Heterogeneous Information Enhanced Prerequisite Learning in Massive Open Online Courses

Tianqi Wang
University at Buffalo
Buffalo, NY, USA
twang47@buffalo.edu

Fenglong Ma
The Pennsylvania State University
University Park, PA, USA
fenglong@psu.edu

Yaqing Wang
Purdue University
West Lafayette, IN, USA
wang5075@purdue.edu

Jing Gao
Purdue University
West Lafayette, IN, USA
jinggao@purdue.edu

2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00155

***Abstract*—The knowledge concept prerequisites describing the dependencies are critical for fundamental tasks such as material recommendations and there are a huge amount of concepts in Massive Open Online Courses (MOOCs). Thus it is necessary to develop automatic prerequisite relation annotation methods. Recently, a few methods have shown their effectiveness in discovering knowledge concept prerequisites in Moocs automatically. However, they suffer from two common issues, i.e., knowledge concepts are not thoroughly learnt, and informative supervision sources are ignored. To overcome these issues, we propose an end-to-end framework to incorporate the rich heterogeneous information in MOOCs, including the semantic, contextual and structural information of the learning materials as well as student video watching behaviors. Such useful information is not only used to derive entity representations but also as supervision to improve the prerequisite learning task. Experimental results on two public datasets show that the proposed framework outperforms state-of-the-art baselines in terms of precision, recall and F1 values and improves up to 9% in terms of F1 metrics. Besides, ablation study demonstrates the effectiveness of the proposed framework.**

***Index Terms*—education data mining, deep learning, prerequisite relation prediction**

I. Introduction

The *knowledge concept prerequisite relation*, which is the dependency between a pair of knowledge concepts and determines the order of knowledge concept learning, not only plays an important role in course organizations and learning routine planning but also helps to improve the performance of the fundamental tasks in the education domain such as learning material recommendation [14] in MOOCs. However, because of the huge volume of learning materials on MOOC platforms, it is inefficient to manually label all the prerequisite pairs. Therefore, it is essential to investigate how to find the knowledge concept prerequisites automatically in MOOCs.

Although several approaches such as [1], [2], [4], [10]–[13], [15], [16], [18], [22] have been proposed to solve this challenging task, they still suffer from the following key issues: insufficient learning of knowledge concepts and ignoring informative supervision sources. Existing methods ignore important information such as taxonomy among knowledge concepts, videos and courses and student video watching behavior, which are highly related to the representation learning of knowledge concepts. Moreover, besides labeled data, there is abundant "weak" supervision information such as a knowledge concept appearing in a lecture video and the order of videos in a course, which provides extra supervision for the prerequisite learning task while is ignored in previous methods.

To address the aforementioned issues, we propose to leverage much richer heterogeneous information in MOOCs via the the **H**eterogeous **I**nformation **E**nhanced **P**rerequisite **L**earning (HIEPL) framework to enhance entity representation learning and introduce supplementary supervisions to further improve the performance of the prerequisite learning task. The HIEPL framework is able to fuse the heterogeneous information effectively and jointly learn the representations of knowledge concepts, lecture videos and courses. In particular, the HIEPL framework consists of five components, including *text encoding*, *hierarchical co-representation learning*, *order aware representation learning*, *heterogeneous graph based reasoning*, and *multitask prerequisite relation learning*. The first four components are used for entity representation learning, and the last one is used to classify the knowledge concept pairs with auxiliary supervisions. Experimental results on two real-world datasets show that the proposed HIEPL framework outperforms state-of-the-art baselines, which demonstrate the effectiveness of the proposed HIEPL framework by incorporating the heterogeneous information into the prerequisite learning task. The contributions of this paper can be summarized as:

- We demonstrate the importance of utilizing richer heterogeneous information by both fusing the information in representation learning and introducing extra supervisions to enhance the performance of the knowledge concept prerequisite learning task.
- We propose a novel end-to-end deep learning framework HIEPL, which is able to learn the embeddings of different types of entities simultaneously combining the heterogeneous information and take informative supervisions into the model learning.
- Comprehensive experiments are performed on two real-world datasets to evaluate the effectiveness of the proposed HIEPL framework and demonstrate the importance of using heterogeneous information in the prerequisite learning task.

