Extracting Prerequisite Relations among Concepts from the Introduction of Online Courses

Zesong Wang
School of Computer Science and Information Engineering
Hubei University
Wuhan, China
201822111920089@stu.hubu.edu.cn

Kui Xiao*
School of Computer Science and Information Engineering
Hubei University
Wuhan, China
xiaokui1024@hotmail.com
(*Corresponding author)

Zeqing Qin
School of Computer Science
Hubei University of Technology
Wuhan, China
qzqhbut@163.com

Shihui Wang
School of Computer Science and Information Engineering
Hubei University
Wuhan, China
wsh@hubu.edu.cn

***Abstract*—Affected by the COVID-19 pandemic, teaching tasks have gradually shifted from offline to online, which expanded online education resources unprecedentedly. "Concept" is a professional vocabulary in the curriculum. Exploring the prerequisite relations among concepts is of great significance to educational planning. This research extracts concepts from online course introduction and proposes a mixed method for extracting concept prerequisite relations. Experiments on public data set show that this method outperforms existing ones. Tests were also carried out on the datasets of eleven schools, which proves that this model has good scalability.**

Keywords- prerequisite relations; educational planning; online education resources

I. INTRODUCTION

Concepts are professional vocabulary covered in the course. Usually, dependencies among concepts exist. This is called prerequisite relations. In recent years, the extracting of prerequisite relations among concepts has become a focus of researchers. Prerequisite relations among concepts have played a significant role in many applied fields of smart education, such as curriculum planning and design [1,2], student knowledge status tracking [3], concept map building [4,5], learner ranking [6,7,8], document reading list generation [9,10], and so on.

With the rise of online education platforms such as MOOC, many universities launched their own courses on it, making online education resources richer in recent years. To make it easier for students choosing courses, online courses are generally equipped with a course introduction, which highlights the key knowledge of the course by condensing its core content. Based on concepts extracted from the online course introduction, a dependency extracting research was carried out. This research proposes a mixed method for extracting prerequisite relations among concepts. By analyzing the course introduction, the attributes of Wikipedia articles in accordance with corresponding concept, 10 different features are built and used to analyze whether prerequisite relations exist.

The structure of this article is as follows: Section 2 reviews the related work of prerequisite relation mining; Section 3 introduces the method of this article and constructs 10 different features. Section 4 conducts an experimental exploration of the proposed method. Finally, a conclusion of this research is drawn in Section 5.

II. RELATED WORK

Talukdar et al. [11] first study prerequisite relations mining between Wikipedia concepts. The author believes that if the Wikipedia article of concept B contains a link to concept A, then A may contain some background knowledge needed to learn before view B, which means A is a prerequisite of B. For these linked concept pairs, the author uses the MaxEnt classifier to predict the prerequisite relations among them.

Liang, C. et al. [12] propose a method based on concept reference distance (RefD) to predict the relations between two Wikipedia concepts. Specifically, each concept in Wikipedia can be replaced by its "set of related concepts". If most of the concepts in the "set of related concepts" of concept B contain a link to concept A, and concept B is rarely cited by the "set of related concepts" of concept A. Then concept A may be a prerequisite of concept B. Zhou, Y et al. [13] use machine learning methods to predict the prerequisite relationship of Wikipedia concepts. The author establishes four sets of features of concept pairs, including link-based, category-based, text-based, and time-based features, six different classifiers are used for experiments. Sayyadiharikandeh et al. [14] propose a method for inferring the prerequisite relation between concepts based on Wikipedia clickstream data. Clickstream is the user's operation log on the Wikipedia platform. This is the first time that researchers have used user interaction behavior to predict the prerequisite relation between concept pairs.

The above methods are all based on the content of the Wikipedia article. Besides, some researchers carry out research on the recognition of the curriculum concept prerequisite relations based on learning resources. Some analyzes the prerequisite relations between the curriculum concepts in the MOOC video [15]. Liang, C et al. [16] analyze the content of the university curriculum introduction to extract the main concepts and infer the prerequisite relationship between them, which is closely related to this research. However, the author only considers the influence of course attributes on prerequisite relations. In this research, course attributes and Wikipedia attributes are all adopted to identify the prerequisite relations between those concepts.

