Research on a Job Competency Profiling and Multi-Agent Collaborative Intelligent Training System for Vocational Learners

Yinghao Ma[1,*], Yifan Li[1], Jingjing Zhou[1], Yaohui Tian[1], Mingming Gong[2]
[1]*School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China*
[2]*Xunfei Lingzhi Talent Training Department, iFlytek, Zhengzhou, Henan, China*
**Corresponding Author*

Abstract: Rapid changes in occupational knowledge and skill requirements create challenges for intelligent training systems, including weak alignment between learner competencies and job requirements, insufficient domain grounding of generated resources, and uncontrolled generation for out-of-knowledge requests. This paper proposes a domain-knowledge-enhanced multi-agent intelligent training system for vocational learners. The system constructs a job competency graph linking positions, competency domains, skills, knowledge points, courses, and practical tasks, and organizes learner experience, skill labels, learning behavior, and practical performance into a job-oriented competency profile. Learning diagnosis, job competency analysis, knowledge retrieval, content generation, quality verification, and learning evaluation agents collaborate to organize training resources. Hybrid retrieval, competency-domain filtering, dynamic Top-K, and a deterministic knowledge-boundary gate are introduced to improve generation reliability. Expert evaluation of 24 course resources produced an overall score of 4.097. On 30 held-out boundary tasks, the gate achieved 93.33% accuracy and 93.27% macro-F1, while increasing the strict termination rate for insufficient-knowledge tasks to 100%. Hybrid retrieval achieved 100% Recall@3; after competency-domain filtering and dynamic Top-K were introduced, precision increased from 0.2000 to 0.5104 and the irrelevant-document rate decreased from 80.00% to 48.96%. These results demonstrate that the proposed system improves domain grounding and generation-boundary control while maintaining effective knowledge retrieval.

Keywords: Vocational Learners; Intelligent Training; Multi-Agent System; Domain Knowledge Base; Retrieval-Augmented Generation; Job Competency Graph

1. Introduction

Digital technologies are continuously transforming the content and competency structure of traditional occupations. Driven by artificial intelligence, big data, cloud computing, and intelligent manufacturing, many jobs have shifted from single-domain knowledge requirements toward integrated requirements involving knowledge, skills, tools, and practical capabilities. For example, an artificial intelligence application developer must not only master Python, machine learning, and data processing, but also be able to work with vector databases, retrieval-augmented generation, agent development, model evaluation, and system deployment. Front-end developers are expected to use HTML, CSS, and JavaScript as well as modern front-end frameworks for page development, component design, and engineering deployment. Operations and maintenance personnel need integrated capabilities in operating systems, databases, middleware, servers, and fault diagnosis.

The rapid evolution of occupational competency structures places vocational learners under continuous pressure to learn. Vocational learners typically differ in educational background, work experience, and practical foundation. Some possess solid theoretical knowledge but lack project experience; some have work experience but an incomplete knowledge system; and others are transitioning from a different discipline. These differences mean that learners should not be assigned identical course sequences and learning resources.

Current online training systems mainly organize resources through job categories, course tags,

and popularity. Although this approach helps learners browse courses quickly, it is difficult to identify their current competency gaps or explain the specific relationship between a course and a target job. Learners often have to select courses, arrange their learning order, and judge whether they have truly mastered the relevant skills. When courses, topic-based learning, and practical tasks are disconnected, learners may complete a course without being able to perform the corresponding job task.

The development of large language models provides new technical conditions for intelligent training. Large language models can generate course explanations, exercises, case analyses, and learning feedback from natural-language instructions, substantially reducing the cost of producing training resources. However, a general-purpose model does not inherently possess complete knowledge of a specific occupation, nor can it guarantee that generated content always conforms to occupational skill standards and engineering practices. Directly using a large language model to generate job-oriented courses may therefore lead to factual errors, version confusion, mismatches with job tasks, and inappropriate difficulty settings [1-5].

An occupational training system should therefore integrate learner modeling, job competency analysis, domain-knowledge retrieval, training-resource generation, practical-task guidance, and learning-outcome evaluation. This study introduces multi-agent collaboration and retrieval-augmented generation into the job-training workflow to address key challenges in personalized training for vocational learners.

This study addresses the following research questions:

1) How can a vocational learner's educational background, project experience, skill tags, course behaviors, and practical-training performance be organized into a computable job competency profile?

