Knowledge graph enhanced Personalized Course Recommendation for MOOCs

No Author Given

No Institute Given

Abstract. To handle the limitations of collaborative filtering-based recommender systems, knowledge graphs (KG) are getting attention as side information. However, it is not appropriate to directly apply the existing KG-based methods to the course recommendations of MOOCs for the following reasons. First, unlike movies or books, the course itself is not included in a knowledge base such as Freebase, so a new way to utilize an external knowledge base is needed. Second, the existing KG-based methods usually consist of the graph embedding module and the recommendation module. However, these two modules are difficult to consider the level of the student. We propose **KPCR**, a framework for **K**nowledge graph enhanced **P**ersonalized **C**ourse **R**ecommendation. In KPCR, internal information of MOOCs and external knowledge base are integrated through user and course-related keywords. In addition, we add the level embedding module that predicts the level of students and courses. The quantitative experiment with real-world datasets demonstrates that our knowledge graph boosts recommendation performance as side information. The results also show that two auxiliary tasks improve the recommendation performance. In addition, we qualitatively evaluate the effectiveness of KPCR through the satisfaction survey of users of the real-world MOOCs platform.

Keywords: E-learning · MOOCs · Personalized Learning · Recommender Systems · Knowledge Graph.

1 Introduction

Massive Open Online Courses (MOOCs) platforms such as Coursera, Udacity, and edX are widely used because of the advantage of providing an environment where users can learn online regardless of time and place[43].

However, despite the growing number of users learning through such platforms, MOOCs have some challenges. Students' retention rates that are less than 10% on average[1] is the main problem with MOOCs, which suggests that learners may not be able to learn through MOOCs efficiently.

One of the major factors that lower retention rates are curricula that do not reflect learners' interests[11]. Too difficult content to follow is another main factor related to the retention rates of MOOCs[21, 43], and [18] stated that providing content appropriate to the learner's level increases retention rates.

Therefore, it is necessary to recommend appropriate content in consideration of the interests and level of learners in MOOCs. [15] suggested that the course recommendation system increases the retention rates of learners in MOOCs, and [11, 12] have shown that recommending personalized courses for learners can increase the efficiency of learning through MOOCs.

For these reasons, studies have been conducted to recommend courses to users in MOOCs in various ways[22, 36, 41], and many of them use collaborative filtering (CF). CF-based methods have been widely adopted because they can capture user preferences effectively and can be easily implemented without the effort to extract features from content-based recommendation systems[28, 30]. However, CF has limitations in that it has low performance in sparse data and has a cold start problem [33, 34, 40]. Utilizing side information is evaluated as a good solution to solve these problems[30], and knowledge graphs (KG) are getting attention as side information[39, 32].

However, it is not appropriate to directly apply the existing KG-based methods to the course recommendations of MOOCs for the following reasons. First, unlike movies or books, the course itself is not included in a knowledge base such as Freebase[2], so a new way to utilize an external knowledge base is needed. Second, the existing KG-based methods usually consist of the graph embedding module and the recommendation module[8]. However, these two modules are difficult to consider the level of the students and the courses.

In this paper, we propose **KPCR**, a framework for **K**nowledge graph enhanced **P**ersonalized **C**ourse **R**ecommendation. We created a knowledge graph by integrating user-course interaction, user interests, course information, and an external knowledge base. When linking internal information in MOOCs and external knowledge bases, users and course-related keywords were used. With the external knowledge base, information related to user interests and course information (e.g., occupations, companies, or related subjects) that are not revealed in the MOOCs platform can be utilized. In addition, we created the user-course level graph containing user-course interaction and the level of users and courses. To improve the recommendation performance, we combined two additional tasks in multi-task learning: knowledge graph embedding task and node classification task, which predicts the level of courses and users.

We demonstrate that our knowledge graph boosts recommendation performance as side information through the experiment with the real-world datasets. The results also show that two auxiliary tasks improve the recommendation performance. In addition, we investigated the user satisfaction of KPCR's recommendations for users of real-world MOOCs platforms.

The organization of this paper is as follows. In Section 2, we perform a literature review. Section 3 describes concepts and notations used in this study. Section 4 presents the proposed approach with a detailed description. We discuss the experimental results on real-world datasets in Section 5 and analyze user satisfaction in Section 6. Finally, section 7 contains the conclusions and the future work.

2 Related work

2.1 Recommender System and KG in MOOCs

Many studies have been conducted on recommender systems in MOOCs platforms. For example, Multi-Layer Bucketing Recommendation (MLBR) is one of the proposed models that is based on CF and has proved its effectiveness in real-world MOOCs datasets[22]. Another model is DBNCF, which uses a highly effective DBN for feature extraction and function approximation, and mitigates the effectiveness of CF in sparse data[41]. Also, [36] created a bipartite graph context CF algorithm and applied it to MOOCs.