DOI reference number: 10.18293/SEKE2021-172

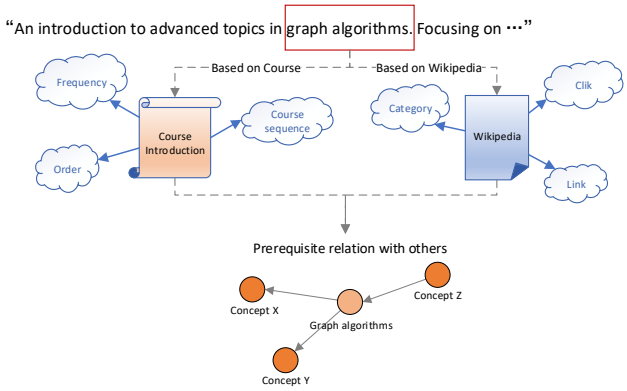

Figure1. The overview of the flow chart in this research

III. METHOD

A concept has many different attributes, such as the frequency and order in the online course introduction. In Wikipedia, each concept is an article with its own content. Links, clickstreams, classification, and other attributes of that article can be used for prediction. To better explore the prerequisite relations between these concepts, features are designed from the two aspects of "Course-based attributes" and "Wikipedia-based attributes". In Fig.1, "graph algorithm" is a concept extracted from course introduction. Fig.1 shows the flow chart of our method.

A. Course-based attributes

In this part, features are designed by using the frequency, position of the concept in the online course introduction, and the learning order between courses. Four features are included in this part. The description of elements is defined as Table I.

TABLE I. ELEMENTS RELATED TO COURSE-BASED ATTRIBUTES

Elements	Description
C_i	One course
w_a	One concept
$Tid(C_i, w_a)$	The *tf-idf* value of w_a in C_i 's course introduction
$con(C_i)$	Set of concepts extracted from C_i 's course introduction
$exist(w_a)$	Course that have w_a in the introduction
$order(w_b, w_a)$	The course whose position of w_b is before the position of w_a in the introduction
$r(C_i, w_a)$	Whether w_a is an important concept of C_i
$Z(C_i, C_j)$	Whether course C_i depends on course C_j

- *Concept appearance*

The introduction of a course can be viewed as a series of "concepts", $exist(w_a)$ represents courses that have w_a in the introduction, $Ca(w_a, w_b)$ represents the probability that w_b appears in the course introduction where w_a appears.

$$Ca(w_a, w_b) = \frac{|exist(w_a) \cap exist(w_b)|}{|exist(w_a)|} \quad (1)$$

When introducing a new concept, the leading concept is introduced at the same time, which is also regarded as background knowledge. So when $Ca(w_a, w_b)$ is larger and $Ca(w_b, w_a)$ is smaller, w_a is more likely to depend on w_b. That is to say, the frequency of w_b in the course introduction that has w_a is very high. On the contrary, w_b as background knowledge, the frequency of w_a in the course introduction has w_b is not obvious. This also reasonably fits the general laws of cognition. Based on this assumption, the first feature is proposed to be "$Caf(w_a, w_b)$":

$$Caf(w_a, w_b) = Ca(w_a, w_b) - Ca(w_b, w_a) \quad (2)$$

- *Concept order*

The concepts contained in the course introduction can be regarded as an ordered list, and the position of each concept can be viewed as its index number.

The same concept may appear multiple times in a course introduction, the first appearance of the concept is taken as its position in this course introduction. $order(w_b, w_a)$ represents the course whose position of w_b is before the position of w_a in the introduction.

$$Co(w_a, w_b) = \frac{|order(w_b, w_a)|}{|exist(w_a) \cap exist(w_b)|} \quad (3)$$

In a course introduction, we believe that w_b is more likely to become the background knowledge of w_a, if the probability of w_b appearing before w_a is higher, and the probability of w_a appearing before w_b is smaller. The second feature is proposed to be "$Cof(w_a, w_b)$":

$$Cof(w_a, w_b) = Co(w_a, w_b) - Co(w_b, w_a) \quad (4)$$

- *Concept in course*

Each course has a corresponding introduction. The course C_i can be represented by a vector on the concept space $(w_1, w_2, ..., w_m)$. The value in the vector is the *tf-idf* value of the different concepts in the C_i's course introduction. E.g:

$$C_1 = \{0, 0.23, 0.014, 0, 0.56, 0, ..., 0.13, 0\}$$

Assuming that w_a appears in the course introduction of C_i, w_b appears in the course introduction of C_j, and C_j needs to be studied before learning C_i, this course sequence possibly means that w_a depends on w_b. For two concepts, if they appear in multiple course pairs with a fixed sequence, then the relationship between these two concepts can be expressed as (5):

$$Cr(w_a, w_b) = \sum_{i=1}^{n} \sum_{j=1}^{n} r(C_i, w_a) \cdot r(C_j, w_b) \cdot z(C_i, C_j) \quad (5)$$