2) How can the relationships among jobs, competency domains, skills, knowledge points, courses, and practical tasks be established so that training resources are organized around learner competency gaps?

3) How can a domain knowledge base and multi-agent collaboration be used to generate course resources with evidential grounding, job relevance, and structural completeness?

4) How can retrieval evaluation and knowledge-boundary gating reduce the entry of irrelevant knowledge chunks into the generation context and prevent out-of-knowledge-base requests from triggering unsupported generation?

The main contributions of this study are as follows:

1) We construct a job competency profiling model for vocational learners that integrates learning experience, skill tags, course behaviors, and practical-training performance.

2) We establish a job competency graph linking jobs, competency domains, skills, knowledge points, courses, practical tasks, and evaluation criteria.

3) We design a domain-knowledge-enhanced workflow for generating courses, topic resources, and practical tasks, while improving traceability through source constraints.

4) We design a collaborative decision process among agents for learning diagnosis, job competency analysis, knowledge retrieval, content generation, practical-task guidance, and learning evaluation.

5) We conduct three experiments on course-resource quality, knowledge-boundary control, and RAG retrieval performance to evaluate the system's resource-generation, boundary-control, and knowledge-retrieval capabilities.

2. Overall System Design

2.1 System Objectives

The system is designed to support the development of job competencies among vocational learners by providing job-oriented courses, online learning, topic-based study, cloud-based practical training, and competency assessment. Specifically, the system enables learners to select a target job and understand its competency requirements; generates a visual competency profile from the learner's prior experience and assessment results; organizes courses, topics, and practical tasks around identified competency gaps; provides course explanations and practical-training feedback under domain-knowledge constraints; and updates the learner's competency profile based on course scores, code execution, and task performance.

2.2 System Architecture

The system consists of a data layer, knowledge layer, agent layer, collaborative decision-making layer, and application layer, as illustrated in Figure 1.

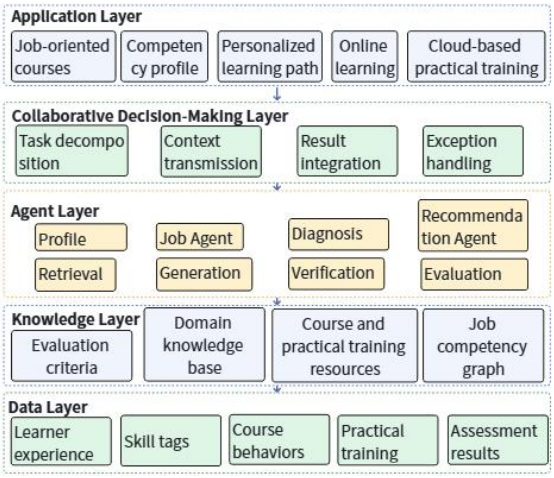

Figure 1. Overall Architecture of Intelligent Training System for Vocational Learners

The data layer stores learner profiles, educational background, work experience, project experience, skill tags, course records, topic-learning records, assessment scores, code-execution records, and practical-training evaluation results. The knowledge layer stores job standards, course materials, technical documents, project cases, practical tasks, and evaluation criteria, and uses a knowledge graph to describe relationships among job competencies.

The agent layer consists of the Learner Profile Agent, Job Competency Agent, Learning Diagnosis Agent, Course Recommendation Agent, Knowledge Retrieval Agent, Content Generation Agent, Practical-Training Guidance Agent, and Learning Evaluation Agent. The collaborative decision-making layer manages task decomposition, context transfer, result integration, and exception handling among agents. The application layer provides job-oriented courses, competency radar charts, personalized learning paths, online courses, and cloud-based practical training for learners, as well as knowledge-base management, course management, and learning-effect analysis for training administrators [6-10].

2.3 Organization of Training Resources

Training resources are organized into three levels: courses, topics, and practical tasks.

Courses are used to build a complete knowledge system and support systematic learning. For an artificial intelligence application development position, for example, the curriculum may include Python programming, fundamentals of data processing, machine learning fundamentals, model evaluation, and retrieval-augmented generation.

Topics are designed to address specific knowledge points or skill-related problems. Vector representation, text chunking, similarity computation, prompt design, and API invocation can be provided as topic resources.

Practical tasks are used to verify whether knowledge can be transferred to authentic work tasks. Learners may, for example, complete an enterprise document question-answering system, a data-analysis dashboard, a front-end management page, or a server-monitoring application.

