Prerequisite Relation Learning for Concepts in MOOCs

Liangming Pan, Chengjiang Li, Juanzi Li* and Jie Tang
Knowledge Engineering Laboratory
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China (* corresponding author)
{panlm14@mails,licj17@mails,lijuanzi,tangjie}tsinghua.edu.cn

Abstract

What prerequisite knowledge should students achieve a level of mastery before moving forward to learn subsequent coursewares? We study the extent to which the prerequisite relation between knowledge concepts in Massive Open Online Courses (MOOCs) can be inferred automatically. In particular, what kinds of information can be leveraged to uncover the potential prerequisite relation between knowledge concepts. We first propose a representation learning-based method for learning latent representations of course concepts, and then investigate how different features capture the prerequisite relations between concepts. Our experiments on three datasets form Coursera show that the proposed method achieves significant improvements (+5.9-48.0% by F1-score) comparing with existing methods.

1 Introduction

Mastery learning was first formally proposed by Benjamin Bloom in 1968 (Bloom, 1981), suggesting that students must achieve a level of mastery (e.g., 90% on a knowledge test) in prerequisite knowledge before moving forward to learn subsequent knowledge concepts. From then on, prerequisite relations between knowledge concepts become a cornerstone for designing curriculum in schools and universities. Prerequisite relations essentially can be considered as the dependency among knowledge concepts. It is crucial for people to learn, organize, apply, and generate knowledge (Laurence and Margolis, 1999). Figure 1 shows a real example from Coursera. The student wants to learn "Conditional Random Field" (in video18 of CS229). The prerequisite knowledge might be "Hidden Markov Model" (in video25 of CS224), whose prerequisite knowledge is "Maximum Likelihood" (in video12 of Math112).

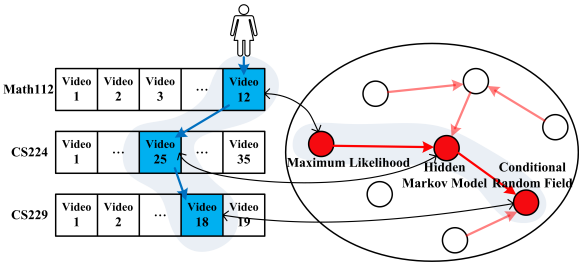

Figure 1: An example of prerequisite relations in MOOCs

Organizing the knowledge structure with prerequisite relations in education improves tasks such as curriculum planning (Yang et al., 2015), automatic reading list generation (Jardine, 2014), and improving education quality (Rouly et al., 2015). For example, as shown in Figure 1, with explicit prerequisite relations among concepts (in red), a coherent and reasonable learning sequence can be recommended to the student (in blue). Before, the prerequisite relationships were provided by teachers or teaching assistants (Novak, 1990); however in the era of MOOCs, it is becoming *infeasible* as the teachers would find that they are facing with hundreds of thousands of students with various background. Meanwhile, the rapid growth of Massive Open Online Courses has offered thousands of courses, and students are free to choose any course from the thousands of candidates. Therefore, there is a clear need for methods to automatically dig out the prerequisite relationships among knowledge concepts from the large course space, so that the students from different background can easily explore the knowledge space and better design their personalized learning schedule.

There are a few efforts aiming to automatically detect prerequisite relations for knowledge base. For example, Talukdar and Cohen (2012) proposed a method for inferring prerequisite relationships between entities in Wikipedia and Liang et al. (2015) presented a more general approach

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1447–1456
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1133

to predict prerequisite relationships. A few other works intend to extract prerequisite relationships from textbooks (Yosef et al., 2011; Wang et al., 2016). However, it is far from sufficient to directly apply these methods to the MOOC environments due to the following reasons. First, the focus of most previous attempts has been on prerequisite inference of Wikipedia concepts (either Wikipedia articles or Wikipedia concepts in textbooks). Many course concepts are not included in Wikipedia (Schweitzer, 2008; Okoli et al., 2014). We can leverage Wikipedia, in particular the existing entity relationships in Wikipedia, but cannot only rely on Wikipedia for detecting prerequisite relations in MOOCs. Second, with the thousands of courses from different universities and also very different disciplinaries, the MOOC scenario is much more complicated — there are not only inter-course concept relationships, but also intra-course and even intra-disciplinary relationships. Moreover, user interactions with the MOOC system might be also helpful to identify the prerequisite relations. How to fully leverage the different information to obtain a better performance for inferring prerequisite relations in MOOCs is a challenging issue.

In this paper, we attempt to figure out what kinds of information in MOOCs can be used to uncover the prerequisite relations among concepts. Specifically, we consider it from three aspects, including course concept semantics, course video context and course structure. First, semantic relatedness plays an important role in prerequisite relations between concepts. If two concepts have very different semantic meanings (e.g., "matrix" and "anthropology"), it is unlikely that they have prerequisite relations. However, statistical features in MOOCs do not provide sufficient information for capturing the concept semantics because of the short length of course videos in MOOCs, we propose an embedding-based method to incorporate external knowledge from Wikipedia to learn semantic representations of concepts in MOOCs. Based on it, we propose one **semantic feature** to calculate the semantic relatedness between concepts. Second, motivated by the *reference distance* (RefD) (Liang et al., 2015), we propose three new **contextual features**, i.e., Video Reference Distance, Sentence Reference Distance and Wikipedia Reference Distance, to infer prerequisite relations in MOOCs based on context information from different aspects, which are more general and informative than RefD and overcome its sparsity problem. Third, we examine different distributional patterns for concepts in MOOCs, including appearing position, distributional asymmetry, video coverage and survival time. We further propose three **structural features** to utilize these patterns to help prerequisite inference in MOOCs.