$r(C_i, w_a)$ indicates whether w_a is an important concept of C_i. When the *tf-idf* value of w_a in C_i is greater than a specified

threshold, it is an important concept of C_i. In case of that, the value of $r(C_i,w_a)$ is 1, otherwise the value is 0. For course C_i, this threshold is defined as the average value of *tf-idf* of the concepts contained in C_i's introduction.

$$r(C_i,w_a)=\begin{cases}1, & \text{if } Tid(C_i,w_a)>\dfrac{\sum_{w_j\in con(C_i)} Tid(C_i,w_j)}{|con(C_i)|}\\ 0, & \text{else}\end{cases} \quad (6)$$

$Z(C_i,C_j)$ represents whether C_i depends on C_j, and the value is 1 or 0, where 1 means that you need to study C_j before learning C_i, and 0 indicates other cases. The third feature "$Crf(w_a,w_b)$" is defined as (7):

$$Crf(w_a,w_b)=Cr(w_a,w_b)-Cr(w_b,w_a) \quad (7)$$

- *Concept related to course*

Because the content of the online course introduction is often limited, some concepts that are closely related to the course cannot appear in the course introduction. For example, "knapsack problem" is the concept often explained in course "Algorithm Design and Analysis", but the online course introduction may only include more coarse-grained concept like "dynamic programming method". Therefore, we have to establish connections between the course and concepts that are not included in its introduction.

For a course C_i and a concept w_a, w_a does not appear in the introduction of C_i. However, w_a may has a strong connection with the concepts in the introduction of the course, their relevance can be expressed as (8):

$$t(C_i,w_a)=\frac{\sum_{w_j\in con(C_i)} \frac{|exist(w_a)\cap exist(w_j)|}{|exist(w_a)|}\cdot Tid(C_i,w_j)}{\sum_{w_j\in con(C_i)} Tid(C_i,w_j)} \quad (8)$$

w_j is a concept extracted from the course introduction of C_i. The more frequent the appearance of w_a and w_j in same time, the higher the correlation between w_a and w_j is. Compare with all concepts included in C_i, $t(C_i,w_a)$ describes a relevance between the concept w_a and the course C_i.

If two concepts w_a and w_b are in correspondence to such courses respectively, and there is a fixed order relationship between the course pair, then the relationship between these two concepts can be expressed as (9):

$$Cs(w_a,w_b)=\sum_{i=1}^{n}\sum_{j=1}^{n} t(C_i,w_a)\cdot t(C_j,w_b)\cdot z(C_i,C_j) \quad (9)$$

On this basis, the fourth feature "$Csf(w_a,w_b)$" is defined as (10):

$$Csf(w_a,w_b)=Cs(w_a,w_b)-Cs(w_b,w_a) \quad (10)$$

B. *Wikipedia-based attributes*

The attributes of concepts in Wikipedia are also used to identify the prerequisite relations between different concepts. Liang et al. [12] propose the idea "set of related concepts" for the first time and believe that for a pair of concepts, if there is a prerequisite relation between their related concept sets, it means that there is also a prerequisite relation between the two concepts.

We have innovated this approach. For w_a, we regard the concepts that both belong to the same Wikipedia category (Category) as w_a and has a link relationship with w_a as the related concept sets of w_a, denoted as $S(w_a)$.

In what follows, we consider the prerequisite relation between concepts from the perspectives of "concept to concept", "concepts to set" and "set to set" respectively. Some elements used in this section are defined as Table II.

TABLE II. ELEMENTS RELATED TO WIKIPEDIA ATTRIBUTES

Elements	Description
$S(w_a)$	Related concept sets of w_a
$C_{\rightarrow}^{(w_a)}$	The set of concepts have clickstream point from w_a
$C_{\leftarrow}^{(w_a)}$	The set of concepts have clickstream point to w_a

- *Category information in Wikipedia*

In Wikipedia, each concept belongs to one or more categories. If the level of one category is above the level of another, the higher-level category usually contains more abstract concepts while the lower-level category usually contains concepts that are more concrete. These concrete concepts often rely on abstract concepts [1,13]. Therefore, we design the following features.

