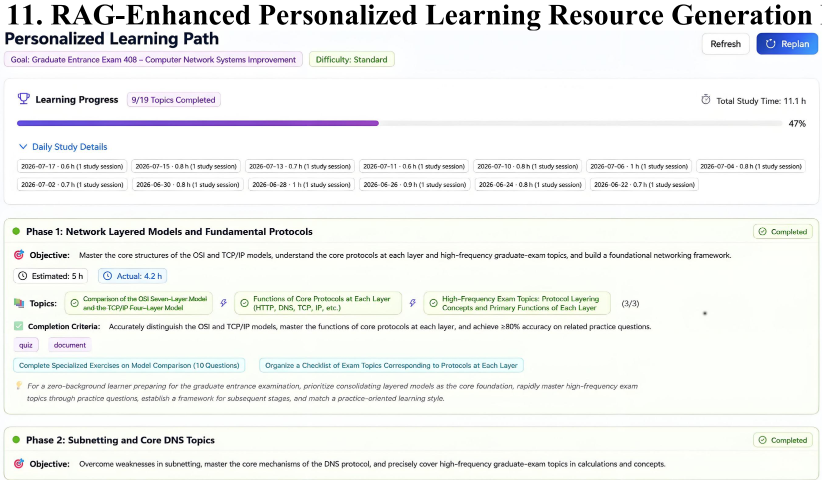


Figure 12. Personalized Learning Path Planning Results

The case study indicates that RAG-based generation produces learning materials more closely aligned with the learner's weak points and relevant domain knowledge. The learning path was divided into stages: basic concept learning, targeted knowledge reinforcement, practical application, and feedback-based review. This structure allows content to adapt to the learner's current knowledge state and progress. Compared with direct LLM generation, the proposed system yielded resources more consistent with the learner profile and a more organized learning sequence. These findings suggest that combining RAG, learner profiling, and path planning enhances the relevance and organization of personalized learning resources in vertical domains.

3.4 Simulation of Personalized Learning Agent

Applications

3.4.1 Learner Profile Analysis Agent

The learner profile analysis agent builds a profile from academic background, learning goals, prior knowledge, weak knowledge points, learning preferences, available learning time, practical ability, and recommended difficulty level. These attributes come from user input and are updated using learning records collected during system use. Based on the profile, the system evaluates prior knowledge, goal clarity, weak points, practical ability, learning time, and difficulty level. The resulting profile serves as input for RAG retrieval, resource generation, and learning path planning.

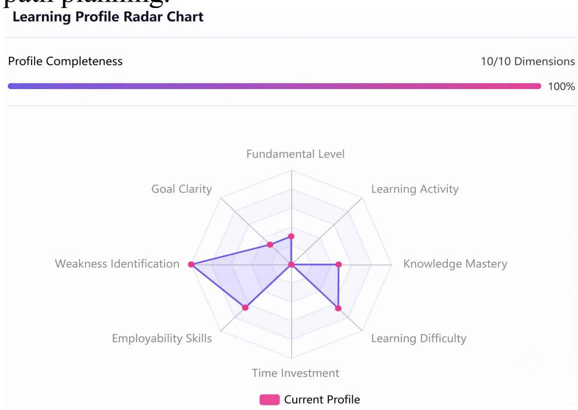


Figure 13. Results of Learner Profile Analysis

A representative profile was used to illustrate the agent's output. The learner had a solid basic knowledge level but struggled with core concept understanding, knowledge integration, and practical application. From these characteristics, the agent identified areas requiring improvement and adjusted recommendations accordingly—emphasizing weak points, adding practical exercises, and aligning resource difficulty with the learner's current ability. Figure 13 presents the profile analysis results.

The learner profile feeds into subsequent resource generation. Beyond knowledge-point information, the system accounts for the learner's current ability, learning goals, weak points, and preferred difficulty. These factors determine the next learning content, resource difficulty, and resource type. Consequently, generated resources adapt to changes in the learner profile.