Meanwhile, several studies have been conducted to build knowledge graphs using user and item data from MOOCs. For example, [44] built a knowledge graph by scraping several MOOCs websites. Entity types of their knowledge graph included course group, lecturer, university, key points, Etc. The relationship types include teach, contrast, Etc. They extracted the course group's key points using TF-IDF, set the course group using K-means, and set the contrastive relationship extraction by calculating the similarity of the two courses using VSM. Another research constructed a course knowledge graph based on a course model including courses nodes, knowledge points nodes, and their relationships, and they visualized it[25].

2.2 Knowledge Graph Embedding

The term "knowledge graph" (KG) was first coined by Google and has recently been used to refer to semantic web knowledge bases such as DBpedia[24]. KGs can be used in a variety of fields, including intelligent QA systems and recommender systems[5].

A KG is a multi-relational graph, consisting of entities that are nodes of the graph and relations that are edges of the graph. Each instance of an edge can be expressed as a triplet $(head\,entity, relation, tail\,entity)$, which is often abbreviated as (h, r, t)[35]. A triplet (h, r, t) means that h has some relation r with t. For instance, $(user, enrolled\_in, course)$ means that the *user* has *enrolled in* the *course* before, and $(course, related\_to, programming$ $language)$ means that the *course* is *related to* a particular *programming language.*

Knowledge graph embedding (KGE) is an effective way to alleviate the problem of data sparsity and computational complexity[6]. KGE generally learns the representation of entities and relations by defining a scoring function over the embedding space. The scoring function $\phi(h, r, t)$ evaluates the correctness of the triplet (h, r, t)[35]. Existing KGE methods can be classified into three branches: tensor-decomposition based methods[31, 37] which regard the KG as 3D tensor and decompose it into low-dimensional vectors, translation-based methods[3, 17] which regard relations in the KG as transformation in the latent space, and deep learning-based methods[7, 20].

2.3 Knowledge Graphs enhanced Recommender Systems

Existing KG-enhanced methods can be classified into three categories. Embedding-based methods[39, 4] utilize the KG to refine the representation of users or items. Usually, two modules(the KGE module and the recommendation module) are associated with the methods. These methods are flexible in that they are suitable for most applications in various domains[8]. Path-based methods leverage the connection patterns in the graph to guide the recommendation. They usually define meta-structure to calculate the similarity between entities or encode the connection patterns[9, 38]. These methods can explain their recommendation, but they rely on manually defined meta-structures and have scalability issues. Propagation-based methods have been proposed to exploit the information in the KG fully and capture high-order connection patterns[13, 32]. These methods usually utilize GNNs to propagate and aggregate embeddings of multi-hop neighbors in the KG. Propagation-based methods can explain the recommendation. However, this method also has the scalability problem, and information loss may occur in the process of neighbor sampling[8].

Our method is embedding-based. To make a recommendation suitable for the education field, we add a module for node classification that predicts users' and courses' levels.

3 Preliminary

In this section, we clarify terminologies used in this study and present our task explicitly.

Table 1. Entity pairs and relation types of internal knowledge graph.

Entity Pair	Relation Type
User−Course	*enrolled_in*
User−Keyword	*interested_in*
Course−Keyword	*related_to*

3.1 Internal Knowledge Graph

The internal knowledge graph $KG_{internal}$ is composed of 3 types of entities and three types of relations. The types of entities are user, course, and keyword.

$$KG_{internal} = \{(h_{in}, r_{in}, t_{in}) \mid h_{in}, t_{in} \in \mathcal{E}_{internal}, \quad r_{in} \in \mathcal{R}_{internal}\}$$
$$\mathcal{E}_{internal} = \mathcal{U} \cup \mathcal{C} \cup \mathcal{K},\ \mathcal{U}\text{:set of users, } \mathcal{C}\text{:set of courses, } \mathcal{K}\text{:set of keywords}$$
$$\mathcal{R}_{internal} = \{enrolled\_in,\ interested\_in,\ related\_to\}$$

The keywords can be extracted from the user's interests and course descriptions provided by the MOOCs platform. A user's interests can be selected during the sign-up process or collected through the course history the user has taken. Course-related keywords can be obtained from course descriptions such as learning objectives, learning topics, and table of contents. Examples of keywords are '*management*', '*artificial intelligence*', and '*social science*'. The entity pairs and relation types of the internal data graph are shown in Table 1.

3.2 External Knowledge Graph

The external knowledge graph $KG_{external}$ is a subset of the external knowledge base, which is associated with the keywords mentioned in 3.1. DBpedia[24], Freebase[2], and YAGO[29] could be adopted as an external knowledge base.

$$KG_{external} = \{(h_{ext}, r_{ext}, t_{ext}) \mid h_{ext}, t_{ext} \in \mathcal{E}_{external}, \quad r_{ext} \in \mathcal{R}_{external}\}$$
$$\mathcal{E}_{external}\text{: set of entities in external knowledge base, } \mathcal{K} \subset \mathcal{E}_{external}$$
$$\mathcal{R}_{external}\text{: set of relations in external knowledge base}$$

The unified knowledge graph $KG_{unified}$ is created by integrating the internal knowledge graph and external knowledge graph through keywords. Fig.1 shows an example of a unified knowledge graph.