To evaluate the proposed method, we construct three datasets, each of which consists of multiple real courses in a specific domain from Coursera [1], the largest MOOC platform in the world. We also compare our method with the representative works of prerequisite learning and make a deep analysis of the feature contribution proposed in the paper. The experimental results show that our method achieves the state-of-the-art results in the prerequisite relation discovery in MOOCs. In summary, our contributions include: a) the first attempt, to the best of our knowledge, to detect prerequisite relations among concepts in MOOCs; b) proposal of a set of novel features that utilize contextual, structural and semantic information in MOOCs to identify prerequisite relations; c) design of three useful datasets based on real courses of Coursera to evaluate our method.

2 Problem Formulation

In this section, we first give some necessary definitions and then formulate the problem of prerequisite relation learning in MOOCs.

A **MOOC corpus** is composed by n courses in the same subject area, denoted as $\mathcal{D} = \{\mathcal{C}_1, \cdots, \mathcal{C}_i, \cdots, \mathcal{C}_n\}$, where $\mathcal{C}_i$ is one course. Each course $\mathcal{C}$ can be further represented as a video sequence $\mathcal{C} = (\mathcal{V}_1, \cdots, \mathcal{V}_i, \cdots, \mathcal{V}_{|\mathcal{C}|})$, where $\mathcal{V}_i$ denotes the i-th teaching video of course $\mathcal{C}$. Finally, we view each video $\mathcal{V}$ as a document of its video texts (video subtitles or speech script), i.e., $\mathcal{V} = (s_1 \cdots s_i \cdots s_{|\mathcal{V}|})$, where s_i is the i-th sentence of the video texts.

Course concepts are subjects taught in the course, i.e., the concepts not only mentioned but also discussed and taught in the course. Let us denote the course concept set of $\mathcal{D}$ as $\mathcal{K} = \mathcal{K}_1 \cup \cdots \cup \mathcal{K}_n$, where $\mathcal{K}_i$ is the set of course concepts in $\mathcal{C}_i$.

Prerequisite relation learning in MOOCs is formally defined as follows. Given a MOOC corpus $\mathcal{D}$ and its corresponding course concepts

[1] https://www.coursera.org/

$\mathcal{K}$, the objective is to learn a function $\mathcal{P} : \mathcal{K}^2 \rightarrow \{0, 1\}$ that maps a concept pair $\langle a, b\rangle$, where $a, b \in \mathcal{K}$, to a binary class that predicts whether a is a prerequisite concept of b.

In order to learn this mapping, we need to answer two crucial questions. How could we represent a course concept? What information regarding a concept pair is helpful to capture their prerequisite relation? We first propose an embedding-based method to learn appropriate semantic representations for each course concept in $\mathcal{K}$. Based on the learned representations, we propose 7 novel features to capture whether a concept pair has prerequisite relation. These features utilize different aspects of information and can be classified into 1 semantic feature, 3 contextual features and 3 structural features. In the following section, we first describe the semantic representations in detail, and then formally introduce our proposed features.

3 Method

3.1 Concept Representation & Semantic Relatedness

We first learn appropriate representations for course concepts. Given the course concepts $\mathcal{K}$ as input, we utilize a *Wikipedia corpus* to learn semantic representations for concepts in $\mathcal{K}$. A Wikipedia corpus $\mathcal{W}$ is a set of Wikipedia articles and can be represented as a sequence of words $\mathcal{W} = \langle w_1 \cdots w_i \cdots w_m \rangle$, where w_i denotes a word and m is the length of the word sequence. Our method consists of two steps: (1) entity annotation, and (2) representation learning.

Entity Annotation. We first automatically annotate the entities in $\mathcal{W}$ to obtain an **entity set** $\mathcal{E}$ and an **entity-annotated Wikipedia corpus** $\mathcal{W}' = \langle x_1 \cdots x_i \cdots x_{m'} \rangle$, where x_i corresponds to a word $w \in \mathcal{W}$ or an entity $e \in \mathcal{E}$. Note that $m' < m$ because multiple adjacent words could be labeled as one entity. Many entity linking tools are available for entity annotation, e.g. TAGME (Ferragina and Scaiella, 2010), AIDA (Yosef et al., 2011) and TremenRank (Cao et al., 2015). However, the rich hyperlinks created by Wiki editors provide a more natural way. In our experiments, we simply use the hyperlinks in Wikipedia articles as annotated entities.

Representation Learning. We then learn word embeddings (Mikolov et al., 2013b,a) on $\mathcal{W}'$ to obtain low-dimensional, real-valued vector representations for entities in $\mathcal{E}$ and words in $\mathcal{W}$. Let us denote v_e and v_w as the vector of $e \in \mathcal{E}$ and $w \in \mathcal{W}$, respectively. For a course concept $a \in \mathcal{K}$, suppose a is a N-gram term $\langle g_1 \cdots g_N \rangle$ and $g_1, \cdots, g_N \in \mathcal{W}$, we obtain its semantic representations v_a as follows.

$$v_a = \begin{cases} v_e, & \text{if } a \equiv e \text{ and } e \in \mathcal{E} \\ v_{g_1} + \cdots + v_{g_N}, & \text{otherwise} \end{cases} \quad (1)$$

It means that if a is a Wikipedia entity, we can directly obtain its semantic representations; otherwise, we obtain its vector via the vector addition of its individual word vectors. In this way, a has no corresponding vector only if any of its constituent word is absence in the whole Wikipedia corpus. This case is unusual because a large online encyclopedia corpus can easily cover almost all individual words of the vocabulary. Our experimental results verify that over 98% of the course concepts have vector representations.