3. Competency Profiling for Vocational Learners

3.1 Competency-Profile Data

The system collects learner information from four aspects.

The first category is static information, including educational background, field of study, years of work experience, and target job. The second category is experience information, including projects, roles, technologies used, and project outcomes. The third category is learning information, including course completion, topic-learning records, study time, and repeated-learning behaviors. The fourth category is assessment information, including objective-test scores, programming-task execution results, practical-task scores, and instructor evaluations.

To reduce inaccuracies in self-reported information, the system does not treat skill tags as the final competency level. Instead, it uses them as initial evidence and calibrates them through assessments and practical tasks.

3.2 Competency Dimensions

Based on job-oriented courses and authentic occupational tasks, the system defines competency dimensions including computational thinking, web development, database operations, platform operations, technical extension, comprehensive professional competence, and middleware and server operations. Different target jobs use different competency weights.

For example, front-end development emphasizes web development, computational thinking, and comprehensive professional competence; AI algorithm positions emphasize algorithmic foundations, data processing, model evaluation, and technical extension; and operations positions place greater emphasis on platform operations,

database operations, middleware, and server management.

3.3 Competency-Score Calculation

Let the learner competency score S_k in a given dimension $k$be determined jointly by assessment, course learning, practical-training performance, and expert evaluation:

$$S_k=100(w_1T_k+w_2C_k+w_3P_k+w_4E_k) \quad (1)$$

The assessment score T_k, course-learning score C_k, practical-training performance P_k, and expert evaluation jointly determine the learner competency score E_k.

$$w_1 + w_2 + w_3 + w_4 = 1 \quad (2)$$

At the initial experimental stage, the weights $w_1 = 0.3, w_2 = 0.2, w_3 = 0.4, w_4 = 0.1$ can be configured to emphasize practical-training performance. In subsequent iterations, the weights can be learned from expert-annotated data.

The system uses a radar chart to display the learner current competency and the target job standard simultaneously. A dimension below the job requirement is marked as a competency gap and traced to specific knowledge points and courses.

3.4 Dynamic Updating of Competency State

After a learner completes a new learning activity, the system updates the learner competency state according to the following rule:

$$S_k^{t+1} = (1 - \eta)S_k^t + \eta R_k^t \quad (3)$$

The activity-level evaluation result R_k^t represents performance k in the current activity, and the update rate η controls the magnitude of the state change. Difficult practical tasks that represent authentic job work receive higher update weights, whereas simple multiple-choice activities have a smaller influence.

4. Job Competency Graph and Course Recommendation

4.1 Job Competency Graph

The job competency graph decomposes job requirements into competency domains, skills, knowledge points, courses, and practical tasks. An AI algorithm engineer position, for example, can be represented as Figure 2.

Each knowledge point contains prerequisite knowledge, applicable jobs, difficulty level, associated courses, and evaluation methods. The system can therefore determine not only that a learner lacks model-evaluation competency, but also whether the learner lacks dataset splitting, evaluation metrics, or error analysis.

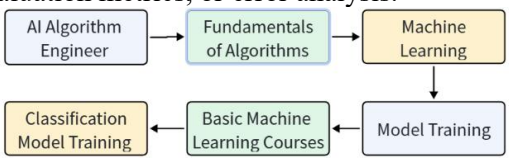

Figure 2. Position Information of AI Algorithm Engineer

4.2 Competency-Gap Calculation

Let the target job requirement for a knowledge point j be $r_j$and the learner current mastery be z_j. The competency gap is defined as:

$$g_j = \max(0, r_j - z_j) \quad (4)$$

Considering the importance of each knowledge point to the job p_j and its prerequisite influenceq_j, the recommendation priority is calculated as follows:

$$\text{Priority}_j = \alpha g_j + \beta p_j + \gamma q_j c \quad (5)$$

High-priority knowledge points enter the near-term training path, whereas low-priority points are scheduled for later stages or offered as extension content.

4.3 Course Recommendation

The course recommendation score is defined as follows:

$$R(c) = \lambda_1 G(c) + \lambda_2 J(c) + \lambda_3 F(c) + \lambda_4 T \quad (6)$$

The recommendation score is composed of the following factors: $G(c)$ Course coverage of the learner competency gaps; $J(c)$course relevance to the target job; $F(c)$ the match between course prerequisites and the learner current knowledge; $T(c)$the match between the course and the learner available study schedule.