II. Notations and Problem Statement

*Definition 1 (**Knowledge Concept**):* A knowledge concept refers to a key term that describes or summarizes the knowledge [9]. The set of all knowledge concepts can be represented

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00155

as $\mathcal{K}$, and $|\mathcal{K}|$ is the number of all the knowledge concepts. For each knowledge concept $k \in \mathcal{K}$, it has a description. Let l_k represent the number of words of the knowledge concept k, and the the description of knowledge concept k is represented by $\mathbf{d}_k = \{w_k^i\}_{i=1}^{l_k}$, where w_k^i is the i_{th} word in the description.

*Definition 2 (**Lecture Video**):* Each lecture video consists of several knowledge concepts and has its subtitle. Let $\mathcal{V}$ denote the set of all lecture videos and $|\mathcal{V}|$ be the number of all lecture videos. For each lecture video $v \in \mathcal{V}$, we use $\mathbf{d}_v = \{w_v^i\}_{i=1}^{l_v}$ to denote its subtitle, where w_v^i is the i_{th} word of the subtitle $\mathbf{d}_v$, and l_v represents the number of the words of the subtitles of the lecture video v.

*Definition 3 (**Course**):* Each course includes a sequence of lecture videos and an introduction of the course, which briefly summarizes the main contents of the course. We use $\mathcal{C}$ to denote the set of all courses and $|\mathcal{C}|$ to represent the number of all courses. For each course $c \in \mathcal{C}$, $\mathbf{d}_c = \{w_c^i\}_{i=1}^{l_c}$ represents the introduction of course c, where w_c^i denotes the i_{th} word in the course introduction $\mathbf{d}_c$, and l_c is the number of words in the course introduction $\mathbf{d}_c$.

*Definition 4 (**Student Watching Behavior**):* Each student watches a series of course videos, which will generate a watching behavior sequence. $\mathcal{S}$ is used to denote the set of all the students, and $|\mathcal{S}|$ is the number of students. For each student $s \in \mathcal{S}$, the sequence of videos watched by the student is denoted as $\mathbf{v}_s = \{v_s^i\}_{i=1}^{l_s^v}$, where $v_s^i \in \mathcal{V}$ represents the i_{th} video watched by the student s, and l_s^v denotes the number of videos watched by the student s.

*Definition 5 (**Knowledge Concept Occurrence**):* For each video $v \in \mathcal{V}$, the sequence of knowledge concepts occurred in the video $\mathbf{k}_v$ can be described as $\mathbf{k}_v = \{k_v^i\}_{i=1}^{l_v^k}$, where k_v^i is the i_{th} knowledge concept in the video v, and l_v^k is the number of total knowledge concept occurrence in the video v.

*Definition 6 (**Lecture Video Occurrence**):* For each course $c \in \mathcal{C}$, the lecture video series $\mathbf{v}_c$ can be denoted as $\mathbf{v}_c = \{v_c^i\}_{i=1}^{l_c^v}$, where v_c^i is the i_{th} video occurred in course c and l_c^v is the number of videos included in course c.

*Definition 7 (**Knowledge Concept Prerequisite**):* If understanding the knowledge concept k_j should be on the basis of understanding the knowledge concept k_i, then concept k_i is the prerequisite of concept k_j, we use $R(k_i \rightarrow k_j) = 1$ to represent that the concept k_i is the prerequisite of the knowledge concept k_j. $\mathcal{R} = \{R(k_i \rightarrow k_j)\}$ is used to denote the known training knowledge concept prerequisite set.

*Problem 1 (**Knowledge Concept Prerequisite Learning**):* Given the knowledge concept set $\mathcal{K}$, the lecture video set $\mathcal{V}$, the course set $\mathcal{C}$, the video watching sequence set $\{\mathbf{v}_s\}_{s=1}^{|S|}$, the knowledge concept sequence set $\{\mathbf{k}_v\}_{v=1}^{|V|}$, the video occurrence sequence set $\{\mathbf{v}_c\}_{c=1}^{|C|}$, the known training knowledge concept prerequisite set $\mathcal{R}$ and two knowledge concepts: $k_1, k_2 \in \mathcal{K}$, the goal is to predict if $R(k_1 \rightarrow k_2) = 1$.

