Exploring the Effectiveness of Student Behavior in Prerequisite Relation Discovery for Concepts

Anonymous ACL submission

Abstract

What knowledge should a student grasp before beginning a new MOOC course? This question can be answered by discovering prerequisite relations of knowledge concepts. In recent years, researchers have devoted intensive efforts to detecting such relations by analyzing various types of information. However, there are still a few explorations of utilizing student behaviors in this task. In this paper, we investigate the effectiveness of student behaviors in prerequisite relation discovery for course concepts. Specifically, we first construct a novel MOOC dataset to support the study. We then verify the effectiveness of student behaviors via serving as additional features for existing prerequisite relation discovery models. Moreover, we explore to better utilize student behaviors via graph-based modeling. We hope our study could call for more attention and efforts to explore the student behavior for prerequisite relation discovery.

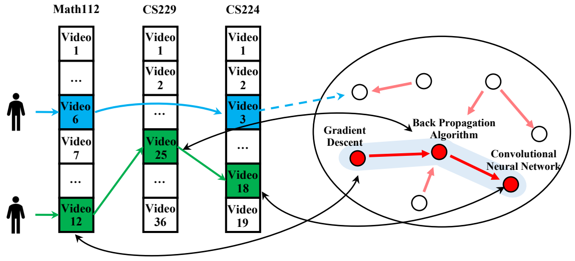

Figure 1: An application example of prerequisite relations. A student who wants to study "Convolutional Neural Network" can be suggested to follow the prerequisite chain (green path) to learn "Gradient Descent" and "Back Propagation Algorithm" first.

1 Introduction

Since the first edition of Robert Gagne's *Principles of Instructional Design* (Gagne and Briggs, 1974) came out, many efforts from pedagogy have suggested that students should grasp prerequisite knowledge before moving forward to learn subsequent knowledge. Such prerequisite relations are described as the dependence among knowledge concepts and are crucial for students to learn, organize, and apply knowledge (Parkay and Hass, 1999). Figure 1 shows an example of the prerequisite relations in Massive Open Online Courses (MOOCs). For a student who wants to learn the concept "Convolutional Neural Network" (CS224:video18), he/she is expected to have had the knowledge of its prerequisite concepts ("Gradient Descent" and "Back Propagation Algorithm").

In the era of intelligent education, prerequisite relations play an essential role in a series of educational applications such as curriculum planning (Agrawal et al., 2015), reading list generation (Gordon et al., 2016), etc. As the example of Figure 1, with explicit prerequisite relations among concepts (red), a coherent and reasonable learning sequence can be recommended to the student (green or blue). However, as the quantity of educational resources proliferates, the explosive growth of knowledge concepts make it expensive and ineffective to obtain fine-grained prerequisite relations by expert annotations (Bergan and Jeska, 1980). Therefore, automatically discovering prerequisite relations becomes a rising topic in recent years.

This task is defined as **Prerequisite Relation Discovery of Concepts in MOOCs**. Despite several attempts on this topic, including extracting such relations from the content of MOOC videos (Pan et al., 2017a; Liang et al., 2018) and the preset orders of MOOC resources (Liang et al., 2017; Roy et al., 2019), it is still far from sufficient to directly apply these methods in the practical applications due to the following challenges.

First, unlike the factual relations of general entities (e.g., *Bill Gates* has the relation `founder` with *Microsoft Inc.*), prerequisite relations are more cognitive than factual, which makes it rarely mentioned explicitly in texts and challenging to be directly captured from MOOC corpus. Second, existing MOOC resources that are considered to be prerequisite clues are noisy, e.g., the order of MOOC videos. As a video usually teaches several concepts, it is common that some of these concepts are not prerequisites to the ones in later videos. Therefore, it is crucial to explore more effective resources to help the discovery of prerequisite relations.

Inspired by the idea from educational psychology that students' learning behaviors are positively related to the cognitive structure of knowledge (Ausubel, 1968), we conduct an investigation on leveraging the *video watching behaviors of students* in the task of discovering prerequisite concepts in MOOCs. For supporting the investigation, we collect student behavior data from real MOOC courses and organize expert annotations to construct a dataset of sufficient and fine-grained prerequisite concept relations. After analyzing typical behavior patterns, we verify this information's effectiveness in improving the existing method.

Furthermore, to explore student behaviors' better modeling, we propose a graph-based solution by building concept graphs from student behaviors and conducting link prediction on them. Experimental results show our proposed method achieves much better performance compared with four representative baselines. We also provide several empirical suggestions for research on related topics.

Our contributions include: (1) An investigation on how to leverage student behaviors to extract prerequisite relations of concepts; (2) Proposal of an effective graph-based model for enhancing prerequisite relation discovery with student behaviors in MOOCs; (3) A manually annotated benchmark of fine-grained prerequisite concepts from real courses of MOOC websites[1].

2 Related Work

Our work mainly follows the efforts in discovering *prerequisite relations* among course concepts, aiming to detect the dependence of concepts from MOOC resources. The task of identifying prerequisite relations originates from educational data mining, which could help in automatic curriculum planning (Parkay and Hass, 1999) and other educational applications (Romero and Ventura, 2007). In the area of education, early works discover general prerequisite structures from students' test performance (Vuong et al., 2011; Scheines et al., 2014; Huang et al., 2015), and these early efforts have mainly focused on discovering the dependence among courses or knowledge units. Talukdar and Cohen (2012) and Liang et al. (2015) further propose to learn more fine-grained prerequisite relations, i.e., the prerequisite relations among concepts. In recent years, detecting prerequisite concepts from courses (especially online courses) has become a rising research topic. Researchers explore various kinds of methods from matrix optimization (Liang et al., 2017), feature engineerings (Pan et al., 2017a) to neural networks (Roy et al., 2019) to consider the static information of MOOCs (course/video) as indispensable clues for discovering such relations.