The system first recommends courses and then recommends topics and practical tasks according to course results, forming a progressive path from foundational knowledge to occupational practice.

5. Domain-Knowledge-Enhanced Training-Resource Generation

5.1 Construction of the Domain Knowledge Base

The knowledge base includes occupational skill standards, job descriptions, course textbooks, technical documents, enterprise cases, and expert-reviewed practical tasks. Knowledge processing includes document parsing, text cleaning, paragraph segmentation, metadata annotation, and vectorization.

Each knowledge chunk stores its source, applicable jobs, competency dimensions, related skills, difficulty level, publication date, and review status. Recording the provenance of knowledge supports explanation and verification of generated training resources.

5.2 Retrieval-Augmented Generation

The system constructs a retrieval query according to the target job, competency gaps, and the learner current level. The RAG Retrieval and Knowledge Boundary Gating Process is shown in Figure 3.

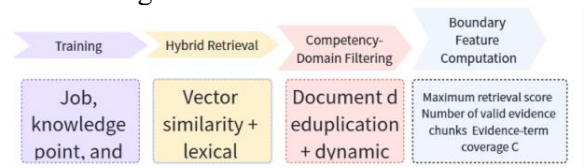

Figure 3. RAG Retrieval and Knowledge Boundary Gating Process

$$q = F(L, G, g, d) \quad (7)$$

The learner profile L, target job G, competency gap g, and target difficulty d are used as query conditions.

Candidate documents are re-ranked according to semantic similarity, job relevance, source reliability, and temporal validity. The ranked results are provided to the Content Generation Agent, which generates knowledge explanations, exercises, cases, and practical tasks [11-16].

5.3 Quality Verification of Generated Content

The Quality Verification Agent checks generated results from four perspectives:

First, it checks whether the generated content is supported by documents in the knowledge base. Second, it checks whether the content is suitable for the target job. Third, it checks whether the difficulty matches the learner competency. Fourth, it checks whether a practical task has clear inputs, outputs, and evaluation criteria.

If the generated content lacks sufficient evidence, has low job relevance, or defines unclear task objectives, the system triggers a new retrieval and regeneration process.

6. Multi-Agent Collaborative Training Method

6.1 Collaborative Workflow

The Learner Profile Agent first analyzes educational background, project experience, course records, and test results to generate the current competency state. The Job Competency Agent analyzes the target job and generates its competency requirements. The Learning Diagnosis Agent compares the two and passes the identified gaps to the Course Recommendation Agent. The Course Recommendation Agent selects appropriate courses and topics; the Knowledge Retrieval Agent provides domain evidence; the Content Generation Agent produces training resources; the Quality Verification Agent reviews the resources; and the Learning Evaluation Agent updates the learner profile according to the final learning results. The multi-agent collaborative training workflow is shown in Figure 4.

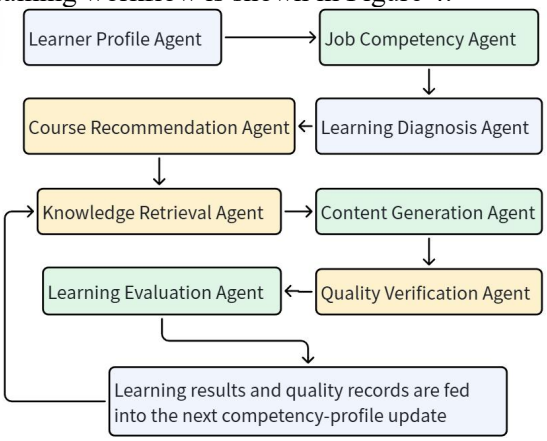

Figure 4. Multi-Agent Collaborative Training Process

6.2 Learning Diagnosis Agent

The Learning Diagnosis Agent analyzes not only incorrect answers but also error types. A syntax error in a programming task may indicate insufficient basic coding ability; executable code with an incorrect result may indicate an algorithmic misunderstanding; and a correct algorithm that cannot complete data loading or deployment may indicate inadequate engineering practice.

Error-type identification prevents the system from attributing every failure to the same knowledge point.