3.4.2 Comprehensive Learning Performance Assessment Agent

The comprehensive performance assessment agent evaluates learning performance using learner profiles, learning progress, mastery levels, test results, error records, and resource usage data. Key evaluation factors include knowledge mastery, learning activity, task completion, practical ability, and improvement in weak knowledge points. An overall score is derived from these indicators, along with separate results for different performance aspects. These results inform learning path updates and subsequent resource recommendations. During a typical learning cycle, performance is evaluated based on task completion and quiz results. These data identify mastered knowledge points, weak points, and recurring errors. The assessment yields an overall score plus separate results for knowledge mastery, learning progress, and changes in identified weaknesses. When low mastery is detected for a specific knowledge point, the system recommends related materials, quizzes, and practical exercises. For knowledge points with high mastery, repeated recommendations are reduced. Figure 14 presents the performance assessment results, and Figure 15 shows the corresponding feedback and resource recommendations.

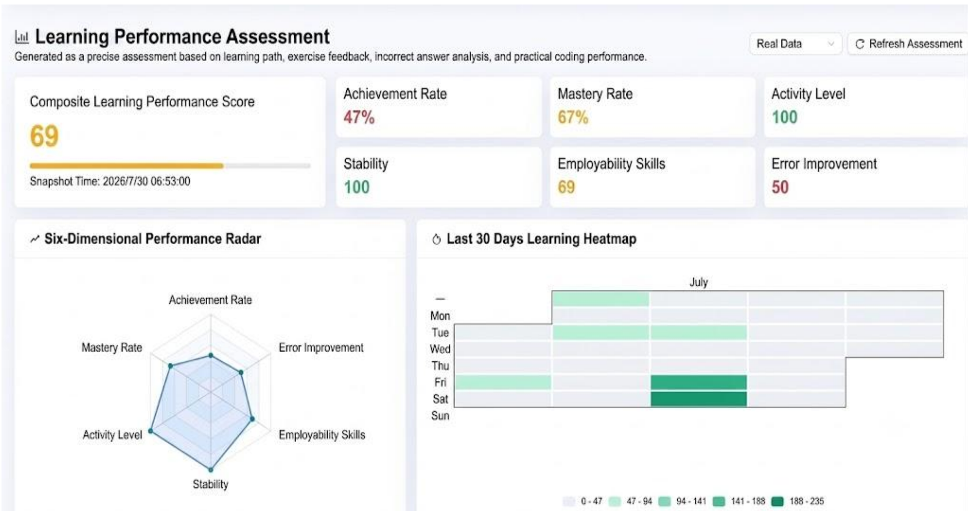


Figure 14. Results of Comprehensive Learning Performance Assessment

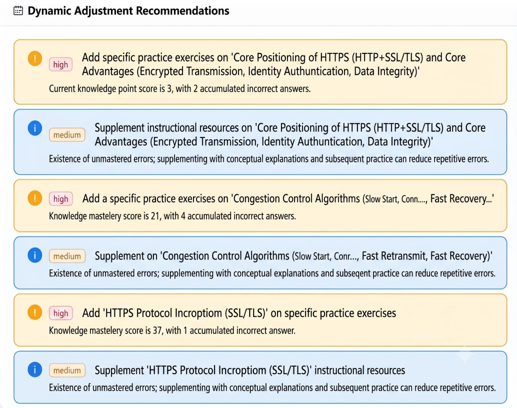


Figure 15. Results of Learning Feedback and Resource Recommendations

Simulation results indicate that the performance assessment agent combines learning process data with evaluation results to support subsequent recommendations. The system integrates progress tracking, performance evaluation, weak-point identification, and learning feedback within a single learning process. As learner performance changes, the system updates assessment results and adjusts recommended resources. This mechanism enables continuous evaluation and supports personalized learning path adjustment and resource recommendation.

4. Conclusion

This study presents a personalized learning system for vertical domains built on RAG and multi-agent collaboration. The system encompasses learner profile construction, knowledge retrieval, resource generation, learning path planning, and learning feedback. Data collected during learning update the learner profile and adjust subsequent resources and paths, connecting all modules within a unified personalized learning process.

