KPCR: Knowledge Graph Enhanced Personalized Course Recommendation*

Heeseok Jung, Yeonju Jang, Seonghun Kim, and Hyeoncheol Kim(✉)

Korea University, Seongbuk-gu Seoul 02841, Republic of Korea
{poco2889,spring0425,ryankim0409,harrykim}@korea.ac.kr

Abstract. To handle the limitations of collaborative filtering-based recommender systems, knowledge graphs are getting attention as side information. However, there are several problems to apply the existing KG-based methods to the course recommendations of MOOCs. We propose **KPCR**, a framework for **K**nowledge graph enhanced **P**ersonalized **C**ourse **R**ecommendation. In KPCR, internal information of MOOCs and an 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. Through the experiments with the real-world datasets, we demonstrate that our knowledge graph boosts recommendation performance as side information. The results also show that the two auxiliary modules improve the recommendation performance. In addition, we evaluate the effectiveness of KPCR through the satisfaction survey of users of the real-world MOOCs platform.

Keywords: MOOCs · Personalized Learning · Recommender Systems.

1 Introduction

Despite the growing number of users learning through Massive Open Online Courses (MOOCs), there is a big challenge that students' retention rates are less than 10% on average [1]. One of the major factors that lower retention rates are curricula that do not reflect learners' interests [7], and the content that is too difficult to follow is another main factor [26]. Meanwhile, [11] stated that providing content appropriate to the learner's level increases retention rates.

For these reasons, studies have been conducted to recommend courses to users in MOOCs in various ways [22, 19], and many of them use collaborative filtering (CF). However, CF has limitations in that it has low performance in sparse data and has a cold start problem [24]. Utilizing side information has been evaluated as a good solution to solve these problems [18], and knowledge graphs (KG) are getting attention as side information [4]. Accordingly, studies have

* This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2018-0-01405) supervised by the IITP(Institute for Information Communications Technology Planning Evaluation).

been conducted to utilize KG for recommendation using the embedding-based [23] , the path-based [5], and the propagation-based method [8, 20] .

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 [4]. 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 courses 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.

Through the experiments with the real-world datasets, we demonstrate 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 investigated the user satisfaction of KPCR's recommendations for users of real world MOOCs platforms.

2 Preliminary

2.1 Internal Knowledge Graph

A Knowledge Graph (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 (h, r, t), which means that h has some relation r with t [21].

The internal knowledge graph $KG_{internal}$ is composed of three 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 that the user has taken. Course-related keywords can be obtained from course descriptions such as learning topics and table of contents. Examples of keywords are '*management*', '*artificial intelligence*', and '*social science*'. In this study, the entity pairs include '*User-Course*', '*User-Keyword*', and '*Course-Keyword*'. The relation types are '*enrolled_in*', '*interested_in*', and '*related_to*'.

2.2 External Knowledge Graph

The external knowledge graph $KG_{external}$ is an external knowledge base, which is associated with the keywords mentioned in 2.1. DBpedia [14], Freebase [2], and YAGO [17] 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.

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

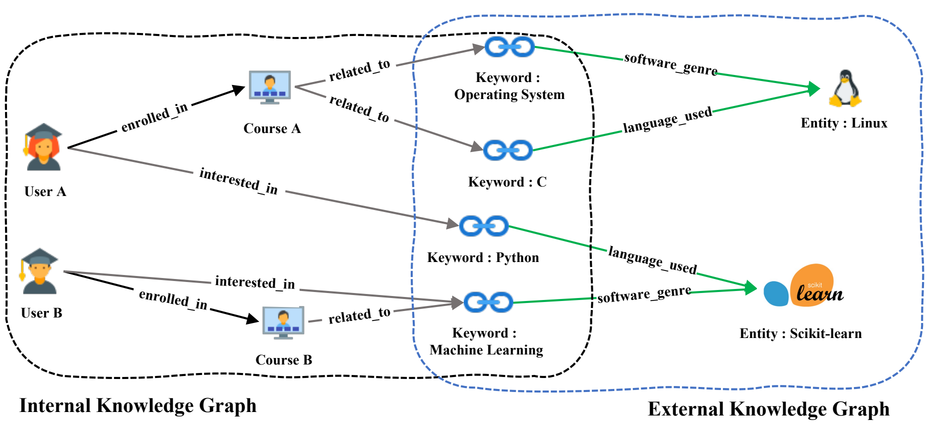

Fig. 1. Example of a unified knowledge graph.