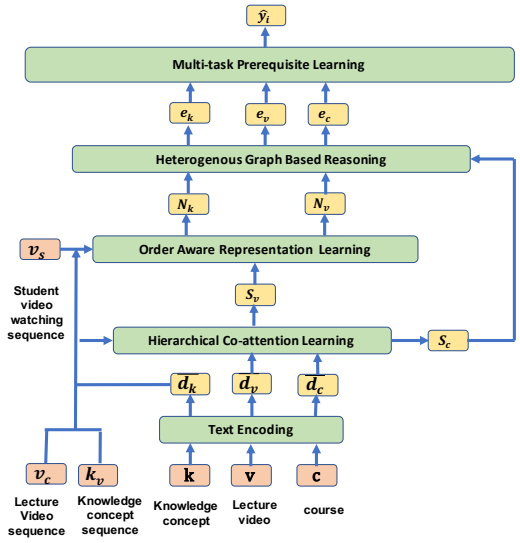

Fig. 1. The overview of the architecture of the proposed HIEPL framework

III. Methodology

A. Overview

The architecture of the proposed HIEPL framework is shown in Figure 1, which consists of five components: a text encoding component, a hierarchical co-representation learning component, an order aware representation learning component, a heterogeneous graph based reasoning component, and a multi-task prerequisite learning component. The first four components are used to fuse the heterogeneous information to learn the representations of different kinds of entities, and the last component is used to determine if one concept from the given concept pair is the prerequisite of the other or not.

B. Text Encoding

To obtain the vector representations of the textual materials, we adopt a pre-trained BERT [3] as the feature extractor. For a given sentence $\mathbf{d} = \{w_i\}_{i=1}^{l}$, where w_i refers to the i_{th} word in the sentence, and l is the number of words, its vector representation extracted by BERT [3] as follows:

$$\tilde{\mathbf{d}} = f_{BERT}(\mathbf{d}), \tag{1}$$

where $f_{BERT}(\cdot)$ represents the average pooling of the output of the last layer of the BERT [3] model, $\tilde{\mathbf{d}} \in \mathbb{R}^{d_b}$ is the outputted vector representations of the sentence $\mathbf{d}$, and d_b is the dimensionality of the vector.

The outputted vector of the BERT model is fed into a fully connected layer to perform dimension reduction:

$$\overline{\mathbf{d}} = FC_t(\tilde{\mathbf{d}}) = \mathbf{W}_t\tilde{\mathbf{d}} + \mathbf{b}_t, \tag{2}$$

where $FC_t(\cdot)$ denotes the fully connected layer function, $\mathbf{W}_t \in \mathbb{R}^{d_b \times h}$ and $\mathbf{b}_t \in \mathbb{R}^h$ are the weight matrix and bias vector need to be learned during the training, and h represents the dimension of the outputted text embedding.

Since we take the heterogeneous information as the input, for the text of the knowledge concept k, the lecture video v and

the course c, their corresponding vector representations can be derived as: $\overline{\mathbf{d}_k} = FC_t(f_{BERT}(\mathbf{d}_k))$, $\overline{\mathbf{d}_v} = FC_t(f_{BERT}(\mathbf{d}_v))$, and $\overline{\mathbf{d}_c} = FC_t(f_{BERT}(\mathbf{d}_c))$, respectively.

C. Hierarchical Co-representation Learning

To capture the inner relations among knowledge concepts, lecture videos and courses. we adopt the co-attention mechanism [23] between the knowledge concepts and the lecture videos and between the lecture videos and the courses.

Given a knowledge concept occurrence sequence $\{k_v^i\}_{i=1}^{l_v^k}$ in lecture video v, we can derive the knowledge concept sequence aware video representation, which can fuse the knowledge concept contextual information and the lecture video v representation using the defined $Coatt(\cdot)$ [1] function in [23]. First, we get the sequence representations $\mathbf{H}_v^k \in \mathbb{R}^{l_v^k \times 2h}$ by feeding it into a bidirectional Long Short-term Memory (BiLSTM) [1] [6],

$$\mathbf{H}_v^k = BiLSTM_c(\mathbf{k}_v) = BiLSTM_c(\{\overline{\mathbf{d}_k^i}\}_{i=1}^{l_v^k}), \quad (3)$$

where $\overline{\mathbf{d}_k^i}$ is the text embedding of $k_v^i \in \mathbf{k}_v$. Similarly, for the video v, which also can be considered as a sequence, we can obtain its representation $\mathbf{H}_v \in \mathbf{R}^{1\times 2h}$ via $BiLSTM_c$:

$$\mathbf{H}_v = BiLSTM_c(\overline{\mathbf{d}_v}). \quad (4)$$

Then $\mathbf{H}_v$ and $\mathbf{H}_v^k$ are fed into the $Coatt(\cdot)$ function to derive the co-representation of the lecture video v:

$$\mathbf{U}_v = Coatt(\mathbf{H}_v, \mathbf{H}_v^k), \quad (5)$$

where $\mathbf{U}_v \in \mathbb{R}^{1\times 4h}$ is the knowledge concept sequence aware representation of the lecture video v.