1) Concept to Concept

root represents the root node in the Wikipedia category. $len(root,w_a)$ represents the shortest path length from w_a to root node, which is also the level of the concept in the Wikipedia system. As is shown in Fig.2, $len(root,w_a)$=2, $len(root,w_b)$=2, $len(root,w_c)$=3.

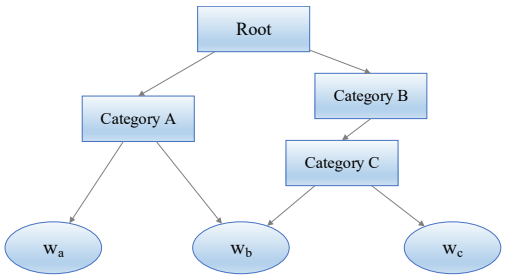

Figure 2. Level of concepts in Wikipedia classification

The larger the value of $len(root,w_a)$, the more concrete the value of w_a is ; The smaller the value of $len(root,w_a)$, the more abstract the content of w_a is . If $len(root,w_b)$ is smaller than $len(root,w_a)$, it means that w_b has a higher level than w_a and its

content is more abstract than w_a. Then w_b may be a prerequisite of w_a, so we define the fifth feature "$Waf(w_a, w_b)$" as (11):

$$Waf(w_a, w_b) = len(root, w_a) - len(root, w_b) \quad (11)$$

2) Concept to Set

The average level between related concept set and concept are explored by using "the set of related concepts". If the level of the set of w_a's related concepts is below w_b on average, and the level of the set of w_b's related concepts is above w_a on average, then we think that w_b is more likely to a prerequisite of w_a. And the sixth feature "$Waf^*(w_a, w_b)$" is defined as (12):

$$Waf^*(w_a, w_b) = \frac{\sum_{w_i \in S(w_a)} Waf(w_i, w_b)}{|S(w_a)|} - \frac{\sum_{w_j \in S(w_b)} Waf(w_i, w_a)}{|S(w_b)|} \quad (12)$$

3) Set to Set

We also design features between sets. We consider that for a pair of concepts (w_a, w_b), if the average level of w_a's related concept set is below the average level of w_b's related concept set, then w_b is more likely to be a perquisite of w_a, So we define the seventh feature "$Waf^{**}(w_a, w_b)$" as (13)

$$Waf^{**}(w_a, w_b) = \frac{\sum_{w_i \in S(w_a)} len(root, w_i)}{|S(w_a)|} - \frac{\sum_{w_j \in S(w_b)} len(root, w_j)}{|S(w_b)|} \quad (13)$$

- *Clickstream in Wikipedia*

Wikipedia usually publishes user clickstream data logs in the past 30 months[1]. Clickstream refers to the action that a user browses a Wikipedia article immediately after another article. User usually clicks on the link of one article to jump to another article to continue brows in, those two articles are often closely related [14]. Clickstream records data of this type.

1) Concept to Concept

After browsing a Wikipedia article for a concept, people often continue to browse other related concepts to view the background knowledge. If there is a clickstream from w_a to w_b, w_b may be a prerequisite of w_a. Therefore, we define the eighth feature "$Wkf(w_a, w_b)$" to identify the prerequisite relations between concepts.

$$Wkf(w_a, w_b) = \begin{cases} 1, & w_b \in C_{\rightarrow}^{(w_a)} \\ 0, & \text{else} \end{cases} \quad (14)$$

2) Concept to Set

Liang et al. [12] use the link in the Wikipedia article of the concept to identify the dependency between concepts. We improve this approach with more precise clickstreams. Clickstream data is different from links, which are made by real users. Since people tend to browse the background knowledge when browsing new knowledge, we believe that if most of the w_a's related concepts having a clickstream pointing to w_b, but w_b is the opposite, then w_b maybe a background knowledge of w_a, w_b is more likely to be a prerequisite of w_a. Therefore, we define the ninth feature "$Wkf^*(w_a, w_b)$" as follows:

$$Wkf^*(w_a, w_b) = \frac{\sum_{w_i \in S(w_a)} Wkf(w_i, w_b)}{|S(w_a)|} - \frac{\sum_{w_j \in S(w_b)} Wkf(w_i, w_a)}{|S(w_b)|} \quad (15)$$