2.3 User-Course Level Graph

To recommend appropriate courses to a user while considering the difficulty of 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*). 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 difficulty of the courses the user has taken so far.

2.4 Task Formulation

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.

3 Methodology

3.1 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 their connectivities. In general, KGE is learned by defining the scoring function of a triplet (h, r, t) [21].

In this study, we use ConvE [3], 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:

$$\phi\left(h, r, t\right) = f\left(vec\left(f\left(\left[\bar{e_h} \parallel \bar{r_r}\right] * \omega\right)\right) W\right) e_t \tag{1}$$

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; $\parallel$ denotes concatenation; $*$ denotes convolution operation; W denotes the weight matrix of the dense layer; f denotes ReLU function [12]; 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:

$$\mathcal{L}_{structural} = \sum_{(h,r,t) \in KG_{unified}} l(h, r, t) \tag{2}$$

$$l_{(h,r,t)} = -\frac{1}{N} \sum_{i}^{N} \left(\ y_{t_i} \cdot \ln\left(s_i\right) + \left(1 - y_{t_i}\right) \cdot \ln\left(1 - s_i\right)\right), s = \sigma(\phi\left(h, r, t\right)) \tag{3}$$

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

3.2 Level Embedding

Like [9], we use two-layer graph convolutional networks (GCNs) for node classification (level prediction) on the user-course level graph. The level prediction is as follows:

$$Z = f(X, A) = \mathrm{softmax}(\hat{A}\mathrm{ReLU}(\hat{A}XW^{(0)})W^{(1)}) \tag{4}$$

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:

$$\mathcal{L}_{level} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf} \tag{5}$$

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

3.3 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:

$$e_u^{CF} = e_u^{level} + e_u^{structural}, e_c^{CF} = e_c^{level} + e_c^{structural} \tag{6}$$

e^{level} denotes the first hidden layer activation of the user or the course in section 3.2. $e^{structural}$ denotes the representation of the user or the course from the entity embedding in section 3.1.

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.

$$p(u, c) = {e_u^{CF}}^{\top} e_c^{CF} \tag{7}$$

We adopt the Bayesian Personalized Ranking (BPR) [15] as the loss function of our recommendation module. BPR assumes that the user prefers the interacted item to the other non-interacted items:

$$\mathcal{L}_{CF} = \sum_{u,c,c'} -\ln \sigma(p(u, c) - p(u, c')) \tag{8}$$

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:

$$\mathcal{L}_{total} = \mathcal{L}_{CF} + \mathcal{L}_{level} + \mathcal{L}_{structural} \tag{9}$$

Specifically, 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.

