Computers and Education: Artificial Intelligence 6 (2024) 100193

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence

ConceptGCN: Knowledge concept recommendation in MOOCs based on knowledge graph convolutional networks and SBERT

Rawaa Alatrash *, Mohamed Amine Chatti, Qurat Ul Ain, Yipeng Fang, Shoeb Joarder, Clara Siepmann

Social Computing Group, Faculty of Computer Science, University of Duisburg-Essen, Germany

ARTICLE INFO

Keywords:
Educational recommender systems
Concept recommendation
MOOCs
Knowledge graphs
Graph neural networks
Graph convolutional networks
Sentence encoders

ABSTRACT

Massive Open Online Courses (MOOCs) have gained popularity in the technology-enhanced learning (TEL) domain. To enhance the learning experience in MOOCs, educational recommender systems (ERSs) can play a crucial role by suggesting courses or learning materials that align with students' knowledge states. Thereby, understanding a student's learning needs and predicting knowledge concepts that the student might be interested in are important to provide effective recommendations. Inspired by the superior ability of knowledge graphs (KGs) in modeling the heterogeneous data in MOOCs and Graph Neural Networks (GNNs) in learning on graph-structured data, few works focusing on GNN-based recommendation of knowledge concepts in MOOCs have emerged recently. However, existing approaches in this domain have limitations mainly related to complexity, semantics, and transparency. To address these limitations, in this paper we propose ConceptGCN, an end-to-end framework that combines KGs, Graph Convolutional Networks (GCNs), and pre-trained transformer language model encoders (SBERT) to provide personalized and transparent recommendations of knowledge concepts in the MOOC platform *CourseMapper*. We conducted extensive offline experiments and an online user study (N = 31), demonstrating the benefits of the ConceptGCN-based recommendation approach, in terms of several important user-centric aspects including accuracy, novelty, diversity, usefulness, overall satisfaction, use intentions, and reading intention. In particular, our results indicate that, if SBERT is used for the initial embeddings of items in the KG, a self-connection operation and a semantic similarity-based score function in the aggregation operation of GCN are not necessarily needed.

1. Introduction

Massive Open Online Courses (MOOCs) have gained popularity in the technology-enhanced learning (TEL) domain as a flexible educational platform that provides more educational opportunities to a global audience (Yousef et al., 2014). However, MOOCs present a new challenge related to the need to guide learners through the growing educational content on MOOC platforms, which often does not meet students' learning needs or knowledge level (Zhao et al., 2021). Therefore, understanding a student's learning needs and predicting knowledge concepts that the student might be interested in are important (Piao, 2021). To address these challenges, educational recommender systems (ERSs) have been studied and developed to filter and personalize the educational content delivered to learners. In particular, there has been growing interest in ERSs on MOOC platforms with respect to different aspects such as course, video, and learning paths (Khalid et al., 2020). More recently, researchers argued that it is beneficial to focus on learner interests regarding specific knowledge concepts, which can capture user interests better and provide the flexibility of choosing learning resources of their interest (Gong et al., 2020, Piao, 2021, Gong et al., 2021). This line of research studied ERSs from a micro perspective and focused on recommending knowledge concepts. In this work, we also focus on the micro perspective for knowledge concept recommendations in MOOC platforms.

In general, the main objective of recommender systems (RS) is to anticipate whether a user will engage with an item from various user-

* Corresponding author.
E-mail addresses: rawaa.alatrash@stud.uni-due.de (R. Alatrash), mohamed.chatti@uni-due.de (M.A. Chatti), qurat.ain@stud.uni-due.de (Q.U. Ain), yipeng.fang@stud.uni-due.de (Y. Fang), shoeb.joarder@uni-duisburg-essen.de (S. Joarder), clara.siepmann@uni-due.de (C. Siepmann).
URL: https://www.uni-due.de/soco (R. Alatrash).

https://doi.org/10.1016/j.caeai.2023.100193
Received 11 July 2023; Received in revised form 7 December 2023; Accepted 9 December 2023
Available online 15 December 2023

item interaction possibilities. Thereby, the main challenge is to learn effective user and item representations from their interactions and side information (if available) (Wu et al., 2022). Considering that users, items, and preferences can be easily modeled as a graph, Graph Neural Networks (GNNs) have emerged as an effective way to encode collaborative information for recommendation tasks (He et al., 2020, Wang, He, Wang, et al., 2019, Berg et al., 2017, Ying et al., 2018). Owing to their interpretability and efficiency, Graph Convolutional Networks (GCNs) quickly become a prevalent formulation of GNNs and are being widely used for recommendation (He et al., 2020). The intuition behind GCN is to refine the user and item representations by aggregating embeddings of multi-hop neighbors in the graph. Modeling multi-hop connectivity from user-item interactions, and the captured signals from high-hop neighbors have been proven to be effective for recommendation (Wang et al., 2021, Wu et al., 2022, Gao et al., 2023). The graph structure and the design of the GNN architecture (e.g., aggregation and update operations, network depth) depends to a large extent on the type of information available in the graph. For example, user-item interactions can be considered either a bipartite graph or two homogeneous graphs (i.e., user-user and item-item graphs). And, a knowledge graph (KG) is inherently a heterogeneous graph with multi-type entities and relations, which requires considering such heterogeneity during propagation (Wu et al., 2022). Recognizing their superiority in graph representation learning, GNNs have been widely utilized in KG-based recommendation to model users and items. Given the user-item interaction information as well as the KG, KG-based recommendation seeks to take full advantage of the rich relational and semantic information in the KG to enhance the user and item representations. This method is based on the idea of embedding propagation, where the entity representation is refined by aggregating embeddings of multi-hop neighbors in the KG. Then, the user's preference can be predicted with the enriched representations of the user and the potential item (Guo et al., 2020). Recent efforts focused on KG-based recommendation of knowledge concepts in MOOCs. (Gong et al., 2020, Piao, 2021, Gong et al., 2021). While these approaches benefit from the advantages of GNNs in capturing high-order structure and semantic information in the KG, they suffer from several limitations. Firstly, existing approaches are complex. They rely heavily on manually designed meta-paths that carry the high-order information and feeding them into a predictive model, and thus they require domain knowledge and are rather labor-intensive. These issues can prohibit the performance and effectiveness of the recommendation model (Wang, Zhao, et al., 2019, Wang, He, Cao, et al., 2019). Additionally, they do not fully capture the semantic information in the KG. In particular, they do not consider the context of the knowledge concepts or the semantic similarity between them during the GNN aggregation and update operations. Further, existing approaches lack transparency in their recommendation process as they do not provide explanation of the generated recommendations, which is beneficial to enhance users' trust and improve the overall acceptance of an RS (Chatti et al., 2023, 2022, Guesmi et al., 2023).

To address these limitations, in this work we present ConceptGCN, a comprehensive end-to-end framework designed to recommend knowledge concepts to learners based on their interests and knowledge state. We combine KGs, GCNs, and pre-trained transformer sentence encoders (SBERT) to incorporate both structural and semantic information and enhance the representations of knowledge concepts. Specifically, we construct a KG for each learning material to capture structured information and relations between a set of entities in this material. Then, we employ propagation-based GCN to represent items by considering multiple hops of neighboring connections in the KG. Inspired by LightGCN (He et al., 2020), we simplify the GCN aggregation and update operations by removing feature transformation and nonlinear activation. Recently, sentence embedding techniques have gained more and more attention due to the good performance they have shown in a broad range of NLP-related scenarios. Sentence embeddings serve to capture the semantic meaning of the sentences or documents and contextual relationships between them, which can be effectively used to extract meaningful data representations, obtain a semantic and relational understanding of the data, and measure semantic similarities between sentences or documents (Hassan et al., 2019). To better capture the semantics of the entities and relations in the KG, we leverage SBERT (Reimers & Gurevych, 2019) as a valuable source of supplementary information to enhance the representations of concepts and distinguish the importance of relations between different concepts. Next, we utilize these enriched representations to build representations of learner models based on the concepts that the learner did not understand (referred to as DNU concepts) for a more personalized recommendation of related concepts to be mastered. Further, to increase transparency, we provide explanations of the recommended concepts using structural and semantic information in the KG. To this end, we introduce a novel weighting method based on different paths connecting learners and recommended concepts. This approach ensures that the recommendations are transparent, allowing learners to understand why specific concepts were suggested to them.

The paper aims to investigate the impact of a knowledge concept recommendation approach that combines KG, GCN, and SBERT on accuracy and students' perceptions of the benefits of this approach. The following research questions guide our investigation:

- RQ1: How to effectively construct a KG that can be used in a MOOC platform to provide personalized and explainable recommendation of knowledge concepts?
- RQ2: What is the potential impact of the proposed ConceptGCN-based recommendation approach on learners' perceptions of the ERS in terms of accuracy, novelty, diversity, usefulness, overall satisfaction, use intentions, and reading intention?

To answer these research questions, we conducted extensive offline experiments as well as an online user study ($N = 31$). Our results suggest that (1) integrating both structural and semantic information from the KG, harnessing SBERT as a valuable source of additional semantic information, and incorporating high-order connectivity through GCN proved to be beneficial in enhancing the representations of knowledge concepts and learner models that ultimately resulted in accurate recommendations, (2) in general, the students had a positive attitude towards the ConceptGCN-based approach with regards to several important user-centric aspects including perceived accuracy, novelty, diversity, usefulness, overall satisfaction, use intentions, and reading intention, and (3) if SBERT is used for the initial embeddings of items in the KG, a self-connection operation and a semantic similarity-based score function in the aggregation operation of GCN are not necessarily required.

To summarize, this work makes the following five main contributions: (1) We propose ConceptGCN, an intuitive end-to-end framework to recommend knowledge concepts to students, based on the concepts that they did not understand (i.e., DNU concepts); (2) We construct a KG to model the various relationships among different types of entities (i.e., learner, learning material, slide, main concept, related concept, category) in the MOOC platform; (3) We combine KG, GCN, and SBERT to derive enriched representations of KG items (i.e., slide, main concept, related concept, category) and learner models; (4) We harness the structural and semantic information in the KG to explain the provided recommendations; (5) We evaluate ConceptGCN in terms of several important user-centric evaluation metrics, and show the effectiveness of our proposed approach.

2. Background and related work

2.1. Graph neural networks for recommendation

Recommender systems (RS) are extensively used to tackle information overload on the web by providing personalized information filtering. The primary objective of an RS is to predict user interactions with items. Collaborative filtering (CF), model-based CF methods (e.g., matrix factorization (Koren et al., 2009)), and neural network-based models (e.g., neural collaborative filtering (He et al., 2017) are key RS approaches that utilize past user-item interactions to make predictions and provide recommendations. However, these methods are limited, as their paradigms of prediction and training ignore the high-order structural information in observed data (Gao et al., 2023). With their advantages in handling the structural data and exploring structural information, Graph Neural Networks (GNNs) have provided an opportunity to address the above issues and have become the new state-of-the-art approaches in RS (Wang et al., 2021, Wu et al., 2022, Gao et al., 2023). GNNs shed light on modeling graph structure, especially high-hop neighbors, to guide the embedding learning (Hamilton et al., 2017, Kipf & Welling, 2016, Veličković et al., 2017). Generally speaking, GNNs are based on the idea of embedding propagation to refine the entity representation by aggregating embeddings of multi-hop neighbors in the graph. By stacking the propagation layers, each node can access high-order neighbors' information, rather than only the first-order neighbors' information as the traditional methods do (Gao et al., 2023).

Different GNN-based approaches have been proposed for CF-based recommendation. In addition to GAT (Veličković et al., 2017) and GraphSAGE (Hamilton et al., 2017), Graph Convolutional Networks (GCNs) represent a popular GNN technique that is increasingly used in the literature on recommender systems. GCNs are a type of Convolutional Neural Networks (CNNs) designed specifically for graphs (Kipf & Welling, 2016). GCNs utilize a sequence of layers that consist of learned filters, similar to CNNs, but adapted for graph structures. These filters are followed by a non-linear activation function, allowing the GCN to effectively learn and represent the graph's features (Wu et al., 2019). Motivated by the strength of GCN, recent efforts adapted GCN to the user-item interaction graph, capturing CF signals in high-hop neighbors for recommendation (He et al., 2020). For example, Wang, He, Wang, et al. (2019) proposed the recommendation framework Neural Graph Collaborative Filtering (NGCF), which exploits the user-item graph structure by propagating embeddings to learn the representation of users and items. NGCF combines the entity embeddings of both the neighbors and the entity itself in each layer to obtain the final entity representation. It incorporates feature transformation and nonlinear activation extensively. NGCF has shown to outperform several methods including the other GCN-based models GC-MC (Berg et al., 2017) and PinSage (Ying et al., 2018). More recently, He et al. (2020) introduced a simplified GCN-based recommender model named LightGCN, including only the most essential component in GCN for collaborative filtering, namely neighborhood aggregation. LightGCN learns user and item embeddings by linearly propagating them on the user-item interaction graph, and uses the weighted sum of the embeddings learned at all layers as the final embedding. The model prediction is defined as the inner product of user and item final embeddings, which is used as the ranking score for recommendation generation. The authors of LightGCN demonstrated that ignoring feature transformation and non-linear activation does not negatively impact the system's performance. Instead, it reduces the complexity of model training while still providing significant improvements (He et al., 2020). However, the limitation of LightGCN is that it typically focuses on user-item interactions and does not extract or utilize side features (side information) associated with users and items. Side features are additional attributes or characteristics associated with users and items in an RS. For example, side features for learning materials might include knowledge concept information. Neglecting side features can lead to less accurate user/item embeddings because it overlooks valuable information that can significantly enhance recommendation quality. In our work, we address these limitations by following a knowledge graph-based recommendation approach that aims at enhancing the entity representation by leveraging semantic relations among entities in the knowledge graph.