$$KG_{unified} = \{(h_{uni}, r_{uni}, t_{uni}) \mid h_{uni}, t_{uni} \in \mathcal{E}_{unified}, \quad r_{uni} \in \mathcal{R}_{unified}\}$$
$$\mathcal{E}_{unified} = \mathcal{E}_{internal} \cup \mathcal{E}_{external}, \ \mathcal{R}_{unified} = \mathcal{R}_{internal} \cup \mathcal{R}_{external}$$
$$\mathcal{E}_{external} \cap \mathcal{E}_{internal} = \mathcal{K}$$

3.3 User-Course Level Graph

To recommend appropriate courses to a user while considering the difficulty of the course and the level of the user, we define the user-course level graph G_{level}. We regard the user-course bipartite graph as a homogeneous graph and label each node with course-level and user-level. The level of user or course is defined in three stages (basic, intermediate, and advanced). The user-course level graph is used in the node classification task, which predicts the level of course or user.

$$G_{level} = \{(u, enrolled\_in, c, l_u, l_c) \mid u \in \mathcal{U}, \quad c \in \mathcal{C}, \quad l_u \in L_U, \quad l_c \in L_C\}$$
$$L_U = \{U_{t_0-lv1},\ U_{t_0-lv2},\ U_{t_0-lv3}, .., \ U_{t_n-lv1},\ U_{t_n-lv2},\ U_{t_n-lv3}\}$$
$$L_C = \{C_{t_0-lv1},\ C_{t_0-lv2},\ C_{t_0-lv3}, .., \ C_{t_n-lv1},\ C_{t_n-lv2},\ C_{t_n-lv3}\}$$

$lv1$, $lv2$, and $lv3$ denote the basic, intermediate, and advanced level respectively. $t_k(k = 0, 1, .., n)$ means category k (e.g, *computer science*), and there are n categories of courses. The level of the course can be extracted from the course description. A user's level can be collected through the self-reported information entered by the users, test scores when registering, or the difficulty of the courses the user has taken so far.

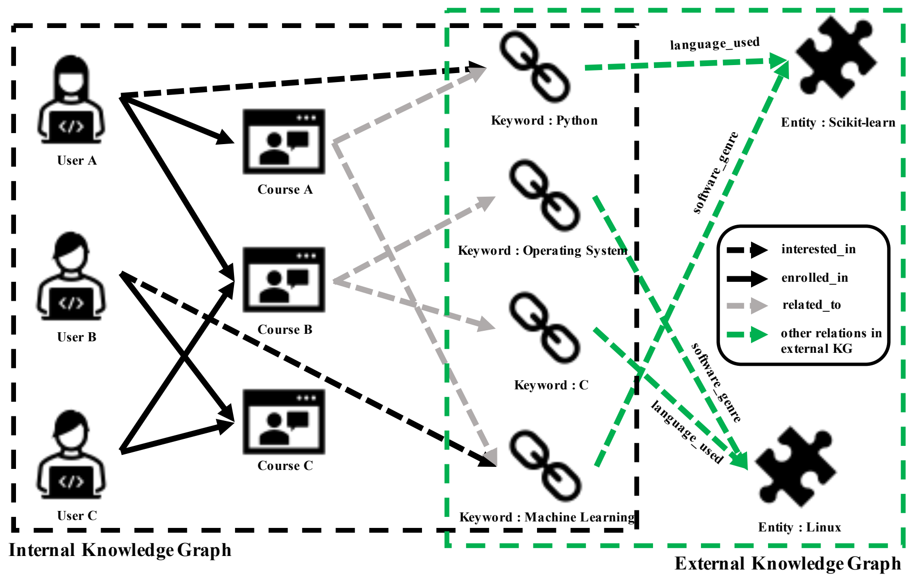

Fig. 1. Example of an unified knowledge graph.

3.4 Task Formulation

The task formulation of this study is as follows: Given the unified knowledge graph and the user-course level graph. We aim to recommend top-K courses in which each user would like to enroll.

4 Methodology

This section presents how we extract the representations of users and courses given the unified knowledge graph and the user-course level graph.

4.1 Structural Embedding

In order to get the structural embedding, we adopt the knowledge graph embedding (KGE). KGE represents the KG components (entities and relations) into low dimensional vectors while preserving the semantic meaning and connectivities. In general, KGE is learned by defining the scoring function of a triplet (h, r, t).

In this study, we use ConvE[7], one of the state-of-the-art methods for KGE. ConvE is known to be highly parameter-efficient and expressive through multiple layers of non-linear features. The scoring function of ConvE is as follows:

$$\phi(h, r, t) = f\left(vec\left(f\left(\left[\bar{e_h} \parallel \bar{r_r}\right] * \omega\right)\right) W\right) e_t$$

e_h, r_r, e_t denote the embedding of h, r, t respectively; $\bar{e_h}$ and $\bar{r_r}$ denote 2D reshaping of e_h and r_r; $\|$ denotes concatenation; $*$ denotes convolution operation; W denotes the weight matrix of the dense layer; f denotes ReLU function[19]; vec denotes reshaping feature map tensor $A \in R^{c\times w\times h}$ into a vector $vec(A) \in R^{cwh}$. The loss function of structural embedding module is as follows:

$$\mathcal{L}_{structural} = \sum_{(h,r,t)\in KG_{unified}} l(h,r,t)$$
$$l_{(h,r,t)} = -\frac{1}{N}\sum_i^N \left(y_{t_i} \cdot \ln(s_i) + (1-y_{t_i}) \cdot \ln(1-s_i)\right),\ s = \sigma(\phi(h,r,t))$$

y_{t_i} means the label vector with dimension $R^{1\times N}$ for 1-N scoring(it's elements are ones if there exists relationships otherwise zeros).