By combining the knowledge concept aware and textual representations of video v, a video summary vector $\mathbf{S}_v \in \mathbb{R}^{1\times h}$ can be derived as:

$$\mathbf{S}_v = FC_s([\mathbf{U}_v \oplus \overline{\mathbf{d}_v}]) = ([\mathbf{U}_v \oplus \overline{\mathbf{d}_v}])\mathbf{W}_s + \mathbf{b}_s, \quad (6)$$

where FC_s denotes the fully connected layer, and $\mathbf{W}_s \in \mathbb{R}^{5h\times h}$ and $\mathbf{b}_s \in \mathbb{R}^{1\times h}$ are the learnable weights and bias.

Similarly, the course summary vector of a course c can be derived given the lecture video sequence $\mathbf{v}_c$. The derivation of the summary vector $\mathbf{S}_c$ is shown as follows:

$$\begin{aligned} &\mathbf{S}_c = FC_s([\mathbf{U}_c \oplus \overline{\mathbf{d}_c}]), \mathbf{U}_c = Coatt(\mathbf{H}_c, \mathbf{H}_c^v), \\ &\mathbf{H}_c^v = BiLSTM_c(\mathbf{v}_c), \mathbf{H}_c = BiLSTM_c(\overline{\mathbf{d}_c}). \end{aligned} \quad (7)$$

D. Order Aware Representation Learning

The occurrence order of the knowledge concepts has been found to be an important indicator, which provides important information in the prerequisite learning [16]. Thus, we learn the order aware representation of the knowledge concepts and the lecture videos by using both the occurrence order in the learning materials and the student video watching behaviors.

[1]The details can be found in the Appendix at https://www.dropbox.com/sh/z9plfdqg6xg10fy/AACL0G0HrE5QqdOiIOYBd7P0a?dl=0

Knowledge Concepts. To learn the order aware representations of the knowledge concepts, the knowledge concept occurrence sequences $\mathbf{k}_v$ are put into a BiLSTM to obtain an order aware embedding first, whose contextual aware representation matrix can be represented as:

$$\mathbf{O}_v = FC_o(BiLSTM_o^k(\mathbf{k}_v)), \quad (8)$$

where the i_{th} knowledge concept k_v^i in the sequence $\mathbf{k}_v$ is represented by its textual embedding $\overline{\mathbf{d}_k}$. $\mathbf{O}_v \in \mathbb{R}^{l_v\times h}$ denotes the contextual matrix which is the concatenation of the hidden state vectors of the sequence $\mathbf{k}_v$ outputted by the $BiLSTM_o^k$. $FC_o(\cdot)$ represents the fully connected layer.

Let $\mathbf{M}_k$ represent a matrix that is the concatenation of all the contextual vectors of knowledge concept k extracted from $\{\mathbf{O}_v : v \in \mathcal{V}\}$, then the contextual aware representation of the knowledge concept can be defined as:

$$\mathbf{N}_k = att(\mathbf{M}_k, \overline{\mathbf{d}_k}), \quad (9)$$

where $\mathbf{N}_k \in \mathbb{R}^{1\times h}$ denotes the contextual aware representation of the knowledge concept k and $att(\cdot)$ is the attention operation defined in Eq. 10.

$$att(\mathbf{M}_{att}, \mathbf{v}_{att}) = softmax(\frac{\mathbf{v}_{att}\mathbf{W}_{\mathbf{att}}\mathbf{M}_{\mathbf{att}}^{\mathbf{T}}}{\sqrt{h}})\mathbf{M}_{att}, \quad (10)$$

where $\mathbf{M}_{att} \in \mathbb{R}^{N_M\times h}$ represents a matrix concatenated by N_M vectors of size h and $\mathbf{v}_{att} \in \mathbf{R}^{1\times h}$ is a vector. $\mathbf{W}_{att} \in \mathbf{R}^{h\times h}$ is the weight matrix needs to be learned.