2.2. Knowledge graph-based recommendation

Heterogeneous Information Networks (HINs) are directed graphs that are able to capture and depict a wide range of entities and relationships found in the data (Yu et al., 2014). Knowledge graphs (KGs) represent a popular instance of HINs. A KG can contain multiple types of entities and relations in the graph. KGs are widely employed to represent large-scale information from multiple domains using the ability of KGs to capture structured information and relations between a set of entities as well as to enhance the user and item representation. As such KGs represent an attractive source of information that could help improve recommendation. KGs are thus increasingly being incorporated into RSs (Chicaiza & Valdiviezo-Diaz, 2021, Guo et al., 2020). KG-based recommendation can bring several benefits. First, the rich semantic relations between items through attributes can help explore high-level relations of entities (Wang et al., 2018). Second, expressing relationships between items through attributes can be leveraged to enhance the item representation (Wu et al., 2022). Third, the connections between a user's historically interacted items and recommended items can support the interpretability of the recommendation results (Yang & Dong, 2020). Despite the above benefits, utilizing KGs in recommendation is rather challenging due to its complex graph structure, i.e., multi-type entities and multi-type relations (Wu et al., 2022).

Mainly three different methods are popular in KG-based recommendation: Embedding-based methods, connection-based methods, and propagation-based methods (Guo et al., 2020). Embedding-based methods rely on KG embedding (KGE) methods to represent entities and their relationships (e.g., (Wang, Zhang, Zhao, et al., 2019, Zhang et al., 2016)). However, the limitation of KGE methods is that they are more suitable for the tasks related to graph, such as graph completion and link prediction rather than recommendation (Wang, Zhao, et al., 2019, Wang, Zhang, Zhang, et al., 2019). Connection-based methods leverage the available connection patterns by mining the relationships between the entities within a graph to drive the recommendation process. These methods mainly utilize the meta-structure (meta-path or meta-graph) of the graph to compute similarities between user and item embeddings. However, these methods face the challenge of designing appropriate meta-paths, which require domain knowledge and are rather labor-intensive for complicated KGs (Wang, Zhao, et al., 2019, Wang, He, Cao, et al., 2019). Propagation-based methods employ GNN architectures to incorporate both the user-item connection patterns and the relationships between them. This approach represents entities by considering their embeddings in relation to the multi-hop neighbors within the structure of the KG. By leveraging the GNN framework, propagation-based methods capture the structural information present in the KG to enhance the recommendation process. This enables a more comprehensive understanding of user-item interactions and can lead to improved recommendations (Guo et al., 2020). Wang, Zhao, et al. (2019) introduced a framework called Knowledge GCN (KGCN) for RSs. KGCN aims to capture both high-order structure and semantic information in a KG along with the user's interests. It utilizes neighborhood aggregation and bias to calculate the representation of entities of multiple hops in the KG. This approach allows KGCN to capture local proximity structure and personalized interests of users in relations. Furthermore, representing the neighborhood of each entity can be extended hierarchically to model high-order dependencies and long-distance interests. Wang et al. (2018) proposed an end-to-end framework named RippleNet that addresses the limitations of existing embedding-based and path-based methods by employing preference propagation, which extends a user's interests iteratively along KG links. By combining multiple "ripples" created by a user's historical interactions, RippleNet predicts the user's preference distribution for candidate items. Experiments demonstrate that RippleNet achieves substantial improvements in movie, book, and news recommendations compared to state-of-the-art methods. However, the size of the ripple set may go unpredictably with the increase of the size of KG, which would lead to heavy computation and storage overhead (Wang, Zhao, et al., 2019). In addition, RippleNet does not take into consideration the importance of relationships between different items in the interaction history. This could limit the ability of RippleNet to effectively capture the user's evolving preferences when encountering various candidate items. This issue has been addressed by Li et al. (2023), who introduced RKGCN, an end-to-end deep learning model that utilizes KGs to enhance user-item representations and make personalized recommendations. However, challenges remain in handling large datasets and mitigating noise interference in KGs, suggesting the need for scalable GNNs and improved sampling methods. Wang, He, Cao, et al. (2019) emphasized the importance of high-order relations in achieving successful recommendations. To address this, the authors proposed Knowledge Graph Attention Network (KGAT) that explicitly models high-order connectivities within KGs. KGAT recursively propagates embeddings from neighboring nodes (users, items, or attributes) to refine node representations and employs an attention mechanism to determine the importance of each neighbor. This approach overcomes the limitations of existing KG-based recommendation methods that either rely on path extraction or implicit regularization for high-order relation modeling. The authors also highlighted the advantages of KGAT compared to path-based and regularization-based methods. The KGAT model consists of an embedding layer, attentive embedding propagation layers, and a prediction layer, which collectively capture and utilize high-order relations for improved recommendations. Despite the mention of linear time complexity in the paper, dealing with massive graphs can still present challenges. The additional complexity hinders the model's ability to achieve improved performance. Wang et al. (2022) proposed a method based on GCN and multi-task called Light Knowledge GCN (LKGCN) that explicitly models the high-order connections between users, items, and entities in a recommendation system. The LKGCN model extends the existing LightGCN model by applying GCN to the tripartite graph consisting of user, item, and entity relations. This allows the model to incorporate information from the k-hop neighborhood of users, items, and entities into their embeddings. The inclusion of entity relations enables connections between users and items that would not be present otherwise. Additionally, an attention mechanism is utilized to strengthen the relationship between users, items, and entities and generate attention scores that can aid in explanation generation.

In summary, propagation-based methods provide a powerful mechanism for KG-based recommendation that effectively combines GNN techniques and KG relations to aggregate embeddings of multi-hop neighbors in the KG. Most of the existing works apply the variants of the traditional GAT (Veličković et al., 2017) over the KG, i.e., the central node is updated by the weighted average of the linked entities, and the weights are assigned according to a score function. For example, KGAT assigns the weight according to the distance between the linked entities in the relation space, such that the closer entities would pass more information to the central node (Wang, He, Cao, et al., 2019). KGCN adopts as a score function the dot product of the user embedding and the relation embedding, such that the entities whose relations are more consistent with users' interests will spread more information to the central node (Wang, Zhao, et al., 2019). Designing a reasonable and effective score function is, however, a complex and challenging task. In particular, the score functions proposed in the literature on KG-bases RS require more computation time and their performance depends on the construction of the KG (Wu et al., 2022, Guo et al., 2020). Our work distinguishes itself from the above propagation-based methods in several key ways. In addition to leveraging both high-order structure and semantic information in the KG, we also harness the semantic information captured by pre-trained transformer language model encoders (SBERT) to enhance the user and item representations. Moreover, to determine the importance of each neighbor, we use SBERT as the basis for a simple score function that assigns weights between linked entities in the KG during the propagation process. Furthermore, we utilize the structural and semantic information in the KG to explain the recommendations to help learners understand why specific concepts were suggested to them.

2.3. Concept recommendation in MOOCs

One challenge in MOOCs is to tailor the recommendations of courses and learning materials to the learner's knowledge state and specific interests. To address this challenge, researchers have shown interest in recommending knowledge concepts. This approach considers learners' individual learning needs at a micro level and emphasizes the significance of understanding their learning requirements and accurately predicting knowledge concepts that align with their interests. Only few approaches have been proposed for recommending knowledge concepts based on KG and GCN. Gong et al. (2020) presented one of the first works for recommending knowledge concepts in MOOCs in a heterogeneous view. The authors treated the number of clicks as ratings and formulated the problem as rating prediction for recommending top-k unknown concepts with higher ratings. They proposed ACKRec, an end-to-end GNN-based approach that leverages both content and context information to learn the representation of entities using GCN. ACKRec constructs a HIN that captures the semantic relationships among different types of entities (e.g., students, knowledge concepts, courses, videos, teachers). Meta-paths on the HIN are employed to guide the propagation of learners' preferences and capture their preference distribution for candidate knowledge concepts. GCNs are used to aggregate nodes on meta-paths to obtain representations of users and concepts. Additionally, an attention mechanism is introduced to adaptively combine context information from different meta-paths, catering to the diverse interests of students. Parameters of the model are learned through extended matrix factorization to predict potential user preferences for concepts in the course. Piao (2021) also addressed the challenge of recommending knowledge concepts to learners in MOOCs, considering the sparsity of learner-concept interactions given a large number of concepts. The authors considered the task of predicting and recommending concepts that a user might be interested in based on their learning history, which includes a set of learned concepts and their contextual information such as courses, videos, etc. Similar to the work in (Gong et al., 2020), Piao (2021) proposed MOOCIR that utilizes a HIN to model information on MOOCs and learn user and concept representations using GCNs based on user-user and concept-concept relationships via meta-paths in the HIN. These representations are then integrated into an extended matrix factorization framework to predict and recommend concept preferences for each user. The study explores different attention mechanisms to derive aggregated user and concept representations, highlighting their importance in achieving better performance. Building upon the work in (Gong et al., 2020), Gong et al. (2021) presented AGMKRec, a reinforced concept recommendation model that addresses the limitations of existing models by leveraging HINs and reinforcement learning. The model effectively incorporates auxiliary information and considers long-term learner interests in concept recommendation tasks within MOOCs. To construct the HIN, the authors utilize a meta-path-based method that automatically identifies relevant meta-paths and multi-hop connections among learners, courses, and concepts. Additionally, the reinforcement learning framework is employed to tackle challenges such as the sparsity of concept clicking rates in MOOCs and the sequential nature of interactions between learners and the recommender agent.

In summary, to obtain embeddings of users and concepts, the existing approaches leverage HINs to model information in MOOCs, construct different meta-paths for users and concepts, and use GCN to aggregate nodes on meta-paths and attention techniques as path weights. Finally, the learned user and concept embeddings are used for predicting the preference scores of concepts for recommendations. The main problem with these approaches is that they rely heavily on meta-path-based representations of users and concepts. Identifying useful meta-paths is a challenging task and has many shortcomings. First, it is hard

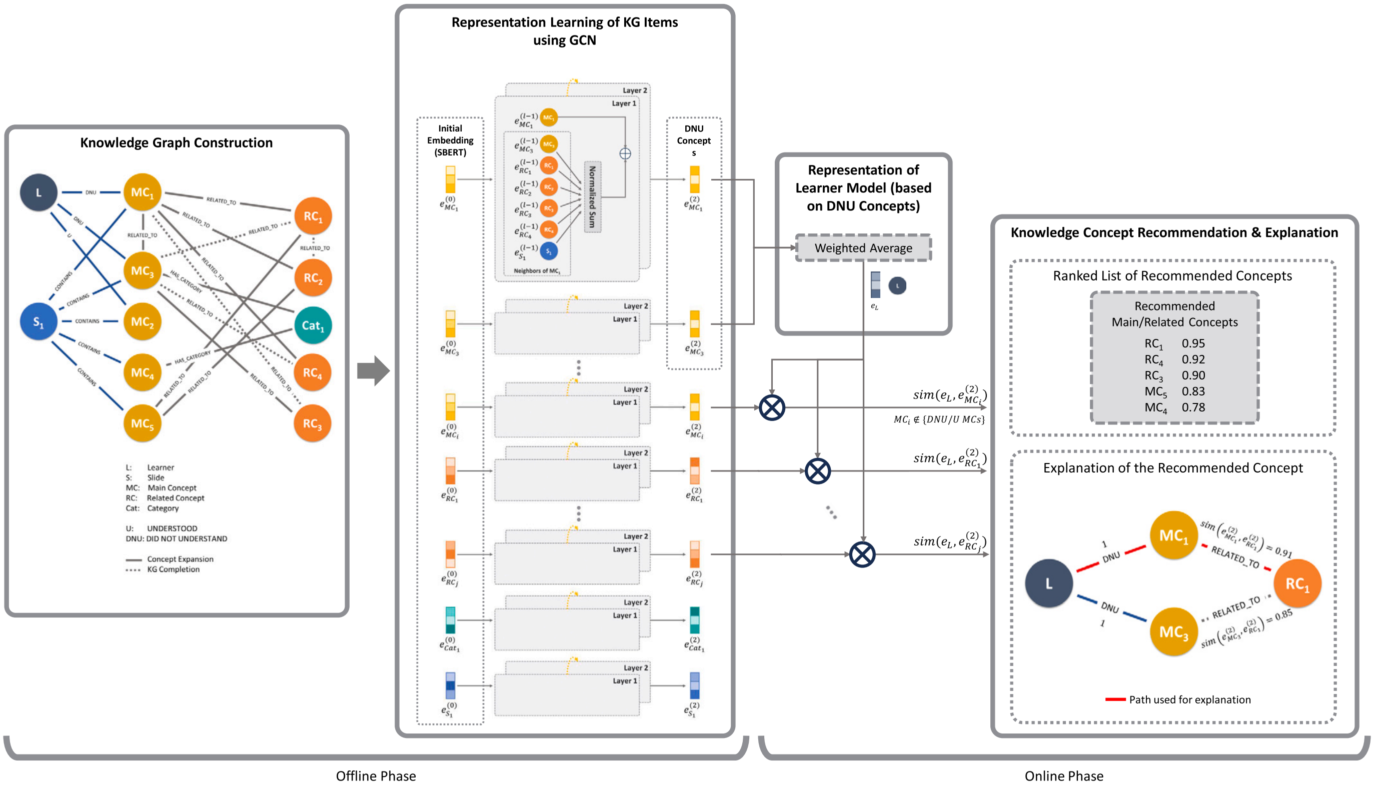


Fig. 1. The conceptual architecture of our proposed ConceptGCN-based knowledge concept recommender system.

to define optimal meta-paths in reality (Wang, Zhao, et al., 2019). Meta-paths are usually diverse for different application scenarios and cannot generalize to new datasets (Guo et al., 2020). Also, it is unavoidable to lose information by decomposing the sophisticated user-item connection pattern into separate linear paths (Guo et al., 2020). Moreover, the potential meta-paths induced from the HIN can be infinite and not all of them are relevant for the recommendation task (Gong et al., 2020). Furthermore, building effective meta-paths manually needs to have specific domain knowledge or experience, and the automatic selection of valid meta-paths in a HIN is relatively complex and more computations are required in enumerating and selecting paths (Guo et al., 2020, Gong et al., 2021, Wang, Zhao, et al., 2019). Finally, the recommended results will be significantly different for different meta-path combinations (Gong et al., 2020). Our work differs from the existing approaches in several aspects. First, we do not consider user-user and concept-concept relationships via different meta-paths. Second, we combine KG, GCN, and SBERT to incorporate both structural and semantic information to enrich the representations of knowledge concepts and specify the edge weights between concepts in the KG. Third, we build representations of learner models based on the concepts that the learner did not understand (referred to as DNU concepts) for a more personalized recommendation. Forth, we introduce a weighting method based on different paths linking learners and recommended concepts to explain the generated recommendations.