From a technical standpoint, the system combines RAG with dynamic learner profiling. The domain knowledge base supplies external information during generation, reducing errors stemming from insufficient domain knowledge in general-purpose LLMs. Learner profile data—prior knowledge, learning goals, and weak points—constrain resource generation. A multi-agent architecture assigns distinct agents to knowledge retrieval, requirement analysis, resource generation, content review, path planning, and feedback. These agents operate within a shared workflow to complete different stages of the personalized learning process. The experimental results indicate that this design improves resource relevance, learning path organization, and the use of learning feedback for subsequent adjustments.

The system generates several types of learning resources, including explanatory materials, quizzes, practical tasks, mind maps, and personalized learning paths. These resources are adjusted according to learner profiles and learning progress. The experimental and simulation results suggest that the proposed method can improve the relevance and organization of personalized learning resources. The system provides an implementation example of combining RAG, learner profiling, and multi-agent collaboration for vertical-domain learning.

In future work, the vertical-domain knowledge base will be further expanded. More learning behavior data will also be collected to support learner profile updating and resource recommendation. In addition, the proposed system will be evaluated in different courses and learning scenarios to further examine its effectiveness and applicability.

Acknowledgments

The authors would like to express their sincere gratitude to iFLYTEK Co., Ltd. for its support, as well as to the reviewers, editors and teachers for their insightful comments and constructive suggestions that helped improve the quality of this paper.

References

[1] Ahmad K A, Iqbal W, El-Hassan A, et al. Data-Driven Artificial Intelligence in Education: A Comprehensive Review. IEEE Transactions on Learning Technologies, 2024, 17: 12-31.

[2] Sajja R, Sermet Y, Cikmaz M, et al. Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education. Information, 2024, 15(10): 596.

[3] Bayly-Castaneda K, Ramirez-Montoya M S, Morita-Alexander A. Crafting Personalized Learning Paths with AI for Lifelong Learning: A Systematic Literature Review. Frontiers in Education, 2024, 9: 1424386.

[4] Kasneci E, Sessler K, Küchemann S, et al. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 2023, 103: 102274.

[5] Xu H, Gan W, Qi Z, et al. Large Language Models for Education: A Survey. arXiv: 2405.13001, 2024.

[6] Du Plooy E, Casteleijn D, Franzsen D. Personalized Adaptive Learning in Higher Education: A Scoping Review of Key Characteristics and Impact on Academic Performance and Engagement. Heliyon, 2024, 10(21): e39630.
[7] Brusilovsky P. Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 2001, 11(1-2): 87-110.
[8] Piech C, Bassen J, Huang J, et al. Deep Knowledge Tracing//Advances in Neural Information Processing Systems. 2015, 28.
[9] Swiecki Z, Khosravi H, Chen G, et al. Assessment in the age of artificial intelligence. Computers and Education: Artificial Intelligence, 2022, 3: 100073.
[10] Chen X, Zou D, Xie H, et al. Past, present, and future of smart learning: a topic-based bibliometric analysis. International Journal of Educational Technology in Higher Education, 2021, 18(1): 1-29.
[11] Lewis P, Perez E, Piktus A, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks//Advances in Neural Information Processing Systems. 2020, 33: 9459-9474.
[12] Mialon G, Dessì R, Lomeli M, et al. Augmented Language Models: a Survey. arXiv: 2302.07842, 2023.
[13] Karpukhin V, Oguz B, Min S, et al. Dense Passage Retrieval for Open-Domain Question Answering//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. 2020: 6769-6781.
[14] Wu Q, Bansal G, Zhang J, et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. arXiv: 2308.08155, 2023.
[15] Sumers T R, Yao S, Narasimhan K, et al. Cognitive architectures for language agents. arXiv: 2309.02427, 2023.
[16] Touvron H, Lavril T, Izacard G, et al. LLaMA: Open and Efficient Foundation Language Models. arXiv: 2302.13971, 2023.