Lecture Videos. Similar with learning the order aware knowledge concept representation, the order aware video representation of lecture video v using only lecture video occurrence sequence $\mathbf{v}_c$ can be represented as:

$$\begin{aligned} \mathbf{O}_c &= FC_o(BiLSTM_o^v(\mathbf{v}_c)), \\ \mathbf{N}_v^c &= att(\mathbf{M}_v^c, \mathbf{S}_v), \end{aligned} \quad (11)$$

where $\mathbf{O}_c$ is the contextual matrix of the sequence which is the concatenation of the hidden state vectors of the sequence $\mathbf{v}_c$ outputted by the $BiLSTM_o^v$. The input embedding for the $BiLSTM_o^v$ of the video v_c^i in the sequence is $\mathbf{S}_v$. $\mathbf{M}_v^c$ represents the matrix which is the concatenation of contextual vectors of video v extracted from $\{\mathbf{O}_c : c \in \mathcal{C}\}$ and $\mathbf{N}_v^c$ is the contextual aware representation of the lecture video v.

Since the order of lecture videos is well organized for learning, we use the lecture video occurrence sequence as the reference sequence and the order aware video representations derived by such sequences as the reference representations. The order aware video representation derived by the video order is used to calculate the attention weight during the process of deriving the order aware video representation by the student behaviors and the final order aware video representation.

Let $\mathbf{M}_v^b$ denote the concatenation of all the vectors of video v in the student video watching behavior sequences outputted by the $FC_o((BiLSTM_o^v())$ function using $\mathbf{N}_v^c$ as the input of v. Then the order aware representation of lecture video v derived by the student video learning behavior sequences can be calculated by:

$$\mathbf{N}_v^b = att(\mathbf{M}_v^b, \mathbf{N}_v^c), \quad (12)$$

where $\mathbf{N}_v^b \in \mathbb{R}^{1\times h}$ is the order aware representation of lecture video v derived by the student behaviors. Thus, the final order aware representation of the video v can be represented as:

$$\mathbf{N}_v = att([\mathbf{N}_v^c \oplus \mathbf{N}_v^b], \mathbf{N}_v^c) \tag{13}$$

where $\mathbf{N}_v \in \mathbb{R}^{1\times h}$ is the order aware representation of the lecture video v.

E. Heterogeneous Graph Based Reasoning

To further exploit the useful structural information among knowledge concepts, we construct a heterogeneous graph and apply graph neural network to pass messages among nodes.

Let $\mathcal{G} =< \mathcal{N}, \mathcal{E} >$ denotes the heterogeneous graph.

$$\begin{aligned} \mathcal{N} &= \mathcal{K} \cup \mathcal{V} \cup \mathcal{C}, \\ \mathcal{E} &= \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3 \cup \mathcal{E}_4 \cup \mathcal{E}_5 \cup \mathcal{E}_6. \end{aligned} \tag{14}$$

$\mathcal{N}$ is the node set and $\mathcal{E}$ represents the edge set which is the union of the video occurrence, video order, knowledge concept occurrence, knowledge concept order, student video watching order and knowledge concept prerequisites [1].

To enable the information propagates over the constructed heterogeneous graph, the message passing mechanism in [5] is applied. The message passing mechanism includes two steps: aggregation and combination. The aggregation step aggregates the information of the neighbors of a node $n \in \mathcal{N}$, which can be represented as:

$$\overline{\mathbf{e}_n^i} = \sum_{r=1}^{6} \frac{1}{|\mathcal{N}_n^r|} \sum_{j\in\mathcal{N}_n^r} FC_r(\mathbf{e}_j^{i-1}), \tag{15}$$

where $\overline{\mathbf{e}_n^i} \in \mathbb{R}^{1\times h}$ is the aggregated representation of node n in layer i. $\mathcal{N}_n^r$ denotes the set of neighbour nodes of the node n in $\mathcal{E}_r$, where $r \in \{1,2,3,4,5,6\}$. $FC_r(\cdot)$ is a fully connected layer.

The combination step combines the representation of the aggregated representation of node n in layer i and the representation of that node in layer $i-1$ to obtain the embedding of node n in layer i, which can be formulated as:

$$\mathbf{e}_n^i = FC_c(\mathbf{e}_n^{i-1}) + \overline{\mathbf{e}_n^i}, \tag{16}$$

where $\mathbf{e}_n^i \in \mathbb{R}^{1\times h}$ denotes the node embedding of node n in layer i. $FC_c(\cdot)$ is a fully connected layer for transformation. The final representation of the node n can be obtained after 3 times message passing, which is: $\mathbf{e}_n = \mathbf{e}_n^3$. Different types of nodes in the graph $\mathcal{G}$ have different initial representations ($\mathbf{e}_n^0$) described as:

$$\mathbf{e}_n^0 = \begin{cases} \mathbf{N}_k, & n \in \mathcal{K}, \\ \mathbf{N}_v, & n \in \mathcal{V}, \\ \mathbf{S}_c, & n \in \mathcal{C}. \end{cases} \tag{17}$$