3. Proposed approach

In this section, we introduce the details of our proposed ConceptGCN approach that integrates KG, GCN, and SBERT to enhance the representations of knowledge concepts and learner models, performs knowledge concept recommendation based on the learned representations, and explains the provided recommendations. The conceptual architecture of our ConceptGCN-based knowledge concept RS is shown in Fig. 1. It consists of an offline and an online phase. The aim of the offline phase is to construct the KG and enhance the representation of KG items (i.e., slide, main concept, related concept, category) using GCN. The aim of the online phase is to represent a learner model based on the enhanced representations of a learner's DNU concepts and use the learner model representation to recommend knowledge concepts and explain the provided recommendations. In the following, we describe each phase along with its components in detail.

3.1. *Offline phase*

The offline phase consists of two main components: (1) *Knowledge Graph Construction* and (2) *Representation Learning of KG Items using GCN*, as shown in Fig. 1.

3.1.1. *Knowledge graph construction*

After a learning material is uploaded to *CourseMapper* (Ain et al., 2022), the process of generating the KG for this learning material begins. Initially, the KG is built for each slide (referred to as Slide-KG), gradually forming a KG for the learning material (referred to as LM-KG). Subsequently, concept expansion takes place, where additional concepts and categories related to the main concepts are added to enhance the KG. Next, KG completion is performed to establish connections between the different entities in the KG.

3.1.1.1. KG entities and relationships The constructed LM-KG consists of different nodes representing the following entities: Learning Material (LM), Slide (S), Main Concept (MC), Related Concept (RC), Category (Cat) and edges representing the following relationships: (LM, *CONSISTS_OF*, Slide), (Slide, *CONTAINS*, MC), (MC, *RELATED_TO*, RC), (MC, *HAS_CATEGORY*, Cat), (Learner, *HAS_READ*, Slide), (Learner, *DNU*, MC), (Learner, *U*, MC), as depicted in Fig. 2. Each main concept (MC) node has three states relative to the learner: "Did not understand", "Understood", and "New", where "New" represents the initial state of the MC, i.e., the learner has not interacted with the main concept. When interacting with the slides of a learning material in *CourseMapper*, the

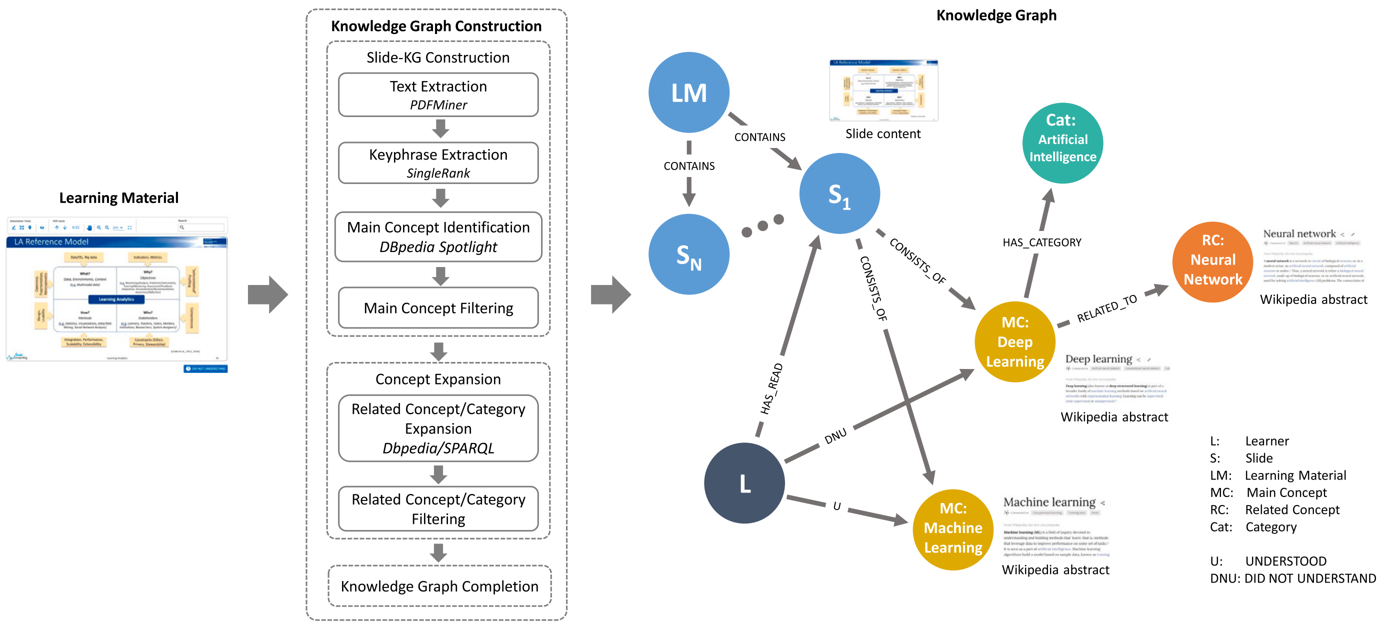


Fig. 2. Knowledge graph construction.

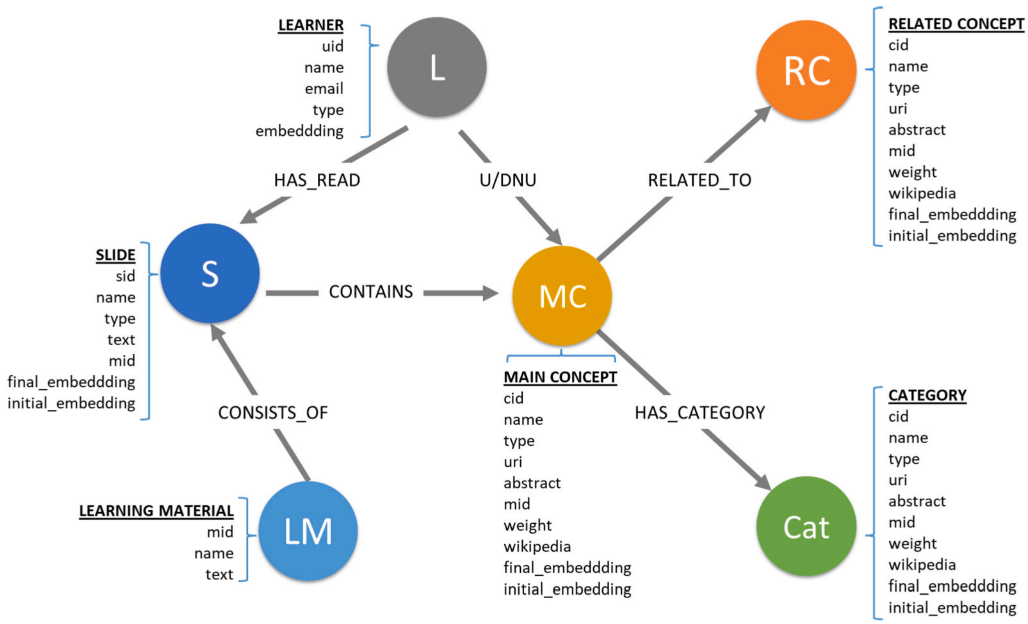


Fig. 3. Knowledge graph illustration in the graph database.

learner can mark the main concepts extracted from each slide as “Understood”, “Did not understand”, or “New”. In this way, the learner model is constructed using the concepts marked as “Did not understand” (i.e., DNU concepts). To store the KG, we use a graph database (Neo4j) which is highly suitable for this purpose as it uses graph structures with nodes and edges to represent and store data. Fig. 3 illustrates the KG entities, relationships, and attributes stored for each KG in the graph database. The process of constructing an LM-KG undergoes three main steps: (1) *Slide-KG Construction*, (2) *Concept Expansion*, and (3) *Knowledge Graph Completion*.

3.1.1.2. Slide-KG construction Each learning material uploaded to *CourseMapper* consists of several slides. Firstly, we create a KG for each slide (Slide-KG). The Slide-KG construction process encompasses four main steps: (1) *Text Extraction*, (2) *Keyphrase Extraction*, (3) *Concept Identification*, and (4) *Concept Filtering*, as shown in Fig. 2. The first step involves extracting the text from the slide using PDFMiner (Shinyama, 2013). Once the text is extracted, the next step is to apply the SingleRank algorithm (Wan & Xiao, 2008) to extract the top-15 keyphrases from the text. These keyphrases serve as candidates for the identification of the main concepts discussed in the slide. To identify the main concepts from these keyphrases, DBpedia Spotlight (Mendes et al., 2011) is utilized as an entity linking service to link the keyphrases to specific entities in the DBpedia knowledge base. This process results in the identification of the main concepts of each slide. Next, the main concepts extracted from the slide are filtered and sorted based on their importance to both the slide and the learning material. This filtering process takes into account the significance of each main concept to the slide’s content as well as its relevance to the learning material. To this end, we employ SBERT (Reimers & Gurevych, 2019) to generate representations of each entity in the KG. SBERT produces embeddings for the learning material (based on its text content), the slide (based on its text content), and the main concepts (based on the abstract of their Wikipedia article). These embeddings capture the semantic meaning and relationships between entities. Using these embeddings, cosine similarity scores are calculated between the main concepts and the slide, as well as between the main concepts and the learning material. These similarity scores quantify the relatedness of the main concepts with the slide and with the learning material. The two similarity scores are then combined by applying summation to derive an overall score for each main concept representing its relatedness with the slide and the learning material. Based on this combined score, the main concepts of the slide are sorted to reflect their relative importance. In this way, top 5 main concepts for each slide are obtained. This process ensures that the most important main concepts (top-5), both in terms of their relevance to the slide and the learning material, are given priority and emphasized. This step is repeated to build the Slide-KG for each slide included in the learning material. Finally, the Slide-KGs for all slides in the learning material are combined to form the LM-KG. At this stage, the KG consists of the following node types: Learning Material (LM), Slide (S), and Main Concept (MC).

3.1.1.3. Concept expansion To enrich the LM-KG, the main concepts are expanded by getting their related concepts and categories using DBpedia Spotlight. These expanded concepts would enhance the structure of the KG and facilitate the learner's concept discovery to help learners further master their knowledge. Expanding the main concepts would also improve the effectiveness of the recommendation process by recommending relevant, novel, and diverse related knowledge concepts to the learner. Concept expansion is performed by querying DBpedia using SPARQL. This query retrieves all the related concepts and categories associated with each main concept. However, within the DBpedia knowledge base, every concept is associated with a large number of related concepts and categories, and as we delve further, these associated concepts tend to become more abstract. Additionally, if all these associated concepts and categories are incorporated into the KG without undergoing a filtering procedure, the KG would extend to an unmanageable size. Once the related concepts and categories are obtained, their embeddings are generated using SBERT. The related concepts are embedded based on the abstract of their associated Wikipedia articles, while the categories are embedded using their names. These embeddings are used to compute the cosine similarities between each main concept and its related concept/category to quantify the relatedness between them. Furthermore, we compute the cosine similarities between each related concept/category and the learning material. These two similarity scores are combined by applying summation to obtain an overall score for each related concept/category. To decide which related concepts and categories should be kept in KG, this overall similarity score is used to sort the most relevant related concepts/categories. In this way, the top-20 related concepts and the top-3 categories for each main concept, are added to the KG. By following this approach, a richer structure of KG is achieved through the expansion and exploration of related concepts and categories.

3.1.1.4. Knowledge graph completion This step completes the missing relations between entities (main concepts, related concepts, and categories) in the KG, which results of a refinement of the KG by taking into consideration bidirectional and transitive relationships in the KG. Expanding main concepts with related concepts and categories might result in missing some relationships between existing entities in the KG, for example, as shown in Fig. 4 where dashed lines indicate missing relationships after the concept expansion step. When conducting concept expansion on MC_1, a "RELATED_TO" relationship to RC_1 is added to the KG because RC_1 is among the top-20 most closely related concepts to MC_1. Since MC_3 is not among the most related concepts to MC_1, there is no connection from MC_1 to MC_3. However, in the other direction, there is a connection established from MC_3 to MC_1 after concept expansion on MC3, as MC_1 is among the top-20 most related concepts to MC_3. During the concept expansion on MC_3, RC_1 is not included in the KG as it is not among the most closely related concepts to MC_3. In order to ensure that the "RELATED_TO" relationship is bidirectional, it is necessary to establish a connection from MC_1 to MC_3 and from RC_1 to MC_1. Moreover, in order to ensure that the "RELATED_TO" relationship is transitive, it is essential to incorporate the bidirectional relationship between MC_3 and RC_1, as MC_3 is connected to MC_1 and MC_1 is connected to RC_1.