6.3 Cloud-Based Practical-Training Guidance Agent

The Practical-Training Guidance Agent connects to a cloud programming environment and reads code-execution logs, error messages, and task-completion status. For learner difficulties, the system adopts a staged prompting strategy: it first provides a conceptual hint, then an operational direction, followed by a partial-code hint, and provides a complete example only after

repeated failed attempts.
This design avoids directly replacing the learner in task completion while preserving necessary learning scaffolding.

6.4 Learning Evaluation Agent

The Learning Evaluation Agent combines course scores, topic scores, code-execution results, task-completion time, and expert ratings to produce a learning report. The report includes mastered competencies, competencies requiring improvement, recommended knowledge points for review, recommended courses for the next stage, recommended practical tasks, and changes in the competency radar chart.

7. System Implementation

7.1 Technical Architecture and Functional Modules

The system adopts a front-end/back-end separated architecture. The front end provides vocational learners with job-oriented courses, competency profiles, learning paths, course resources, and cloud-based practical training. The back end manages learner data, job-course relationships, knowledge-base indexing, resource generation, and experimental logs. A relational database stores structured data on learners, jobs, competencies, courses, practical tasks, and learning behaviors. Domain documents are segmented and vectorized into searchable knowledge chunks.

7.2 RAG Retrieval and Course-Resource Generation

The knowledge-retrieval module constructs queries from the training topic, target knowledge points, and resource type, and then computes vector-similarity and lexical-matching scores. The system supports keyword retrieval, vector retrieval, and hybrid retrieval, and adds competency-domain filtering and dynamic Top-K selection to hybrid results. Retrieved results retain chunk identifiers, source documents, content, and scores as the evidence context for the Content Generation Agent; these fields are also written to resource metadata to support experimental reproduction and provenance tracking [11-16].

7.3 Knowledge-Boundary Gating

Before retrieved results are passed to the Content Generation Agent, the system calculates the maximum retrieval score, the number of valid evidence chunks, and evidence-term coverage. If no result is retrieved, evidence-term coverage is below 0.10, the maximum retrieval score is below 0.20, or the number of valid chunks is insufficient, the system returns KNOWLEDGE_INSUFFICIENT and skips generation. For partially covered requests, it generates only evidence-supported content and explicitly marks the uncovered portion. The gate is integrated into ordinary, streaming, and PPT-resource generation workflows [11-17].

7.4 Multi-Agent Collaboration and Result Logging

The Learner Profile Agent and Job Competency Agent provide the learner state and target-job requirements. The Learning Diagnosis Agent calculates competency gaps; the Course Recommendation Agent organizes courses and topics; the Knowledge Retrieval Agent provides domain evidence; the Content Generation Agent produces resources; and the Quality Verification Agent checks sources, structure, and job relevance. Retrieval hits, knowledge-boundary status, resource type, difficulty, and quality evaluations are written to resource metadata so that generation remains traceable. The experiments directly evaluate course-resource quality, knowledge-boundary classification, and retrieval performance; profile updating, path recommendation, and practical-task guidance are implemented functions and are not claimed to have produced measured learning gains in this study [6-10].

8. Experimental Design and Results

8.1 Experimental Objectives and Research Questions

To evaluate the effectiveness of the intelligent training system for vocational learners at the levels of course-resource generation, knowledge-boundary control, and knowledge retrieval, we designed three interconnected experiments and treated retrieval improvement as an extension of the third experiment. Experiment 1 examines the professionalism, knowledge coverage, structural completeness, job relevance, and task executability of generated course resources. Experiment 2 examines whether the system can identify knowledge-coverage status and prevent unsupported course generation when knowledge is insufficient. Experiment 3 evaluates the

effects of different RAG retrieval strategies on the ranking of relevant knowledge chunks [9, 14-16].

The experiments address three questions: whether domain-knowledge enhancement generates course resources with high professional accuracy, knowledge coverage, job relevance, and task executability; whether knowledge-boundary gating reduces unsupported generation caused by out-of-knowledge-base requests; and whether hybrid retrieval and competency-domain filtering reduce irrelevant knowledge entering the generation context while preserving recall. The experimental evidence chain is shown in Figure 5.