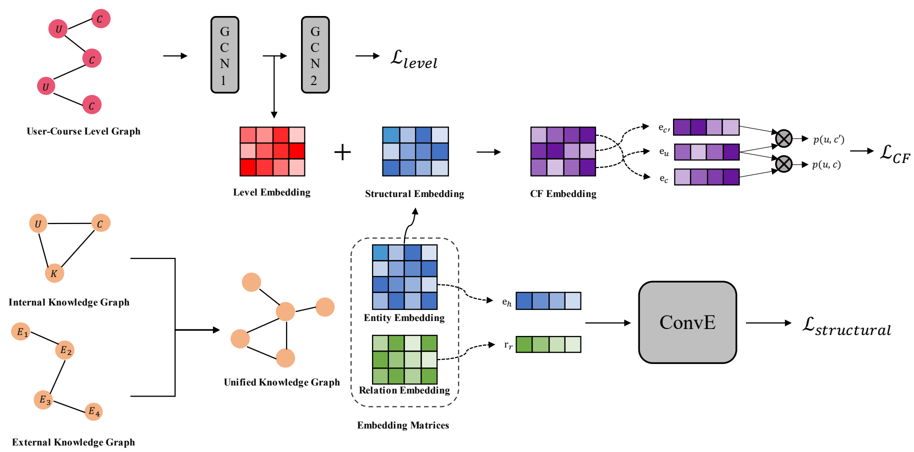

Fig. 2. Illustration of training process for our proposed KPCR.

4 Experiment 1

4.1 Datasets

We used two datasets, ESOF[1] and XuetangX[2], to demonstrate the effectiveness of KPCR. We collected a real-world MOOCs dataset from ESOF, which is a public MOOCs platform for software education in the Republic of Korea. To verify that our method is effective on another dataset, we compared the performance using the XuetangX dataset [25]. XuetangX is one of the biggest MOOCs platforms in China.

To ensure the quality of the data, we remove users and courses with less than three interactions for both datasets. For the ESOF dataset, we conduct 3-fold cross-validation to evaluate the performance. Regarding the XuetangX dataset, we sampled about 25% of all users and 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. Table 1 illustrates the detailed statistics of the datasets.

[1] https://www.ebssw.kr/
[2] https://www.xuetangx.com/

Table 1. Detailed statistics of the datasets.

		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

4.2 Knowledge Graph Construction

For the ESOF dataset, we extract the user-related keywords from the information entered by the user during the sign-up process. The course-related keywords are selected by the lecturers. Regarding the XuetangX dataset, since the last update of this dataset was in 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.

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

4.3 User-Course Level Graph Construction

For the ESOF dataset, the difficulty level of the course specified in the course description was used for the course level. The user level is set as the average level of the courses that the user has taken so far. In the case of the XuetangX dataset, we were not able to access some course descriptions. So 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 basic (level 1), and if it includes 'advanced', the course's level is set as advanced (level 3). If the title does not include difficulty information, the level of the course is set to intermediate (level 2). The user's level is set as the average level of the courses that the user has taken so far.

4.4 Experimental Settings

To investigate the effectiveness of recommendations, we opted to use Recall@K and NDCG@K [16] as evaluation metrics. 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].

For both datasets, we fix the embedding size of all models to 64. For our model, we use 16 convolution filters with (3×3) size for ConvE. 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 [6]. The learning rate is set as 0.001. We select Adam Optimizer [8] for all models, and use the early stopping technique with 10-epoch patience according to Recall@10.

We demonstrate the performance of KPCR comparing it with the baseline models below:

BPRMF [15] : BPR-based Matrix Factorization (BPRMF) is a KG-free CF method based on pairwise preference.

CKE [23] : 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 [20] : 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 that uses both the level embedding module and the structural module to assist the recommendation module.

KPCR(S) : A simple version of our proposed method that does not use the level embedding module.

KPCR(S_i) : Another version of KPCR(S) that uses internal KG only.

4.5 Results

Table 2 shows the overall performance comparison. The experimental results showed similar trends in both datasets.

Table 2. 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

Our methods (KPCR(S), KPCR(S_i), and KPCR($L + S$)) outperforms the KG-free method (BPRMF), while some other KG-enhanced methods (CKE,

CKE-ConvE) showed lower performance than BPRMF. This suggests that CKE and CKE-ConvE could not fully utilize KG [8], while our method effectively utilized the knowledge graph as side information.

CKE-ConvE is higher than that of the original CKE. This 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 a lower performance compared to KPCR(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, KPCR(S) uses KG including user-course interactions and user-related information as well.

KPCR(S) showed a better performance than KGAT, which uses the same type of KG. There might be two reasons for this: 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 [4].

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.