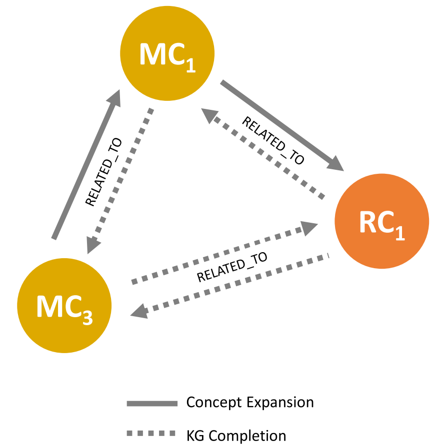


Fig. 4. Example of KG completion.

3.1.2. Representation learning of KG items using GCN

After constructing the LM-KG, the next step is to harness the structural and semantic information in the KG to enhance the representation of KG items (i.e., slide, main concept, related concept, category) using GCN. This is achieved by following three core steps: (1) Construct initial embedding matrix, (2) Construct the adjacency matrix, and (3) Construct final embedding matrix, as depicted in Fig. 5.

3.1.2.1. Construct initial embedding matrix The initial embedding matrix is composed of the initial embeddings of the KG items (i.e., slide, main concept, related concept, category), which is derived from the embedding of the item's textual content using SBERT. Specifically, for nodes categorized as slide (s), the content of the slide serves as the representation of the slide node. Whereas, for nodes classified as main concept (mc) or related concept (rc), the abstract provided in their linked Wikipedia articles is employed to represent them. Furthermore, for nodes of type category (cat), the category name itself is utilized to represent the category node, as expressed in the following equations where s is the set of slide nodes in the KG and n is the number of slides belonging to the learning material.

$$s = \{s_1, s_2, s_3, ..., s_n\} \tag{1}$$

The embedding e_{s_n} of the slide node s_n is generated by applying SBERT on the slide content:

$$e_{s_n} = SBERT\,\{slide\ content_n\} \tag{2}$$

Then the set of main concepts mc identified from s_n is presented as follows, where i is the i^{th} main concept in mc:

$$mc = \{mc_1, mc_2, mc_3, ..., mc_i\} \tag{3}$$

The embedding of the main concept mc_i based on the abstract available in its Wikipedia article is generated using SBERT:

$$e_{mc_i} = SBERT\,\{Wikipedia\ abstract_i\} \tag{4}$$

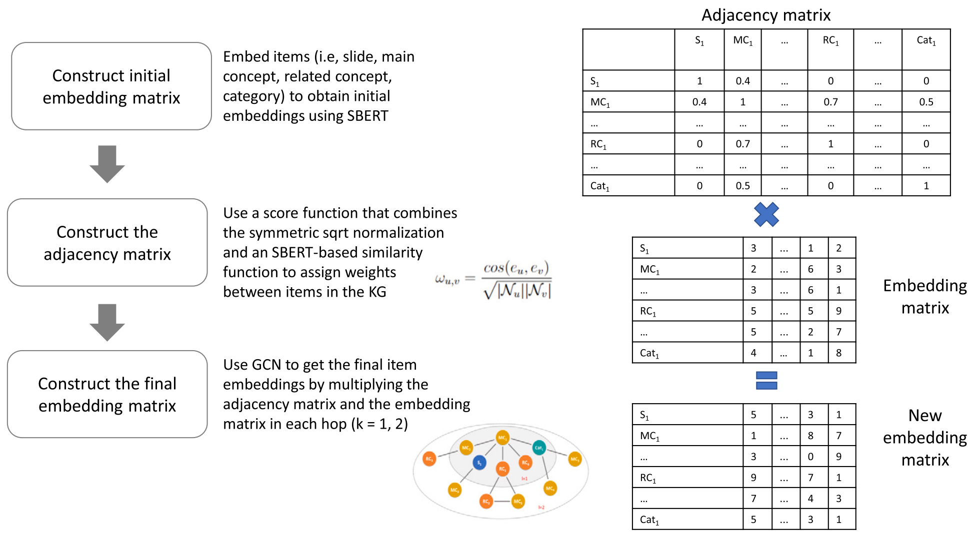


Fig. 5. Enhanced representation of KG items using GCN.

After that, each mc_i is expanded to get the list of its related concepts (rc_j) and categories (cat_k) as follows:

$$rc = \{rc_1, rc_2, rc_3, ..., rc_j\} \tag{5}$$

$$cat = \{cat_1, cat_2, ..., cat_k\} \tag{6}$$

where j is the number of related concepts in the expanded list of mc_i, and k is the number of categories that the mc_i has. The embedding of the related concept rc_j is generated in the same manner:

$$e_{rc_j} = SBERT\{Wikipedia\ abstract_j\} \tag{7}$$

lastly, the embedding of the category cat_k is generated using its name:

$$e_{cat_k} = SBERT\{category\ name_k\} \tag{8}$$

3.1.2.2. Construct the adjacency matrix In general, the construction of an adjacency matrix is based on the binary polarity relationship between nodes. A relationship value of 1 signifies a direct connection, indicating adjacency between the nodes. Conversely, a relationship value of 0 indicates the absence of a direct connection, implying non-adjacency. This approach solely considers the presence or absence of a direct relationship between nodes, disregarding the degree of influence of neighboring nodes. However, it is essential to account for the varying degrees of influence that different neighbors exert on a node. In our work, we incorporate the degree of influence between KG items through relationship weights that we compute based on the cosine similarity between the item embeddings, where higher similarity indicates stronger influence from the neighbor. Conversely, lower similarity indicates weaker neighbor influence.

To assign weights between directly connected items in the KG, we propose a new simple score function ("attention mechanism") that combines the symmetric sqrt normalization used in LightGCN (He et al., 2020) (see Section 3.3 for details regarding this normalization term) and an SBERT-based semantic similarity function. The score function that characterizes the importance of the relationship between two directly connected items u and v in the KG is computed as follows:

$$\omega_{u,v} = \frac{cos(e_u, e_v)}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}} \tag{9}$$

where e represents the item embedding, $\mathcal{N}_u$ and $\mathcal{N}_v$ denote the set of items directly connected to u and v, respectively, and cos is the cosine similarity between two embedding vectors. The adjacency matrix is computed as follows:

$$ADJm = \begin{cases} \omega_{u,v}, \text{ if } v \in \mathcal{N}_u \\ 0, \text{ otherwise} \end{cases} \tag{10}$$

3.1.2.3. Construct the final embedding matrix We apply GCN to enhance the representations of items in the KG. We use the embeddings obtained at the last layer (i.e., layer 2 in our case) as the final embeddings.

Fig. 6 gives an illustrative example of a two-layer receptive field for a given item MC_1, where l is set as 2. The left subgraph shows an example of KG structure where Slide 1 ($S1$) contains five main concepts (MC) that are related to four related concepts (RC) and one category (Cat). A learner did not understand (DNU) MC_1 and MC_3 and understood (U) MC_2. The right subgraph presents the high-order connectivity where the target node is MC_1. The high-order connectivity (i.e., two-layer receptive field in our example) denotes the path that connects MC_1 with its 2-hop neighbors in the KG. Such high-order connectivity contains rich semantics that carry information between items to get an enhanced representation of MC_1. Specifically, at layer $l+1$ an aggregation function is used to obtain the updated item representation by aggregating weighted embeddings of the direct neighboring items as well as the item itself (i.e., self-connection) from layer l as follows:

$$e_u^{(l+1)} = e_u^{(l)} + \sum_{v \in \mathcal{N}_u} \omega_{u,v}\, e_v^{(l)} \tag{11}$$

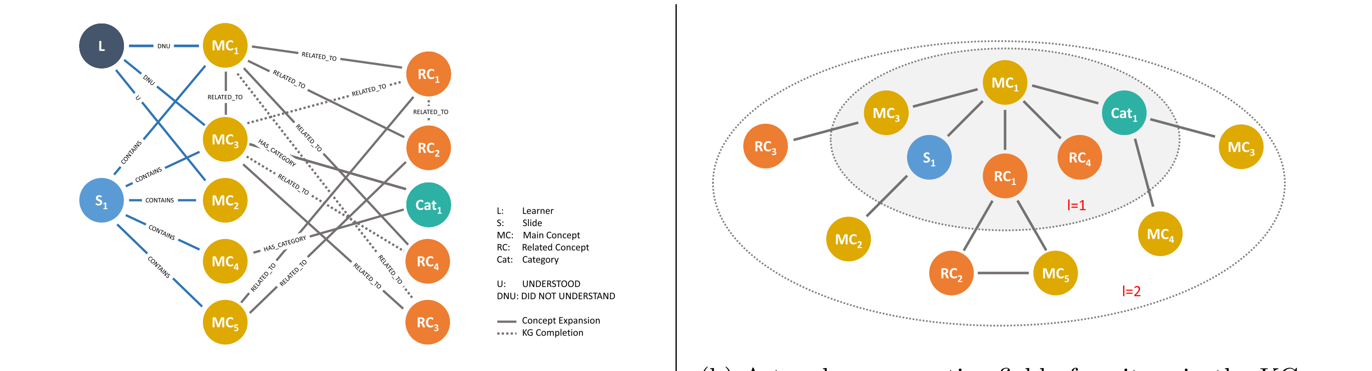
The enhanced representations of the items in the KG are achieved by multiplying the adjacency matrix $ADJm$ with the initial embedding matrix to get the new embedding matrix at layer 1. At layer 2, the adjacency matrix is multiplied with the new embedding matrix to get the final embedding matrix containing the enriched item embeddings which will be used for recommendation in the online phase (Fig. 5).

3.2. Online phase

The online phase consists of three main components: (1) *Representation of Learner Model* and (2) *Knowledge Concept Recommendation*, and (3) *Recommendation Explanation*, as shown in Fig. 1. The technical architecture related to the user interaction with the ConceptGCN-based knowledge concept recommender system is depicted in Fig. 7.

3.2.1. Representation of learner model

The learner model L is based on the concepts that the learner understood (U) and did not understand (DNU) when interacting with the slides of a learning material in *CourseMapper*. In this way, L is a vector where a concept marked as DNU (U) is represented as 1 (0). In our example from Fig. 6,



(a) Sample KG structure

(b) A two-layer receptive field of an item in the KG

Fig. 6. Illustration of a knowledge graph and high-order connectivity to enhance the embedding of the target node MC_1.

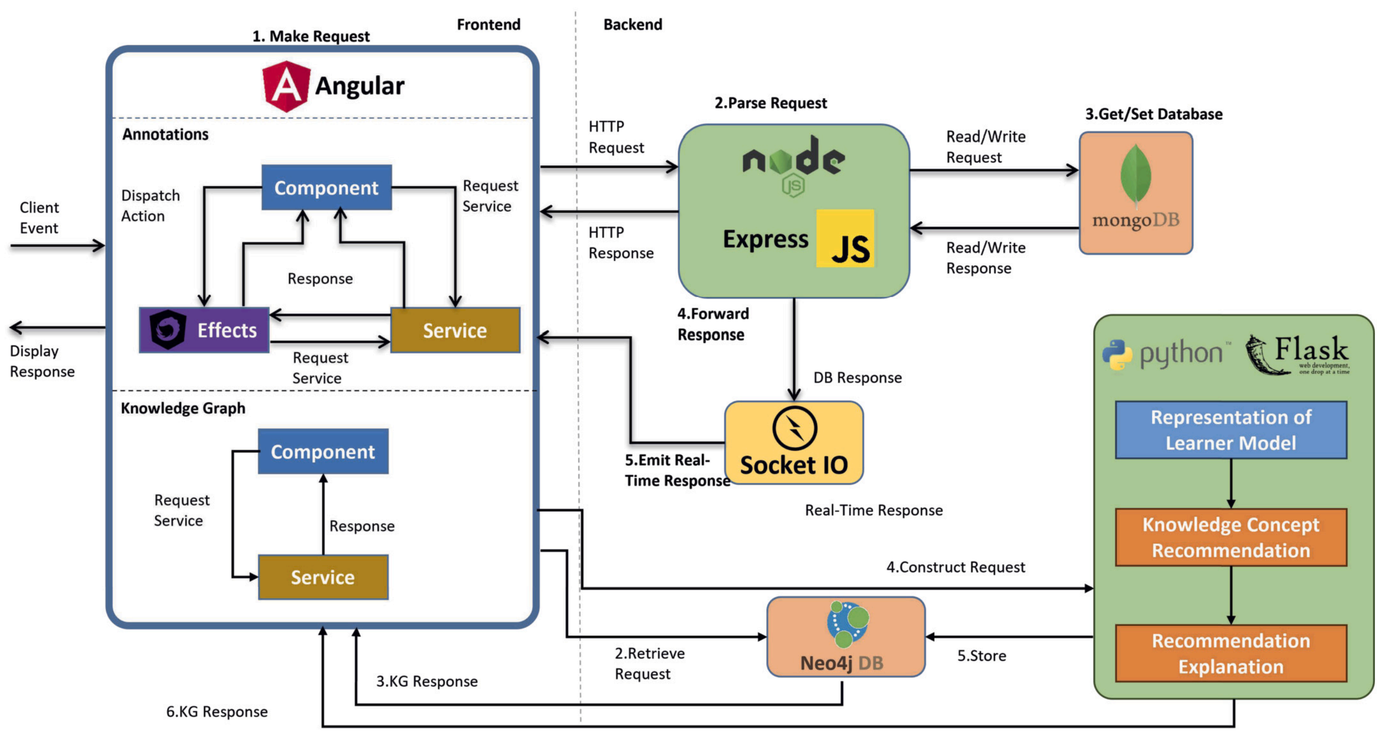


Fig. 7. Technical architecture of our proposed ConceptGCN-based recommender system.

$$L = [1, 0, 1] \tag{12}$$

where the learner did not understand (DNU) MC_1 and MC_3 and understood (U) MC_2. Given that the value of a concept understood by the learner is 0, the learner model can be represented only by the concepts that the learner did not understand. Therefore, we represent the learner model as a weighted average of the learner's DNU concepts, where the weight of a concept is computed as the cosine similarity score between its embedding and the learning material embedding (based on the content of the learning material). This way, concepts that are more relevant to the learning material are assigned higher weights than less relevant ones. These concepts indicate that the learner needs to master those concepts first, consequently making them more important for the learner. Utilizing the enriched embeddings of the learner's DNUs, along with their respective weights, the learner model's embedding is obtained as follows:

$$e_L = \left[\frac{1}{\omega_{sum}} \sum_{c \in DNU} \omega_c \, e_c \right]; \; \omega_c = cos(e_c, e_{lm}); \; \omega_{sum} = \sum_{c \in DNU} \omega_c \tag{13}$$

where e_c is the embedding of concept c and ω_c is its weight in the learning material lm.