There are also some attempts to extract prerequisite relations from other resources, e.g., paper citation networks (Gordon et al., 2016) and textbooks' unit sequences and titles (Labutov et al., 2017). Recently, the user clickstream of Wikipedia pages (Sayyadiharikandeh et al., 2019) are also proven to indicate concept dependence. This inspires us to improve prerequisite prediction by considering the user behaviors in MOOCs, which contains more behavior details and is relevant to the cognitive learning process.

3 Problem Formulation

In this section, we give some basic definitions and formulate the problem of discovering prerequisite relations among course concepts in MOOCs.

A MOOC corpus is composed of courses from MOOCs, denoted as $\mathcal{M} = \{\mathcal{C}_i\}_{i=1}^{|\mathcal{M}|}$, where $\mathcal{C}_i$ indidates the i-th course. Each course includes a sequence of videos, i.e., $\mathcal{C}_i = [v_{ij}]_{j=1}^{|\mathcal{C}_i|}$, where v_{ij} refers to a video with its subtitles from the course. And the **Student Behavior** that we use in this paper is the Video Watching Behaviors $\mathcal{S} = \{(u, v, t)\}$, where each behavior records student $u \in U$ started to watch the video v at time t, and U is the set of all students.

Course dependence is defined as a prerequisite relation between courses (Liang et al., 2017), denoted as $\mathcal{D} = \{(\mathcal{C}_i, \mathcal{C}_j)\,|\mathcal{C}_i, \mathcal{C}_j \in \mathcal{M}\}$, which indicates that course $\mathcal{C}_i$ is a prerequisite course of $\mathcal{C}_j$. This information is often provided by the teachers when setting up new courses.

Course Concepts are the subjects taught in a course (e.g., "LSTM" is a concept of the Deep Learning course). We respectively denote the concepts of a certain video, a course and the whole MOOC corpus as $\mathcal{K}^v$, $\mathcal{K}^c$ and $\mathcal{K}$. The video con-

[1] The dataset will be publicly available after review.

cepts $\mathcal{K}^v_{ij} = \left\{c_1, ..., c_{|\mathcal{K}^v_{ij}|}\right\}$ is the concepts taught in course video v_{ij}. As a course is consist of several videos, the course concept $\mathcal{K}^c_i = \mathcal{K}^v_{i1} \cup ... \cup \mathcal{K}^v_{i|\mathcal{C}_i|}$. And all the concepts of the MOOC corpus is $\mathcal{K} = \mathcal{K}^c_1 \cup ... \cup \mathcal{K}^c_{|\mathcal{M}|}$.

Discovering prerequisite relation of course concepts in MOOCs is formulated as: Given the MOOC corpus $\mathcal{M}$, course dependence $\mathcal{D}$, student behaviors $\mathcal{S}$ and the corresponding course concepts $\mathcal{K}$, the objective is to learn a function $\mathcal{L} : \mathcal{K}^2 \rightarrow \{0, 1\}$ that maps a concept pair (c_a, c_b), where $c_a, c_b \in \mathcal{K}$, to a binary class that indicates whether c_a is a prerequisite concept of c_b.

4 The MOOC Dataset

Although there are a few datasets for mining prerequisite relations from online courses (Pan et al., 2017a; Li et al., 2019; Yu et al., 2020), they still cannot adequately support our investigation due to the following reasons. (1) Lack of student behavioral data: Most of the existing datasets do not collect relevant student behavior data, and such data are difficult to supplement due to accessibility. (2) Sparsity: Datasets with student behavior data, such as MOOCCube (Yu et al., 2020), only use distant supervision methods to automatically label prerequisite relationships. This makes its high-confidence prerequisite relationships too sparse to support fine-grained quantitative analysis.

Therefore, with the consideration of user privacy protection[2], we collect data of courses, videos, and student behaviors from a large MOOC website[3], and organize multi-stage annotations to construct a fine-grained, rich-connectivity prerequisite dataset.

Stage 1: MOOC Information Collection: We select 12 sample courses in three domains to collect information of MOOCs, including "Basic Knowledge of Computer Science" (**CS**), "Programming Languages" (**PL**), and "Artificial Intelligence" (**AI**). These courses are selected because their concepts are highly relevant, lifting the connectivity of course concepts in the dataset. Then we collect course and student data in three steps: (1) downloading all course materials, which include the video orders and subtitles; (2) obtaining the video watching logs of students who participated in these courses during 2017-2019 as user behavior data source, which could help us to infer a student's learning frequency, watching duration, and other information of a particular video; (3) annotating the dependence of courses.

		CS	PL	AI	ALL
#Course		4	4	4	12
#Video		312	222	233	767
#Concept		369	227	377	700
#Pair	*+pos*	672	673	267	1,612
	-neg	1,258	539	218	2,015
#Student		12,094	12,014	3,541	17,587
#Behavior		430,769	337,953	39,136	807,858
Kappa		0.765	0.737	0.769	0.754

Table 1: Statistics of our dataset. As course concepts and students may overlap in different courses, their total number is not a simple numerical addition.

Stage 2: Data Processing: Regarding all the subtitles of selected courses as the MOOC corpus, we employ a widely-used concept extraction method (Pan et al., 2017b) in MOOC-related tasks to obtain concept candidates. For each candidate, two annotators label it as "not course concept" or "course concept", and the disagreements are confirmed by the teacher. Each labeled concept's Wikipedia abstract is dumped as side-information for the reproduction of baseline methods.

Stage 3: Prerequisite Relation Annotation: We manually annotate the prerequisite relations among the labeled course concepts. A critical challenge in the annotation is the giant quantity and sparsity. If the concept number is n, the candidate pair number is $n(n-1)/2$, which requires arduous human labeling work. Therefore, we present a two-step strategy to reduce the workload:

- Step 1: The teacher of the corresponding course leads the annotators to cluster the concepts to several groups, which may maintain possible prerequisite relations. After this step, we get 28 clusters of 700 course concepts, where the largest contains 210 concepts, and the smallest contains 14 concepts. We organize the following annotations within these concept clusters.