Feature 1: *Semantic Relatedness*

For a given concept pair $\langle a, b\rangle$, the **semantic relatedness** between a and b, denoted as $\omega(a, b)$, is our first feature (the only semantic feature). With learned semantic representations, semantic relatedness of two concepts can be easily reflected by their distance in the vector space. We define $\omega(a, b) \in [0, 1]$ as the normalized cosine distance between v_a and v_b, as follows.

$$\omega(a, b) = \frac{1}{2}(1 + \frac{v_a \cdot v_b}{\|v_a\| \cdot \|v_b\|}) \quad (2)$$

3.2 Contextual Features

Context information in course videos provides important clues to infer prerequisite relations. In videos where concept A is taught, if the teacher also mentions concept B for a lot but not vice versa, then B is more likely to be a prerequisite of A than A of B. For example, "gradient descent" is a prerequisite concept of "back propagation". In teaching videos of "back propagation", the concept "gradient descent" is frequently mentioned when illustrating the optimization detail of back propagation. On the contrary, however, "back propagation" is unlikely to be mentioned when teaching "gradient descent". A similar observation also exists in Wikipedia, based on which Liang et al. (2015) proposed an indicator, namely *reference distance* (RefD), to infer prerequisite relations among Wikipedia articles. However, RefD is computed based on the link structure of Wikipedia, thus is only feasible for Wikipedia

concepts and is not applicable in plain text. We overcome the above shortcomings of RefD to propose three novel features, which utilize different aspects of context information—course videos, video sentences and Wikipedia articles—to infer prerequisite relations in MOOCs.

Feature 2: ***Video Reference Distance***

Given a concept pair $\langle a, b\rangle$ where $a, b \in \mathcal{K}$, we propose the **video reference weight** (Vrw) to quantify how b is referred by videos of a, defined as follows.

$$Vrw\left(a,b\right)=\frac{\sum\limits_{\mathcal{C}\in\mathcal{D}}\sum\limits_{\mathcal{V}\in\mathcal{C}}f\left(a,\mathcal{V}\right)\cdot r\left(\mathcal{V},b\right)}{\sum\limits_{\mathcal{C}\in\mathcal{D}}\sum\limits_{\mathcal{V}\in\mathcal{C}}f\left(a,\mathcal{V}\right)} \tag{3}$$

where $f\left(a,\mathcal{V}\right)$ indicates the term frequency of concept a in video $\mathcal{V}$, which reflects how important is concept a to this video. $r\left(\mathcal{V},b\right) \in \{0,1\}$ denotes whether concept b appears in video $\mathcal{V}$. Intuitively, if b appears in more important videos of a, $Vrw\left(a,b\right)$ tends to be larger, and the range of $Vrw\left(a,b\right)$ is between 0 and 1. Then, the **video reference distance** (Vrd) is defined as the difference of Vrw between two concepts, as follows.

$$Vrd\left(a,b\right)=Vrw\left(b,a\right)-Vrw\left(a,b\right) \tag{4}$$

In practice, this feature may be too sparse if the MOOC corpus is small. For an arbitrary concept pair, they may have no co-occurrence in all course videos. We expend the video reference distance to a more general version by considering the semantic relatedness among concepts. Besides the conditions in which A refers to B, we also consider the cases in which A-related concepts refer to B. We first define the **generalized video reference weight** ($GVrw$) as follows.

$$GVrw\left(a,b\right)=\frac{\sum_{i=1}^{M}Vrw\left(a_i,b\right)\cdot\omega\left(a_i,b\right)}{\sum_{i=1}^{M}\omega\left(a_i,b\right)} \tag{5}$$

where $a_1,\cdots,a_M \in \mathcal{K}$ are the top-M most similar concepts of a, measured by the semantic relatedness function $\omega(\cdot,\cdot)$ in feature 1. $GVrw$ is the weighted average of $Vrw\left(a_i,b\right)$, indicating how b is referred by a-related concepts in their corresponding videos. Note that $a_1 = a$, thus $GVrw\left(a,b\right) \equiv Vrw\left(a,b\right)$ when $M = 1$. Similarly, we define the **generalized video reference distance** ($GVrd$) as follows.

$$GVrd\left(a,b\right)=GVrw\left(b,a\right)-GVrw\left(a,b\right) \tag{6}$$

Intuitively, if most of b-related concepts refer to a but not vice versa, then a is likely to be a prerequisite of b. For example, it is plausible for the related concepts of "gradient descent", e.g., "steepest descent" and "Newton's method", to mention "matrix" but clearly not vice versa.