3) Set to Set

From the perspective of set to set relations, we use $Out(w_a, w_b)$ to indicate the intersection of all clickstream from $S(w_a)$ and $S(w_b)$; $In(w_a, w_b)$ represents the intersection of all clickstream to $S(w_a)$ and $S(w_b)$.

$$Out(w_a, w_b) = \left| \left(\bigcup_{w_i \in S(w_a)} C_{\rightarrow}^{(w_i)} \right) \cap S(w_b) \right| \quad (16)$$

$$In(w_a, w_b) = \left| \left(\bigcup_{w_i \in S(w_a)} C_{\leftarrow}^{(w_i)} \right) \cap S(w_b) \right| \quad (17)$$

If $Out(w_a, w_b)$ is larger and $In(w_a, w_b)$ is smaller, it means that users often browse related concepts of w_b after browsing related concepts of w_a, but rarely continue to browse related concepts of w_a after browsing the related concepts of w_b, which shows that w_b may be a prerequisite of w_a. So we define the tenth feature as "$Wkf^{**}(w_a, w_b)$".

$$Wkf^{**}(w_a, w_b) = Out(w_a, w_b) - In(w_a, w_b) \quad (18)$$

IV. EXPERIMENT

A. Datasets

Liang et al. [16] crawled the data of 654 computer science courses from the online course websites of 11 Well-known universities, which include the introduction of each course and the learning order between courses[2]. Among these courses, 861 pairs of courses have fixed learning sequence. We will verify the proposed method on this data set. The data set were cleaned, we obtain 1312 concept pairs with dependencies and 2448 concept pairs without dependencies. Table 3 shows the detailed information of this set, "Concept prerequisite relations" represents the number of concept pairs that have dependencies.

TABLE 3. UNIVERSITY COURSE DATA SET

Universities	#Courses	#Course pairs	#Concept prerequisite relations
Caltech	41	56	461
Illinois	72	97	554
CMU	65	78	618
Iowa	38	50	395
Maryland	34	54	455

1 https://dumps.wikimedia.org/other/clickstream/

2 https://github.com/harrylclc/eaai17-cpr-recover

MIT	165	220	712
MSU	33	59	390
Princeton	16	20	292
PSU	77	98	479
Purdue	20	16	282
Stanford	93	113	711

B. Evaluation Results

This research uses five cross-validations to evaluate the proposed method. Six commonly used machine learning classifiers were used to predict the prerequisite relations among concepts. They are Random Forest (RF), Naive Bayes (NB), Multilayer Perceptron (MLP), and Support Vector Machine (SVM), Logistic Regression (LR), and AdaBoost. All classifiers are implemented using python program and sklearn library, and the parameters are default ones. Specific experimental results are shown in Table 4.

It can be seen from Table 4 that the prediction results of different classifiers are quite different. The random forest classifier has the best performance. It is better than other classifiers in metrics such as Accuracy, Precision, Recall, and F1, reaching 84.18%, 80.66%, 73.18%, and 76.26%, respectively, which is similar to the conclusions of related studies [13,15].

Support vector machines (SVM) performs poorly, with index values such as Recall and F1 being only 15.13% and 24.16%. It is estimated that because the feature values are all specific values, and the range of these values is quite different, it is difficult to form a better hyperplane in the two types of samples to classify the samples. We will use random forests for following experiments.

TABLE 4 CLASSIFICATION RESULTS OF THE METHOD PROPOSED IN THIS PAPER(%)

Classifier	Accuracy	Precision	Recall	F1
RF	**84.18**	**80.66**	**73.18**	**76.26**
NB	73.54	55.48	66.67	47.51
MLP	74.20	55.09	69.59	45.59
SVM	69.90	60.00	15.13	24.16
LR	75.23	66.67	43.70	52.79
AdaBoost	74.70	63.33	47.90	54.55

C. Comparison With Baselines

We select three baseline methods for comparison. The first is the method of calculating the concept reference distance (RefD) proposed in [12]. The author used two ways to define the weight of each related concept. One is equal (the weight of all related concepts is 1), the other is tf-idf (the weight of all related concepts is their tf-idf value). We name them "RefD-equal" and "RefD-tfidf".

The second method is a concept dependency recognition method based on optimization technology(CPR) proposed in [16]. This is the first time that the course learning sequence is used to calculate the prerequisite relations of concepts. The third method is the concept prerequisite relations prediction method (EPR) proposed in [13] using features such as link, category, text, and creation time. The specific experimental results are shown in Table 5.