- Step 2: We generate the candidate concept pairs within the clusters and sample a small scale of them as golden standard (300). Then we employ them to train existing baselines (i.e. MOOC-RF (Pan et al., 2017b), GlobalF (Liang et al., 2018), PREREQ (Roy et al., 2019) and CPR-Recover) as candidate filters. To ensure the Recall, we only filter out the pair if none of the above classifiers predict it to be a prerequisite, and preserve the remaining pairs into the annotation.

[2] The details of data privacy protection, annotation and quality control can be found in Ethical Section and Appendix.

[3] Anonymous for blind review.

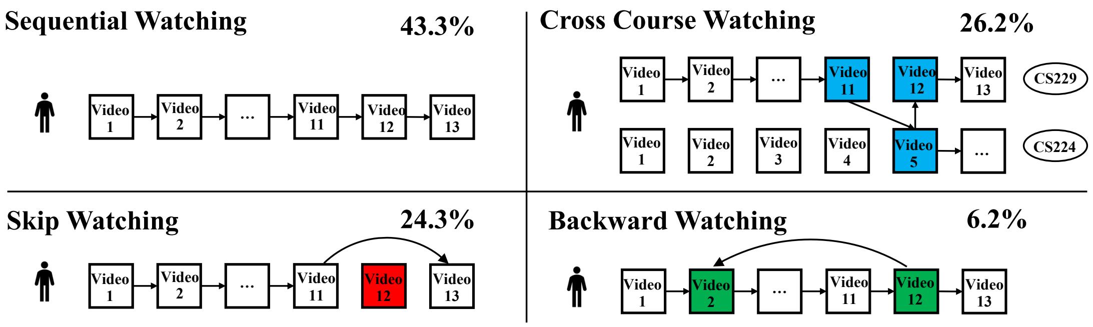

Figure 2: Four typical patterns when a student watches course videos. The figure shows the proportion of video pairs that match each pattern in all behaviors.

In the annotation process, two annotators from corresponding domains are asked to label whether concept A is a prerequisite of concept B, i.e., the annotator need to answer the question of "whether A is helpful for understanding B". A pair is labeled as positive only when the two annotators are in agreement. The statistics of our dataset are shown in Table 1, where *#Course*, *#Video*, *#Concept*, *#Pair*, *#Student* are the number of corresponding items and *#Behavior* is the number of video watching records. The *Kappa* statistics of the inter-annotator agreement is 0.754, showing the reliability of the annotation results??.

5 Effectiveness of Student Behavior

In this section, we first explore whether the student behaviors in MOOCs are useful in the task of discovering prerequisite concepts. To this end, we present a feature-based method to model student behaviors and investigate whether this information can improve the existing methods. The design of our approach is based on the following cognitive learning hypothesis:

Hypothesis. *Students tend to follow the prerequisite cognitive structure to learn new knowledge.*

The hypothesis was proposed in educational psychology (Ausubel, 1968), and was widely applied in prerequisite-driven instructional design (Parkay and Hass, 1999; Romero and Ventura, 2007). We extend this hypothesis by analyzing the clues of the prerequisite concepts implied in student learning orders. Surprisingly, although MOOCs preset the video order, our observation on student behavior data indicates that students often learn MOOC videos in their own orders. As shown in Figure 2, we summarize four typical behavior patterns, and the out-of-pre-order learning behaviors are even more than half of total (56.72%). To leverage student behaviors in this task, we analyze the causes of these patterns and build several features to model prerequisite relations from student behaviors.

5.1 Student Behavior Patterns and Prerequisite Features

We first construct a video watch behavior sequence $\mathcal{S}_u$ for each user u from student behavior record $\mathcal{S}$, where the video watch behaviors are sorted in time order. Comparing the preset video order of each $\mathcal{C}_i$ with $\mathcal{S}_u$, we summarize four typical patterns of students' video watching behaviors: *Sequential Watching*, *Cross Course Watching*, *Skip Watching* and *Backward Watching* as shown in Figure 2. Before introducing the modeling details, we first define the behavior patterns as follows.

Definition 1 (Behavior Pattern). *A behavior pattern $\mathcal{P}$ is formed by one or more video pairs. A video pair (v_i, v_j) belongs to a pattern $\mathcal{P}$ when it matches the corresponding conditions.*

As the student behavior patterns are at *video* level, we infer the prerequisite features of a concept pair $c_a \in \mathcal{K}_i^v$ and $c_b \in \mathcal{K}_j^v$ by considering videos as bags of course concepts, where $\mathcal{K}_i^v$, $\mathcal{K}_j^v$ correspond to the concepts taught in v_i, v_j, and a concept may be taught in more than one videos.

Over 72.5% of the students' behavior records contain all the four typical patterns in Figure 2, indicating that they are not accidental. Therefore, we attempt to speculate the causes of these four patterns from the cognitive perspective, and build prerequisite features $f^{\mathcal{P}}$ to model them.

Sequential Watching Pattern. Sequential watching indicates that a student watches videos in the course's preset video order, which indicates that the concepts taught in these videos are in accordance with the prerequisite cognitive structure. To leverage this pattern, we assign prerequisite feature

$f_1^{\mathcal{P}}$ for the concepts c_a and c_b as:

$$f_1^{\mathcal{P}}(c_a, c_b) = \sum_{u \in U} \sum_{v_i, v_j \in \mathcal{S}_u} \alpha^{|j-i|} \cdot \frac{\mathtt{Seq}(u, v_i, v_j)}{\max(|\mathcal{K}_i^v|, |\mathcal{K}_j^v|)}, \quad (1)$$

where function $\mathtt{Seq}(u, v_i, v_j) = 1$ holds when 1) v_i, v_j are the i-th, j-th videos of a student's watching record $\mathcal{S}_u$ and are in the same course, $j > i$; 2) $c_a \in \mathcal{K}_i^v$ and $c_b \in \mathcal{K}_j^v$ (Otherwise $\mathtt{Seq}(u, v_i, v_j) = 0$).