Feature 3: ***Sentence Reference Distance***

Sentence reference distance is similar to feature 2, but stands on the sentence level. Following the same design pattern of feature 2, we define the **sentence reference weight** (Srw) and **sentence reference distance** (Srd) as follows.

$$Srw\left(a,b\right)=\frac{\sum\limits_{\mathcal{C}\in\mathcal{D}}\sum\limits_{\mathcal{V}\in\mathcal{C}}\sum\limits_{s\in\mathcal{V}}r(s,a)\cdot r(s,b)}{\sum\limits_{\mathcal{C}\in\mathcal{D}}\sum\limits_{\mathcal{V}\in\mathcal{C}}\sum\limits_{s\in\mathcal{V}}r(s,a)} \tag{7}$$

$$Srd\left(a,b\right)=Srw\left(b,a\right)-Srw\left(a,b\right) \tag{8}$$

where $r\left(s,a\right) \in \{0,1\}$ is an indicator of whether concept a appears in sentence s. $Srw(a,b)$ calculates the ratio of B appearing in the sentences of a. We also define **generalized sentence reference weight** ($GSrw$) and **generalized sentence reference distance** ($GSrd$) as follows.

$$GSrw\left(a,b\right)=\frac{\sum_{i=1}^{M}Srw\left(a_i,b\right)\cdot\omega\left(a_i,b\right)}{\sum_{i=1}^{M}\omega\left(a_i,b\right)} \tag{9}$$

$$GSrd\left(a,b\right)=GSrw\left(b,a\right)-GSrw\left(a,b\right) \tag{10}$$

Feature 4: ***Wikipedia Reference Distance***

Contextual information of Wikipedia is also useful for detecting prerequisite relations. As mention before, RefD is not general enough to be applied in our settings, because it is limited to Wikipedia concepts. Therefore, we improve this indicator to a more general one, which is also suitable for non-wiki concepts.

Specifically, for a concept $a \in \mathcal{K}$, let us denote the top-M most related wiki entities of a as $\mathcal{R}_a = \langle e_1,\cdots,e_M\rangle$, where $e_1,\cdots,e_M \in \mathcal{E}$. Because concepts in $\mathcal{K}$ and entities in $\mathcal{E}$ are jointly embedded in the same vector space in Section 3.1, we can easily obtain $\mathcal{R}_a$ with the semantic relatedness metric $\omega(\cdot,\cdot)$ in Feature 1. We then define the **wikipedia reference weight** (Wrw) as follows.

$$Wrw\left(a,b\right)=\frac{\sum\limits_{e\in\mathcal{R}_a}Erw\left(e,b\right)\cdot\omega\left(e,a\right)}{\sum\limits_{e\in\mathcal{R}_a}\omega\left(e,a\right)} \tag{11}$$

where $Erw(e,a)$ is a binary indicator, in which $Erw(e,a) = 1$ if the Wikipedia article of e refers to any entity in $\mathcal{R}_a$, and $Erw(e,a) = 0$ otherwise. $Wrw\left(a,b\right)$ measures how frequently that a-related wiki entities refer to b-related wiki entities. Finally, **wikipedia reference distance** (Wrd) is

defined as the difference of Wrw between a and b, i.e., $Wrd(a,b) = Wrw(b,a) - Wrw(a,b)$.

3.3 Structural Features

Since course concepts are usually introduced based on their learning dependencies, the structure of MOOC courses also significantly contribute to prerequisite relation inference in MOOCs. However, structure-based features for prerequisite detection have not been well-studied in previous works. In this section, we investigate different structural information, including appearing positions of concepts, learning dependencies of videos and complexity levels of concepts, to propose three novel features to infer prerequisite relations in MOOCs. Before introducing these features, let us define two useful notations as follows. $\mathcal{C}(a)$ are the courses in which a is a course concept, i.e., $\mathcal{C}(a) = \{\mathcal{C}_i | \mathcal{C}_i \in \mathcal{D}, a \in \mathcal{K}_i\}$. $\mathcal{I}(\mathcal{C}, a)$ are the video indexes that contain concept a in course $\mathcal{C}$. For example, if a appears in the first and the 4-th video of $\mathcal{C}$, then $\mathcal{I}(\mathcal{C}, a) = \{1, 4\}$.

Feature 5: *Average Position Distance*

In a course, for a specific concept, its prerequisite concepts tend to be introduced before this concept and its subsequent concepts tend to be introduced after this concept. Based on this observation, for a concept pair $\langle a, b\rangle$, we calculate the distance of the average appearing position of a and b as one feature, namely **average position distance** (Apd). If $\mathcal{C}(a) \cap \mathcal{C}(b) \neq \emptyset$, $Apd(a,b)$ is formally defined as follows.

$$Apd(a,b) = \frac{\sum_{\mathcal{C} \in \mathcal{C}(a) \cap \mathcal{C}(b)} \left| \frac{\sum_{i \in \mathcal{I}(\mathcal{C},a)} i}{|\mathcal{I}(\mathcal{C},a)|} - \frac{\sum_{j \in \mathcal{I}(\mathcal{C},b)} j}{|\mathcal{I}(\mathcal{C},b)|} \right|}{|\mathcal{C}(a) \cap \mathcal{C}(b)|} \tag{12}$$

If $\mathcal{C}(a) \cap \mathcal{C}(b) = \emptyset$, we set $Apd(a,b) = 0$.

Feature 6: *Distributional Asymmetry Distance*

We also use the learning dependency of course videos to help infer learning dependency of course concepts. Based on our observation, the chance that a prerequisite concept is frequently mentioned in its subsequent videos is larger than that a subsequent concept is talked about in its prerequisite videos. Specifically, if video $\mathcal{V}_a$ is a precursor video of $\mathcal{V}_b$, and a is a prerequisite concept of b, then it is likely that $f(b, \mathcal{V}_a) < f(a, \mathcal{V}_b)$, where $f(a, \mathcal{V})$ denotes the term frequency of a in video $\mathcal{V}$. We thus define another feature, namely **distributional asymmetry distance** (Dad), to calculate the extent that a given concept pair satisfies this distributional asymmetry pattern. Formally, in course $\mathcal{C}$, for a given concept pair $\langle a, b\rangle$, we first define $\mathcal{S}(\mathcal{C}) = \{(i,j) | i \in \mathcal{I}(\mathcal{C}, a), j \in \mathcal{I}(\mathcal{C}, b), i < j\}$, i.e., all possible video pairs of $\langle a, b\rangle$ that have sequential relation. Then, the distributional asymmetry distance of $\langle a, b\rangle$ is formally defined as follows.