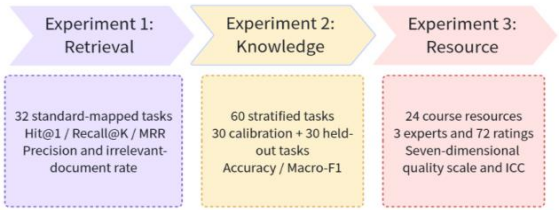

Figure 5. Experimental Verification Framework and Evidence Chain

8.2 Experimental Environment and Dataset

The experiments used the system's current domain corpus of 25 Markdown documents covering Python and NumPy, probability and statistics, machine learning, deep learning, model engineering, and fault analysis. Following the system's current chunking rules, the documents formed 25 searchable retrieval chunks. The course-resource experiment used 24 generated domain resources; the knowledge-boundary experiment used 60 stratified tasks, including 30 calibration tasks and 30 independent held-out tasks; and the retrieval experiment used 32 sufficient- or partially covered tasks with established mappings to gold-standard relevant documents.

Knowledge-boundary tasks were divided into sufficient-knowledge, partial-coverage, and insufficient-knowledge categories. Each category accounted for one third of both the calibration and held-out sets. Partial-coverage tasks combined in-domain terminology with occupational subgoals unsupported by the knowledge base, whereas insufficient-knowledge tasks were entirely outside the current knowledge scope. All automated experiments used a fixed knowledge base and fixed task sets. Retrieval vectors were generated using deterministic hash mapping, without random sampling.

8.3 Course-Resource Generation Quality Experiment

This experiment compares a general large-model generation workflow with a workflow augmented by domain-knowledge retrieval. Each resource was generated around an occupational learning topic and included learning objectives, knowledge points, learning steps, a practical task, and evaluation requirements. Resource quality was evaluated along seven dimensions: professional accuracy, knowledge coverage, structural completeness, job relevance, difficulty fit, task executability, and source reliability. Each dimension used a 1-5 Likert scale [13].

A score of 5 indicates accurate, complete, and directly usable content; 4 indicates basically usable content requiring minor editing; 3 indicates acceptable content requiring local revision; 2 indicates obvious defects; and 1 indicates content unsuitable for training. Knowledge coverage was converted from the proportion of standard knowledge points covered: 90-100% corresponds to 5 points, 75-89% to 4, 60-74% to 3, 40-59% to 2, and below 40% to 1. A resource was considered unsuccessful if any of professional accuracy, job relevance, or source reliability scored below 3.

This experiment selected 24 domain course resources, which were independently rated by three experts with relevant professional backgrounds, yielding 72 rating records. To reduce order effects, resources were presented using neutral identifiers, and each expert rated independently without access to the other raters' results. We report the resource-level mean, standard deviation, 95% confidence interval, and inter-rater reliability. Results are shown in Table 1.

The overall resource score was 4.097, with a standard deviation of 0.148. Knowledge coverage and professional accuracy were relatively high, indicating that domain retrieval anchored generated content in reference materials. Job relevance and task executability were relatively lower, suggesting that expansion into neighboring knowledge may weaken the correspondence between training tasks and job requirements. Inter-rater reliability was ICC(2, 1)=0.355 for a single rater and ICC(2, k)=0.623 for the average of three raters [14], indicating that averaging multiple raters is more appropriate than relying on a single judgment.

Table 1. Experimental Results of Course Resource Generation Quality

Evaluation Dimension	Mean	Standard Deviation	95% Confidence Interval
Professional Accuracy	4.22	0.23	[4.13, 4.32]
Knowledge Coverage	4.40	0.24	[4.31, 4.50]
Structural Completeness	4.10	0.18	[4.02, 4.17]
Job Relevance	3.94	0.50	[3.75, 4.14]
Difficulty Fit	4.01	0.12	[3.97, 4.06]
Task Executability	3.93	0.22	[3.84, 4.02]
Overall Source Reliability	4.07	0.39	[3.91, 4.23]

8.4 Knowledge-Boundary Control Experiment

The gate takes the Top-5 retrieval results as input and calculates the maximum retrieval score, the number of valid relevant chunks, and evidence-term coverage.

Evidence-term coverage C is formally defined as:

$$C = |Tq \cap Te|/|Tq| \quad (8)$$

where Tq denotes the set of query terms after removing generic terms such as "course", "generation" and "explanation", and Te refers to the term set contained in the Top-5 retrieved evidence. If no result is retrieved, coverage is below 0.10, the maximum score is below 0.20, or valid evidence is insufficient, the system returns KNOWLEDGE_INSUFFICIENT and skips course generation. When the minimum evidence requirement is met but coverage is below 0.425, the request is classified as partially covered and only the explicitly supported portion is generated. Thresholds were determined by grid search on 30 calibration questions; the held-out set was not used for tuning.