Considering there are multiple concepts taught in each video, we employ $\max(|\mathcal{K}_i^v|, \left|\mathcal{K}_j^v\right|)$ to normalize the feature of a certain concept pair. Furthermore, since the distance between watching videos corresponds to their relatedness, we employ an attenuation coefficient of $\alpha \in (0, 1)$ to capture distant dependence from long sequences in this pattern.

Cross Course Watching Pattern. Besides watching in one course, there is a phenomenon that some students choose to watch videos in other courses before continuing on the present study. The main reason is that the knowledge provided by other courses' videos is helpful to study this course. Hence, cross course watching behavior could reflect the dependence between concepts from different courses. The prerequisite feature $f_2^{\mathcal{P}}$ for c_a and c_b is calculated as:

$$f_2^{\mathcal{P}}(c_a, c_b) = \sum_{u \in U} \sum_{v_i, v_j \in \mathcal{S}_u} \alpha^{|j-i|} \cdot \frac{\mathtt{Crs}(u, v_i, v_j)}{\max(|\mathcal{K}_i^v|, |\mathcal{K}_j^v|)}, \quad (2)$$

where function $\mathtt{Crs}(u, v_i, v_j) = 1$ holds when 1) v_i, v_j are the i-th, j-th videos of a student's record $\mathcal{S}_u$ and are in the different courses; 2) $c_a \in \mathcal{K}_i^v$ and $c_b \in \mathcal{K}_j^v$ (Otherwise $\mathtt{Crs}(u, v_i, v_j) = 0$).

Skipping Watching Pattern. An abnormal student behavior is skipping some videos when learning a course, which drops a hint that the "skipped videos" are not so necessary for latter videos' comprehension. Given a student behavior sequence $\mathcal{S}_u$ and course video orders $\mathcal{C} = [v_1..v_i...]$, we can detect the skipped video pairs and assign a negative $f_3^{\mathcal{P}}$ for the concept pair c_a and c_b as:

$$f_3^{\mathcal{P}}(c_a, c_b) = -\sum_{u \in U} \sum_{v_i, v_j \in \mathcal{S}_u} \alpha^{|j-i|} \cdot \frac{\mathtt{Skp}(u, v_i, v_j)}{\max(|\mathcal{K}_i^v|, |\mathcal{K}_j^v|)}, \quad (3)$$

where function $\mathtt{Skp}(u, v_i, v_j) = 1$ holds when 1) v_i, v_j are the i-th, j-th videos of a same course, and $i < j$; 2) v_j is watched by user u but v_i is not watched; 3) $c_a \in \mathcal{K}_i^v$ and $c_b \in \mathcal{K}_j^v$ (Otherwise $\mathtt{Skp}(u, v_i, v_j) = 0$).

Backward Watching Pattern. This pattern means a student goes back to a video that he/she watched before. A possible explanation is he/she jumps back to a video for re-learning prerequisite knowledge of the current video. Based on this assumption, we adjust the equation for the feature $f_4^{\mathcal{P}}$ between c_a and c_b.

$$f_4^{\mathcal{P}}(c_a, c_b) = \sum_{u \in U} \sum_{v_i, v_j \in \mathcal{S}_u} \alpha^{|j-i|} \cdot \frac{\mathtt{Bck}(u, v_i, v_j)}{\max(|\mathcal{K}_i^v|, |\mathcal{K}_j^v|)}, \quad (4)$$

where function $\mathtt{Bck}(u, v_i, v_j) = 1$ holds when 1) v_i, v_j are the i-th, j-th videos of a student behavior record $\mathcal{S}_u$, and $i < j$; 2) v_i is watched again after v_j; 3) $c_a \in \mathcal{K}_i^v$ and $c_b \in \mathcal{K}_j^v$ (Otherwise $\mathtt{Bck}(u, v_i, v_j) = 0$).

5.2 Experiment: Enhancing Exiting Methods

To verify our assumption of the accordance of student behavior and prerequisite structures, we conduct experiments to explore whether the extracted features $f_k^{\mathcal{P}}(k = 1, 2, 3, 4)$ can help discover prerequisite relations among concepts. Specifically, we enhance existing prerequisite discovery models by adding student behavior features. We select following typical baselines for the experiments:

- **MOOC-RF**: A widely-used method (Pan et al., 2017a), which extracts seven features from the video and subtitle corpus of MOOCs. We reproduce this method and select Random Forest as the classifier to match its claimed best performance.
- **GlobalF**: This method (Liang et al., 2018) extract the graph-based and text-based features for each concept pair. The graph-based features are based on Wikipedia Anchor Links, and the text-features are based on the description of concepts.
- **PREREQ**: This method (Roy et al., 2019) utilizes course dependence and video orders to find prerequisite relations through a siamese network.
- **LSTM**: Recently, some researchers try to utilize neural approaches to extract prerequisite relations from text. We reproduce the LSTM model in (Alzetta et al., 2019) to encode the concepts' texts as prerequisite features.

For enhancing existing models, we concatenate the student behavior features $f_k^{\mathcal{P}}(k = 1, 2, 3, 4)$ with original features and then utilize the same classifiers in the respective papers to obtain experimental results.

Result Analysis We summarize the results in Table 2, where $^{+sf}$ represents the results of models enhanced with student behavior features. We apply

	P	R	$F1$	Δ
MOOC-RF	0.749	0.584	0.656	-
MOOC-RF^{+sf}	0.755	0.639	0.691	**+3.5**
GlobalF	0.679	0.631	0.650	-
GlobalF^{+sf}	0.710	0.657	0.680	**+3.0**
PREREQ	0.468	0.792	0.567	-
PREREQ^{+sf}	0.511	0.712	0.595	**+2.8**
LSTM	0.706	0.743	0.723	-
LSTM^{+sf}	0.707	0.736	0.720	-0.3

Table 2: Performance of student behavior enhanced baselines. P, R and $F1$ represent *precision*, *recall*, and *F1 score* respectively, and Δ represents the improvement of F1 score after adding student behavior features. $^{+sf}$: enhanced with student behavior features.

