Journal of Electronic Research and Application, 2026, Volume 10, Issue 9
https://bbwpublisher.com/journal/JERA
Online ISSN: 2208-3510
Print ISSN: 2208-3502

Construction of a University Computer Resource Repository Based on Knowledge Graph

Han Wang[1], Aihe Zhou[2]

[1]School of Big data, Baoshan University, Baoshan 678000, Yunnan, China
[2]Mahasarakham University, Mahasarakham 44150, Thailand

Copyright: © 2026 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: University computer courses produce a large number of teaching materials in various forms. Slides, videos, exercises, experiment reports and project cases are usually in different systems and do not have semantic links; therefore, retrieval is inefficient and resource reuse is not possible. This paper employs a knowledge graph to construct a repository of university computer resources. A course ontology is designed to organize concepts, competencies, and resources within a unified framework. Entity and relation extraction are performed on course materials, followed by knowledge fusion and human-in-the-loop quality control. A graph-based repository was constructed in Neo4j and tested with real data from three computer courses. The constructed graph has 1864 entities and 5213 triples. The F1 score for knowledge extraction is 0.87, and the mean query response time has fallen from 2.8 seconds to 0.6 seconds after switching to keyword search.

Keywords: Knowledge graph; Resource repository; Computer education; Ontology; Resource recommendation

Online publication: October 30, 2026

1. Introduction

The contents of university computer courses are quite extensive, covering the basics of programming, data structures, operating systems, artificial intelligence, etc. Teaching materials for these courses are created by different teachers and take many forms. Slides, video recordings, exercise banks and experiment guides are usually in the learning management system, cloud drive and local folder. When students need to learn about a certain topic, they have to check many places for it. Teachers have the same problem of preparing lectures and designing homework. The problem is not a lack of resources but the absence of an organized system that reveals how knowledge points are connected.

A knowledge graph represents facts in an organized manner, where entities are connected through typed relations. In an educational context, it can capture relationships among concepts and link them to relevant teaching materials. Systematic reviews have shown that knowledge graphs are being used in education to plan learning paths, recommend teaching materials and carry out intelligent assessment [1]. A knowledge

graph can be used to organize the materials of the course resources in a more intelligent way. A Knowledge Graph contains the meaning of all resources and is not a simple file or a relational database. The association among the two ideas has been saved in the system, and it can answer questions that a folder structure cannot, such as which concepts need to be mastered before learning dynamic programming. A primary reason why knowledge graphs are considered a strong foundation for digital teaching resources is their rich semantic structure [2].

Some ways have been used to construct course knowledge graphs. According to the recent survey of the application of knowledge graphs and large language models in education, most construction tasks and extraction technologies have been identified; however, a prerequisite for their actual use is the matching of resources with knowledge points [3]. Based on the above results, this paper will focus on one particular problem: how to build a resource repository for university computer courses based on a knowledge graph. There are three reasons. A Course Ontology for Concepts, Competencies and Resources is Designed. There is now an extraction and fusion pipeline with human intervention. Deploy the working repository and then, according to the actual course data, carry out the evaluation. The rest of this paper is organized as follows. Section 2 is the related work on educational knowledge graphs and digital resource platforms. Section 3 is the Ontology Design and the Extraction and Fusion Pipeline. Section 4 is the System Architecture and Recommendation Mechanism. Section 5 is the evaluation results section, and Section 6 is the conclusion.

2. Related work

Recently, educational knowledge graphs have gained the attention of scholars. CourseKG is an educational knowledge graph based on course information for precise teaching. Based on the data from the course, this approach can obtain teaching ideas and design personalized study plans to improve the overall quality of education [4]. The other way is automatic construction. Intelligent tutoring systems have been used to automatically build multimodal knowledge graphs, and with the help of retrieval-augmented generation, structured knowledge has been extracted from course videos and textbooks; it has been shown that end-to-end graph construction for complex course content is feasible.

Knowledge Graphs can help organize courses and improve teaching. Recently, a study has put forward a knowledge-graph-based Design for digital-intelligent curriculum modules in auditing education and found that graph-structured curriculum organization can promote the integration of interdisciplinary content [5]. Therefore, it has been concluded that knowledge graphs can provide semantics for digital teaching materials.

