Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 3, 2026

A Job-Oriented Knowledge Graph for Computer Networking Courses in Higher Vocational Education

Huaxin Zheng

Wenzhou Polytechnic, Wenzhou, China

* **Corresponding author**: huaxin179@163.com

Abstract: Students in higher vocational computer networking programs often learn technical topics such as TCP/IP, virtual local area networks, access control lists, firewall configuration, wireless access points, Linux services, and network monitoring without clearly understanding how these topics support real occupational roles. Teachers also need evidence-based methods to explain why a course matters and how its modules are connected with job abilities. To address this problem, this paper proposes a job-oriented knowledge graph for computer networking courses in higher vocational education. The proposed system separates raw job evidence from the student-facing teaching graph. Public and low-risk job evidence is organized in a controlled evidence layer, while the published graph presents normalized jobs, abilities, course modules, knowledge points, tools, certificates, tasks, and vendors. The student-facing explanation process connects target jobs with required abilities, course modules, and knowledge points, allowing students to understand how course learning supports career preparation. A Zhejiang pilot system was implemented using Vue 3, ECharts, Express APIs, and Neo4j. The current pilot graph contains 123 nodes and 367 relationships, including 6 job nodes, 10 ability nodes, 7 course module nodes, 34 knowledge nodes, and 98 course-to-knowledge teaching relationships. The system supports target-job course path recommendation, quiz-based ability gap diagnosis, personalized learning route generation, remedial resource recommendation, and teacher-facing curriculum improvement analysis. The pilot results show that the proposed graph can provide explainable learning paths and curriculum evidence, although larger job samples, formal teacher coding agreement, and real classroom evaluation are still required.

Keywords: Ability gap diagnosis; Computer networking; Job-oriented learning path, Knowledge graph.

1. Introduction

Computer networking is a core area in higher vocational information technology education. Its curriculum usually covers network fundamentals, routing and switching, network services, wireless networking, security operations, data center operations, cloud networking, and system maintenance. Although these topics are important for technical training, students often ask a practical question: “Why do I need to learn this course, and which real jobs use these knowledge points?” This question is especially important in higher vocational education because courses are expected to connect directly with employability, occupational abilities, certificates, and practical technical tasks.

Traditional course resource platforms usually organize materials by chapters, slides, experiments, and exercises. This organization helps teachers manage teaching resources, but it does not fully explain the relationship between job demand and course knowledge. Job postings, certificates, and competition tasks may contain useful evidence about required abilities, yet they are often stored separately from curriculum documents and student assessment data. As a result, students may not clearly see the value of a course, and teachers may lack a traceable basis for curriculum adjustment.

Knowledge graphs provide a useful method for representing heterogeneous educational and occupational entities. A knowledge graph can connect jobs, abilities, courses, knowledge points, tools, certificates, tasks, and evidence sources through explicit relationships. Compared with a simple list of course chapters or job skills, a knowledge graph can answer path-based questions such as “Which course modules support the ability required by a network operations role?” and “Which knowledge points should be reviewed if a student wants to become a security operations engineer?” Knowledge graphs have been widely studied for representation, acquisition, refinement, and application in different domains [1–5]. In education, learning analytics and educational data mining also emphasize the value of connecting learning evidence with teaching decisions [6, 7].

This paper presents a job-oriented knowledge graph for higher vocational computer networking courses. The main contributions are as follows. First, a layered architecture is proposed to separate controlled job evidence from the student-facing teaching graph. Second, the graph schema defines a teaching explanation path from job roles to abilities, course modules, and knowledge points. Third, deterministic graph services are designed for course path recommendation, ability gap diagnosis, personalized learning route generation, remedial resource recommendation, and teacher-facing curriculum improvement. Fourth, a Zhejiang pilot system is implemented and evaluated through graph construction results and path completeness validation.

2. System Design and Methodology

2.1. Project Scope and Data Governance

The project focuses on computer networking-related job roles suitable for higher vocational students and junior technical career paths. The pilot scope is limited to Zhejiang Province. Typical target roles include network operations engineer, network engineer or network administrator, network implementation or system integration engineer, IDC or NOC operations engineer, network security operations or SOC engineer, cloud networking or cloud operations engineer, and wireless network engineer.

The system does not treat job postings as the graph itself.

Instead, recent job demand is used as evidence. Raw postings, short evidence snippets, source records, and review items are stored in a controlled evidence layer. The student-facing graph only publishes normalized concepts and reviewed relationships. This design reduces data risk and avoids turning the system into a recruitment database.

The data governance principles are as follows. Public or authorized low-risk sources are preferred. Personal identifiers, recruiter contact information, private communication records, resumes, hidden API data, login-only content, and full job descriptions are not published. The allowed outputs are normalized nodes and relationships, aggregated evidence counts, short evidence snippets when necessary, source references, retrieval dates, confidence values, and review status.