10-fold cross-validation and balance the training set by oversampling the positive instances[4]. From the presented results, we can infer the following insights: 1) **Student behaviors are effective in prerequisite relation discovery**. MOOC-RF, GlobalF, and PREREQ gain significant improvement after adding the extracted student behavior features. It preliminarily proves that the student behaviors imply clues of prerequisite concepts and are useful to prerequisite relation discovery. 2) **Shortcoming**. Feature-based behavior modeling meets a bottleneck in improving state-of-the-art LSTM-based baseline. A possible explanation is that the course concepts and their prerequisite relations naturally form a dependence graph structure (Gordon et al., 2016), so the sequence-based LSTM reaches the limit of performance and is difficult to be effectively improved. Therefore, in the next sections, we explore how to utilize student behaviors in the graph structure.

6 Explore Graph-based Modeling of Student Behavior

Building *concept graphs* is a common idea in concept mining tasks, including concept extraction and expansion (Pan et al., 2017b; Yu et al., 2019). Since the prerequisite relations among concepts are transitive, i.e., if $a \rightarrow b$, $b \rightarrow c$ then $a \rightarrow c$, previous works also often employ a directed graph to describe the dependence on a set of concepts (Brunskill, 2011; Gordon et al., 2016). This inspires us to leverage student behaviors better by building a *concept graph*, defined as:

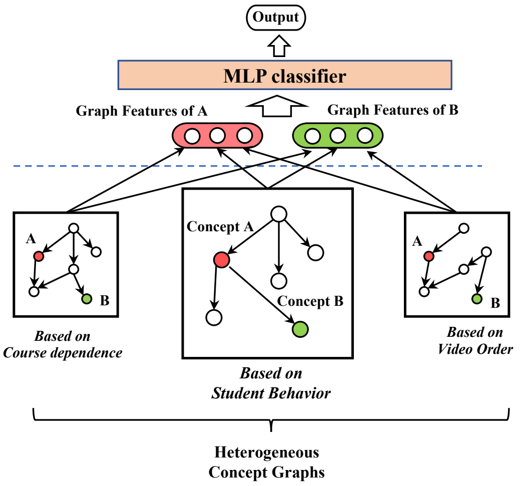

Figure 3: The framework of our graph-based model. The nodes of each graph are course concepts. Features of concepts are the concatenation of the corresponding node embeddings learned by GCN.

Definition 2 (**Concept Graph**). *A concept graph* $\mathcal{G} = (\mathcal{K}, E)$ *is a weighted directed graph, whose nodes are course concepts* $\mathcal{K}$ *and each edge* $e = (c_a \rightarrow c_b) \in E$ *is associated with a weight* w_e.

Regarding the prerequisite relation learning as a link prediction problem in a graph, we are able to leverage the student behavior better by utilizing Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) to model information propagation of the concepts. Meanwhile, as several types of MOOC information have been applied to detect prerequisite concepts in previous research, including course dependence (Liang et al., 2017; Roy et al., 2019), video order (Pan et al., 2017a), we also design similar concept graphs for these resources. By comparing the model performance of different graphs, we can explore the role of student information more fairly (excluding the factors of the graphical modeling). In this section, we introduce the construction of concept graphs and how to conduct prerequisite relation learning on them for employing student behaviors better.

6.1 Concept Graph Construction

As shown in Figure 3, we design a concept graph $\mathcal{G}^s$ based on student behaviors as while as $\mathcal{G}^c$ based on course dependence graph and $\mathcal{G}^v$ based on video order graph. As the nodes of these concept graphs are the same course concept $\mathcal{K}$, the only difference is the setting of their edges.

Our graph construction stage's main idea is to assign edge weight for each concept pair in these

[4] The following experiments are also set up by the same settings, and more details are in Appendix.

graphs. After calculating all edges' weights in a graph, we only preserve the edges with positive weights, for they are helpful for relation reasoning.

Concept Graph Based on Student Behavior. To build concept graph from student behaviors, a straightforward idea is to model the prerequisite clues by combining the extracted features in Section 5. Hence, we assign the weight w_e^s for the edge $e = (c_a \rightarrow c_b)$ in this graph $\mathcal{G}^s$ as:

$$w_e^s = \sum_{i=1}^{4} f_i^{\mathcal{P}}(c_a, c_b) \times \frac{\log(|U|)}{|U|}, \tag{5}$$

where $f_i^{\mathcal{P}}(i = 1, 2, 3, 4)$ denotes the features of the concept c_a and c_b from the four behavior patterns. $log(|U|)/|U|$ is used to normalize the weight to combine with other two user-independent graphs.

Except for student behaviors, we also build concept graphs for existing static MOOC prerequisite clues through similar methods, including the dependence among courses and the preset order of videos. By modeling these information, we can more fairly compare the contribution of these clues in graphs and explore whether they can be integrated to further enhance the model.

Concept Graph Based on Course Dependence. Course dependence is widely used in prerequisite learning. When a course is certain to be a prerequisite course of another one, there must be dependence relations between some of their concepts. So we build a concept graph $\mathcal{G}^c$ based on course dependency to exploit this information. Suppose c_a and c_b are respectively concepts of course $\mathcal{C}_i$ and $\mathcal{C}_j$, for an edge $e = (c_a \rightarrow c_b)$ of this concept graph, we can calculate its weight w_e^c as:

$$w_e^c = \sum_{\mathcal{C}_i, \mathcal{C}_j \in \mathcal{M}} \frac{\texttt{CD}(\mathcal{C}_i, \mathcal{C}_j)}{\max(|\mathcal{K}_i^c|, |\mathcal{K}_j^c|)}, \tag{6}$$

where function $\texttt{CD}(\mathcal{C}_i, \mathcal{C}_j) = 1$ only when pair $(\mathcal{C}_i, \mathcal{C}_j)$ is in course dependence set $\mathcal{D}$ (otherwise $\texttt{CD}(\mathcal{C}_i, \mathcal{C}_j) = 0$). We also use $\max(|\mathcal{K}_i^c|, \left|\mathcal{K}_j^c\right|)$ to normalize such information to concept-level.