$$Dad(a,b) = \frac{\sum_{\mathcal{C} \in \mathcal{C}(a) \cap \mathcal{C}(b)} \frac{\sum_{(i,j) \in \mathcal{S}(\mathcal{C})} f(a, \mathcal{V}_i^{\mathcal{C}}) - f(b, \mathcal{V}_j^{\mathcal{C}})}{|\mathcal{S}(\mathcal{C})|}}{|\mathcal{C}(a) \cap \mathcal{C}(b)|} \tag{13}$$

where $\mathcal{V}_i^{\mathcal{C}}$ denotes the i-th video of course $\mathcal{C}$. If $\mathcal{C}(a) \cap \mathcal{C}(b) = \emptyset$, we set $Dad(a,b) = 0$.

Feature 7: *Complexity Level Distance*

Two related concepts with prerequisite relationship tend to have a difference in their complexity level, meaning that one concept is basic while another one is advanced. For example, "data set" and "training set" have learning dependencies and the latter concept is more advanced than the former one. However, "test set" and "training set" have no such relation when their complexity levels are similar. Complexity level of a course concept is implicit in its distribution in courses. Specifically, we observe that, for a concept in MOOCs, if it covers more videos in a course or it survives longer time in a course, then it is more likely to be a basic concept rather than an advanced one. We then formally define the **average video coverage** (avc) and the **average survival time** (ast) of a concept a as follows.

$$avc(a) = \frac{1}{|\mathcal{C}(a)|} \sum_{\mathcal{C} \in \mathcal{C}(a)} \frac{|\mathcal{I}(\mathcal{C}, a)|}{|\mathcal{C}|} \tag{14}$$

$$ast(a) = \frac{1}{|\mathcal{C}(a)|} \sum_{\mathcal{C} \in \mathcal{C}(a)} \frac{max(\mathcal{I}(\mathcal{C}, a)) - min(\mathcal{I}(\mathcal{C}, a)) + 1}{|\mathcal{C}|} \tag{15}$$

where $max/min(\mathcal{I}(\mathcal{C}, a))$ obtains the video index where a appears the last/first time in course $\mathcal{C}$. Based on the above equations, we define the **complexity level distance** (Cld) between concept a and b as follows.

$$Cld(a,b) = avc(a) \cdot ast(a) - avc(b) \cdot ast(b) \tag{16}$$

4 Experiments

4.1 Data Sets

In order to validate the efficiency of our features, we conducted experiments on three MOOC corpus with different domains: "Machine Learning" **(ML)**, "Data Structure and Algorithms" **(DSA)**, and "Calculus" **(CAL)**. To the best of our knowledge, there is no public data set for mining

Dataset	#courses	#videos	#concepts	#pairs		κ
				$-$	$+$	
ML	5	548	244	5,676	1,735	0.63
DSA	8	449	201	3,877	1,148	0.65
CAL	7	359	128	1,411	621	0.59

Table 1: Dataset Statistics

prerequisite relations in MOOCs. We created the experimental data sets through a three-stage process.

First, for each chosen domain, we select its relevant courses from Coursera, one of the leading MOOC platforms, and download all course materials using *coursera-dl* [2], a widely-used tool for automatically downloading Coursera.org videos. For example, for ML, we select 5 related courses [3] from 5 different universities and obtain a total of 548 course videos. Then, we manually label course concepts for each course: (1) Extract candidate concepts from documents of video subtitles following the method of Parameswaran et al. (2010). (2) Label the candidates as "course concept" or "not course concept" and obtain a set of course concepts for this course.

Finally, we manually annotate the prerequisite relations among the labeled course concepts. If the number of course concepts is n, the number of all possible pairs to be checked could reach $n \times (n-1)/2$, which requires arduous human labeling work. Therefore, for each dataset, we randomly select 25 percent of all possible pairs for evaluation. For each course concept pair $\langle a, b \rangle$, three human annotators majoring in the corresponding domain were asked to label them as "a is b's prerequisite", "b is a's prerequisite" or "no prerequisite relationship" using their own knowledge background and additional textbook resources. We take a majority vote of the annotators to create final labels and access the inter-annotator agreement using the average of pairwise κ statistics (Landis and Koch, 1981) between all pairs of the three annotators.

The statistics of the three datasets are listed in Table 1, where *#courses* and *#videos* are the total number of courses and videos in each dataset and *#concepts* is the number of labeled course concepts. The *#pairs* denotes the number of labeled concept pairs for evaluation, in which '$+$' denotes the number of positive instances, i.e. pairs who have prerequisite relations, and '$-$' denotes the number of negative instances.