4.2 Level Embedding

Like [14], we use two-layer GCNs for node classification(level prediction) on the user-course level graph. The level prediction is as follows:

$$Z = f(X, A) = \mathrm{softmax}(\hat{A}\mathrm{ReLU}(\hat{A}XW^{(0)})W^{(1)})$$

Here, $\hat{A}$ is a self-loop added and normalized adjacency matrix of the user-course level graph. $W^{(0)}$ is input-to-hidden weight matrix and $W^{(1)}$ is hidden-to-output weight matrix. X is initial node data (we used the one-hot label of the nodes' ID as X). The loss function of the level embedding module is as follows:

$$\mathcal{L}_{level} = -\sum_{l\in\mathcal{Y}_L}\sum_{f=1}^F Y_{lf}\ln Z_{lf}$$

F denotes the number of node labels, $\mathcal{Y}_L$ is the set of node indices that have labels. We use the first hidden layer activation ($\hat{A}XW^{(0)}$) as the level embedding of users and courses.

4.3 Model Optimization and Prediction

Since the level and structural embedding both contain educationally important side information, we define the final embedding of the user u and the course c as follows:

$$e_u^{CF} = e_u^{level} + e_u^{structural},\ e_c^{CF} = e_c^{level} + e_c^{structural}$$

e^{level} denotes the first hidden layer activation of the user or the course in section 4.2. $e^{structural}$ denotes the representation of the user or the course from the entity embedding in section 4.1.

For the final prediction, we estimate the matching score between the user u and the course c by conducting inner product of e_u^{CF} and e_c^{CF}. According to this score, we recommend top-K courses for the users.

$$p(u,c) = {e_u^{CF}}^\top e_c^{CF}$$

We adopt the BPR[26] as the loss function of our recommendation module. BPR assumes that the user prefers the interacted item to the other non-interacted items:

$$\mathcal{L}_{CF} = \sum_{(u,c,c')\in R} - \ln \sigma(p(u,c) - p(u,c'))$$
$$R = \{(u,c,c')\,|(u, enrolled\_in, c) \in KG_{unified}, (u, enrolled\_in, c') \notin KG_{unified}\}$$

u denotes user in the train dataset; c and c' denote positive (observed) and negative (unobserved) course in the train dataset.

Our final objective function is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{CF} + \mathcal{L}_{level} + \mathcal{L}_{structural}$$

In specific, we optimize $\mathcal{L}_{CF}$ and $\mathcal{L}_{structural}$ jointly (since these two tasks are similar) and optimize $\mathcal{L}_{level}$ alternatively. Fig.2 describes the training process of our model framework.

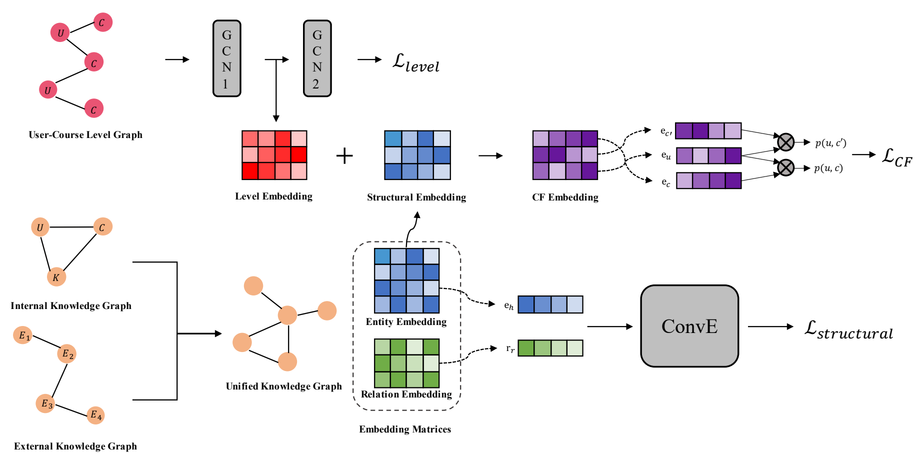

Fig. 2. Illustration of training process for our proposed KPCR.

5 Experiment 1

In this section, we demonstrate the effectiveness of our proposed method with two real-world datasets.

5.1 Dataset 1 : ESOF

We collected a real-world MOOCs dataset from ESOF, a public MOOCs platform for software education in Republic of Korea. There are five course categories: *Computing System*, *Network & Internet*, *Data Representation & Applications*, *Algorithm & Programming*, and *Understanding of Information Society*.