Concept Graph Based on Video Order. Video order indicates the dependence between videos. In general, the previous videos in a course are helpful for the latter ones (Roy et al., 2019) and such dependence is stronger when two videos are closer. Based on this assumption, when calculating the weight for the concept graph $\mathcal{G}^v$ based on video order, we also apply the attenuation coefficient α to obtain edge weight w_e^v for the edge e between concept c_a and c_b:

$$w_e^v = \sum_{u \in U} \sum_{v_i, v_j \in \mathcal{S}_u} \alpha^{|j-i|} \cdot \frac{\texttt{VO}(u, v_i, v_j)}{\max(|\mathcal{K}_i^v|, |\mathcal{K}_j^v|)}, \tag{7}$$

where function $\texttt{VO}(v_i, v_j) = 1$ only when 1) v_i, v_j are the i-th, j-th videos of a same course; 2) $c_a \in \mathcal{K}_i^v$ and $c_b \in \mathcal{K}_j^v$ (otherwise $\texttt{VO}(v_i, v_j) = 0$).

6.2 Prerequisite Relation Learning

After building concept graphs $\mathcal{G}^c$, $\mathcal{G}^v$, and $\mathcal{G}^s$, we utilize GCNs to reason prerequisite relations in these graphs. In particular, we initialize the adjacency matrix A of the graph and the feature matrix X of the concept nodes for each graph. The adjacency matrix A, with a size of $|\mathcal{K}|^2$, can be derived from edge weights, e.g., for the adjacency matrix A^s of the student behavior graph $\mathcal{G}^s$, we have $A_{ij}^s = w_e^s$, where w_e^s is the weight of edge $e = (c_i \rightarrow c_j)$. And the $|\mathcal{K}| \times d$ sized feature matrix X of the concept nodes in all graphs is initialized by a pre-trained d-dimension language model , i.e., X_i is the word embedding of the text concept c_i.

The training of GCNs on our directed concept graphs follows the propagation rule shown below, which is an adapted version for directed graphs:

$$Z = \hat{D}^{-1} \hat{A} X \Theta, \tag{8}$$

where Θ is a matrix of filter parameters, Z is the convolved signal matrix, $Z_i = h_i$ is the graph embedding of concept c_i, $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ and the Laplacian is $\hat{A} = I_N + A$. The other settings are the same with (Kipf and Welling, 2017).

After the graph-training stage, we input the graph embeddings h_a,h_b of a concept pair (c_a, c_b) into a two-layer MLP followed with a sigmoid function to do classification:

$$\Pr(\mathcal{L}(c_a, c_b) = 1) = \sigma(\max(0, (h_a \oplus h_b)\mathbf{W_1})\mathbf{W_2}), \tag{9}$$

where Pr is the probability, $\sigma(\cdot)$ is the sigmoid function, $\mathbf{W_1} \in \mathcal{R}^{2d \times d}$ and $\mathbf{W_2} \in \mathcal{R}^{d \times 1}$ are trainable matrices, and $\oplus$ denotes vector concatenation.

6.3 Experiment: Graph-based Modeling

We conduct experiments on our newly presented dataset and apply the same settings to evaluate the performance of our proposed graph-based method.

Table 3 summarizes the comparing results of different methods. $^{+cv}$ denotes $\mathcal{G}^c$ and $\mathcal{G}^v$ are used, $^{+s}$ denotes only $\mathcal{G}^s$ is used, and $+cvs$ denotes all the three graphs are used. We analysis the performance in the following aspects: (1) *Advantage of*

Model	P	R	F_1
MOOC-RF^{+sf}	0.755	0.639	0.691
GlobalF^{+sf}	0.710	0.657	0.680
PREREQ^{+sf}	0.511	0.712	0.595
LSTM	0.706	0.743	0.723
GCN^{+cv}	0.789	0.792	0.790
GCN^{+s}	0.762	0.784	0.772
GCN^{+cvs}	**0.792**	**0.814**	**0.802**

Table 3: Overall performance. $^{+sf}$: enhanced with student behavior features (Section 5.1); $^{+cv}$: $\mathcal{G}^c$ and $\mathcal{G}^v$ are used; $^{+s}$: only $\mathcal{G}^s$ is used; $^{+cvs}$: all $\mathcal{G}^c$, $\mathcal{G}^v$ and $\mathcal{G}^s$ are used.

	P	R	F_1
GCN^{+cv}	-8.5	-11.1	-9.9
GCN^{+s}	-7.7	-11.8	-9.8
GCN^{+cvs}	**-10.5**	**-14.5**	**-12.6**

Table 4: Absolute performance decline when dropping the edge weights of the concept graphs.

Graph-based Modeling. GCN-based models perform better than previous state-of-the-art methods, which indicates that modeling concept dependence in graphs is effective. The prerequisites can be obtained through proper reasoning, and develop a more advanced graph-based model is a promising direction. (2) *The effectiveness of Student Behavior in Graph Modeling*. GCN^{+s} performs better than LSTM and has a competitive performance among all baselines. Further, GCN^{+cvs} performs better than GCN^{+cv}, which indicates that except for the improvement of graph-based modeling, the student behavior is still beneficial in advanced attempts of prerequisite relation discovery.

6.4 Analysis of Graph Modeling

As the graph modeling further improve the performance, we present experimental results to analyze the role of its different components. And more experimental discussions are in Appendix.