Three-class performance was evaluated using accuracy, per-class precision, per-class recall, and macro-F1. Generation control was evaluated using strict termination rate and false-termination rate. Strict termination rate is the proportion of insufficient-knowledge tasks that do not enter the generation stage; false-termination rate is the proportion of sufficient-knowledge or partially covered tasks that are incorrectly blocked. 95% Wilson confidence intervals were calculated for the main proportions [15]. Results are shown in Table 2.

Table 2. Experimental Results of Knowledge Boundary Control

Metric	Before Improvement	After Improvement
Held-Out Three-Class Accuracy	33.33%	93.33%
Held-Out Macro-F1	16.67%	93.27%
Strict Termination Rate for Insufficient-Knowledge Tasks	0%	100%
False-Termination Rate for Non-Insufficient Tasks	0%	0%

Among the 30 independent held-out tasks, all 10 sufficient-knowledge tasks and all 10 insufficient-knowledge tasks were correctly classified. Of the 10 partially covered tasks, 8 were correctly classified and 2 were classified as sufficient. The overall accuracy was 93.33%, with a 95% Wilson confidence interval of [78.7%, 98.2%]. The strict termination rate was 100%, with a 95% Wilson confidence interval of [72.2%, 100%]. Both errors occurred in the partial-coverage category, indicating that when most terms in a request are supported by the knowledge base, a small uncovered subgoal may be masked by aggregate coverage.

Further analysis showed that the "civilian drone pilot license" task received a maximum similarity score of 0.2901 because of a low-dimensional hash-vector collision, although its evidence-term coverage was 0. Adding a minimum evidence-term coverage constraint allowed the task to be correctly classified as insufficient knowledge, demonstrating that similarity thresholds should be combined with interpretable textual evidence.

8.5 RAG Retrieval Performance and Improvement Experiment

Using the same knowledge base, queries, and Top-5 setting, we compared keyword retrieval, vector retrieval, and hybrid retrieval. We then added competency-domain filtering and dynamic Top-K selection to the hybrid method. Metrics included Hit@1, Recall@3, Recall@5, MRR, Precision@5, and irrelevant-document rate [16-17]. Results are shown in Table 3.

Table 3. Experimental Results of RAG Retrieval Performance and Improvement Experiment

Method	Hit@1	Recall@3	Recall@5	MRR	Precision@5	Irrelevant-Document Rate@5
Keyword Retrieval	81.25%	96.88%	90.62%	0.8750	0.2708	72.92%
Vector Retrieval	62.50%	90.62%	90.62%	0.7745	0.1938	80.63%
Hybrid Retrieval	90.62%	100.00%	90.62%	0.9531	0.2000	80.00%
Hybrid Retrieval + Competency-Domain Filtering + Dynamic Top-K	90.62%	90.62%	90.62%	0.9062	0.5104	48.96%

Hybrid retrieval outperformed both single-method baselines in recall and ranking, with Recall@3 and Recall@5 both reaching 100%. After competency-domain filtering and dynamic Top-K selection were added, Precision@5 increased from 0.2000 to 0.5104 and the irrelevant-document rate decreased from 80.00% to 48.96%; however, Recall@3 decreased from 100% to 90.62%. These results indicate that filtering reduces prompt contamination but overly strict filtering may sacrifice recall. In deployment, filtering should therefore be determined dynamically according to the maximum relevance score and competency-domain consistency of candidate documents.

8.6 Integrated Discussion and Experimental Limitations

The three experiments form an evidence chain from retrieval to boundary decision and then resource generation. The retrieval experiment shows that hybrid retrieval improves the ranking quality of relevant knowledge. The boundary experiment shows that deterministic gating before generation can substantially reduce unsupported generation for insufficient-knowledge tasks. The course-resource experiment shows that knowledge enhancement produces resources with good professional accuracy and knowledge coverage, while job relevance and task executability remain the main areas for improvement.

Two limitations remain. First, the boundary and retrieval tasks were constructed by the researchers from the current knowledge base and may contain distribution bias; future work should validate the method on real user queries and external cross-job test sets. Second, the experiments mainly evaluate course-resource quality, knowledge retrieval, and generation-boundary control. They do not yet include pre/post learning tests or job-task transfer, so this paper does not make claims about long-term educational effectiveness beyond the experimental scope.