Table 2. Detailed statistics of the dataset.

		ESOF	XuetangX
Internal knowledge graph	#user	3,703	10,714
	#course	276	339
	#keyword	75	5
	#user-course interaction	27,502	56,029
	#user-keyword interaction	3,355	16,621
	#course-keyword interaction	1,431	339
External knowledge graph	#entities	14,578	9,042
	#relations	310	135
	#edges	28,612	18,284

Statistics of the Dataset To ensure the quality of the data, we remove users and courses with less than three interactions. Thus, there remain 3,703 users, 276 courses, and 27,502 interactions in the dataset. Table 2 illustrates the detailed statistics of the dataset. We conduct 3-fold cross-validation with this dataset to evaluate the performance.

Knowledge Graph Construction We extract the user-related keywords from the information entered by the user during the sign-up process. The user-related keywords consist of the SW field of interest and programming language of interest. The lecturers select the course-related keywords. These keywords are about the learning goal, the related SW field, the related programming language, Etc.

Meanwhile, we used Freebase[2] as an external knowledge base to create the external knowledge graph. Freebase is a database system composed of entities and relations in the real world, designed to be used as a public repository of the world's knowledge[2].

User-Course Level Graph Construction The difficulty level of the course specified in the course description was used for the course level. And the user level is set as the average level of the courses that the user has taken so far.

5.2 Dataset 2 : XuetangX

To verify that our method is effective on the dataset with various course categories, we compared the performance using the XuetangX dataset[42]. XuetangX is one of the biggest MOOCs platforms in China. We have selected and used the five most popular categories of courses provided by this platform. The course categories selected are *Computer Science*, *Medicine and Health*, *Social Science*, *Engineering*, and *Management*.

Statistics of the Dataset We sampled about 25% of all users and applied 3-core settings like section 5.1. Thus, there remain 10,714 users, 339 courses, and 56,029 interactions. Table 2 illustrates the detailed statistics of the dataset. We randomly sampled 30% of the interactions as the test set. Also, we randomly sampled 10% of the train set as a validation set for hyper-parameter tuning.

Knowledge Graph Construction Since the last update of this dataset was October 2018, we could not access the descriptions of some courses. Therefore, the course-related keywords were set as the name of the course category. The user-related keywords are the union of the course-related keywords taken by each user. As an external knowledge graph, Freebase[2] was used as in section 5.1.

User-Course Level Graph Construction Since we are not accessible to some course descriptions, the difficulty level of the course was determined according to the title of each course. For example, if a course title includes 'introduction', we regard the course's level as the basic (level 1), and if it includes 'advanced', the course's level is set as the advanced (level 3). If the title does not include difficulty information, the course level is set to the intermediate (level 2). The user's level is set as the average level of the courses the user has taken so far.

5.3 Experimental Settings

Evaluation Metrics To investigate the effectiveness of recommendations, we opted to use Recall@K and NDCG@K[27]. Recall@K (Rec@K) is the ratio of courses selected by the user among the top-K recommended list to the total number of courses that the user enrolled in. While Recall@K equivalently considers the items ranked within the top-K, NDCG@K also considers the predicted positions. We set K in [3, 5, 10].

Compared Methods The compared methods used in our experiments are as follows:

BPRMF[26] : Bayesian Personalized Ranking based Matrix Factorization (BPRMF) is a KG-free CF method based on pairwise preference.
CKE[39]: Collaborative Knowledge Base Embedding (CKE) is a method that utilizes textual, visual, and structural knowledge to refine item embedding. In our experiment, we only use a structural knowledge module consisting of TransR.
CKE-ConvE: A model that changed TransR to ConvE in CKE. We experiment with this model to check the efficiency of ConvE.
KGAT[32]: Knowledge Graph Attention Network (KGAT) is a propagation-based method that refines the embedding of users and items with knowledge-aware attention.
KPCR($L + S$): Our proposed method which uses both the level embedding module and the structural module to assist the recommendation module.
KPCR(S): Simple version of our proposed method, which does not use the level embedding module.
KPCR(S_i) : Another version of KPCR(S) which uses internal knowledge graph only.

Parameter Settings For both datasets, we fix the embedding size of all models to 64. We use 16 convolution filters with a (3x3) size in the ConvE module for our model. The dropout ratio of embedding, convolution feature map, and dense layer are [0.2, 0.1, 0.3]. To avoid overfitting, we also use batch normalization[10]. The learning rate is set as 0.001. We select Adam Optimizer[13] for all models and use early stopping technique with 10-epoch patience according to Recall@10.

5.4 Results

Table 3 shows the overall performance comparison. The experimental results showed similar trends in both datasets (see Fig.3 and Fig.4).

Table 3. Overall performance comparison.