3.2.2. Knowledge concept recommendation

We recommend relevant knowledge concepts to the learners based on their learner model L. A candidate knowledge concept (c_{cand}) for recommendation is a main concept or a related concept that the learner did not mark as U or DNU. We compute the cosine similarity $cos(e_L, e_{c_{cand}})$ between the embedding of the learner model (e_L) and the embedding of each candidate knowledge concept ($e_{c_{cand}}$). The top-5 ranked candidate knowledge concepts with the highest similarity score will then be recommended to the learner.

3.2.3. Recommendation explanation

We harness the structural and semantic information in the KG to explain the recommendations to help learners understand why specific knowledge concepts were suggested to them. To this end, we propose a weighting method to weigh the paths between the learner and the recommended concepts in the KG, wherein the path with the high-

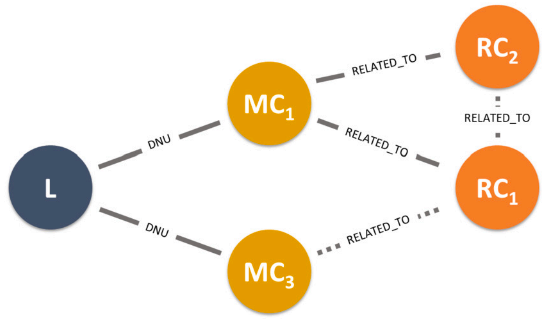


Fig. 8. Example for recommendation explanation.

est weight is utilized as the explanation for the recommendation. For example, if the learner did not understand MC_1 and MC_3 and the recommended concept is RC_1, as shown in Fig. 8, then a possible reason for recommending RC_1 could be that it is the related concept of both MC_1 and MC_3, or that it has the same related concept RC_2 as MC_1. In this work, we adopt various paths in the KG for recommendation explanation. Particularly, four types of paths between the learner and the recommended concept are applied for recommendation explanation. These paths are differentiated based on the type of the recommended concept (i.e., MC or RC). Fig. 9 presents the four possible paths for recommendation explanation.

The paths of type $Path1$ are constructed by establishing connections between the learner L and the recommended concept of type MC, where both MC and the recommended concept REC_{mc} are connected with the same slide node S as follows:

$$Path1 : L \overset{DNU}{\longrightarrow} MC \overset{\omega_{mc,s}}{\longrightarrow} S \overset{\omega_{s,rec_{mc}}}{\longrightarrow} REC_{mc} \tag{14}$$

The paths of type $Path2$ are formulated based on the connections between the learner L and the recommended concept of type MC/RC (i.e., MC or RC), where both MC and the recommended concept $REC_{mc/rc}$ have the same related concept RC as follows:

$$Path2 : L \overset{DNU}{\longrightarrow} MC \overset{\omega_{mc,rc}}{\longrightarrow} RC \overset{\omega_{rc,rec_{mc/rc}}}{\longrightarrow} REC_{mc/rc} \tag{15}$$

The formation of paths of type $Path3$ relies on the connections between the learner L and the recommended concepts of type MC/RC, where both MC and the recommended concept $REC_{mc/rc}$ have the same category Cat as follows:

$$Path3 : L \overset{DNU}{\longrightarrow} MC \overset{\omega_{mc,cat}}{\longrightarrow} Cat \overset{\omega_{cat,rec_{mc/rc}}}{\longrightarrow} REC_{mc/rc} \tag{16}$$

The construction of paths of type $Path4$ involves establishing connections between the learner L and the recommended concept of type MC/RC, where the recommended concept $REC_{mc/rc}$ represents the related concept of MC as follows:

$$Path4 : L \overset{DNU}{\longrightarrow} MC \overset{\omega_{mc,rec_{mc/rc}}}{\longrightarrow} REC_{mc/rc} \tag{17}$$

In all paths above, $\omega_{u,v}$ (u and v can be a slide s, main concept mc, related concept rc, or category cat) represents the cosine similarity score $cos(e_u, e_v)$ between the final embeddings of u and v. If the recommended concept type is main concept, then there are four types of paths: $Path1$, $Path2$, $Path3$, and $Path4$. If the recommended concept type is related concept, there are only three types of paths: $Path2$, $Path3$, $Path4$. The reason that there is no path of type $Path1$ is that there is no direct connection between a slide and a related concept in the KG. We find the different paths between the learner and the recommended concept in the KG for explanation. In case that there are more than one path of the same type, we select the one with the highest weight score. The weights for the different path types are calculated as follows:

$$\omega_{path1} = \omega_{mc,s} + \omega_{s,rec_{mc}} \tag{18}$$

$$\omega_{path2} = \omega_{mc,rc} + \omega_{rc,rec_{mc/rc}} \tag{19}$$

$$\omega_{path3} = \omega_{mc,cat} + \omega_{cat,rec_{mc/rc}} \tag{20}$$

$$\omega_{path4} = \omega_{mc,rec_{mc/rc}} \tag{21}$$

In the example in Fig. 10, RC_1 is recommended to the learner L. Between L and RC_1, there is one path of type $Path2$ and two paths of type $Path4$. Since there is only one path of type $Path2$ (i.e., $L \longrightarrow MC_1 \longrightarrow RC_2 \longrightarrow RC_1$), it is selected for explanation. Since there are two paths of type $Path4$, only the path $L \longrightarrow MC_1 \longrightarrow RC_1$ is selected for explanation, because it has a higher weight than the path $L \longrightarrow MC_3 \longrightarrow RC_1$. Finally, we provide visual and textual explanations of the recommendations. While the visual explanation shows a KG subgraph containing the paths used for explanation (Fig. 11), textual explanation describes these paths in textual format (Fig. 12).

3.3. Relation with LightGCN

It is worth mentioning that LightGCN (He et al., 2020) inspired us developing ConceptGCN. Similar to LightGCN, ConceptGCN simplifies the GCN update process by removing feature transformation and nonlinear activation. However, ConceptGCN differs from LightGCN in many aspects. First, LightGCN is more appropriate for collaborative filtering (CF) recommendation. Specifically, it learns both user and item embeddings by linearly propagating them on the user-item interaction graph. ConceptGCN, by contrast, uses GCN to refine only item (i.e., main concept, related concept, category, slide) embeddings. User (i.e., learner model) embeddings are built online during the learner's interaction with the MOOC platform based on the generated embeddings of DNU concepts.

Second, while LightGCN uses the weighted sum of the embeddings learned at all layers as the final embedding, ConceptGCN uses the embeddings obtained at the last layer (i.e., layer 2) as the final embedding. We made this choice for two reasons: (1) The evaluation of LightGCN showed that the best performance was achieved on layer 2, while after that it drops quickly to worst point of layer 4. This indicates that smoothing a node's embedding with only its first-order and second order neighbors is very useful (He et al., 2020). (2) Finding the appropriate weight which can be used to combine the embeddings learned at different propagation layers to form the final embedding is not a straightforward task. This weight denoting the importance of the k-th layer embedding in constituting the final embedding should be treated as a hyperparameter to be tuned manually, or as a model parameter to be optimized automatically (He et al., 2020). The authors of LightGCN did not design special component to optimize this weight. They noted that setting this weight uniformly as 1/(K+1) leads to good performance in general. Its optimal setting, however, needs further investigation.

Third, LightGCN removes the self-connection operation from GCN. That is, it aggregates the connected neighbors without integrating the target node itself. The authors of LightGCN pointed out that the layer combination operation (i.e., weighted sum of the embeddings at each layer to obtain the final embedding) essentially captures the same effect as self-connection. Since in ConceptGCN we are not doing layer combination, we add self-connection into the adjacency matrix.

Fourth, LightGCN uses an user-item interaction graph for CF, where each node (user or item) is only described by an ID feature, which has no concrete semantics besides being an identifier. In contrast, ConceptGCN uses as input a KG, where each node has rich attributes as input features. To capture the semantics of the different items in the KG (i.e., main concept, related concept, category, slide), we leverage SBERT for the starting features (i.e., the 0-th layer embeddings).

Fifth, LightGCN requires model training to provide recommendations, which can be computationally expensive. The trainable model parameters are the embeddings at the 0-th layer. In contrast, ConceptGCN utilizes pre-trained SBERT-based semantic input features and thus does not require training data and optimization mechanisms, making ConceptGCN simpler than LightGCN.

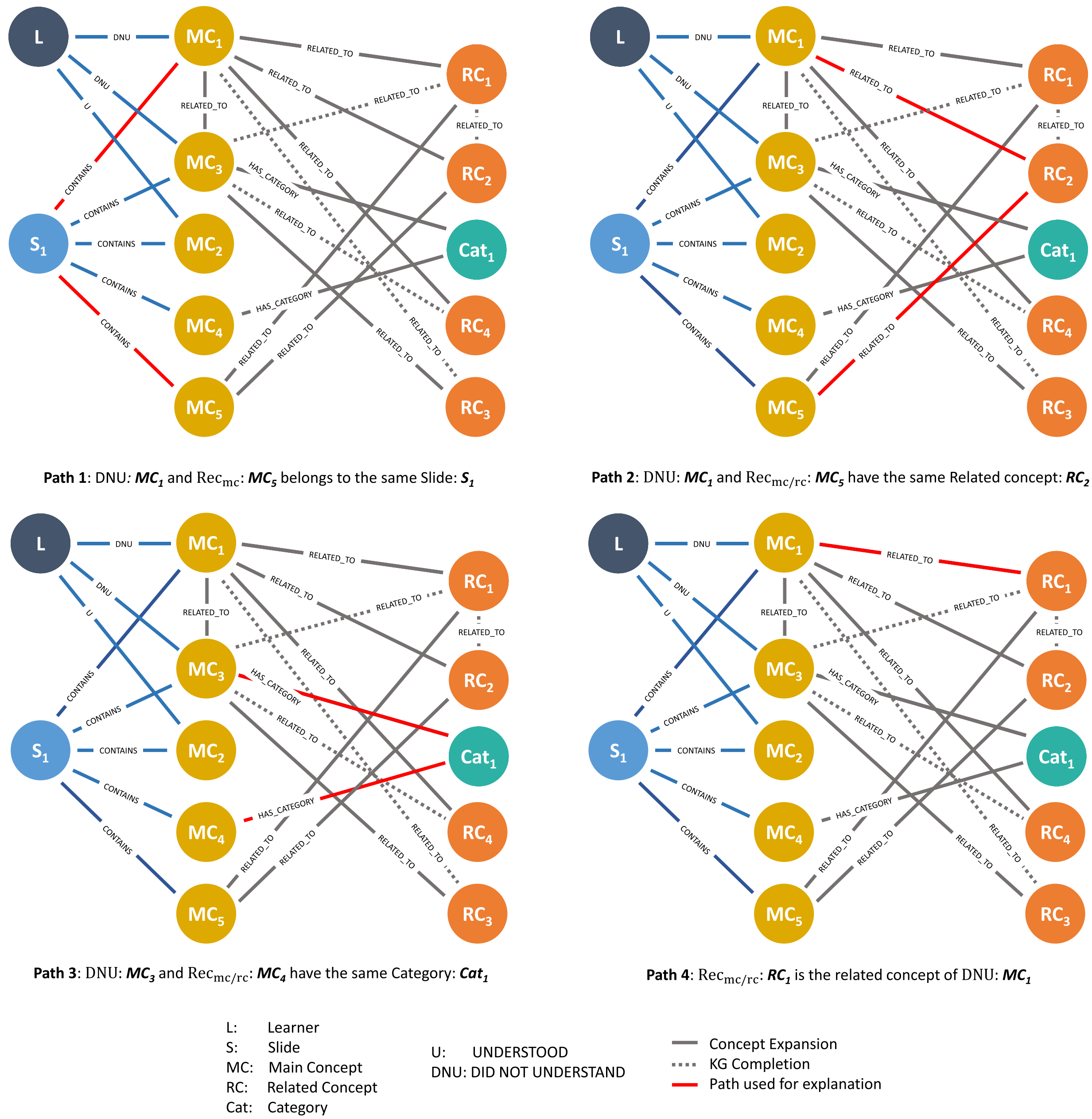


Fig. 9. The four types of paths used for recommendation explanation.