Necessity of Edge Weights. We set all the edge weights to 1 to convert the three concept graphs into unweighted ones, and present the corresponding results in Table 4. The performance of all the three GCN-based models declines severely, especially the most competitive GCN^{+cvs}, indicating the necessity of edge weights.

Impact of Different Behavior Patterns. We also investigate the impact of four behavior patterns by only using some patterns when building graph $\mathcal{G}^s$. The changes in performance after adding part of the features are shown in Table 5, which provides some insights for understanding the student behaviors: (1) Sequential watching covers high-quality prerequisite concept pairs, resulting in a significant improvement of the precision (P). However, such a pattern is not so effective for those do not match with the preset order of courses, resulting in relatively small improvement of recall (R); (2) The other three patterns, improves recall significantly, indicating that they are complementary for discovering prerequisite relations those not covered by sequential watching; (3) Therefore, the four behavior patterns are complementary, and all of them are helpful for discovering prerequisite concepts.

Pattern	P	R	F_1
Seq.	**+0.6**	+0.5	+0.5
Crs.	+0.3	+2.1	+1.2
Skp.	+0.1	+2.3	+1.1
Bck.	+0.1	**+2.6**	**+1.3**

Table 5: Performance improvement of GCN^{+cvs} compared with GCN^{+cv} when only using one type of student behavior pattern while building the concept graph $\mathcal{G}^s$. Seq: sequential watching, Crs: cross course watching, Bck: backward watching, and Skp: skip watching.

7 Conclusion and Future Work

In this work, we conduct an investigation on employing the students' video watching behaviors in the task of discovering prerequisite relations of concepts in MOOCs. To support the study, we collect student behaviors and conduct data annotations to build a novel dataset for this task. After analyzing the typical patterns, we propose a feature-based method and experimentally verify the student behaviors' effectiveness in enhancing existing models. Then we propose a graph-based method and experimentally show that GCNs are more beneficial to model student behaviors.

We also present several promising future directions, including 1) A more detailed analysis of the relationship between user behavior and prerequisite concepts, e.g., divide the typical patterns into a finer-grained level for analysis. 2) More advanced graph-based models to discover high-quality prerequisite relations, e.g. employing graph attention mechanism in this task. 3) Developing more interactive applications to collect more kinds of user behaviors for prerequisite relation discovery, such as learning path recommendation, games, etc.

Ethical Consideration

Our datasets are from real MOOC scenarios. Therefore, we carefully consider the legitimacy of the data and the protection of user privacy during the whole process of collection.

Certification and User privacy. All data collected is licensed by the platform. Considering the protection of user privacy, we strictly abide by the agreement between the platform and the users, remove sensitive personal information, and anonymize the users into UserIDs. Meanwhile, we utilize static masking techniques (Ghinita et al., 2007) for further data security protection.

References

Rakesh Agrawal, Behzad Golshan, and Evangelos Papalexakis. 2015. Datadriven synthesis of study plans. *Data Insights Laboratories*.

Chiara Alzetta, Alessio Miaschi, Giovanni Adorni, Felice Dell'Orletta, Frosina Koceva, Samuele Passalacqua, and Ilaria Torre. 2019. Prerequisite or not prerequisite? that's the problem! an nlp-based approach for concept prerequisites learning.

Ausubel. 1968. Educational psychology: A cognitive view.

John R Bergan and Patrick Jeska. 1980. An examination of prerequisite relations, positive transfer among learning tasks, and variations in instruction for a seriation hierarchy. *Contemporary Educational Psychology*.

Emma Brunskill. 2011. Estimating prerequisite structure from noisy data.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert M Gagne and Leslie J Briggs. 1974. *Principles of instructional design.* Holt, Rinehart & Winston.

Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. 2007. Fast data anonymization with low information loss. In *Proceedings of the 33rd international conference on Very large data bases*, pages 758–769.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–875, Berlin, Germany. Association for Computational Linguistics.

Xiaopeng Huang, Kyeong Yang, and Victor B Lawrence. 2015. An efficient data mining approach to concept map generation for adaptive learning. In *Industrial conference on data mining*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Igor Labutov, Yun Huang, Peter Brusilovsky, and Daqing He. 2017. Semi-supervised techniques for mining learning outcomes and prerequisites. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 907–915. ACM.

Irene Li, Alexander R Fabbri, Robert R Tung, and Dragomir R Radev. 2019. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning. In *Proceedings of Thirty-Third AAAI Conference on Artificial Intelligence*, volume 33, pages 6674–6681.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. 2015. Measuring prerequisite relations among concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1668–1674, Lisbon, Portugal. Association for Computational Linguistics.

Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C Lee Giles. 2018. Investigating active learning for concept prerequisite learning. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence*, pages 7913–7919.

Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, and C Lee Giles. 2017. Recovering concept prerequisite relations from university course dependencies. In *Proceedings of Thirty-First AAAI Conference on Artificial Intelligence*, pages 4786–4791.

Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. 2017a. Prerequisite relation learning for concepts in MOOCs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, Vancouver, Canada. Association for Computational Linguistics.

Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, and Jie Tang. 2017b. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 875–884, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Forrest W Parkay and Glen Hass. 1999. *Curriculum planning: A contemporary approach.* Allyn & Bacon, Incorporated.

Cristobal Romero and Sebastian Ventura. 2007. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*.

Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. 2019. Inferring concept prerequisite relations from online educational resources. In *Proceedings of Thirty-Third AAAI Conference on Artificial Intelligence*, volume 33, pages 9589–9594.

Mohsen Sayyadiharikandeh, Jonathan Gordon, Jose-Luis Ambite, and Kristina Lerman. 2019. Finding prerequisite relations using the wikipedia clickstream. In *Companion Proceedings of The 2019 World Wide Web Conference*, page 1240–1247. ACM.

Richard Scheines, Elizabeth Silver, and Ilya M Goldin. 2014. Discovering prerequisite relationships among knowledge components. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 355–356.