3. Knowledge graph construction method

3.1. Ontology design

An ontology is the scheme of the repository. The four kinds of entities are course, concept, competency and resource. A course is a single unit of study, such as Data Structures or Operating Systems. A concept is a point of knowledge that can be taught and tested, such as binary tree traversal. A competency is an ability that students are expected to have, such as algorithm design. A resource is an actual teaching material, such as slides, videos, exercises, experiment reports and project cases.

The first five kinds of relationships are prerequisite, belongs-to, trains, covers and attaches-to. The first connects the two ideas and shows the order of learning; for example, graph traversal needs queue operations.

The trains relation links a competency to the concepts that support it, and the covers relation links a course to the concepts covered in that course. An Attachment-to-record links a material to the concepts it covers, and also has two other attributes: difficulty and approximate study time. All the entities have a small number of attributes. A concept has a definition, a difficulty level and an optional list of aliases. A Resource stores the form, length, upload time and use count. The above are the filter conditions for retrieval; for example, only videos can be chosen by students. The ontology is relatively small, and teachers do not need to be trained to add to it. It is also a general rule that the schema of a course graph should be based on teaching goals rather than on the convenience of data storage.

3.2. Entity and relation extraction

The three steps in processing course documents are text cleaning, entity extraction and relation extraction. Delete the header, footer and other repeated boilerplate content from the text in the slides. Build a dictionary and a pre-trained language model, then conduct entity extraction. The dictionary contains course syllabi and textbook glossaries collected by the teaching group, and there are about 800 domain terms in total. The language model can handle words that are not in the dictionary, such as new frameworks mentioned in the recent lecture notes. Rules are established for the structure of typical sentences, and then, with a small amount of annotated data, fine-tuning is performed on the model for relation extraction. The output of this stage is a set of candidate triples and their source positions; they will be verified later.

The similarity of two entity mentions is a weighted sum of name similarity, semantic similarity and context similarity:

$$sim(a,b) = \lambda_1 sim_{name}(a,b) + \lambda_2 sim_{emb}(a,b) + \lambda_3 sim_{ctx}(a,b) \tag{1}$$

And the weights satisfy $\lambda 1 + \lambda 2 + \lambda 3 = 1$, and they have been optimised on a validation set. String matching is the similarity of names; word embeddings are used to compute semantic similarity; contextual similarity compares sets of co-occurring concepts. Pairs with similarity above the threshold are merged after manual confirmation. Learning path recommendation systems based on knowledge tracing and large language models have shown that the alignment of graph construction with teaching goals is necessary for the latter [6].

Knowledge representation and link prediction are done with the TransE model. All the entities are mapped to a low-dimensional vector, and all the relations are expressed as translations. The score function is:

$$f(h,r,t) = \left\| h + r - t \right\| \tag{2}$$

where h, r, and t are the head entity, the relation, and the tail entity, respectively. A small value indicates a high probability of a triple, and margin-based ranking loss is used to train the model on the extracted set of triples. The learned embeddings are used to compute the semantic distance of the resources for recommendation [7,8].

3.3. Knowledge fusion and quality control

Gather the resources and knowledge from all the courses into a single repository graph. According to the construction experience of knowledge graph-based learning systems, ontology-level mapping and instance-level alignment are combined to integrate resources from all the courses [9]. Align the equivalent concepts in all the courses and resolve the conflicts among them according to the priority rules. For example, the idea of

recursion can be found in Data Structures and Artificial Intelligence, and the two cases can be combined into a single node with the same set of resources. The concept definition is taken from the course that covers this topic in more detail, and the other instance is saved as an alias.

A poor quality of triples will be misleading for the students. The Repository has a two-tier lock. The first is automatic; rule checks will verify that the type, cardinality and value range are correct. A triple that directly links a course node to a resource node without an intermediate concept is considered a schema violation. The second level is human review, and the course teacher will correct any incorrect relationships in the merged graph. According to the latest systematic reviews of hybrid knowledge tracing models, adding knowledge representations to students' response data can increase the accuracy of learning assessment. Human-in-the-loop mode will ensure that the final graph is consistent with the actual teaching logic of the courses. Every triple records the source document, extraction method and reviewer so that any errors can be traced back to their origin. After the fusion round, the number of duplicate concept nodes has been reduced by 38% compared to the union of the three separate course graphs, and a uniform naming convention for all courses has been maintained in the merged graph.