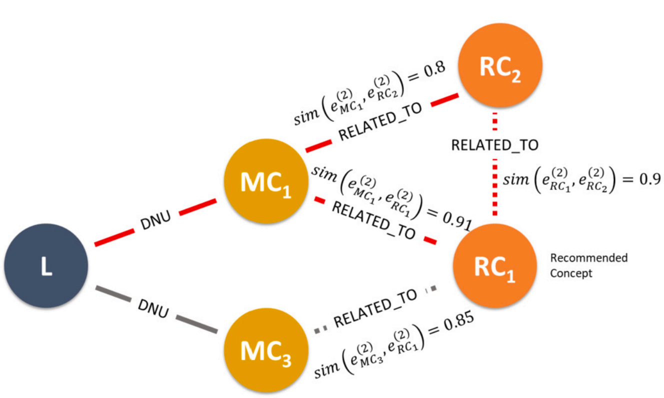


Fig. 10. Selection of the paths used for recommendation explanation.

Finally, LightGCN primarily leverage the graph structure information to obtain entity representations, neglecting the significance of semantic information. ConceptGCN, by contrast, takes full advantage of the rich structural and semantic information in the KG to enhance the entity representations. Unlike LightGCN which during the propagation process employs symmetric sqrt normalization as a score function to assign weights between linked entities in the graph based on their degree information (Equation (22)), ConceptGCN combines the symmetric sqrt normalization with an SBERT-based semantics-aware aggregation function (Equation (9)) to aggregate information from linked entities based on both degree information and semantic similarities to determine the importance of each neighbor.

$$e_u^{(l+1)} = \sum_{v \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_v|}} e_v^{(l)} \tag{22}$$

Fig. 11. Visual explanation.

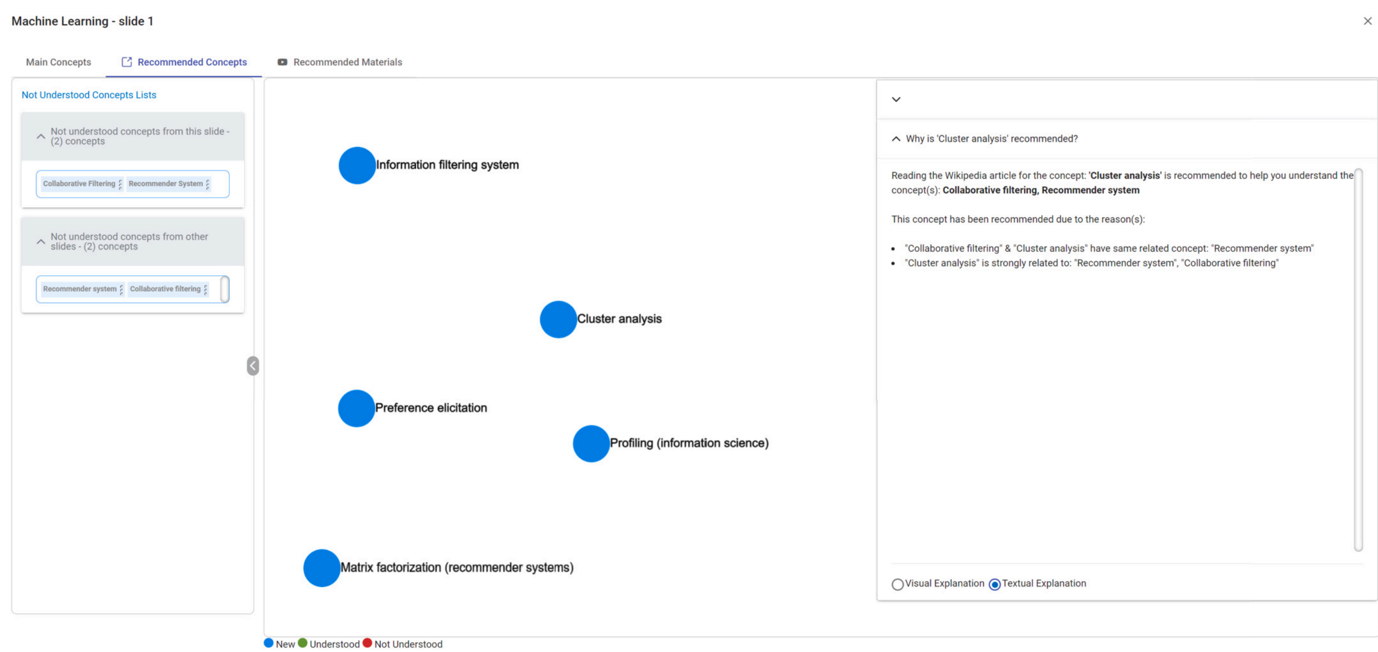


Fig. 12. Textual explanation.

where $\mathbf{e}_u^{(l+1)}$ represents the item representation at the $(l+1)^{th}$ layer, $\mathcal{N}_u$ denotes the neighbors of item u, and $\mathbf{e}_v^{(l)}$ represents the item representation of neighbor v at the l^{th} layer. The symmetric sqrt normalization factor $\frac{1}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}$ accounts for the influence of the degree of nodes u and v. It follows the design of standard GCN (Kipf & Welling, 2016) and aims at avoiding the scale of embeddings increasing with graph convolution operations (He et al., 2020).

4. Experiments and results

We conducted offline experiments focused on investigating the effectiveness of our proposed KG construction process. Moreover, we conducted an online user study to evaluate the accuracy and learners' perceptions of the benefits of our proposed ConceptGCN-based recommendation approach.

4.1. Evaluation of knowledge graph construction

To answer our first research question ("How to effectively construct a KG that can be used in a MOOC platform to provide personalized and explainable recommendation of knowledge concepts?"), we conducted extensive offline experiments to evaluate the effectiveness of our proposed KG construction process. Specifically, this evaluation aims to ascertain the optimal combination of methods to optimize the construction of the KG.

Table 1
Evaluation results on Inspec & SemEval2017 datasets.

K	Method	Inspec			SemEval2017		
		P	R	F1	P	R	F1
5	$SingleRank$	36.76	18.71	24.80	41.26	11.92	18.50
	$SIFRank_{SqueezeBERT}$	**44.16**	**22.48**	**29.79**	**47.79**	**13.81**	**21.43**
10	$SingleRank$	33.22	33.53	33.37	38.11	22.03	27.92
	$SIFRank_{SqueezeBERT}$	**39.46**	**39.68**	**39.57**	**43.83**	**25.34**	**32.11**
15	$SingleRank$	30.05	44.24	35.80	34.90	30.23	32.40
	$SIFRank_{SqueezeBERT}$	**34.19**	**49.78**	**40.54**	**39.65**	**34.32**	**36.79**

4.1.1. Evaluation of keyphrase extraction

To evaluate the keyphrase extraction step in the Slide-KG construction process, offline experiments are conducted on two keyphrase extraction techniques, namely SingleRank (Wan & Xiao, 2008) and $SIFRank_{SqueezeBERT}$ (Ain et al., 2023), with a focus on precision and performance speed. $SIFRank_{SqueezeBERT}$ is an enhanced version of the SIFRank keyphrase extraction method proposed in (Sun et al., 2020) by adopting SqueezeBERT (Iandola et al., 2020), a transformer model for word embedding. To conduct the experiments, two well-known benchmark datasets are used:

- **Inspec dataset** (Hulth, 2003) contains a total of 2000 publication abstract of journal papers from Inspec database of computer science and information technology. Each abstract has two kinds of keyphrases: controlled keyphrases restricted by a dictionary and uncontrolled keyphrases annotated by experts. The document set is composed of three sets: a training set of 1,000 summaries, a validation set of 500 summaries, and a test set of 500 summaries. Following the evaluation method in (Sun et al., 2020, Hulth, 2003, Mihalcea & Tarau, 2004), both controlled and uncontrolled keyphrases were used as ground truth for the evaluation. 76.2% of the uncontrolled keyphrases were present in the abstracts, while only 18% of the controlled keyphrases were present in the abstracts. The average length of the abstracts was 134.4 tokens and the average number of keywords per abstract was 9.8.
- **SemEval2017 dataset** (Augenstein et al., 2017) is a double-annotated document set of 493 passages extracted from 500 ScienceDirect journal articles. These articles cover the fields of computer science, materials science, and physics. Each article has a number of keyphrases assigned by an undergraduate student and an expert annotator. In case of disagreement between the two annotators, the expert annotator's annotation is given priority. The average length of paragraphs was 194.7 tokens and the average number of keywords per paragraph was 17.3.

The evaluation is conducted according to the approximate matching strategy (Zesch & Gurevych, 2009). This strategy considers relevant keyphrases to be subsets or substrings of the annotated keyphrases. To accomplish this, the precision (P), recall (R), and F1 score measures are calculated utilizing the stemmed keyphrases, which include both the extracted keyphrases and the annotated keyphrases. Table 1 shows the results of precision, recall, and F1 score for top 5, 10 and 15 keyphrases extracted using SingleRank and $SIFRank_{SqueezeBERT}$ on Inspec and SemEval2017 datasets.

The results show that $SIFRank_{SqueezeBERT}$ performed slightly better than SingleRank in extracting the top-5, top-10 and top-15 keyphrases for both datasets. To measure the time performance of both methods on constructing the KG, a learning material consisting of 126 slides was chosen. This learning material contained a combination of text, images, mathematical formulas, and code snippets. The time measured for both approaches consists of the time needed for the keyphrase extraction from a slide, Slide-KG construction, and LM-KG construction. The results can be found in Table 2. In extracting keyphrases from a slide, SingleRank takes only 0.1∼0.3 s. Whereas, $SIFRank_{SqueezeBERT}$ takes 8∼9 s, which makes it ten times slower than SingleRank. Moreover, using SingleRank to build the LM-KG takes nearly twenty minutes less than $SIFRank_{SqueezeBERT}$, thus making SingleRank a better choice for the keyphrase extraction task.

4.1.2. Evaluation of main concept filtering

During the Slide-KG construction process, the main concepts undergo a filtering procedure to ensure the retention of the most relevant ones, i.e., those closely related to the slide they are identified from and the learning material in general. Considering that computing similarities of main concepts with both slide and learning material might have a negative impact on the time required to construct the LM-KG, we experimented with three different options to filter the main concepts: (1) based on the similarity between the main concept and the slide, (2) based on the similarity between the main concept and the learning material, or (3) based on a combination of both similarities. To compare these three options, the same learning material consisting of 126 slides was chosen and the LM-KG was constructed for it. We assessed each option based on its impact on the time required for LM-KG construction. The result can be found in Table 3. The observed time difference between all three options is minimal. Therefore, we selected the combination of both slide similarity and learning material similarity because it considers both types of similarities, thus leading to the retention of more relevant main concepts.

4.1.3. Evaluation of concept expansion

The result of the concept expansion step of the KG construction process are related concepts and categories which will build the base for recommendation in the online phase. Considering that the time performance of the recommender system is impacted by how the main concepts are expanded, we experimented with two options for the concept expansion step. The first option entails expanding all the main concepts, prioritizing a more complete enrichment of the LM-KG. This option, however, would require more time in the offline phase. The second option involves expanding solely the top-15 main concepts from the learning material, prioritizing time efficiency in the offline phase. However, this option requires to perform further concept expansion and KG completion in the online phase. Specifically, if a learner did not understand a main concept from a slide and this concept is not contained in the expanded top-15 main concepts, the related concepts and categories associated with that particular main concept would be absent from the LM-KG and would need to be expanded during the interaction with the learning material. This would lead to significant waiting time for the learner, thus negatively impacting the user experience. To compare the impact of both options on the time needed for concept expansion (offline) and recommendation generation (online), the same learning material with 126 slides was chosen. The time performance results are shown in Table 4. Expanding all main concepts (first option) requires 8x more time than only expanding the top-15 main concepts. On the other hand, the second option demands 40x more time for recommendation generation than if all concepts have been expanded before the recommendation takes place. To ensure faster recommendation and better user experience, we selected the option of expanding all main concepts in the offline phase.

Table 2
Time performance using $SingleRank$ and $SIFRank_{SqueezeBERT}$.

	Keyphrase extraction	Slide-KG construction	LM-KG construction
$SingleRank$	**0.1∼0.3 s**	**11.5 s**	**1445.62 s**
$SIFRank_{SqueezeBERT}$	8∼9 s	20.4 s	2573.27 s

Table 3
Time spent for LM-KG construction based on three options of main concept filtering.

Similarity(MC,Slide)	Similarity(MC,LM)	LM-KG construction time
✓	x	1445.62 s
x	✓	1471.51 s
✓	✓	**1333.12 s**

Table 4
Time spent based on two options of concept expansion.

Expanded main concepts	Concept expansion time (offline)	Recommendation generation time (online)
All	16086.51 s	2.70 s
Top-15	1999.151 s	106.21 s

Table 5
Time spent for three options of related concepts/categories filtering.

Similarity(RC/Cat,MC)	Similarity(RC/Cat,LM)	LM-KG construction time
✓	x	2028.65 s
x	✓	1829.45 s
✓	✓	1999.151 s

4.1.4. Evaluation of related concept and category filtering

Within the concept expansion process, a critical aspect involves filtering the related concepts and categories to preserve the most significant ones associated with the main concept. We considered three different options to filter related concepts and categories: (1) based on the similarity between the related concepts/categories and the main concept, (2) based on the similarity between the related concepts/categories and the learning material, or (3) based on a combination of both the similarities. These three filtering options are experimentally evaluated based on the time taken to construct the LM-KG. The results in Table 5 show that the difference in time needed to filter the related concepts/categories is minimal. However, we opted for combining both similarities, as considering both similarities is expected to lead to the retention of more relevant related concepts and categories.