F. Multitask Prerequisite Learning

Given two knowledge concepts k_1 and k_2, a two-layer fully connected network can be used to determine if k_1 is the prerequisite of k_2, which is represented as:

$$\hat{R}(k_1 \to k_2) = FC_p([\mathbf{e}_{k_1} \oplus \mathbf{e}_{k_2}]) \tag{18}$$

$\hat{R}(k_1 \to k_2)$ is the predicted probability that k_1 is a prerequisite of k_2. $FC_p(\cdot)$ is a two-layer fully connected network where the activation functions are the ReLU and the sigmoid function separately. $\mathbf{e}_{k_1}$ and $\mathbf{e}_{k_2}$ are the knowledge concept representations of k_1 and k_2 defined in Eq. 16 with $i = 3$.

To learn the parameters in the proposed model, we introduce three auxiliary tasks in addition to the knowledge concept prerequisite learning. Those three auxiliary tasks are:

- Predict if a video appears in a course.
- Predict if a video appears ahead the other video.
- Predict if a knowledge concept appears in a video.

Let $\hat{R(i)}_j$ denote the predicted result of pair j of the Task i, which can be formulated as:

$$\hat{R}(i)_j = FC_p^i([\mathbf{e}_{j_1} \oplus \mathbf{e}_{j_2}]), \tag{19}$$

where $\mathbf{e}_{j_1}$ and $\mathbf{e}_{j_2}$ are the representations of the entity pair defined in Eq. 16 with $i = 3$. $FC_p^i(\cdot)$ share the same architecture as $FC_p(\cdot)$. In addition, the parameters of the first layer are shared between $FC_p^1(\cdot)$ and $FC_p^3(\cdot)$ as well as between $FC_p(\cdot)$ and $FC_p^2(\cdot)$.

The average cross-entropy defined in Eq. 20 is used as the loss function for each task.

$$\mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}}) = -\frac{1}{|\mathcal{Y}|} \sum_{y_j\in\mathcal{Y}} (y_j log(\hat{y}_j) + (1-y_j) log(1-\hat{y}_j), \tag{20}$$

where $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ are the ground truth label set and the predicted probability set for each task. y_j and $\hat{y}_j$ denote the ground truth label and predicted probability of the sample j. $|\mathcal{Y}|$ represents the number of samples in set $\mathcal{Y}$.

The total loss is formulated as the weighted sum of the losses for those tasks, which can be denoted as:

$$\mathcal{L} = \alpha\mathcal{L}_p + \frac{1-\alpha}{3} \sum_i \mathcal{L}_i, \tag{21}$$

where $\mathcal{L}_p$ and $\mathcal{L}_i$ denotes the loss represented by Eq. 20 for the knowledge concept prerequisite learning task and the auxiliary Task i ($i = \{1,2,3\}$). $0 \le \alpha \le 1$ is the balance parameter. The model parameters are learned during the training phase by optimizing the loss function $\mathcal{L}$.

IV. Experiments

A. Datasets

We construct two datasets for evaluation from the public available MOOCCube dataset [22][2]. We divide the labeled prerequisites into two subsets according to the domains they belong to and obtain two datasets with positive prerequisite samples in the mathematics and computer science. The reverse of the positive pairs are considered as the negative samples. Since the prerequisite relations are sparse, we also randomly sample the knowledge concept pairs in each of the domain to obtain more negative samples with a pre-defined positive sample ratio. The statistics of the original MOOCCube and the two constructed datasets are shown in the Appendix [1].