[2]https://github.com/coursera-dl/coursera-dl

[3]These courses are: "Machine Learning (Stanford)", "Machine Learning (Washington)", "Practical Machine Learning (JHU)", "Machine Learning With Big Data (UCSD)" and "Neural Networks for Machine Learning (UofT)"

Classifier		ML		DSA		CAL	
	M	1	10	1	10	1	10
SVM	P	63.2	60.1	60.7	62.3	61.1	61.9
	R	68.5	72.4	**69.3**	67.5	**67.9**	68.3
	F_1	65.8	65.7	64.7	64.8	64.3	64.9
NB	P	58.0	58.2	62.9	62.6	60.1	60.6
	R	58.1	60.5	62.3	61.8	61.2	62.1
	F_1	58.1	59.4	62.6	62.2	60.6	61.3
LR	P	66.8	67.6	63.1	62.0	62.7	63.3
	R	60.8	61.0	64.8	66.8	63.6	64.1
	F_1	63.7	64.2	63.9	64.3	61.6	62.9
RF	P	**68.1**	**71.4**	**69.1**	**72.7**	**67.3**	**70.3**
	R	**70.0**	**73.8**	68.4	**72.3**	67.8	**71.9**
	F_1	**69.1**	**72.6**	**68.7**	**72.5**	**67.5**	**71.1**

Table 2: Classification results of the proposed method(%).

4.2 Evaluation Results

For each dataset, we apply 5-fold cross validation to evaluate the performance of the proposed method, i.e., testing our method on one fold while training the classifier using the other 4 folds. Usually, there are much fewer positive instances than negative instances, so we balance the training set by oversampling the positive instances (Yosef et al., 2011; Talukdar and Cohen, 2012). In our experiments, we employ 4 different binary classifiers, including NaïveBayes (*NB*), Logistic Regression (*LR*), SVM with linear kernel (*SVM*) and Random Forest (*RF*). We use precision (P), recall (R), and F1-score (F_1) to evaluate the prerequisite classification results. The experimental results are presented in Table 2.

Contextual features are shaped by the parameter M, i.e., the number of related concepts being considered. In our experiments, we tried different settings of M and report the results when M=1 and M=10 in Table 2. As for the semantic representation, we use the latest publicly available Wikipedia dump [4] and apply the skip-gram model (Mikolov et al., 2013b) to train word embeddings using the Python library gensim [5] with default parameters.

As shown in Table 2, the evaluation results varies by different classifiers. It turns out that NaïveBayes performs the worst. This seems to be caused by the fact that the independence assumption is not satisfied for our features; for

[4]https://dumps.wikimedia.org/enwiki/20170120/

[5]http://radimrehurek.com/gensim/

example, Feature 2 and Feature 3 both utilize the local context information, only with different granularity, thus are quite co-related. Random Forest beats others, with best F_1 across all three datasets. Its average F_1 outperforms SVM, NB and LR by 7.0%, 11.1% and 8.3%, respectively (M=10). The reason is as follows. Instead of a simple descriptive feature, each of our proposed feature determines whether a concept pair has prerequisite relation from a specific aspect; its function is similar to an independent weak classifier. Therefore, rather than using a linear combination of features for classification (e.g., SVM and LR), a boosting model (e.g., Random Forest) is more suitable for this task. The performance is slightly better when M=10 for all classifiers, with +0.20% for SVM, +0.53% for NB, +0.73% for LR and +3.63% for RF, with respect to the average F_1. The results verify the effectiveness of considering related concepts in contextual features. We use RF and set M=10 in the following experiments.

4.3 Comparison with Baselines

We further compare our approach with three representative methods for prerequisite inference.

4.3.1 Baseline Approaches

Hyponym Pattern Method (HPM). Prerequisite relationships often exists between hyponym-hypernym concept pairs (e.g., "Machine Learning" and "Supervised Learning"). As a baseline, we adopt the 10 lexico-syntactic patterns used by Wang et al. (2016) to extract hyponym relationships between concepts. If a concept pair matches at least one of these patterns in the MOOC corpus, we judge them to have prerequisite relations.

Reference Distance (RD) We also employ the RefD proposed by Liang et al. (2015) as one of our baselines. However, this method is only appliable to Wikipedia concepts. To make it comparable with our method, for each of our datasets, we construct a subset of it by picking out the concept pairs $\langle a, b\rangle$ in which a and b are both Wikipedia concepts. For example, we find 49% of course concepts in ML have their corresponding Wikipedia articles and 28% percent of concept pairs in ML meet the above condition. We use the new datasets constructed from ML, DSA and CAL, namely **W-ML**, **W-DSA**, and **W-CAL**, to compare our method with RefD.

Supervised Relationship Identification (SRI) Wang et al. (2016) has employed several features to infer prerequisite relations of Wikipedia concepts in textbooks, including 3 Textbook features and 6 Wikipedia features. Based on these features, they performed a binary classification using SVM to identify prerequisite relationships and has achieved state-of-the-art results. Because the Wikipedia features can only be applied to Wikipedia concepts, in order to make a comparison, we create two versions of their method: (1) **T-SRI**: only textbook features are used to train the classifier and (2) **F-SRI**: the original version, all features are used. We compare the performance of our method with T-SRI on ML, DSA and CAL datasets; we also compare our method with F-SRI on W-ML, W-DSA and W-CAL datasets.

Method		ML	DSA	CAL	W-ML	W-DSA	W-CAL
HPM	P	67.3	71.4	69.5	**79.9**	72.3	73.5
	R	18.4	14.8	16.5	25.5	27.3	23.3
	F_1	29.0	24.5	26.7	38.6	39.6	35.4
RD	P	–	–	–	73.4	**77.8**	**74.4**
	R	–	–	–	42.8	44.8	43.1
	F_1	–	–	–	54.1	56.8	54.6
T-SRI	P	61.4	62.3	62.5	58.1	60.1	62.7
	R	62.9	64.6	65.5	67.6	65.3	67.9
	F_1	62.1	63.4	64.0	62.5	62.6	65.2
F-SRI	P	–	–	–	64.3	64.3	64.8
	R	–	–	–	62.1	65.6	65.2
	F_1	–	–	–	63.2	64.9	65.0
MOOC	P	**71.4**	**72.7**	**70.3**	72.8	68.4	71.4
	R	**73.8**	**72.3**	**71.9**	**71.3**	**72.0**	**70.8**
	F_1	**72.6**	**72.5**	**71.1**	**72.0**	**70.2**	**71.1**

Table 3: Comparison with baselines(%).