4.1.5. Final KG construction

The final LM-KG is constructed by choosing the most suitable option from each step evaluated in the previous sections. Concretely, for keyphrase extraction, SingleRank was chosen based on its time efficiency. For filtering the main concepts, the combination of both the similarity between the main concepts and the slide and the similarity between the main concepts and the learning material were considered as only minimal time difference could be observed. In the concept expansion step, we opted to expand all the main concepts because this reduces the waiting time required to get the recommendations compared to only expanding the top-15 main concepts. Lastly, for filtering the related concepts and categories, a combination of both the similarity between the slide and the related concepts/categories and the similarity between the learning material and the related concepts/categories was selected. An overview of all the chosen options for each step can be found in Table 6.

Table 6
The selected options for final KG construction.

KG construction step	Selected option
Keyphrase extraction	SingleRank
Main concept filtering	Similarity(MC,Slide)+Similarity(MC,LM)
Concept expansion	All main concepts
Related concept/category filtering	Similarity(RC/Cat,MC)+Similarity(RC/Cat,LM)

4.2. Evaluation of the recommendation benefits

To answer our second research question ("What is the potential impact of the proposed ConceptGCN-based recommendation approach on learners' perceptions of the ERS in terms of accuracy, novelty, diversity, usefulness, overall satisfaction, use intentions, and reading intention?"), we conducted an online user study (N = 31) to gauge the accuracy and perceived benefits of ConceptGCN. We evaluated ConceptGCN against a baseline variant of LightGCN that we adapted to the context of our MOOC platform. We chose LightGCN as a baseline because it is the method that is most relevant with ConceptGCN and it has shown to outperform several GCN-based recommendation methods (He et al., 2020). Similar to LightGCN, the implemented variant also removes the self-connection, feature transformation, and nonlinear activation operations. Moreover, it uses the symmetric sqrt normalization as a score function in the aggregation operation. However, the implemented variant adapts LightGCN in three ways: (1) It generates only item (i.e., main concept, related concept, category, slide) embeddings, (2) it employs SBERT for the initial item embeddings, and (3) it uses the embeddings obtained at layer 2 as the final item embeddings. For the sake of brevity, we will refer to this implemented variant as LightGCN in the following.

4.2.1. User study

We conducted an online user study to assess the accuracy and perceived benefits of ConceptGCN. We chose not to rely on offline evaluation for two main reasons. Firstly, our dataset did not possess a sufficiently large scale, making it challenging to obtain reliable and statistically significant results through offline evaluation methods. Secondly, offline evaluations have known limitations. Previous studies including those by Beel et al. (2013), Chatti et al. (2013), and McNee et al. (2006) have revealed conflicting findings when comparing user studies to offline evaluations, leading to doubts regarding the validity of offline evaluation in capturing real user preferences. Furthermore, offline evaluations lack direct user feedback, which limits their ability to capture the dynamic and subjective aspects of user satisfaction and preferences. Thus, we opted for an online user study to obtain more accurate and insightful evaluations of the recommendations generated by our ConceptGCN-based ERS.

4.2.1.1. Participants A total of 31 participants took part in the evaluation, comprising 16 males, 14 females, and one participant who opted not to disclose their gender. Among the participants, only one individual was over 35 years of age, while the remaining participants fell within the 20-35 age range. Regarding educational background, participants exhibited diverse levels of attainment, including Bachelor's degrees, Master's degrees, and Ph.D. degrees. The majority of participants had pursued studies in the field of computer engineering, while some had backgrounds in computer science, applied computing science, or other related disciplines. It is worth noting that the participants demonstrated a strong familiarity with recommender systems, as their prior academic studies had extensively covered this subject.

4.2.1.2. Measurement To evaluate the participants' perceptions of the recommendations, the ResQue framework (Pu et al., 2011) is utilized. ResQue is a user-centric evaluation framework specifically designed to evaluate the experience of users during their interaction with RSs. It encompasses four main categories: *perceived system qualities*, *beliefs*, *attitudes*, and *behavioral intentions*. Each category is further dived into subcategories. In this study, to evaluate the perceived system quality the subcategories *perceived accuracy*, *novelty*, and *diversity* were chosen. To evaluate the beliefs about the system the subcategory *perceived usefulness* was chosen, for the attitudes toward the system it was decided to choose the subcategory *overall satisfaction*, and finally, for behavioral intentions, the subcategories *use intentions* and *reading intention* were chosen. By incorporating these constructs, we aimed to evaluate the recommendation model from a holistic perspective, considering the learners' perceptions of recommendation qualities, beliefs, attitudes, and intentions to use the system.

4.2.1.3. Procedure The study was conducted remotely via Microsoft Teams. In the initial phase, participants were provided with a comprehensive explanation of the evaluation's purpose and content. An assurance of response anonymity and non-disclosure of data to third parties was given to participants. Consent was obtained before commencing the session and recording the videos. Subsequently, participants were introduced to the system through an interactive video. After confirming their understanding of the system, participants were requested to complete their demographic profiles. Following this, participants engaged in a task on our platform via remote screen control. The task involved reading a learning material focused on "Recommender Systems" which was sourced from a course offered by our department. The learning material comprises a combination of text, images, and mathematical formulas/code snippets. As the participants had previously taken part in that course, they were familiar with its content. During the task, participants were requested to identify concepts within the learning material that they did not understand (i.e., DNU concepts). The identified concepts were collected, and the system presented participants with recommended concepts that should help them understand the previously not understood concepts. Participants were then instructed to see the recommended concepts, read their corresponding Wikipedia descriptions, and complete a questionnaire related to the recommended concepts. This is repeated for both models meaning that all participants see recommendations generated from both models. The questionnaires were administered using Google Forms and were designed based on the ResQue framework, focusing on different criteria, namely accuracy, novelty, diversity, perceived usefulness, overall satisfaction, use intentions, and reading intention. A single question was designed for each criterion, utilizing statements from the ResQue framework.

- **Recommendation Accuracy**: The items recommended to me matched my interests.
- **Recommendation Novelty**: The recommender system helped me discover new concepts.
- **Recommendation Diversity**: The items recommended to me are diverse.
- **Perceived Usefulness**: The recommender gave me good suggestions.
- **Overall Satisfaction**: Overall, I am satisfied with the recommender.
- **Use Intentions**: I will use this recommender frequently.
- **Reading Intention**: I would read about the recommended concepts, given the opportunity.

The responses were collected using a 5-point Likert scale ranging from "strongly disagree" to "strongly agree". Additionally to the items taken from the ResQue framework, we asked participants for each model how many of the recommended items they found relevant using the question "How many of the recommended concepts do you feel are relevant?". The responses to this question were used to calculate the precision (Precision@5) for each model. We use the paired *t-test* for testing the significance where the significance level of α is set to 0.05. The study took approximately one hour per participant.

4.2.2. Results and analysis

We evaluated the recommendation models based on the seven criteria perceived accuracy, novelty, diversity, perceived usefulness, overall satisfaction, use intentions, and reading intentions. Fig. 13 shows the results and the comparison between the two models. The mean and standard deviation for each criterion are shown in Fig. 14. No statistically significant differences between the two models across all seven criteria can be observed. Concretely, the results indicate that both models deliver similar good results regarding perceived accuracy: ConceptGCN (agreement 55% vs. disagreement 16%), LightGCN (agreement 65% vs. disagreement 6%); novelty: ConceptGCN (agreement 80% vs. disagreement 6%), (agreement 88% vs. disagreement 0%); diversity: ConceptGCN (agreement 71% vs. disagreement 6%), LightGCN (agreement 62% vs. disagreement 6%); perceived usefulness: ConceptGCN (agreement 61% vs. disagreement 13%), LightGCN (agreement 61% vs. disagreement 13%); overall satisfaction: ConceptGCN (agreement 55% vs. disagreement 19%), LightGCN (agreement 61% vs. disagreement 16%); use intentions: ConceptGCN (agreement 39% vs. disagreement 19%), LightGCN (agreement 39% vs. disagreement 19%); and reading intention: ConceptGCN (agreement 77% vs. disagreement 10%), LightGCN (agreement 81% vs. disagreement 0%). Overall, the LightGCN model exhibited slightly better results compared to ConceptGCN. Only with regards to diversity, ConceptGCN was perceived more effective in providing recommendations that encompassed more diverse concepts. One reason we deem responsible for this result is that, in addition to structural information in the KG, ConceptGCN employs semantic similarities between entities when performing neighborhood aggregation.

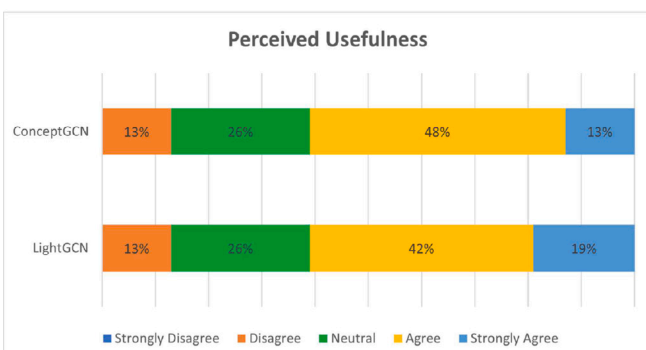
For further investigation, we calculated the precision (Precision@5) of the two models using the response to the question "How many of the recommended concepts do you feel are relevant?". Both ConceptGCN and LightGCN models achieved relatively high precision scores. The results further indicate that the LightGCN model provides slightly more accurate recommendations compared to ConceptGCN (69% vs. 63%), as illustrated in Fig. 15. However, the results were not statistically significant.

In summary, the results of our user study demonstrate the benefits of the ConceptGCN-based recommendation approach, in terms of several important user-centric aspects including accuracy, novelty, diversity, usefulness, overall satisfaction, use intentions, and reading intention. Overall, except for diversity, the LightGCN model performed slightly better than ConceptGCN. The main difference between the two models was that ConceptGCN uses a self-connection operation and an SBERT-based score function in the aggregation operation. The results suggest that, if SBERT is used for the initial embeddings of items (i.e., main concept, related concept, category, slide), the self-connection and the semantic similarity-based score function are not necessarily needed. We speculate that self-connection is not needed because connected neighbors of an item in the KG are semantically similar and consequently have already SBERT representations similar to the target item. Thus, aggregating SBERT-based embeddings of connected neighbors subsumes the effect of self-connection. Further, the SBERT-based score function is not required because connected items in the KG have already high semantic similarities.

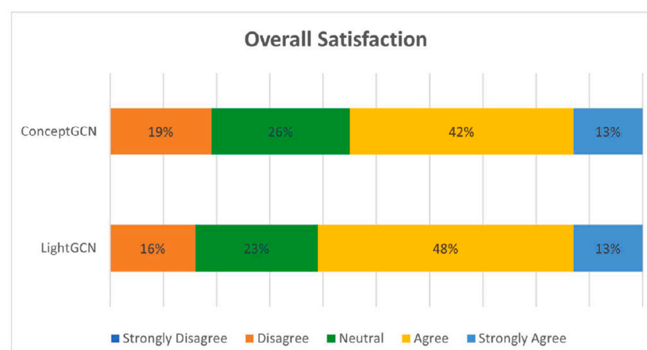
5. Conclusion and future work

Owing to the superiority of knowledge graphs (KGs) in modeling the heterogeneous data in technology-enhanced learning (TEL) environments and Graph Neural Networks (GNNs) in learning on graph data, utilizing KG and GNN techniques in educational recommender systems (ERS) has gained increasing interest. In particular, few works

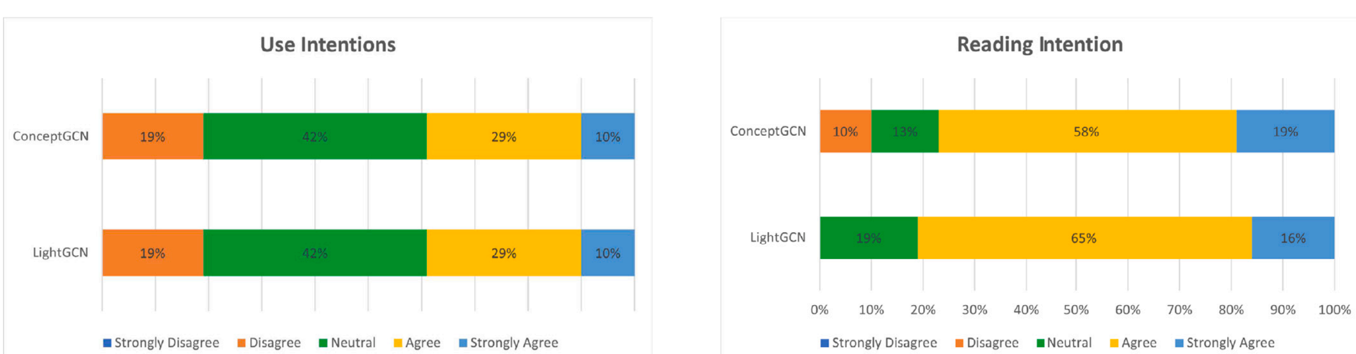
(a) Perceived System Qualities.



(b) Beliefs.



(c) Attitudes.



(d) Behavioral Intentions.

Fig. 13. Results from the ResQue questionnaire.