5 Experiment 2

ESOF allows authorized users to create their own homepage and load courses from ESOF. The authorized users can monitor the learning progress of the homepage members. We created a homepage and provided lists of recommended courses using KPCR and KGAT to the homepage members. Afterwards, we investigated user satisfaction for each list of recommended courses. Participants were recruited through the ESOF platform, and a total of 129 volunteers participated in the experiment. Fig.3 shows the first page of the homepage we created. The thumbnails of the courses have been blurred due to copyright issue.

5.1 Instruments

We investigated user satisfaction by measuring satisfaction with 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 sample t-tests were performed.

User satisfaction with personalized recommendations and the value of the recommender system were measured by questions adapted from the customized service part of SERVQUAL [13] and the questions used in [10]. The internal reliability of the instruments shows a Cronbach's alpha of 0.839 and 0.888 respectively.

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,
Q3. whether the system provides adaptive recommendations,
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.

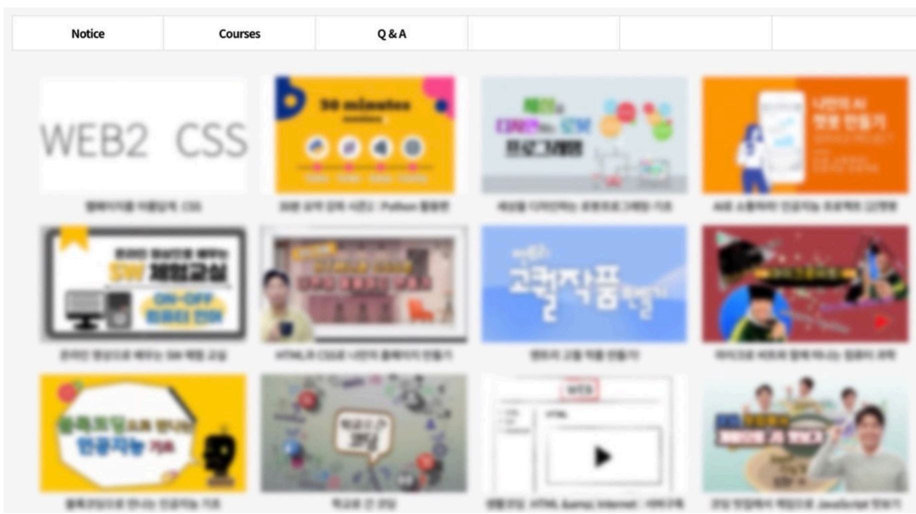

Fig. 3. The first page of the created homepage (thumbnails have been blurred due to copyright issues).

5.2 Results

Satisfaction with personalized recommendation and value of the recommender system were analyzed using the average score of each related questions (Q1-Q3 for satisfaction with personalized recommendation, and Q4-Q6 for value of the recommender system). Recommendations using KPCR obtained high average scores in both areas. As a result of independent sample t-tests, both areas showed statistically significant differences (personalized recommendation: p=0.034, value of recommender system: p=0.002). That is, KPCR showed a statistically significant higher user satisfaction than KGAT. Table 3 demonstrates the detailed results of independent sample t-tests on user satisfaction.

Table 3. Results of independent sample 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.

6 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 built and the two KG-related auxiliary modules improved recommendation performance through the experiments with the real-world datasets and the investigation of user satisfaction.

For future works, we can study ways to utilize more diverse educational side information, such as the learning style of the users and 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. 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)
4. 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)
5. 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)
6. 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)
7. 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)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. 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)
11. Loyd, E., Prante, M., Messinger, L.: Evaluation of charlotte-mecklenburg schools personalized learning initiative—year 2 (2017)
12. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010)
13. 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)
14. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
15. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
16. 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)
17. 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)
18. 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)
19. Symeonidis, P., Malakoudis, D.: Multi-modal matrix factorization with side information for recommending massive open online courses. Expert Systems with Applications **118**, 261–271 (2019)
20. 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)
21. 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)
22. 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)
23. 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)
24. 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)
25. 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)
26. 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)