	ESOF				XuetangX			
	Rec@5	Rec@10	NDCG@5	NDCG@10	Rec@5	Rec@10	NDCG@5	NDCG@10
KPCR($L+S$)	0.633	0.728	0.628	0.659	0.503	0.609	0.450	0.489
KPCR(S)	0.618	0.712	0.618	0.648	0.498	0.595	0.448	0.483
KPCR(S_i)	0.612	0.701	0.610	0.639	0.495	0.589	0.447	0.481
KGAT	0.602	0.701	0.593	0.626	0.452	0.548	0.404	0.439
CKE-ConvE	0.527	0.617	0.510	0.542	0.368	0.446	0.326	0.355
CKE	0.450	0.544	0.433	0.467	0.347	0.423	0.306	0.335
BPRMF	0.589	0.676	0.583	0.611	0.435	0.518	0.394	0.424

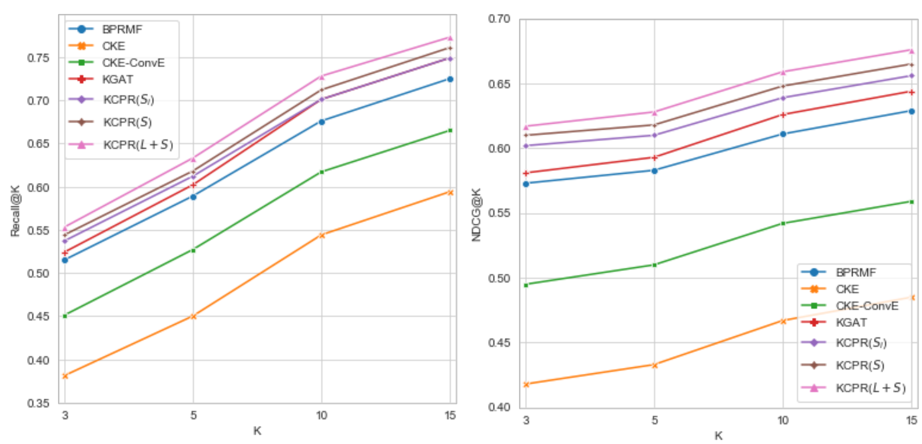

Fig. 3. Recommendation performances of ESOF dataset.

Our methods(KCPR(S), KPCR(S_i), KCPR($L+S$)) outperforms the KG-free method(BPRMF), while some other KG-enhanced methods(CKE, CKE-ConvE) showed lower performance than BPRMF. It suggests that CKE and CKE-ConvE could not fully utilize KG[13], while our method effectively utilized the knowledge graph as side information.

CKE-ConvE is higher than that of the original CKE. It can suggest that ConvE learns expressive features better than TransR. In addition, the number of parameters of CKE is 2.5M (ESOF) and 1.9M (XuetangX), while the number of parameters of CKE-ConvE is 1.3M (ESOF) and 1.4M (XuetangX). This also suggests that ConvE is parameter efficient.

CKE-ConvE showed lower performance compared to KCPR(S). The performance gap seems to be due to the difference in the type of KG used by the two models. Unlike CKE-ConvE, which uses KG including only item (course)-related information, KCPR(S) uses KG, including user-course interactions and user-related information.

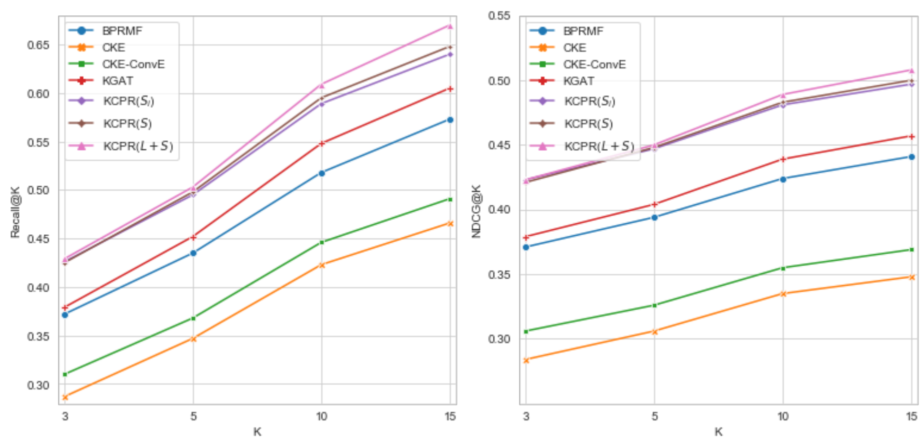

Fig. 4. Recommendation performances of XuetangX dataset.

KPCR(S) showed better performance than KGAT, which uses the same type of KG. There might be two reasons: First, KGAT uses the score function of TransR when calculating knowledge-aware attention. As mentioned earlier, TransR has limitations in learning expressive features. Second, there is a possibility that information loss occurred in the neighbor sampling process of KGAT[8].

In order to investigate whether external knowledge improves performance in educational recommendations, we compare the performance between KPCR(S_i) and KPCR(S). The former used the internal KG defined in section 3.1, and the latter used the unified KG defined in section 3.2. The result suggests that external knowledge was influential in the task of recommending courses.

Finally, we compare the performances of KPCR($L + S$) and KPCR(S). The result shows the effectiveness of using the level information in education regardless of the number of categories.