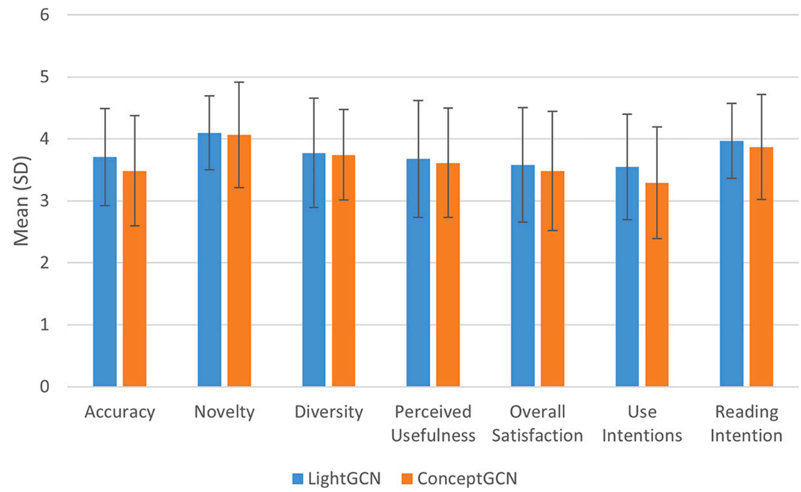


Fig. 14. Mean and standard deviation for each criterion from the ResQue questionnaire.

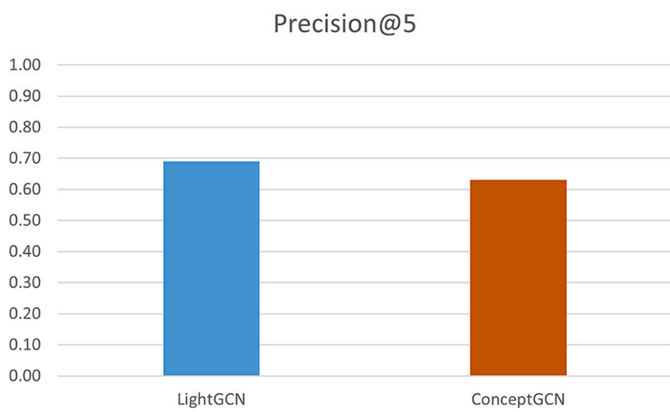


Fig. 15. Accuracy (Precision@5) of LightGCN and ConceptGCN.

have applied these techniques to recommend knowledge concepts in MOOCs. These approaches, however, have limitations mainly related to complexity, semantics, and transparency. To address these issues, in this work we presented ConceptGCN, a comprehensive framework for recommending knowledge concepts to learners. Conceptually, our approach combines KGs, Graph Convolutional Networks (GCNs), and transformer sentence encoders (SBERT) to enhance the representations of knowledge concepts and learner models and provide personalized and explainable recommendation of concepts. To evaluate our approach, we conducted extensive offline experiments to investigate the most performing methods to construct a KG in the MOOC platform *CourseMapper*. Moreover, we conducted an online user study (N = 31) to gauge the accuracy and investigate the impact of a ConceptGCN-based recommendation approach on learners' perceptions of the ERS in terms of several important user-centric aspects including accuracy, novelty, diversity, usefulness, overall satisfaction, use intentions, and read intention. Our results indicate that, in general, combining KG, GCN, and SBERT provides a simple, yet effective method to provide accurate and explainable recommendation of knowledge concepts in MOOCs. In future work, we are planning to optimize the KG construction pipeline to identify more accurate knowledge concepts which is essential for improved learner modeling and recommendation. Moreover, we will investigate alternative GNN-based recommendation approaches and weighting techniques to improve the overall effectiveness of the ERS. Another interesting direction in future work would be to develop different GNN-based explanation types and explore their effects on the learners' perceptions of the explainable ERS, in terms of different explanation aims including efficiency, effectiveness, persuasiveness, transparency, satisfaction, scrutability, and trust.

Acronyms

Massive Open Online Courses (MOOCs)
Technology-Enhanced Learning (TEL)
Educational Recommender Systems (ERSs)
Knowledge Graphs (KGs)
Graph Neural Networks (GNNs)
Graph Convolutional Networks (GCNs)
Sentence Bidirectional Encoder Representations from Transformers (SBERT)
Graph Convolutional Networks (GCNs)
Recommender Systems (RS)
Natural Language Processing (NLP)
Did Not Understand (DNU)
Collaborative Filtering (CF)
Graph Attention Networks (GAT)
Neural Graph Collaborative Filtering (NGCF)
Graph Convolutional Matrix Completion (GC-MC)
Heterogeneous Information Networks (HINs)
Knowledge Graph Embedding (KGE)
Knowledge Graph Convolutional Network (KGCN)
Ripple Knowledge Graph Convolutional Networks (RKGCN)
Knowledge Graph Attention Network (KGAT)
Light Knowledge Graph Convolutional Network (LKGCN)
MOOC Interest Recommender (MOOCIR)
Learning Material Knowledge Graph (LM-KG)
Slide Knowledge Graph (Slide-KG)
Learning Material (LM)
Slide (S)
Main Concept (MC)
Related Concept (RC)
Category (Cat)
Learner (L)
Understood (U)
Adjacency Matrix (ADJm)

Statement on open data and ethics

All procedures in the study were conducted in accordance with applicable laws and institutional guidelines. Informed consent was obtained from all participants, and their privacy rights were strictly observed.

CRediT authorship contribution statement

Rawaa Alatrash: Conceptualization, Methodology, Visualization, Writing – original draft, Writing – review & editing, Software, Formal analysis, Investigation, Validation. **Mohamed Amine Chatti:** Conceptualization, Project administration, Supervision, Writing – original draft, Writing – review & editing, Formal analysis, Methodology, Validation, Visualization. **Qurat Ul Ain:** Validation, Writing – review & editing, Conceptualization, Methodology, Supervision, Visualization. **Yipeng Fang:** Data curation, Formal analysis, Software, Validation, Conceptualization. **Shoeb Joarder:** Software, Visualization. **Clara Siepmann:** Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We acknowledge the support from the Open Access Publication Fund of the University of Duisburg-Essen.

References

Ain, Q. U., Chatti, M. A., Joarder, S., Nassif, I., Wobiwo Teda, B. S., Guesmi, M., & Alatrash, R. (2022). Learning channels to support interaction and collaboration in CourseMapper. In *Proceedings of the 14th international conference on education technology and computers* (pp. 252–260).

Ain, Q. U., Chatti, M. A., Bakar, K. G. C., Joarder, S., & Alatrash, R. (2023). Automatic construction of educational knowledge graphs: A word embedding-based approach. *Information*, *14*. https://doi.org/10.3390/info14100526. https://www.mdpi.com/2078-2489/14/10/526.

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications (pp. 546–555). https://doi.org/10.18653/v1/S17-2091.

Beel, J., Genzmehr, M., Langer, S., Nürnberger, A., & Gipp, B. (2013). A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In *Proceedings of the international workshop on reproducibility and replication in recommender systems evaluation* (pp. 7–14).

Berg, R. v. d., Kipf, T. N., & Welling, M. (2017). Graph convolutional matrix completion. arXiv preprint, arXiv:1706.02263.

Chatti, M. A., Dakova, S., Thüs, H., & Schroeder, U. (2013). Tag-based collaborative filtering recommendation in personal learning environments. *IEEE Transactions on Learning Technologies*, *6*, 337–349.

Chatti, M. A., Guesmi, M., & Muslim, A. (2023). Visualization for recommendation explainability: A survey and new perspectives. arXiv preprint, arXiv:2305.11755.

Chatti, M. A., Guesmi, M., Vorgerd, L., Ngo, T., Joarder, S., Ain, Q. U., & Muslim, A. (2022). Is more always better? The effects of personal characteristics and level of detail on the perception of explanations in a recommender system. In *Proceedings of the 30th ACM conference on user modeling, adaptation and personalization* (pp. 254–264).

Chicaiza, J., & Valdiviezo-Diaz, P. (2021). A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information*, *12*, 232.

Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., et al. (2023). A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, *1*, 1–51.

Gong, J., Wang, C., Zhao, Z., & Zhang, X. (2021). Automatic generation of meta-path graph for concept recommendation in MOOCs. *Electronics*, *10*, 1671.

Gong, J., Wang, S., Wang, J., Feng, W., Peng, H., Tang, J., & Yu, P. S. (2020). Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 79–88).

Guesmi, M., Chatti, M. A., Joarder, S., Ain, Q. U., Alatrash, R., Siepmann, C., & Vahidi, T. (2023). Interactive explanation with varying level of details in an explainable scientific literature recommender system. *International Journal of Human-Computer Interaction*, 1–22. https://doi.org/10.1080/10447318.2023.2262797.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., & He, Q. (2020). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, *34*, 3549–3568.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *30*.

Hassan, H. A. M., Sansonetti, G., Gasparetti, F., Micarelli, A., & Beel, J. (2019). BERT, ELMo, USE and InferSent sentence encoders: The panacea for research-paper recommendation? In *RecSys (late-breaking results)* (pp. 6–10).

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 639–648).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. https://doi.org/10.3115/1119355.1119383.

Iandola, F. N., Shaw, A. E., Krishna, R., & Keutzer, K. W. (2020). SqueezeBERT: What can computer vision teach NLP about efficient neural networks? arXiv preprint, arXiv:2006.11316.

Khalid, A., Lundqvist, K., & Yates, A. (2020). Recommender systems for MOOCs: A systematic literature survey (January 1, 2012–July 12, 2019). *International Review of Research in Open and Distributed Learning*, *21*, 255–291.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint, arXiv:1609.02907.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*, 30–37.

Li, C., Cao, Y., Zhu, Y., Cheng, D., Li, C., & Morimoto, Y. (2023). Ripple knowledge graph convolutional networks for recommendation systems. arXiv preprint, arXiv:2305.01147.

McNee, S. M., Kapoor, N., & Konstan, J. A. (2006). Don't look stupid: Avoiding pitfalls when recommending research papers. In *Proceedings of the 2006 20th anniversary conference on computer supported cooperative work* (pp. 171–180).

Mendes, P. N., Jakob, M., Garcia-Silva, A., & Bizer, C. (2011). DBpedia spotlight: Shedding light on the web of documents. In *International conference on semantic systems*.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *EMNLP*.

Piao, G. (2021). *Recommending knowledge concepts on MOOC platforms with meta-path-based representation learning*. International Educational Data Mining Society.

Pu, P., Chen, L., & Hu, R. (2011). A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on recommender systems* (pp. 157–164).

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. arXiv preprint, arXiv:1908.10084.

Shinyama, Y. (2013). PDFMiner. https://unixuser.org/~euske/python/pdfminer/.

Sun, Y., Qiu, H., Zheng, Y., Wang, Z., & Zhang, C. (2020). SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, *8*, 10896–10906. https://doi.org/10.1109/ACCESS.2020.2965087.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint, arXiv:1710.10903.

Wan, X., & Xiao, J. (2008). Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *International conference on computational linguistics*.

Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., & Guo, M. (2018). RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 417–426).

Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., & Wang, Z. (2019). Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 968–977).

Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., & Guo, M. (2019). Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference* (pp. 2000–2010).

Wang, H., Zhao, M., Xie, X., Li, W., & Guo, M. (2019). Knowledge graph convolutional networks for recommender systems. In *The world wide web conference* (pp. 3307–3313).

Wang, Q., Tragos, E., Hurley, N., Smyth, B., Lawlor, A., & Dong, R. (2022). Entity-enhanced graph convolutional network for accurate and explainable recommendation. In *Proceedings of the 30th ACM conference on user modeling, adaptation and personalization* (pp. 79–88).

Wang, S., Hu, L., Wang, Y., He, X., Sheng, Q., Orgun, M., Cao, L., Ricci, F., & Yu, P. (2021). Graph learning based recommender systems: A review. In Z. H. Zhou (Ed.), *International joint conferences on artificial intelligence*. *Proceedings of the thirtieth international joint conference on artificial intelligence* (pp. 4644–4652).

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T. S. (2019). KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 950–958).

Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 165–174).

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *International conference on machine learning* (pp. 6861–6871). PMLR.

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, *55*, 1–37.

Yang, Z., & Dong, S. (2020). Hagerec: Hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. *Knowledge-Based Systems*, *204*, Article 106194.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 974–983).

Yousef, A. M. F., Chatti, M. A., Schroeder, U., & Wosnitza, M. (2014). What drives a successful MOOC? an empirical examination of criteria to assure design quality of MOOCs. In *2014 IEEE 14th international conference on advanced learning technologies* (pp. 44–48). IEEE.

Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., & Han, J. (2014). Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on web search and data mining* (pp. 283–292).

Zesch, T., & Gurevych, I. (2009). Approximate matching for evaluating keyphrase extraction. In *RANLP*.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016). 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).

Zhao, Y., Ma, W., Jiang, Y., & Zhan, J. (2021). A MOOCs recommender system based on user's knowledge background. In *Knowledge science, engineering and management: 14th international conference, Proceedings, Part I* (pp. 140–153). Springer.

Update

Computers and Education: Artificial Intelligence

Volume 6, Issue , June 2024, Page

DOI: https://doi.org/10.1016/j.caeai.2024.100235

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence

Erratum to "ConceptGCN: Knowledge concept recommendation in MOOCs based on knowledge graph convolutional networks and SBERT" [Computers and Education: Artificial Intelligence 6 (June 2024) 100193]

Rawaa Alatrash *, Mohamed Amine Chatti, Qurat Ul Ain, Yipeng Fang, Shoeb Joarder, Clara Siepmann

Social Computing Group, Faculty of Computer Science, University of Duisburg-Essen, Germany

"When this article was first published, the associated Ethical Statement was inadvertently omitted. This has now been added to the article, in compliance with the journal's policy on research involving human subjects.

The publisher would like to apologise for any inconvenience caused.

DOI of original article: https://doi.org/10.1016/j.caeai.2023.100193.

* Corresponding author.
E-mail address: rawaa.alatrash@stud.uni-due.de (R. Alatrash).

https://doi.org/10.1016/j.caeai.2024.100235

Available online 17 May 2024