4. Repository system design

4.1. System architecture

The repository consists of three layers. The Data Layer contains the real data and metadata. The graph layer stores the knowledge graph in Neo4j, and each resource is connected to one or more concept nodes. The top part is for the search, navigation, recommendation and statistics of a website. The three layers have a good division of labor and will not be affected by changes to the storage engine.

A small front-end and a set of RESTful services will be employed for the Web interface. The structure of the concept browser is a tree, and a search box can be used for natural language query to find relevant concept nodes. The Statistics module will list the coverage deficiencies, such as concepts that do not have corresponding exercises or prerequisite relationships. Automatically generate the report and send it to the teaching group every two weeks. The front end is responsive; therefore, it can be used in various browsers on computers and mobile phones, and students can study the materials in the laboratory or on their way to school.

4.2. Resource organization and governance

All the resources have attributes, such as titles, types, formats, difficulties, approximate learning durations, and related knowledge, etc. The connection of attachments is the link to resources and knowledge. When a student clicks on a concept node, all the resources related to that node and the resources of adjacent nodes will be shown together. The new way is to match knowledge semantics and not filenames for retrieval.

Data governance should have an audit trail and be regularly audited, and the same will be extended to the new graph-based educational recommendation system [10]. The repository will save the history of all changes, and an audit will be conducted regularly at the end of every six months. Mark the deprecated resources as such and do not delete them; otherwise, the old references will be lost. The statistics of the constructed graph are shown in **Table 1**. The graph has three courses and twelve chapters, a total of 1864 entities and 5213 triples.

Table 1. Statistics of the constructed repository knowledge graph

Course	Chapters	Concepts	Entities	Triples
Data Structures	5	214	702	2058
Operating Systems	4	168	618	1766
Artificial Intelligence	3	130	544	1389
Total	12	512	1864	5213

4.3. Resource recommendation

Resource recommendation is built on top of the graph. If the learning goal is a set of concepts, then each resource will be given a relevance score. The score combines the semantic distance between the resource and the goal concepts with the quality level of the resource:

$$score(q,r) = \sum_{c \in C} w(c) \cdot \left(1 - d(c,r)\right) \cdot q(r) \tag{3}$$

where C is the concept set of the goal, d(c, r) is the normalized graph distance between concept c and resource r, w(c) is the importance weight of concept c, and q(r) is the quality score of resource r in the range [0, 1]. The top-rated resources will be shown in the recommendations. The recommendation list will show the matching concepts in bold so that students know why a particular resource has been recommended. When a student sets a learning goal, for example, to study for the final exam in sorting algorithms, the system will gather relevant knowledge for this purpose, rank suitable resources according to Formula 3, and return a short list of slides, exercises and videos based on their saved preferences in the profile. Formula 3 weights are initially set to be the same and will be adjusted based on teacher feedback after the first month of use.

5. Evaluation

5.1. Dataset and setup

The repository is based on the data from the three computer courses in one semester. In all, there are 46 slide decks, 38 videos, 520 exercises and 64 experiment reports. Six teachers and 120 students participated in the evaluation. The set of 800 manually annotated test sentences will be employed to evaluate the extraction results. All the experiments were carried out on a server with an Intel Xeon processor and 32GB of RAM, and each query was repeated ten times to reduce the variance. There are three courses: Data Structures is required for Operating Systems, and both will be the foundation for Artificial Intelligence. This Design will cover the chain of prerequisites in the evaluation of different courses.