TABLE 5 COMPARISON WITH BASELINE METHOD(%)

	RefD-equal	RefD-tfidf	CPR	EPR	Proposed method
Accuracy	62.257	60.651	50.927	79.755	**84.176**
Precision	65.490	64.223	56.083	72.044	**80.658**
Recall	62.437	64.166	62.277	68.685	**73.180**
F1	63.254	64.194	55.953	70.291	**76.264**

The proposed method outperforms other methods in all metrics. It should be noted that neither RefD nor CPR uses machine learning classifiers in the classification task. The EPR method uses conventional classifiers to classify as in this article, and their performances are significantly better than RefD and CPR methods, which shows that artificially established features can indeed provide effective help for the recognition of dependencies between concepts.

D. Analysis of Feature Contribution

TABLE 6 FEATURE CONTRIBUTION ANALYSIS(%)

Course-based attributes	$Caf(w_a, w_b)$	**80.701(-4.13)**	79.610 (-5.42)	79.610 (-5.42)
	$Cof(w_a, w_b)$	81.850(-2.76)		
	$Crf(w_a, w_b)$	**81.970(-2.62)**		
	$Csf(w_a, w_b)$	81.503(-3.18)		
Wikipedia-based attributes	$Waf(w_a, w_b)$	82.022(-2.56)	80.459 (-4.41)	**76.375 (-9.27)**
	$Waf^{*}(w_a, w_b)$	82.026(-2.55)		
	$Waf^{**}(w_a, w_b)$	81.649(-3.00)		
	$Wkf(w_a, w_b)$	81.975(-2.61)	80.390 (-4.50)	
	$Wkf^{*}(w_a, w_b)$	81.426(-3.27)		
	$Wkf^{**}(w_a, w_b)$	**81.293(-3.43)**		

To explore the importance of each feature in the classification task, we analyze the contribution of them. Table 6 shows the changes in the average accuracy after removing each feature in turn. The contribution of "Wikipedia-based Attributes" is greater than that of "Course-based Attributes". When the features based on Wikipedia were removed, the average accuracy falls by 9.27%, and when the features based on course attributes were removed, the average accuracy falls by 5.42%. This may be due to the number of features in "Based on Wikipedia" is slightly more than that in "Based on course".

Among the features of "Course-based Attributes", $Caf(w_a, w_b)$ contributed the most, and the average accuracy falls by 4.12% after removal. The contribution of $Crf(w_a, w_b)$ is the smallest, which may be due to that only few concept pairs were involved.

Among the features of "Wikipedia-based Attributes", when the features of "Category" are removed, the average accuracy falls by 4.41%. While the features of "Clickstream" are removed, the average accuracy falls by 4.50%. There is little difference between these two groups. Compared with individual concepts, the use of "related concept sets" improves feature's contributions greatly.

E. Cross-School Testing

The overall data set is composed of data from 11 schools. To figure out the results of the intersection of different school data sets and the average accuracy under cross-school conditions. We first use the data sets of 11 schools to train the classifiers to explore the average accuracy of each school respectively. And a "cross-school test" is conducted to explore the scalability and adaptability of the model in the cross-school situation.

Fig.3 shows the experimental results. Take Caltech as an example, "In-school Training" represents the use of Caltech's data set to train the classifier and test the accuracy of the school's internal prediction; "Out-of-school Training" means using the data sets of other ten schools as the training set, and Caltech's data set as the testing set to verify the accuracy of prediction.

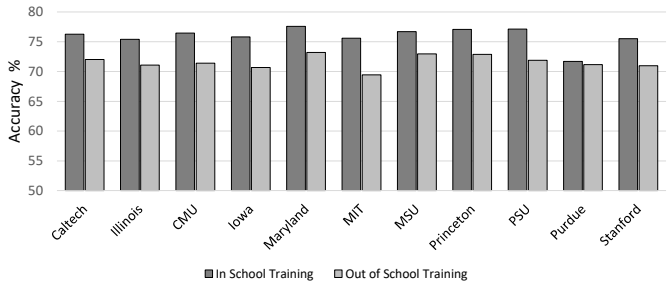

Figure 3. Cross-school testing

It can be seen from Fig.3 that compared with the accuracy of training the classifier on the overall data set (84.18%), the accuracy of training the classifier on a single school data set of "In-school Training" is generally low. This is also because the size of the data set decreases when trainings were performed in single school.