6 Experiment 2

ESOF allows authorized users to create their homepage and load courses from ESOF. The authorized users can monitor the learning progress of their homepage members. We created a homepage and provided a list of recommended courses using KPCR and KGAT to the homepage members, respectively. Afterward, user satisfaction for each list of recommended courses was investigated. Figure 5 shows the first page of the homepage we created. The thumbnails of the courses have been blurred due to the copyright issue.

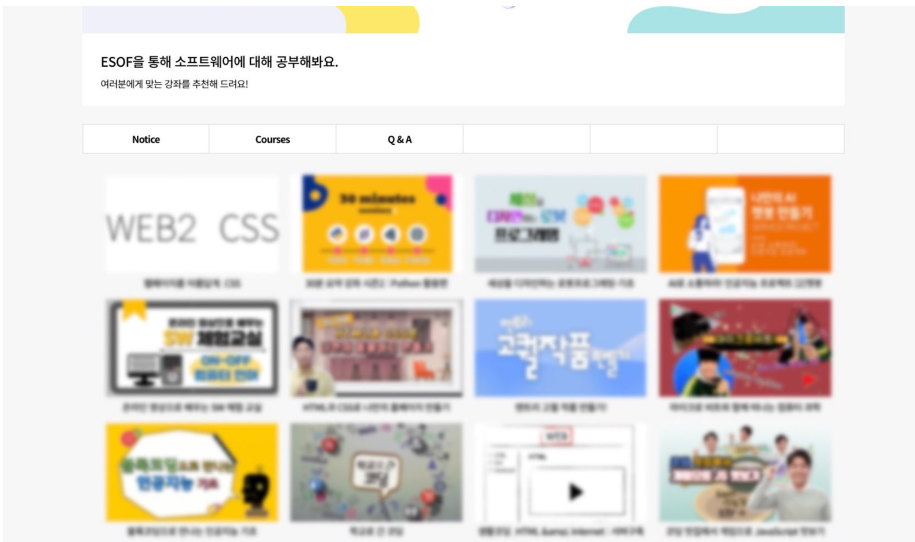

Fig. 5. The first page of the created homepage.

6.1 Participants

Participants in this study consisted of volunteers recruited through the ESOF platform. A total of 129 volunteers participated in the experiment, and the age distribution was N(10's)=64, N(20's)=42, N(30's)=21, N(more than 40's)=2. And among the total participants, 54 were female, and 75 were male.

6.2 Instruments

We investigated user satisfaction by measuring personalized services and system values. The questions were measured on a 5-point Likert scale ranging from (1) strongly disagree to (5) strongly agree. Data were statistically analyzed using SPSS 25.0, and the alpha level was set at 0.05. Separated independent samples t-tests were performed.

User Satisfaction with Personalized Service User satisfaction with personalized recommendation was measured by questions adapted from the customized service part of SERVQUAL[23]. The internal reliability of the instrument has Cronbach's alpha of 0.839. The questions are as follows:
Q1. whether the recommender system pays attention to the user needs,
Q2. whether the recommender system captures the user's interests, and
Q3. whether the system provides adaptive recommendations.

The Value of the Recommender System The value of the recommender system was adapted from the questions used in [16]. The internal reliability of the instrument has Cronbach's alpha of 0.888. The questions are as follows:
Q4. whether the recommender system is useful,
Q5. whether the recommender system finds interesting courses efficiently, and
Q6. the overall satisfaction of the recommender system.

6.3 Results

Satisfaction with personalized recommendation and value of the recommender system was analyzed using the average score of each related question (Q1-Q3 for personalized recommendation, Q4-Q6 for the value of the recommender system). Recommendations using KPCR obtained high average scores in both areas. As a result of analyzing the scores of the two models with independent samples t-tests, both areas showed statistically significant differences (personalized recommendation: p=.034, the value of recommender system: p=.002). That is, KPCR showed a statistically significant higher user satisfaction than KGAT. Table 4 demonstrates the detailed results of the independent samples t-test on user satisfaction.

Table 4. Results of independent samples t-tests on user satisfaction.

	Mean			
	KPCR	KGAT	t-value	significance
personalized recommendation	4.008	3.778	2.137	0.034*
value of recommender system	4.191	3.876	3.165	0.002**

* denotes p<0.05, ** denotes p<0.01.

7 Conclusion and Future Work

In this study, we proposed KPCR, a framework for Knowledge graph enhanced Personalized Course Recommendation. KPCR creates an integrated KG through keywords, and based on this, provides recommendations that also consider the level of learners. We demonstrate that the KG we build and the two KG-related auxiliary modules improved recommendation performance through experiments with real-world datasets and user satisfaction investigation.

For future works, we can explore various ways to explain the recommendation of our method. Also, we can study ways to utilize more diverse educational side information, such as the learning style of the users and the degree of social interaction between users in the course.