Partha Talukdar and William Cohen. 2012. Crowdsourced comprehension: Predicting prerequisite structure in Wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 307–315, Montréal, Canada. Association for Computational Linguistics.

Annalies Vuong, Tristan Nixon, and Brendon Towle. 2011. A method for finding prerequisites within a curriculum. In *Proceedings of the 4th International Conference on Educational Data Mining*, pages 211–216.

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, et al. 2020. Mooccube: a large-scale data repository for nlp applications in moocs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142.

Jifan Yu, Chenyu Wang, Gan Luo, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2019. Course concept expansion in MOOCs with external knowledge and interactive game. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4292–4302, Florence, Italy. Association for Computational Linguistics.

A Data Annotation and Quality Control

Annotation. The group of annotators consists of 1 experienced CS professor, 2 CS Ph.Ds, and 8 Students who finished these courses. We also employ the teachers of our selected courses as consultants to deal with the disagreements in the annotation process. The annotation requires each course pair to be labeled by two students. If there are conflicts, two Ph.D. students will further label the result. And the final results are confirmed by the professor and corresponding teachers.

Data quality. The data quality includes the quality of the annotation and student behavior data. From our experience, the annotation of prerequisite is not easy to determine a perfect standard. E.g., "Stack" and "Queue": Grasping one of them indeed help the understanding of the other, but someone may think these two concepts are not prerequisite. We finally control the Kappa over 0.7, which indicates a good quality of the final prerequisite dataset with the help of corresponding teachers, students and our *multi-round annotate-check* annotation process.

Meanwhile, to alleviate the noises introduced by users' random operations, we filter out the student behavior data by: (i) Remove students who have less than 2 elective courses and watch less than 10 videos. (ii) Delete behavior records that the student who watched less than 30% of the video.

B Implementation Details

Running Environment The experiments in this paper are conducted on a single Linux server with an Intel(R) Xeon(R) CPU E5-2669 v4 @ 2.20GHz, 256G RAM, and 8 NVIDIA GeForce TITAN X (Pascal). The codes of our proposed models are implemented with Pytorch 1.3.1 in Python 3.7.

Experimental Settings When training the GCNs for evaluation, we utilize a dropout with drop rate 0.2. All hyper-parameters are tuned on the validation set. The word vectors of all baseline methods are initialized using BERT (Devlin et al., 2019). As the training dataset is not big, we reduce the dimensionality of these word vectors by PCA to prevent overfitting. The attenuation coefficient α is set to 0.3.

C Model Analysis

Attenuation Coefficient α. It is a parameter to model the impact of long-range dependence on the concept relationships. As shown in Figure 4, α is effective in reducing noises for student behavior modeling. As for the settings of the combined graph model, the performance is more stable with different α. Both GCN^{+s} and GCN^{+cvs} perform best under the setting of a 0.3 α. GCN^{+cv} perform best with a 0.7 α but it is not so sensitive to this hyper-parameter. Since α affect the state of the constructed graph, and the overall performance trend of the three models does not change, we choose 0.3 as the setting value to keep the built graphs same in our experiments.

Qualitative Analysis. Furthermore, we manually analyze which previous error cases are corrected by graph modeling, and find some fascinating phenomena. Here we list 30 sampled cases in Table 6 as the supplement of Qualitative analysis. Compared with one of the strongest baseline *LSTM*, graph modeling perform better in two main cases: (1) Hypernymy(41.3%) (e.g. “Linked list”-“Doubly linked list”), which has been discussed as an important cause of prerequisite relation in previous work (Liang et al., 2015). As the hypernymy relations are organized in a directed acyclic graph(usually a tree), the graph modeling can capture the global features better. (2) Theory-Application pairs(27.4%) (e.g. “Instant messaging”-“Advanced mobile phone network”). Such concept pairs have no apparent structural or semantic features like others, which is the main reason that baselines cannot handle such cases well. As our method can figure them out, we conjecture that such improvement is provided by the proper modeling of student behaviors.

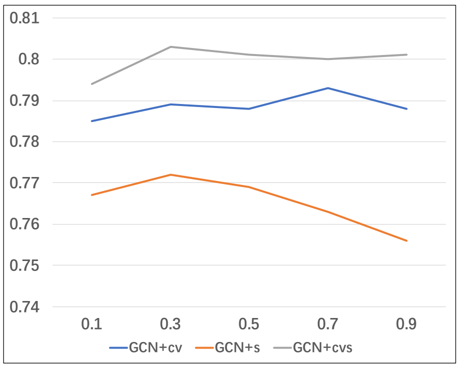

Figure 4: Effect of α.

Type	ConceptA	ConceptB
Hypernymy	Ethernet	Switched Ethernet
	Ethernet	Fast Ethernet
	Data Encryption Standard	Advanced Encryption Standard
	Maximum	Global Maximum
	Linked List	Single Linked List
	Linked List	Doubly Linked List
	Iterative Loop	Iterative Calculation
	Recurrence Relationship	Recursion
	Computer Vision	Image Classification
	Tree Algorithm	Tree Search Algorithm
	Depth First Search	Eight Queens Problem
	Linear Regression	Linear Regression Model
	Network Attacks	Replay Attack
	Data	Meta Data
Theory-Application	Divide	Factorization
	Effective digits	Run Length Coding
	Continuous Time System	Kalman Filter
	Scheduling Strategy	Resource Allocation
	Reasoning Method	Automatic Reasoning
	System Structure	Service Data Unit
	Multiple Input/ Multiple Output	Digital Subscriber Line Access Multiplexer
	Interconnection Network	One-Arm router
	Fourier Transform	Convolution
Other	Transitivity	Inequality
	White Box Testing	Integration Testing
	Computational Complexity	Graphical Method
	Network Delay	Instant Messaging
	Conditional Distribution	Posterior Distribution
	Feasible Solution	NP Hard Problem
	Binary	Assembly Language

Table 6: The baselines' error cases corrected by graph modeling We divide them into three categories.