[2]http://moocdata.cn/data/MOOCCube

TABLE I
MODEL PERFORMANCE ON THE PREREQUISITE PREDICTION TASK

	CS			Math		
Model	recall	precision	F1	recall	precision	F1
MOOC-LR	0.467	0.640	0.540	0.482	0.650	0.553
MOOC-XG	0.500	0.563	0.530	0.510	0.565	0.536
DNN	0.542	0.580	0.560	0.579	0.579	0.579
PREREQ	0.676	0.580	0.624	0.668	0.593	0.628
HIEPL-B	0.680	0.660	0.670	0.678	0.660	0.669
HIEPL-C	0.678	0.654	0.665	0.680	0.665	0.672
HIEPL-S	0.542	0.580	0.560	0.575	0.573	0.574
HIEPL	**0.688**	**0.676**	**0.682**	**0.694**	**0.680**	**0.687**

B. Methods for Comparison

The proposed HIEPL is compared with the several methods [1] to evaluate its effectiveness. Since the proposed framework works in a supervised learning setting, only the supervised knowledge concept prerequisite learning models proven to achieve better performance are compared. The models we compared include: MOOC-LR [16], MOOC-XG [16], PREREQ [18] and Deep Neural Network (DNN). To investigate the effects of different components of the HIEPL, we also compare it with its three variations: HIEPL-B, HIEPL-C and HIEPL-S by removing the student video watching behaviors, co-representation learning and supplementary supervision component in the HIEPL respectively.

C. Experimental Setting

The HIEPL framework is implemented using PyTorch [17]. The model parameters are randomly initialized, and then the Adam optimizer [7] is used to learn the model parameters with the learning rate of 0.0001. The batch size is set to 32 for training and 128 for testing, and the maximum number of epoch is 500. To avoid the issue of overfitting, dropout [20] with a rate of 0.2 and early stop are applied. The size of the outputted representation vectors by the text encoder is set to 32 ($h = 32$). The balance parameter in Eq. 21 is set to 0.5 ($\alpha = 0.5$) [3] .

To evaluate the effectiveness of the proposed model, we construct the datasets in the two domains by sampling the negative samples with a positive sample ratio of 0.2. For each positive sample, we need to randomly sample three negative pairs in addition to the reverse pair of it. For each domain, we sample three different datasets. Five-fold cross validation is used to evaluate the performance of those models. The performance of the models are measured by precision, recall and F1 value. The average metrics of the fifteen runs in each domain are reported.

D. Experimental Results

The experimental results are shown in Table. I. We can observe that *HIEPL* outperforms all the baselines. Although the *MOOC-LR* and the *MOOC-XG* model consider features in three different aspects: semantic, context and structure, the performance of those two models is still among the worst ones. The inferior performance of those two models may indicate the limited ability of manually designed features to capture useful information related with the prerequisite learning task.

The performance of the *DNN* model is better than *MOOC-LR* and *MOOC-XG*, which demonstrates that the pre-trained word embeddings can capture the signal related with the prerequisite learning. However, it is beaten by the *PREREQ* model, which also uses the textual information. The reason may be that the pre-trained embeddings are trained on a large scale corpus with great generalization ability on different tasks but not specifically optimized for the prerequisite learning task. Thus, those embeddings may capture the shared characteristics of different NLP tasks such as the similarity of the input, which is also informative in the prerequisite learning task. However, it can not capture the informative signal specific of the prerequisite learning task such as the dependencies utilized in the *PREREQ*.

As for the *PREREQ*, it learns the knowledge concept representations by jointly modeling the dependencies and semantic information and achieves the best performance among all those four baselines. The superior performance of *PREREQ* among all the baselines shows that both the semantic information and dependencies are useful for prerequisite learning and fusing information from different sources effectively helps improve the knowledge concept prerequisite learning task.

E. Ablation Study

To investigate the effects of different parts of the proposed HIEPL framework, we perform ablation study by comparing the HIEPL framework with its three variations. The results are shown in Table. I. The performance of all reduced models degrades compared with the full model, which shows the effects of related parts in the proposed HIEPL framework.

The decreased performance of the HIEPL-B compared with HIEPL shows the impacts of the student watching behaviors in the prerequisite learning task, where the student video watching behaviors can help bridge the knowledge concepts in different courses. The behavior patterns can also help reveal the prerequisite relations and further improve the performance of the knowledge concept prerequisite learning task. The hierarchical co-representation learning can help transfer and fuse information among courses, lecture videos and knowledge concepts and therefore benefit the prerequisite learning task, which may account for the worse performance of the HIEPL-C compared with the HIEPL framework. The HIEPL-S is the worst model among those three reduced models, which demonstrates the importance and necessity of utilizing auxiliary supervision to optimize the HIEPL framework with only a small number of labeled prerequisites.