References

1. Alraimi, K.M., Zo, H., Ciganek, A.P.: Understanding the moocs continuance: The role of openness and reputation. Computers & Education **80**, 28–38 (2015)
2. Bollacker, K., Tufts, P., Pierce, T., Cook, R.: A platform for scalable, collaborative, structured information integration. In: Intl. Workshop on Information Integration on the Web (IIWeb'07). pp. 22–27 (2007)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS). pp. 1–9 (2013)
4. Cao, Y., Wang, X., He, X., Hu, Z., Chua, T.S.: Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In: The world wide web conference. pp. 151–161 (2019)
5. Chen, X., Jia, S., Xiang, Y.: A review: Knowledge reasoning over knowledge graph. Expert Systems with Applications **141**, 112948 (2020)
6. Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics **9**(5), 750 (2020)
7. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
8. Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., He, Q.: A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering (2020)
9. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1531–1540 (2018)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
11. Jdidou, Y., Khaldi, M.: Increasing the profitability of students in moocs using recommendation systems. International Journal of Knowledge Society Research (IJKSR) **7**(4), 75–85 (2016)
12. Jdidou, Y., Khaldi, M.: Using recommendation systems in mooc: An innovation in education that increases the profitability of students. In: Enhancing Knowledge Discovery and Innovation in the Digital Era, pp. 176–190. IGI Global (2018)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

15. Labarthe, H., Bachelet, R., Bouchet, F., Yacef, K.: Increasing mooc completion rates through social interactions: a recommendation system. Research Track p. 471 (2016)
16. Liang, T.P., Lai, H.J., Ku, Y.C.: Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. Journal of Management Information Systems **23**(3), 45–70 (2006)
17. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29 (2015)
18. Loyd, E., Prante, M., Messinger, L.: Evaluation of charlotte-mecklenburg schools personalized learning initiative—year 2 (2017)
19. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010)
20. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121 (2017)
21. Onah, D.F., Sinclair, J., Boyatt, R.: Dropout rates of massive open online courses: behavioural patterns. EDULEARN14 proceedings **1**, 5825–5834 (2014)
22. Pang, Y., Jin, Y., Zhang, Y., Zhu, T.: Collaborative filtering recommendation for mooc application. Computer Applications in Engineering Education **25**(1), 120–128 (2017)
23. Parasuraman, A., Zeithaml, V.A., Berry, L.: Servqual: A multiple-item scale for measuring consumer perceptions of service quality. 1988 **64**(1), 12–40 (1988)
24. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
25. Qiao, L., Yin, C., Chen, H., Sun, H., Rong, W., Xiong, Z.: Automated constmction of course knowledge graph based on china mooc platform. In: 2019 IEEE International Conference on Engineering, Technology and Education (TALE). pp. 1–7. IEEE (2019)
26. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
27. Rodriguez, M., Posse, C., Zhang, E.: Multiple objective optimization in recommender systems. In: Proceedings of the sixth ACM conference on Recommender systems. pp. 11–18 (2012)
28. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in artificial intelligence **2009** (2009)
29. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706 (2007)
30. Sun, Z., Guo, Q., Yang, J., Fang, H., Guo, G., Zhang, J., Burke, R.: Research commentary on recommendations with side information: A survey and research directions. Electronic Commerce Research and Applications **37**, 100879 (2019)
31. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning. pp. 2071–2080. PMLR (2016)
32. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 950–958 (2019)
33. Wang, X., He, X., Feng, F., Nie, L., Chua, T.S.: Tem: Tree-enhanced embedding model for explainable recommendation. In: Proceedings of the 2018 World Wide Web Conference. pp. 1543–1552 (2018)

34. Wang, X., He, X., Nie, L., Chua, T.S.: Item silk road: Recommending items from information domains to social users. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 185–194 (2017)
35. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 28 (2014)
36. Xiaoyan, Z., Jie, B.: Research on mooc system based on bipartite graph context collaborative filtering algorithm. In: Proceedings of the 2019 8th International Conference on Software and Computer Applications. pp. 154–158 (2019)
37. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
38. Yu, X., Ren, X., Sun, Y., Sturt, B., Khandelwal, U., Gu, Q., Norick, B., Han, J.: Recommendation in heterogeneous information networks with implicit user feedback. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 347–350 (2013)
39. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 353–362 (2016)
40. Zhang, H., Huang, T., Lv, Z., Liu, S., Yang, H.: Moocrc: A highly accurate resource recommendation model for use in mooc environments. Mobile Networks and Applications **24**(1), 34–46 (2019)
41. Zhang, H., Yang, H., Huang, T., Zhan, G.: Dbncf: Personalized courses recommendation system based on dbn in mooc environment. In: 2017 International Symposium on Educational Technology (ISET). pp. 106–108. IEEE (2017)
42. Zhang, J., Hao, B., Chen, B., Li, C., Chen, H., Sun, J.: Hierarchical reinforcement learning for course recommendation in moocs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 435–442 (2019)
43. Zheng, S., Rosson, M.B., Shih, P.C., Carroll, J.M.: Understanding student motivation, behaviors and perceptions in moocs. In: Proceedings of the 18th ACM conference on computer supported cooperative work & social computing. pp. 1882–1895 (2015)
44. Zheng, Y., Liu, R., Hou, J.: The construction of high educational knowledge graph based on mooc. In: 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). pp. 260–263. IEEE (2017)