9. Conclusion

This paper presents an intelligent training system for vocational learners that integrates job competency profiling, a job competency graph, domain-knowledge enhancement, and multi-agent collaboration. The system converts learner experience, course behaviors, and practical-training performance into a computable competency state, and establishes links among job requirements, knowledge points, course resources, and practical tasks through the competency graph. RAG provides traceable domain evidence, while specialized agents collaboratively perform competency diagnosis, course recommendation, resource generation, practical-task guidance, and learning evaluation.

The results show that domain-knowledge enhancement improves the professional accuracy and knowledge coverage of course resources. On 30 independent held-out tasks, knowledge-boundary gating achieved 93.33% three-class accuracy and 93.27% macro-F1, while the strict termination rate for insufficient-knowledge tasks reached 100%. Hybrid retrieval achieved Recall@3 of 100%; after competency-domain filtering and dynamic Top-K selection, Precision@5 increased from 0.2000 to 0.5104 and the irrelevant-document rate decreased to 48.96%. These findings indicate that occupational training systems must jointly address retrieval recall, generation-boundary control, and the job executability of training resources.

The study completes a system prototype and three controlled experiments. Future work will continuously expand the knowledge base using authentic job standards, introduce external independent test sets, and further evaluate educational effectiveness through pre/ post tests with vocational learners, practical-task completion rates, and job-task performance.

Acknowledgments

This paper is supported by College Students' Innovation Training Program of Henan University of Technology (No. 202610463051).

References

[1] Anagnostou G, Doumanas D, Kotis K. LLM4KGen: A framework for developing KG-based semantic applications with LLMs, RAG and AI agents.Data & Knowledge Engineering, 2026, 165102626-102626. DOI: 10. 1016/J. DATAK. 2026. 102626.

[2] TLILI A, SHEHATA B, ADARKWAH M A, et al. What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. Smart Learning Environments, 2023, 10(1): 15.

[3] LO C K. What is the impact of ChatGPT on education? A rapid review of the literature. Education Sciences, 2023, 13(4): 410.

[4] OGUNLEYE B, FOSTER C, et al. A

systematic review of generative AI in higher education. International Journal of Educational Technology in Higher Education, 2024, 21: 30.

[5] DWIVEDI Y K, KSHETRI N, HUGHES L, et al. So what if ChatGPT wrote it? Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI. International Journal of Information Management, 2023, 71: 102642.

[6] XI Z, CHEN W, GUO X, et al. The rise and potential of large language model based agents: A survey. arXiv preprint arXiv:2309.07864, 2023.

[7] HONG S, ZHENG X, TANG J, et al. MetaGPT: Meta programming for multi-agent collaborative framework//International Conference on Learning Representations. 2024.

[8] QIAN C, LIU X, FENG H, et al. ChatDev: Communicative agents for software development//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024: 15174-15186.

[9] Skulskiy N Y, Cheremisina N E ,Kirov F E , et al.Adaptive Automated Response System for Virtual Computer Lab and LMS Moodle Using LLM, RAG, and Serverless Architecture. Physics of Particles and Nuclei, 2026, 57(4): 578-580.

[10] WANG L, MA C, FENG X, et al. A survey on large language model based autonomous agents. Frontiers of Computer Science, 2024, 18(6): 186345.

[11] ASAI A, WU Z, WANG Y, et al. Self-RAG: Learning to retrieve, generate, and critique through self-reflection//International Conference on Learning Representations. 2024.

[12] YAN S, GUI H, ZHANG J, et al. Corrective retrieval augmented generation. arXiv preprint arXiv:2401.15884, 2024.

[13] GAO Y, XIONG Y, GAO X, et al. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2023.

[14] ES S, JHA S, et al. RAGAS: Automated evaluation of retrieval augmented generation//Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024: 150-158.

[15] SAAD-FALCON J, HOU Y, et al. ARES: An automated evaluation framework for retrieval-augmented generation systems// Proceedings of NAACL-HLT. 2024: 338-354.

[16] RU D, QIAN Y, et al. RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation//Advances in Neural Information Processing Systems. 2024, 37.

[17] DEEPSEEK-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv: 2501. 12948, 2025.

