Learning Path Recommendation for MOOC Platforms Based on a Knowledge Graph

Hui Chen[1], Chuantao Yin[2(✉)], Xin Fan[2], Lei Qiao[2], Wenge Rong[1], and Xiong Zhang[1]

[1] Computer Science and Engineering School, Beihang University, Beijing, China
{chenhui,w.rong,xiongz}@buaa.edu.cn
[2] Sino-French Engineer School, Beihang University, Beijing, China
chuantao.yin@buaa.edu.cn, rachel.fan.xin@centralepekin.cn

Abstract. With the development of Internet technologies and the increasing demand for knowledge, increasingly more people choose online learning platforms as a way to acquire knowledge. However, the rapid growth in the types and number of courses makes it difficult for people to make choices, which leads to a series of problems, such as unsystematic learning processes and a low learning efficiency. Based on the current course situation of MOOC (massive open online courses) platforms, this paper proposes a new automated construction method for course knowledge graphs. A course knowledge graph is constructed by annotating the pre-knowledge of each course and calculating the similarity between courses, and it is displayed using the Neo4j graph database platform. After completion of the course knowledge graph, the knowledge graph of the courses is used to study learning path recommendation algorithms, including rule-based and machine learning based algorithms, and to perform a comparative analysis using the higher education formation program of a university.

Keywords: Smart learning · MOOC · Knowledge graph · Learning path recommendation · Machine learning

1 Introduction

Nowadays, MOOCs (massive open online courses) have been hot topics in the field of education. Moocs ease learning tasks and enable users to learn at their own pace and comfort [1], but they face a few challenges. On one hand, the unstructured learning resources have brought unprecedented problems of cognitive overload to learners. On the other hand, the course recommendation model ignores the semantic relation between the actual content of the course and the keywords.

Supported by the National Natural Science Foundation of China (No. 61977003).

H. Qiu et al. (Eds.): KSEM 2021, LNAI 12816, pp. 600–611, 2021.
https://doi.org/10.1007/978-3-030-82147-0_49

The course knowledge graph can solve the first problem. The knowledge graph is a graph-based data structure, composed of nodes and edges, where nodes refer to entities and edges refer to relations between entities. It integrates scattered courses with knowledge points, and fully reflects the relation between courses and knowledge points [2]. To solve the second problem, the learning path recommendation can select a suitable learning path to the learner according to the learner's information.

We propose a learning path recommendation for MOOC platforms based on the course knowledge graph. It can provide learners with personalized learning programs and with greater autonomy by flexibly arranging teaching methods.

The main contributions of this paper are as follows: proposition of an improved course text vector calculation method based on TF-IDF by using the course information of MOOC; establishment of course knowledge graph according to the course model, which includes courses nodes, knowledge points nodes, and their relations; visualization of course knowledge graph with Neo4j graph database platform; classification of courses in the course knowledge graph by using machine learning classification algorithms; and course recommendation based on the knowledge graph and student information.

2 Related Studies

The course knowledge graph construction process includes the definition of the course knowledge graph model, entity recognition, and relation extraction.

2.1 Course Knowledge Graph Model

Yongyou Zhong et al. [3] proposed a method for constructing the ontology of course knowledge points. The relation of course knowledge points includes inheritance relation, integral part relation, instance relation, dependency relation and parallel relation.

Pingyi Zhou et al. [4] proposed a method for automatically constructing a course knowledge graph based on a semantic relation model. The method is mainly composed of three parts: the extraction of the semantic entities, the expression of the course semantics and topic semantics, and the construction of the course dependencies.

2.2 Entity Recognition

Entity recognition [5] automatically discovers the named entities from Internet texts. The main methods include rule-based methods, statistical machine learning-based methods, and deep learning-based methods.

Rau et al. [6] proposed for the first time a method of combining manually written rules with heuristic ideas, which realized the automatic extraction of named entities of company name type from the text.