5.2. Extraction performance

Table 2 reports the extraction results. The proposed method achieves an overall precision of 0.90 and a recall of 0.84, and the F1 score is 0.87. The F1 score is computed as:

$$F1 = 2PR / \left(P + R\right) \tag{4}$$

where P and R are the precision and recall, respectively. Relation extraction is slightly less effective than entity extraction because it is more sensitive to sentence structure. Based on the error analysis of the test set, most of the incorrect relations are caused by passive voice constructions and long coordinate clauses. Manual verification accuracy of a random sample of 200 triples is 93.5%, and therefore it can be concluded that the graph is reliable enough for classroom use [10].

Table 2. Knowledge extraction performance

Task	Precision	Recall	F1
Entity extraction	0.92	0.88	0.90
Relation extraction	0.88	0.80	0.84
Overall	0.90	0.84	0.87

5.3. Query performance

Table 3 is Query Performance. The mean response time of graph-based retrieval is 0.6 seconds; it is much lower than the 2.8 seconds for keyword search. The advantage of the first is that graph traversal only needs to visit the nodes related to the target concept, and keyword search must scan the entire text index. P95 time has dropped from 4.6 seconds to 1.2 seconds, and now the system can handle a large number of users. For complex queries that contain two or more concepts, the graph-based method is even better; that is, the join operation can be replaced by a path search in the graph.

Table 3. Query response time comparison

Retrieval method	Average time (s)	P95 time (s)
Keyword search	2.8	4.6
Graph-based retrieval	0.6	1.2

6. Conclusion

A Knowledge Graph will be used in this paper to build a university computer resource repository. A course ontology, an extraction and fusion pipeline, and a graph-based retrieval and recommendation system have been developed and realized. Based on experiments with real course data, the constructed graph has 1864 entities and 5213 triples, the extraction F1 score is as high as 0.87, and graph-based retrieval is about four times faster than keyword search. Users have told us that the repository will save them some time in their studies. Based on the experience of deployment, the participation of course teachers in the review loop is a reason for the quality of the graph, and a light ontology is relatively simple to maintain compared with a complex one. In the future, I will add more courses to the repository, automatically update the materials according to the learning results, and explore how to use large language models for resource creation.

Funding

Joint Special Fund for Basic Research of Local Undergraduate Universities in Yunnan Province (202101BA070001-200)

Disclosure statement

The authors declare no conflict of interest.

Reference

[1] Abu-Salih B, Alotaibi S, 2024, A Systematic Literature Review of Knowledge Graph Construction and Application in Education. Heliyon, 10(3): e25383.

[2] Li Y, Liang Y, Yang R, et al., 2024, CourseKG: An Educational Knowledge Graph Based on Course Information for Precision Teaching. Applied Sciences, 14(7): 2710.

[3] Deng C, Yuan B, 2026, Research on an Intelligent Tutoring System Based on Automatic Construction of Multimodal Knowledge Graphs and Retrieval-Augmented Generation. Frontiers in Computer Science, 8: 1777749.

[4] Ren J, Wang X, 2026, Knowledge Graph-Based Design of Digital-Intelligent Curriculum Modules and Teaching Reform in Auditing. Frontiers in Education, 11: 1757383.

[5] Chen G, Song T, Wang Q, et al., 2025, Knowledge Graph and Large Language Model Integration With an Emphasis on Educational Applications: A Survey. Neurocomputing, 654: 131230.

[6] Lin Y, Wu Z, 2025, Learning Path Recommendation Based on Knowledge Tracing and Large Language Models. Electronics, 14(22): 4385.

[7] Luo G, Gu H, Dong X, et al., 2025, HA-LPR: An Adaptive Learning Path Recommendation. Education and Information Technologies, 30(10): 14597–14627.

[8] Zanellati A, Di Mitri D, Gabbrielli M, et al., 2024, Hybrid Models for Knowledge Tracing: A Systematic Literature Review. IEEE Transactions on Learning Technologies, 17: 1021–1036.

[9] Zhou L, Wang Y, 2025, Simulation of Personalized English Learning Path Recommendation System Based on Knowledge Graph and Deep Reinforcement Learning. Scientific Reports, 15(1): 34554.

[10] Yang Y, Peng X, Chen M, et al., 2025, An Explainable Graph-Based Course Recommendation Model Based on Multiple Interest Factors. Expert Systems with Applications, 264: 125889.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.