2.2. System Architecture

The proposed system consists of four layers: the evidence layer, the normalization layer, the knowledge graph layer, and the learning service layer. The evidence layer records source registration, job samples, evidence snippets, and review items. The normalization layer standardizes job names, skill aliases, tool names, vendor names, certificate names, and ambiguous terms through dictionaries and human review. The knowledge graph layer publishes normalized jobs, abilities, course modules, knowledge points, tools, certificates, tasks, and vendors. The learning service layer supports course path recommendation, ability gap diagnosis, personalized learning routes, remedial resource recommendation, and teacher-facing improvement reports.

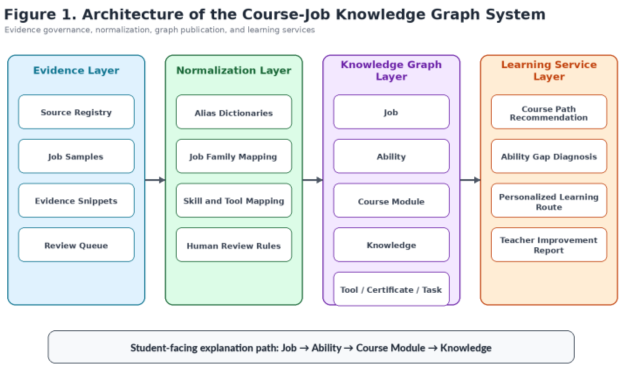

Figure 1. Architecture of the Course-Job Knowledge Graph System

The prototype uses Vue 3 and Vite for the frontend, ECharts for graph visualization, Express APIs for service access, and Neo4j for graph storage. When the backend service is unavailable, the frontend can fall back to local mock graph data. This design supports both classroom demonstration and later backend integration.

2.3. Graph Schema and Teaching Path

The graph schema is designed around two types of entities. The first type is the student-facing concept layer, including Job, Occupation Family, Ability, Course Module, Knowledge, Tool, Certificate, Task, and Vendor. The second type is the controlled evidence layer, including Source, Posting, Evidence Snippet, Ingest Run, and Review Item.

The main relationships include Job requires Ability, Course Module trains Ability, Course Module teaches Knowledge, Ability is supported by Knowledge, Job uses Tool, Job prefers Certificate, Job performs Task, Task applies Knowledge, and Tool is made by Vendor. For student-facing explanation, this paper defines the learning-path notation as:

Job → Ability → Course Module → Knowledge

The arrows in this notation indicate the explanation order presented to students, not necessarily the physical direction of every stored graph edge. In other words, a student first selects a target job, then sees the abilities required by that job, then sees the course modules that train those abilities, and finally sees the knowledge points taught in those modules.

More formally, the explanation path is represented by the validated triples (j,requires,a), (c,trains,a), and (c,teaches,k), where j, a, c, and k denote a job, an ability, a course module, and a knowledge point, respectively. Since the stored relationship (c,trains,a) points from a course module to an ability, the student-facing path uses a pedagogical traversal order rather than a strict edge-direction statement. This distinction allows the system to remain graph-model consistent while still presenting a simple learning path to students.

Course-to-knowledge relationships are further marked by teaching priority: core, support, or context. Core knowledge points must be taught, support knowledge points help students complete technical tasks, and context knowledge points provide extension or application background.

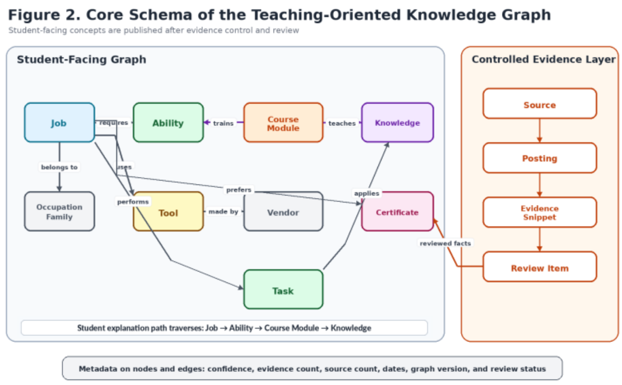

Figure 2. Core Schema of the Teaching-Oriented Knowledge Graph

Each node and relationship can include metadata such as confidence, evidence count, source count, posting count, source type, first-seen date, last-seen date, graph version, and review status. These metadata fields are important for traceability, review, and later graph quality evaluation.

2.4. Graph Construction Procedure

The graph construction process includes ten steps. First, job-related sources are registered with source type, permission basis, risk level, and collection scope. Second, allowed job evidence is stored as metadata and short evidence snippets. Third, job samples are filtered by region, education level, and technical relevance. Fourth, job descriptions are divided into requirements, responsibilities, benefits, and contact-related sections. Fifth, entities such as roles, abilities, knowledge points, tools, vendors, certificates, tasks, and courses are extracted. Sixth, aliases are normalized. For example, “TCP IP” and “TCP-IP” are normalized as “TCP/IP,” and vendor aliases are mapped to canonical vendor names. Seventh, duplicate or near-duplicate records are removed. Eighth, normalized entities and relationships are converted into graph nodes and edges. Ninth, ambiguous or low-confidence terms are sent to a human review queue. Tenth, reviewed graph data are published as a versioned graph for

Neo4j and frontend visualization.