In recent years, many works have proposed the use of neural networks to automatically capture effective features from text, and then complete named entity recognition.

2.3 Relation Extraction

Relation extraction [7] automatically extracts the relations between the entities from the Internet text. The main methods include template-based and machine learning-based methods.

For the first time, Kambhatla et al. [8] used a feature vector-based method to integrate entity context information, syntactic analysis trees, dependency relations and other features, and combine vocabulary, syntax and semantic features with the maximum entropy model to classify relations.

Hasegawa et al. [9] first proposed the use of unsupervised methods for relation extraction. They first obtained named entity recognition and its context information, then clustered similar named entity pairs, and finally selected core vocabulary to label various semantic relations.

2.4 Learning Path Recommendation

Learning path design is the main activity of learning design, which is usually a very complex task for learners. The learning recommendation system can help learners find suitable learning objects and establish effective learning paths in the learning process. Durand earlier proposed a graph-based learning path recommendation method [10].

Lumy Joseph et al. [11] believe that in an online learning environment, the system ignores the student's personal information and provides predefined learning content for everyone, which will affect the student's learning process.

Zhu et al. [12] proposed a learning path recommendation based on knowledge graph. They first divided the learning scenario into four different models, and then adjusted the variables and weights according to the scenario's preference for learning paths. Finally, the knowledge graph is used as the basis of the recommendation algorithm, and the learning path is recommended, and a good recommendation effect is achieved.

3 Construction of Course Knowledge Graph

First, this paper collected data from the Chinse MOOCs. Then, this paper proposed a method for constructing the MOOC course knowledge graph. Based on the course knowledge graph, this paper recommended the learning path for the learner. The process is shown in Fig. 1.

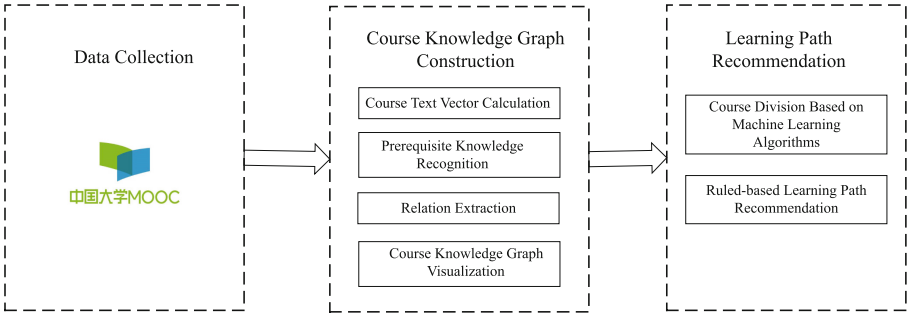

Fig. 1. The research process.

3.1 Course Knowledge Graph Model

This paper used the course text from a Chinese MOOC, took the courses and knowledge points as nodes, took the similarity and prerequisite relations as edges. The attributes and relations of the course nodes and the knowledge point nodes are shown in Fig. 2.

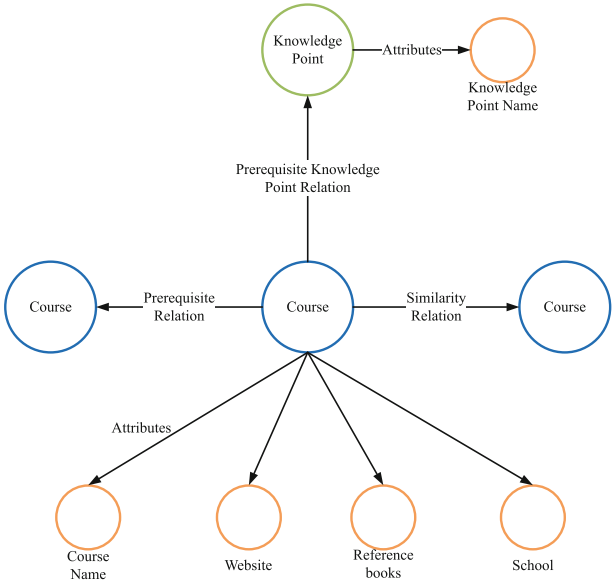

Fig. 2. The diagram of course knowledge graph model.