4.3.2 Performance Comparison

In Table 3 we summarize the comparing results of different methods across different datasets ("MOOC" refers to our method). We find that our method outperforms baseline methods across all six datasets [6]. For example, the F_1 of our method on ML outperforms T-SRI and HPM by 10.5% and 43.6%, respectively. Specifically, we have the following observations. First, HPM achieves relatively high precision but low recall. This is because when A "is a" B, a prerequisite relation often exists from B to A, but clearly not vise versa. Second, T-SRI has certain effectiveness for learning prerequisite relations, with F_1 ranging from 62.1 to 65.2%. However, T-SRI only considers relatively simple features, such as the sequential and co-occurrence among concepts. With more

[6]The improvements are all statistically significant tested with bootstrap re-sampling with 95% confidence.

comprehensive feature engineering, the F_1 of our method significantly outperforms T-SRI (+10.5% on ML, +9.1% on DSA and +7.1% on CAL). Third, incorporating Wikipedia-based features (F-SRI) achieves certain promotion in performance (+0.93% comparing with T-SRI in average F_1).

4.4 Feature Contribution Analysis

In order to get an insight into the importance of each feature in our method, we perform a contribution analysis with different features. Here, we run our approach 10 times on the ML dataset. In each of the first 7 times, one feature is removed; in each of the rest 3 times, one group of features are removed, e.g., removing contextual features means removing $Gvrd$, $Gsrd$ and Wrd at the same time. We record the decrease of F1-score for each setting. Table 4 lists the evaluation results after ignoring different features.

According to the decrement of F1-scores, we find that all the proposed features are useful in predicting prerequisite relations. Especially, we observe that Cld (Feature 7), decreasing our best F1-score by 7.4%, plays the most important role. This suggests that most concepts do exist difference in complexity level. For two concepts, the difference of their coverage and survival times in courses are important for prerequisite relation detection. On the contrary, with 1.9% decrease, Sr (Feature 1) is relatively less important. We may easily find two concepts which have related semantic meanings (e.g., "test set" and "training set") but have no prerequisite relationship. However, semantic relatedness is critical for the contextual features because it overcomes the problem of the sparsity of context in calculation. We experience a decrease of 5.4% when we further do not consider related concepts in contextual features, i.e., set M=1. As for the feature group contribution, we observe that Structural Features, with a decrease of 9.2%, has a greater impact than the other two groups. This is as expected because it includes Cld. Among the three structural features, Apd makes relatively less contribution. The reason is that sometimes the professor may frequently mention a prerequisite concept after introducing a subsequent concept orally, for helping students better understand the concept.

5 Related Works

To the best of our knowledge, there has been no previous work on mining prerequisite relations among concepts in MOOCs. Some researchers have been engaged in detecting other type of prerequisite relations. For example, Yang et al. (2015) proposed to induce prerequisite relations among courses to support curriculum planning. Liu et al. (2011) studied learning-dependency between knowledge units, a special text fragment containing concepts, using a classification-based method. In the area of education, researchers have tried to find general prerequisite structures from students' test performance (Vuong et al., 2011; Scheines et al., 2014; Huang et al., 2015). Different from them, we focus on more fine-grained prerequisite relations, i.e., the prerequisite relations among course concepts.

Among the few related works of mining prerequisite relations among concepts, Liang et al. (2015) and Talukdar and Cohen (Talukdar and Cohen, 2012) studied prerequisite relationships between Wikipedia articles. They assumed that hyperlinks between Wikipedia pages indicate a prerequisite relationship and design several useful features. Based on these Wikipedia features plus some textbook features, Wang et al. (Wang et al., 2016) proposed a method to construct a concept map from textbooks, which jointly learns the key concepts and their prerequisite relations. However, the investigation of only Wikipedia concepts is also the bottleneck of their studies. In our work, we propose more general features to infer prerequisite relations among concepts, regardless of whether the concept is in Wikipedia or not. Liang et al. (2017) propose an optimization based framework to discover concept prerequisite relations from course dependencies. Gordon et al. (2016) utilize cross-entropy to learn concept dependencies in scientific corpus. Besides local statistical information, our method also utilize external knowledge to enrich concept semantics, which is more informativeness.

	Ignored Feature(s)	P	R	F_1
Single	Sr	69.6	72.9	71.2(**-1.4**)
	GVrd	68.8	71.4	70.1(-2.5)
	GSrd	67.9	71.4	69.6(-3.0)
	Wrd	70.1	72.1	71.1(-1.5)
	Apd	69.7	70.8	70.2(-2.4)
	Dad	69.2	69.5	69.4(-3.2)
	Cld	64.9	65.6	65.2(**-7.4**)
Group	Semantic	69.6	72.9	71.2(**-1.4**)
	Contextual	66.4	68.9	67.6(-5.0)
	Structural	63.7	64.2	63.4(**-9.2**)

Table 4: Contribution analysis of different features(%).