The accuracy rate of each school in the "Out-of-school Training" training is slightly lower than that of the "In-school Training", but the gap is not very significant. This may be caused by the unbalanced division of the data set. On average, the accuracy of the "Out-of-school Training" to "In-school Training" ratio in 11 schools is 94.34%, close to 95%. This proves that our model has excellent scalability and universality. If the data volume is larger, the trained model can better adapt to the prediction of the prerequisite relation in an unknown situation.

V. Conclusion And Future Work

This research proposes a new method to extracting prerequisite relations among concepts from online course introduction, using the course attributes and Wikipedia attribute design features together. Experiments show that this method is superior to existing baselines.

Due to the limitation of the data set, this research only conduct experiment in the field of computer science. In the future, we will create concept pair data sets from online courses in different majors and languages, further verify and improve the model we proposed. Besides, we will also try to analyze various types of learning resources such as online video subtitle data and textbooks, using them to extract the prerequisite relationships between concepts.

Acknowledgment

This research was supported by The National Natural Science Foundation of China (No.61977021), The Technology Innovation Special Program of Hubei Province(No.2018-ACA139,No.2019ACA144), The Research Project of Hubei Provincial Department of Education (No.D20191002).

References

[1] Agrawal, R., Golshan, B., & Papalexakis, E. (2016). Toward Data-Driven Design of Educational Courses: A Feasibility Study. Journal of Educational Data Mining, 8(1), 1-21.

[2] Limongelli, C., Gasparetti, F., & Sciarrone, F. (2015, June). Wiki course builder: a system for retrieving and sequencing didactic materials from wikipedia. In 2015 International Conference on Information Technology Based Higher Education and Training (ITHET) (pp. 1-6). IEEE.

[3] Chen, P., Lu, Y., Zheng, V. W., & Pian, Y. (2018, November). Prerequisite-driven deep knowledge tracing. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 39-48). IEEE.

[4] Yang, Y., Liu, H., Carbonell, J., & Ma, W. (2015, February). Concept graph learning from educational data. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (pp. 159-168).

[5] ALSaad, F., Boughoula, A., Geigle, C., Sundaram, H., & Zhai, C. (2018). Mining MOOC Lecture Transcripts to Construct Concept Dependency Graphs. International Educational Data Mining Society..

[6] Gasparetti, F., De Medio, C., Limongelli, C., Sciarrone, F., & Temperini, M. (2018). Prerequisites between learning objects: Automatic extraction based on a machine learning approach. Telematics and Informatics, 35(3), 595-610.

[7] Manrique, R., Sosa, J., Marino, O., Nunes, B. P., & Cardozo, N. (2018, December). Investigating learning resources precedence relations via concept prerequisite learning. In 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI) (pp. 198-205). IEEE.

[8] Vuong, A., Nixon, T., & Towle, B. (2011, July). A Method for Finding Prerequisites Within a Curriculum. In EDM (pp. 211-216).

[9] Gordon, J., Aguilar, S., Sheng, E., & Burns, G. (2017, September). Structured generation of technical reading lists. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (pp. 261-270).

[10] Koutrika, G., Liu, L., & Simske, S. (2015, April). Generating reading orders over document collections. In 2015 IEEE 31st International Conference on Data Engineering (pp. 507-518). IEEE.

[11] Talukdar, P.P., Cohen, W.W.: Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In: Proceedings of the Seventh Workshop on Building Educational Applications Using NLP pp. 307–315. Association for Computational Linguistics, (2012).

[12] Liang, C., Wu, Z.h., Huang, W.y., Giles, C.L.: Measuring prerequisite relations among concepts. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing pp. 1668–1674(2015).

[13] Zhou, Y., Xiao, K.: Extracting Prerequisite Relations Among Concepts in Wikipedia. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE, (2019, July).

[14] Sayyadiharikandeh, M., Gordon, J., Ambite, J. L., Lerman, K.: Finding Prerequisite Relations using the Wikipedia Clickstream. In Companion Proceedings of The 2019 World Wide Web Conference (pp. 1240-1247). (2019, May).

[15] Pan, L., Li, C., Li, J., & Tang, J. (2017, July). Prerequisite relation learning for concepts in moocs. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1447-1456).

[16] Liang, C., Ye, J., Wu, Z., Pursel, B., & Giles, C. (2017, February). Recovering concept prerequisite relations from university course dependencies. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).