3.2 Data Collection

This paper used the Selenium tool [13] to obtain information from all 1808 courses from the MOOC, including the course name, website, summary, school, reference books, agenda, preliminary knowledge, and course category. The number of courses in each category is shown in Table 1.

Table 1. Number of courses in each category

Course category	Number of courses
Computer	203
Management	259
Psychology	20
Language	53
History	142
Design	101
Engineering	420
Science	310
Biology	220
Philosophy	20
Law	60
Total	**1808**

3.3 Course Text Vectorization

The main methods of constructing course text vectors are: TF-IDF algorithm and Word2Vect algorithm. Word2Vec is a commonly used method, which mainly includes Continuous Bag-of-Words Model (CBOW) and Skip-gram. This type of method mainly completes the prediction and classification task by training a neural network.

In this paper, we choose the Skip-gram model to train the word vector, and then construct the text vector based on the word vector. As shown in Eq. 1, we add the word vectors of all words in the text and take the average value as the vector of the text s_i, where $e_{i,j}$ is the word vector of the jth word in the text d_i, and l_i is the number of words.

$$s_i = \frac{1}{l_i} \sum_{j=1}^{l_i} e_{i,j} \tag{1}$$

Due to the serious unevenness of the number of courses in different categories, this paper proposes to use LDA (Linear Discriminant Analysis) [14] to reduce

the dimensionality of the course text vector. Based on supervised learning, LDA projects the data to the low-dimensional feature space while minimizing the variance of data belonging to the same category, and maximizing the variance of data belonging to different categories. Therefore, we can use LDA to obtain dense course text vectors that incorporate course category information.

We take two categories of data as examples to illustrate the principle of the LDA algorithm. Given a dataset $X = \{x_1, ..., x_m\}$, where x_i represents the n-dimensional feature vector of the i sample, $Y = \{y_1, ..., y_m\}$ represents the category label corresponding to each sample, and $y_i \in \{0, 1\}$ is the binary label of the ith sample. X_i, N_i are the set of samples in the i class and the number of samples in the i class, and μ_i and Σ_i are the mean value and feature of the feature vector of the i class samples. The calculation process is shown in Eqs. 2 and 3.

$$\mu_i = \frac{1}{N_i} \sum_{x \in X_i} x \tag{2}$$

$$\Sigma_i = \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T \tag{3}$$

w is the project vector, and the projections of the mean vector of class 0 and class 1 are respectively $w^T\mu_0$ and $w^T\mu_1$, LDA maximizes $\left\|w^T\mu_0 - w^T\mu_1\right\|_2^2$ to make the distance between classes as large as possible, and at the same time minimizes $w^T\Sigma_0 w + w^T\Sigma_1 w$ to make the distance between the features in same category as close as possible. Therefore, the optimization goals of LDA are:

$$\arg\max_w J(w) = \frac{\left\|w^T\mu_0 - w^T\mu_1\right\|_2^2}{w^T\Sigma_0 w + w^T\Sigma_1 w} = \frac{w^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)w}{w^T(\Sigma_0 + \Sigma_1)w} \tag{4}$$

Then, we calculated the silhouette coefficient [15] of TF-IDF algorithm, Word2Vect algorithm and the related algorithms that add LDA. The value of the silhouette coefficient is between -1 and 1. The closer to 1, the better the cohesion and separation.

3.4 Relation Extraction

For a course, the paper first established the prerequisite relations according to the preliminary knowledge tag of the courses. If there is a course that matches the preliminary knowledge tag, a prerequisite relation between courses will be established; if there is no matching course, a prerequisite relation between the course and preliminary knowledge points will be established. In addition, we directly add the course series to the prerequisite relations, such as analysis 1, analysis 2, etc. Finally, the similarity relations between courses were calculated based on the cosine similarity.