The construction process emphasizes review and determinism. Low-confidence extraction results are not directly published. Human review decisions determine whether a candidate term enters the student-facing graph or remains only in the evidence layer.

2.5. Recommendation and Diagnosis Methods

The course path recommendation service uses graph traversal over the student-facing explanation path, namely Job → Ability → Course Module → Knowledge. Given a target job, the system finds required abilities, identifies the course modules that train those abilities, and ranks course modules according to ability coverage and teaching priority. In the current prototype, core, support, and context knowledge points can be weighted as 3, 2, and 1, respectively. The output includes recommended course modules, covered abilities, knowledge point priority distribution, recommendation reasons, and uncovered abilities if any.

The ability diagnosis service maps quiz questions to knowledge points and abilities. After a student completes a quiz, question scores are aggregated into knowledge mastery scores and then into ability mastery scores. A score of 80 or above is treated as mastered, a score from 60 to 79 is treated as needing consolidation, and a score below 60 is treated as weak. The diagnosis result is then connected back to the target job and course modules, allowing the system to generate a personalized route.

The personalized learning route follows four stages: prerequisite review, core module learning, weak-point practice, and reassessment. Remedial resources are recommended only for weak knowledge points or missing prerequisites. Each recommendation is explainable through a graph path rather than an opaque model score.

3. Discussion

The proposed knowledge graph has three main advantages. First, it connects job evidence with course teaching through explicit graph paths. This makes the value of course modules easier to explain to students. Second, it separates evidence governance from student-facing graph publication. Raw job postings and evidence snippets remain controlled, while students only see normalized and reviewed concepts. Third, the system supports both student learning and teacher curriculum improvement. Students receive explainable course paths and personalized routes, while teachers receive quality summaries and improvement evidence.

The graph also reflects the teaching characteristics of higher vocational education. Courses are not treated as abstract academic subjects. Instead, they are linked to occupational abilities and practical knowledge. The Job → Ability → Course Module → Knowledge explanation path matches the way students understand career goals and the way teachers organize classroom teaching.

However, several limitations remain. The first limitation is sample size. The pilot currently uses a manually normalized seed dataset and has not reached a large-scale evidence base. The second limitation is coding reliability. A formal two-teacher coding process and inter-rater agreement calculation are still needed. The third limitation is student evaluation. Current diagnosis and route functions are mainly prototype functions and should be tested with real students. The fourth limitation is graph coverage. The current knowledge point scale is useful for a pilot, but more reviewed knowledge points are needed for a complete curriculum graph.

Future work should expand the job sample size, introduce a formal ability coding workflow, integrate certificate and competition task mappings, and conduct classroom trials with real student data. The system should also improve teacher-facing import, export, and review functions so that graph maintenance can become part of normal curriculum development.

4. Conclusion

This paper proposed a job-oriented knowledge graph for computer networking courses in higher vocational education. The system uses job demand as evidence, but it does not expose raw job postings as the teaching graph. Instead, it builds a controlled evidence layer, normalizes job and skill concepts, and publishes a student-facing graph containing jobs, abilities, course modules, knowledge points, tools, certificates, tasks, and vendors. The main explanation path is defined as Job → Ability → Course Module → Knowledge, which helps students understand why a course matters and which real jobs use its knowledge points.

A Zhejiang pilot system was implemented with Vue 3, ECharts, Express APIs, and Neo4j. The pilot graph contains 123 nodes and 367 relationships and supports course path recommendation, ability gap diagnosis, personalized learning route generation, remedial resource recommendation, and teacher-facing curriculum improvement. The results show that the proposed graph can provide explainable links between occupational demand and course learning. Future work should expand the evidence base, complete formal teacher coding validation, integrate certificate and competition requirements, and evaluate the system in real classroom settings.

Acknowledgment

We thank the teachers, students, and technical reviewers who participated in project discussion, course knowledge organization, graph validation, and prototype testing. This work may be supported by a school-level teaching reform project or professional group construction project. The specific funding information can be added after confirmation.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” Scientific American, vol. 284, no. 5, pp. 34–43, 2001.

[2] A. Hogan, E. Blomqvist, M. Cochez, et al., “Knowledge graphs,” ACM Computing Surveys, vol. 54, no. 4, article 71, 2021.

[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A survey on knowledge graphs: Representation, acquisition, and applications,” IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 2, pp. 494–514, 2022.

[4] H. Paulheim, “Knowledge graph refinement: A survey of approaches and evaluation methods,” Semantic Web, vol. 8, no. 3, pp. 489–508, 2017.

[5] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs,” Proceedings of the IEEE, vol. 104, no. 1, pp. 11–33, 2016.

[6] G. Siemens, “Learning analytics: The emergence of a discipline,” American Behavioral Scientist, vol. 57, no. 10, pp. 1380–1400, 2013.

[7] C. Romero and S. Ventura, “Educational data mining and learning analytics: An updated survey,” WIREs Data Mining and Knowledge Discovery, vol. 10, no. 3, article e1355, 2020.