V. RELATED WORK

The prerequisite mining methods can be divided into statistics based methods and learning based methods. In the statis-

[3]The implementation details can be found in the Appendix. Both the Appendix and the code are available at https://www.dropbox.com/sh/z9plfdqg6xg10fy/AACL0G0HrE5QqdOiIOYBd7P0a?dl=0

tics based methods, such as [1], [2], metrics are manually designed and thresholds are applied to determine the prerequisite relationship. Although using different features and ways to make decision, the workflow of existing learning learning based prerequisite mining methods [10]–[13], [15], [16], [18], [19] can be summarized as two steps: the knowledge concept representation learning step and then prerequisite relationship determination step. Recently, several publicly available datasets [4], [11], [22] are released, which also help advance the development of supervised prerequisite learning methods. Different from the aforementioned prerequiste learning methods, the proposed HIEPL framework can automatically learn the knowledge concept representations by fusing rich information from heterogeneous sources in an end-to-end manner and utilize the heterogeneous information as extra supervision.

Our work is also related to the co-attention and graph neural network. The co-representation learning is inspired by the method in [23] which has been proven effective in capturing the relevant information. The graph neural networks [5], [8], [21] have achieved great success by its ability of transforming information among nodes in a graph to meaningful representations. In the proposed HIEPL framework, we use the message passing mechanism proposed in [5] to transform the information among different nodes.

VI. Conclusions

Identifying the knowledge concept prerequisite relations is important for many fundamental tasks in MOOCs such as learning material recommendation. Although some efforts have been devoted to this topic, the existing methods still suffer from two issues: insufficient learning of knowledge concepts and ignoring informative supervision sources, which can be addressed by utilizing heterogeneous information, including the semantic, contextual and structural information as well as student behavior. To this end, we propose a novel end-to-end HIEPL framework, which can fully exploit the heterogeneous information in context of obtaining the entity representations and additional sources for supervision. Experiments conducted on two real-world datasets show that the proposed HIEPL framework outperforms the baselines in terms of multiple evaluation metrics. These results confirm the effectiveness of the proposed framework of incorporating the heterogeneous information into the prerequisite learning task.

VII. Acknowledgement

This work is sponsored by NSF IIS-2226108 and NSF IIS-2141037. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] F. ALSaad, A. Boughoula, C. Geigle, H. Sundaram, and C. Zhai. Mining mooc lecture transcripts to construct concept dependency graphs. In *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[2] M. C. Aytekin, S. Räbiger, and Y. Saygin. Discovering the prerequisite relationships among instructional videos from subtitles. In *Proceedings of the 13th International Conference on Educational Data Mining, 2020*.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[4] A. R. Fabbri, I. Li, P. Trairatvorakul, Y. He, W. T. Ting, R. Tung, C. Westerfield, and D. R. Radev. Tutorialbank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation. *arXiv preprint arXiv:1805.04617*, 2018.

[5] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proceedings of the Advances in neural information processing systems*, 2017.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8), 1997.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5), 2012.

[10] I. Li, A. Fabbri, S. Hingmire, and D. Radev. R-vgae: Relational-variational graph autoencoder for unsupervised prerequisite chain learning. *arXiv preprint arXiv:2004.10610*, 2020.

[11] I. Li, A. R. Fabbri, R. R. Tung, and D. R. Radev. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[12] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles. Investigating active learning for concept prerequisite learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[13] C. Liang, J. Ye, Z. Wu, B. Pursel, and C. Giles. Recovering concept prerequisite relations from university course dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[14] Q. Liu, S. Tong, C. Liu, H. Zhao, E. Chen, H. Ma, and S. Wang. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[15] W. Lu, Y. Zhou, J. Yu, and C. Jia. Concept extraction and prerequisite relation learning from educational data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 01, 2019.

[16] L. Pan, C. Li, J. Li, and J. Tang. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 8026–8037. Curran Associates, Inc., 2019.

[18] S. Roy, M. Madhyastha, S. Lawrence, and V. Rajan. Inferring concept prerequisite relations from online educational resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[19] M. Sayyadiharikandeh, J. Gordon, J.-L. Ambite, and K. Lerman. *Finding Prerequisite Relations Using the Wikipedia Clickstream*. Association for Computing Machinery, New York, NY, USA, 2019.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 2014.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[22] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, J. Luo, C. Wang, L. Hou, J. Li, Z. Liu, et al. Mooccube: A large-scale data repository for nlp applications in moocs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[23] V. Zhong, C. Xiong, N. Keskar, and R. Socher. Coarse-grain fine-grain coattention network for multi-evidence question answering. *arXiv preprint arXiv:1901.00603*, 2019.