4 Learning Path Recommendation Based on a Course Knowledge Graph

Based on the course knowledge graph, this paper combines students' learning information, such as information about the courses they already studied, to recommend personalized and complete learning paths for students.

4.1 Division of Course Levels Based on Machine Learning Algorithms

This paper considers the process of recommending courses in the initial state as a multi-classification problem of assigning courses to four levels.

The division is divided into two steps. The first thing to do is to create the prior set. We assign certain courses to four different levels according to certain rules to establish the prior set. The rules are as follows.

- All courses without prerequisite courses are listed as the first level.
- If a course has a prerequisite course in the $k-1$ level, it is listed as the k level.

In the second step, we use a machine learning classification algorithm to complete the multi-classification of the courses that cannot be assigned according to the rules. The class labels of the divided courses are used as the training set to train the classifier, and the trained classifier is used to classify the courses without class labels.

4.2 Rule-Based Learning Path Recommendation

After students have completed a certain number of courses, we will formulate the rules based on the knowledge graph and students' MOOC information to complete the learning path recommendation. We can define the degree of recommendation of a course as Eq. 5, where n_{pass} indicates the number of prerequisite courses passed of the course, n_{total} means the total number of prerequisite courses, $avg\left(\frac{note_{obtain}}{note_{pass}}\right)$ means the average value of the ratio of the score obtained of the prerequisite course and the passing score of the prerequisite course, $n_{similar}$ represents the number of failed courses whose cosine similarity with the course is greater than 0.2, $n_{total\_non\_pass}$ indicates the total number of failed courses, and $avg(\frac{note_{similar}}{note_{pass}})$ represents the average of the score obtained and passing score of similar failed courses. Finally, this paper uses the Min-Max Normalization method [16] to calculate the recommendation degree, and recommends a list of courses sorted by recommendation degree to learners.

$$r = (\frac{n_{pass}}{n_{total}}) * avg(\frac{note_{otain}}{note_{pass}}) + (\frac{n_{similar}}{n_{total\_non\_pass}}) * avg(\frac{note_{similar}}{note_{pass}}) \quad (5)$$

5 Results and Discussions

5.1 Knowledge Graph Construction Results

Course Text Vector Construction. This paper used the text of the course name, summary, and agenda to calculate the course text vectors. The Silhouette coefficients of the text vectors constructed by TF-IDF algorithm, Skip-gram algorithm, TF-IDF+LDA algorithm and Skip-gram+LDA algorithm are 0.0117, 0.0370, 0.1445, and 0.2454 respectively. Therefore, this article chooses the Skip-gram+LDA algorithm.

Course Knowledge Graph Construction and Visualization. In the course similarity relation construction, we calculated the similarity between courses. Through statistics, we found that the similarity between courses is widely distributed, and the similarity between courses over 90% is less than 0.05. For one course, this article chooses the course with a similarity of more than 0.2 as its similar course.

5.2 Results of Course Division Based on Machine Learning

This paper compares the results of classification based on KNN [17] and logistic regression on the basis of the course vector constructed by the Skip-gram+LDA algorithm. When using the KNN algorithm for multi-classification, we will use 60% of the courses in the dataset to train the classifier and 40% of the courses as the test set, and set the coefficient $k = 5$. When using the logistic regression, we select one level as a positive example and the other three levels as negative examples to complete multi-classification. The result is shown in Table 2. This paper selected logistic regression for multi-classification.

Table 2. The results of KNN classifier and logistic regression on Skip-gram+LDA.