Our work is also related to the study of automatic relation extraction. Different research lines have been proposed around this topic, including hypernym-hyponym relation extraction (Ritter et al., 2009; Wei et al., 2012), entity relation extraction (Zhou et al., 2006; Fan et al., 2014; Lin et al., 2015) and open relation extraction (Fader et al., 2011). However, previous works mainly focus on factual relations, the extraction of cognitive relations (e.g. prerequisite relations) has not been well studied yet.

6 Conclusions and Future Work

We conducted a new investigation on automatically inferring prerequisite relations among concepts in MOOCs. We precisely define the problem and propose several useful features from different aspects, i.e., contextual, structural and semantic features. Moreover, we apply an embedding-based method that jointly learns the semantic representations of Wikipedia concepts and MOOC concepts to help implement the features. Experimental results on online courses with different domains validate the effectiveness of the proposed method. Promising future directions would be to investigate how to utilize user interaction in MOOCs for better prerequisite learning, as well as how deep learning models can be used to automatically learn useful features to help infer prerequisite relations.

Acknowledgments

This work is supported by 973 Program (No. 2014CB340504), NSFC Key Program (No. 61533018), Fund of Online Education Research Center, Ministry of Education (No. 2016ZD102), Key Technologies Research and Development Program of China (No. 2014BAK04B03) and NSFC-NRF (No. 61661146007).

References

Benjamin Samuel Bloom. 1981. *All our children learning: A primer for parents, teachers, and other educators*. McGraw-Hill Companies.

Yixin Cao, Juanzi Li, Xiaofei Guo, Shuanhu Bai, Heng Ji, and Jie Tang. 2015. Name list only? target entity disambiguation in short texts. In *Proceedings of EMNLP*. pages 654–664.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*. pages 1535–1545.

Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y. Chang. 2014. Distant supervision for relation extraction with matrix completion. In *Proceedings of ACL*. pages 839–849.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of CIKM*. pages 1625–1628.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of ACL*.

Xiaopeng Huang, Kyeong Yang, and Victor B. Lawrence. 2015. An efficient data mining approach to concept map generation for adaptive learning. In *Proceedings of ICDM*. pages 247–260.

James Gregory Jardine. 2014. *Automatically generating reading lists*. Ph.D. thesis, University of Cambridge, UK.

RJ Landis and GG Koch. 1981. The measurement of interrater agreement. *Statistics methods for rates and proportions* 2:212–236.

Stephen Laurence and Eric Margolis. 1999. Concepts and cognitive science. *Concepts: core readings* pages 3–81.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. 2015. Measuring prerequisite relations among concepts. In *Proceedings of EMNLP*. pages 1668–1674.

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

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*. pages 2181–2187.

Jun Liu, Lu Jiang, Zhaohui Wu, Qinghua Zheng, and Ya-nan Qian. 2011. Mining learning-dependency between knowledge units from text. *The VLDB Journal* 20(3):335–345.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *International Journal of CoRR* abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*. pages 3111–3119.

Joseph D. Novak. 1990. Concept mapping: A useful tool for science education. *International Journal of Research in Science Teaching* 27(10):937C949.

Chitu Okoli, Mohamad Mehdi, Mostafa Mesgari, Finn Årup Nielsen, and Arto Lanamäki. 2014. Wikipedia in the eyes of its beholders: A systematic review of scholarly research on wikipedia readers and readership. *International Journal of the American Society for Information Science and Technology (JASIST)* 65(12):2381–2403.

Aditya G. Parameswaran, Hector Garcia-Molina, and Anand Rajaraman. 2010. Towards the web of concepts: Extracting concepts from large datasets. *Proceedings of the VLDB Endowment (PVLDB)* 3(1):566–577.

Alan Ritter, Stephen Soderland, and Oren Etzioni. 2009. What is this, anyway: Automatic hypernym discovery. In *Proceedings of AAAI*. pages 88–93.

Jean Michel Rouly, Huzefa Rangwala, and Aditya Johri. 2015. What are we teaching?: Automated evaluation of CS curricula content using topic modeling. In *Proceedings of ICER*. pages 189–197.

Richard Scheines, Elizabeth Silver, and Ilya M. Goldin. 2014. Discovering prerequisite relationships among knowledge components. In *Proceedings of EDM*. pages 355–356.

Nick J Schweitzer. 2008. Wikipedia and psychology: Coverage of concepts and its use by undergraduate students. *International Journal of Teaching of Psychology* 35(2):81–85.

Partha Pratim Talukdar and William W Cohen. 2012. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. pages 307–315.

Annalies Vuong, Tristan Nixon, and Brendon Towle. 2011. A method for finding prerequisites within a curriculum. In *Proceedings of EDM*. pages 211–216.

Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 2016. Using prerequisites to extract concept maps fromtextbooks. In *Proceedings of CIKM*. pages 317–326.

Bifan Wei, Jun Liu, Jian Ma, Qinghua Zheng, Wei Zhang, and Boqin Feng. 2012. MOTIF-RE: motif-based hypernym/hyponym relation extraction from wikipedia links. In *Proceedings of ICONIP*. pages 610–619.

Yiming Yang, Hanxiao Liu, Jaime G. Carbonell, and Wanli Ma. 2015. Concept graph learning from educational data. In *Proceedings of WSDM*. pages 159–168.

Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. 2011. AIDA: an online tool for accurate disambiguation of named entities in text and tables. *Proceedings of the VLDB Endowment (PVLDB)* 4(12):1450–1453.

Guodong Zhou, Jian Su, and Min Zhang. 2006. Modeling commonality among related classes in relation extraction. In *Proceedings of ACL*.