Level		Precision	Recall	F1
First level	KNN	0.77	0.78	0.78
	Logistic regression	0.91	0.53	0.67
Second level	KNN	0.23	0.21	0.22
	Logistic regression	0.20	0.38	0.26
Third level	KNN	0.25	0.19	0.21
	Logistic regression	0.19	0.53	0.28
Fourth level	KNN	0.36	0.41	0.38
	Logistic regression	0.34	0.68	0.45

The learning path recommendation results based on the Logistic Regression are shown in Fig. 3. The recommended learning path is shown in the form of a graph. The connections between the courses indicate the prerequisite relations.

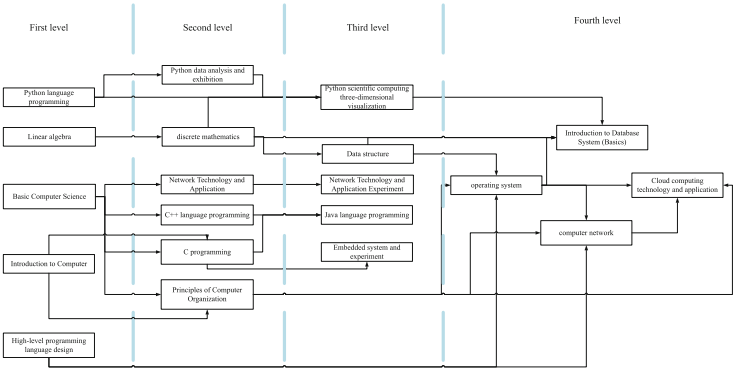

Fig. 3. Logistic regression-based learning path recommendation results.

5.3 Rule-Based Learning Path Recommendation Results

In order to verify the effectiveness of the learning path recommendation algorithm based on the course knowledge graph proposed in this paper, this paper uses two hypothetical student cases to simulate the results of the recommendation.

For student A, it is assumed that he has taken two courses at the beginning, and his scores are 78 points for Basic Computer Science and 89 points for data structure, of which 60 is a passing score. The recommended list is shown in the Table 3:

Table 3. Results of learning path recommendation for student A.

Course	Scores for course recommendation for student A
Principles of Computer organization	0.20
C programming	0.20
C++ language programming	1.00
Network technology and application	1.00
Introduction to database system (Basics)	0.00
Operating system	0.00

Later, it is assumed that student A has passed two more courses. The current academic scores are 78 points for Basic Computer Science, 89 points for data structure, 63 points for computer organization principles, and 93 points for high-level language programming. The list of recommended courses is shown in the Table 4:

Table 4. Results of learning path recommendation for Student A.

Course	Scores for course recommendation for student A
Principles of Computer organization	0.35
C programming	0.35
C++ language programming	1.00
Network Technology and Application	1.00
Introduction to Database System (Basics)	0.12
Operating system	0.88
Computer network	0.45
Cloud computing technology and application	0.00

When the student's learning information changes, the recommendation result also changes. It realizes the student's personalized course path recommendation, and also confirms the effectiveness of the recommendation degree for the student's course recommendation.

For student B, the scores are as follows: 56 points for Python programming, 66 points for discrete mathematics, 56 points for C programming, and 53 points for operating systems. There are 3 failed courses. The list of recommended courses is shown in the Table 5:

Table 5. Results of learning path recommendation for Student B.

Course	Scores for course recommendation for student B
Python data analysis and exhibition	0.79
Data structure	1.00
Java language programming	0.22
Introduction to Database System (Basics)	0.64
Computer network	0.00
Cloud computing technology and application	0.00

Later, student B passed the 2 failed courses and got 65 points for Python language programming, 79 points for discrete mathematics, 70 points for C programming, and 59 points for operating system. The list of recommended courses is shown in the Table 6:

From the table, when students have failed courses, such as student B's Python language programming, C programming, operating system, etc., the algorithm

Table 6. Results of learning path recommendation for student B.

Course	Scores for course recommendation for student B
Python data analysis and exhibition	0.76
Data structure	1.00
Embedded system and experiment	0.85
Java language programming	0.25
Introduction to database system (Basics)	0.61
Computer network	0.00
Cloud computing technology and application	0.00

will no longer recommend subsequent courses of these courses, but recommend similar courses of the failed courses. When failed course is passed, the algorithm can normally recommend subsequent courses. This verifies the effectiveness of the recommendation algorithm in recommending similar courses and subsequent courses.

6 Conclusion

With the rapid development of Internet technology, increasingly more people choose online platforms for learning. To solve the problem of the explosion in the number of course resources on an online learning platform, this paper combined the relevant machine learning algorithms and used the course information on a Chinese MOOC platform to analyze and model the courses. Then, this paper constructed a course knowledge graph based on the MOOCs. The course knowledge graph better shows the relations between the courses on the MOOC website. Then, this paper completed the learning path recommendation based on the constructed course knowledge graph. This paper provided an application basis for online learning platforms and had important practical significance.

In the future, we will try to include multiple data sources, such as the Baidu Encyclopedia, open source databases, etc. We will also compare the recommended results with other modern course recommendation method.

References

1. Nabizadeh, A.H., Gonçalves, D., Gama, S., et al.: Adaptive learning path recommender approach using auxiliary learning objects. Comput. Educ. **147**, 103777 (2020)
2. Morsi, R., Ibrahim, W., Williams, F.: Concept maps: development and validation of engineering curricula. In: 2007 37th Annual Frontiers in Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports. IEEE (2007). T3H-18-T3H-23

3. Yongyou, Z.: Ontology-based curriculum knowledge point modeling of major of information management and information system. Inf. Res. **8**, Article no. 28 (2013)
4. Zhou, P., Liu, J., Yang, X., et al.: Automatically constructing course dependence graph based on association semantic link model. Personal Ubiquit. Comput. **20**(5), 731–742 (2016)
5. Bikel, D.M., Schwartz, R., Weischedel, R.M.: An algorithm that learns what's in a name. Mach. Learn. **34**(1), 211–231 (1999)
6. Rau, L.F.: Extracting company names from text. In: Proceedings of the Seventh IEEE Conference on Artificial Intelligence Application. IEEE Computer Society (1991). 29, 30, 31, 32–29, 30, 31, 32
7. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. **3**(2003), 1083–1106 (2003)
8. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions, pp. 178–181 (2004)
9. Hasegawa, T., Sekine, S., Grishman, R.: Discovering relations among named entities from large corpora. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004), pp. 415–422 (2004)
10. Durand, G., Belacel, N., LaPlante, F.: Graph theory based model for learning path recommendation. Inf. Sci. **251**, 10–21 (2013)
11. Joseph, L., Abraham, S.: Instructional design for learning path identification in an e-learning environment using Felder-Silverman learning styles model. In: 2017 International Conference on Networks and Advances in Computational Technologies (NetACT), pp. 215–220. IEEE (2017)
12. Zhu, H., Tian, F., Wu, K., et al.: A multi-constraint learning path recommendation algorithm based on knowledge map. Knowl. Based Syst. **143**, 102–114 (2018)
13. De Medio, C., Gasparetti, F., Limongelli, C., et al.: Automatic extraction and sequencing of Wikipedia Pages for smart course building. In: 2017 21st International Conference Information Visualisation (IV), pp. 378–383. IEEE (2017)
14. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
15. Thalamuthu, A., Mukhopadhyay, I., Zheng, X., et al.: Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics **22**(19), 2405–2412 (2006)
16. Panda, S.K., Bhoi, S.K., Singh, M.: A collaborative filtering recommendation algorithm based on normalization approach. J. Ambient Intell. Humanized Comput. **3**, 1–23 (2020)
17. Fresco, R., Pederiva, A.: An approach to the process maps based on semantic web methodologies. In: Meersman, R., Tari, Z. (eds.) OTM 2003. LNCS, vol. 2889, pp. 98–108. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39962-9_22