Research Article

Mining Precedence Relations among Lecture Videos in MOOCs via Concept Prerequisite Learning

Kui Xiao 1,2, Youheng Bai 1,2 and Shihui Wang 1,2

1 School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China
2 Hubei Province Educational Informatization Engineering Research Center, Hubei University, Wuhan 430062, China

Correspondence should be addressed to Kui Xiao; xiaokui@hubu.edu.cn

Received 23 April 2021; Revised 22 July 2021; Accepted 6 August 2021; Published 13 August 2021

Academic Editor: Zhen Liu

Copyright © 2021 Kui Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, MOOC has gradually become an important way for people to learn knowledge. But the knowledge background of different people is quite different. Moreover, the precedence relations between lecture videos in a MOOC are often not clearly explained. As a result, some people may encounter obstacles due to lack of background knowledge when learning a MOOC. In this paper, we proposed an approach for mining precedence relations between lecture videos in a MOOC automatically. First, we extracted main concepts from video captions automatically. And then, an LSTM-based neural network model was used to measure prerequisite relations among the main concepts. Finally, the precedence relations between lecture videos were identified based on concept prerequisite relations. Experiments showed that our concept prerequisite learning method outperforms the existing methods and helps accurately identify the precedence relations between lecture videos in a MOOC.

1. Introduction

Nowadays, more and more people around the world are learning through online education platforms. The number of learners and courses of the top MOOC providers has also been rising rapidly. As of December 2020, 76 million learners around the world have already registered on Coursera, and the provider has also launched over 4600 courses (https://www.classcentral.com/report/mooc-stats-2020/).

As is well known, a MOOC consists of a series of lecture videos, and each lecture videos is an independent learning resource. The sequence of the learning resources in a MOOC is fixed. But every online learner has a different background, and not everyone wants to learn from the first lecture to the last one. Some learners prefer to choose videos of interest for learning, but they are not sure whether they have the appropriate background knowledge. The required background knowledge may be addressed in some former lectures, but the precedence relations between lecture videos in a course are usually ambiguous, and as a result, the learners do not know where to start.

A precedence relation between a pair of lecture videos establishes which one addresses more simple concepts and thus should be delivered first. In other words, the precedence relation between two resources is determined by the prerequisite relations among the concepts addressed by these resources. Here, a prerequisite refers to a concept or requirement before one can proceed to the next one. The prerequisite relation exists as a natural precedence among concepts in cognitive processes when people learn, organize, apply, and generate knowledge [1]. Concept prerequisite learning has played an important role in many fields, including precedence relation identification [2–4], concept graph extraction [5, 6], reading list generation [7, 8], curriculum planning [9], course sequencing [10], course design [11], knowledge tracing [12], and automatic assessment [13].

In this paper, we present a new approach for identifying precedence relations between lecture videos in a MOOC based on concept prerequisite learning. First of all, we extracted main concepts from the captions of every lecture video, and then measure prerequisite relations among concepts with an LSTM-based neural network model. After
that, the precedence relations between lecture videos were inferred according to the concept prerequisite relations.

What we do can help learners not only know which main concepts are addressed in each lecture video but also understand the dependencies between different lecture videos. The dependencies between these videos can let learners know what background knowledge is needed before learning a video and let them choose the previous related videos to learn according to their personal circumstances.

Our main contributions are as follows:

(i) A new approach to measure concept prerequisite relations outperforms other baseline methods.

(ii) A novel metric to identify precedence relations between lecture videos in a MOOC based on concept prerequisite relations.

(iii) A new dataset containing 1873 video pairs in “Food and Health,” “Algorithms,” “Calculus,” and “Global History.”

The rest of the paper is organized as follows. Section 2 discusses related work; Section 3 gives a brief introduction of our approach; Section 4 introduces the concept prerequisite learning algorithms; Section 5 describes the method for identifying precedence relations between lecture videos in a MOOC; Section 6 describes the new datasets we collected for this paper and reports our experimental results; in Section 7, there is the conclusion and a discussion of future work.

2. Related Work

Mining precedence relations between learning resources is a task that has gained much attention in recent years. Gasparetti et al. [2] presented an approach to predict a prerequisite relation among two learning resources. Given a pair of resources, a feature vector is built based on the Wikipedia pages of the concepts found in the learning resource text. Chen et al. [3] divided each lecture video into a length of 20-second-long segments and inferred precedence relations between the segments. Manrique et al. [4] also identified precedence relations between MOOC videos with their main concepts. However, their main concepts were extracted by experts manually. Obviously, it is a time-consuming work. This manual approach is not feasible when the number of videos is large.

The problem of discovering prerequisite relations between concepts has also been addressed by some researchers. Talukdar and Cohen [14] proposed an approach to model the prerequisite structure between Wikipedia concepts. They assumed that hyperlinks between Wikipedia pages indicate a prerequisite relation and designed several features, such as PageRank score and Random Walk with Restart (RWR) score via the hyperlink network. At last, they used a Maximum Entropy (MaxEnt) classifier to predict prerequisite relations between concepts. Liang et al. [15] also use the hyperlinks for measuring prerequisite relations between Wikipedia concepts. In their work, each concept was represented by a list of related concepts, and the authors calculated the reference distance (RefD) between two concepts to infer whether one concept is a prerequisite of another or not. Sayyadharikandeh et al. [16] introduced an approach that inferred prerequisite relations between concepts with Wikipedia clickstream data, but a limitation of the approach is that Wikipedia clickstream data usually cannot cover all the concept pairs. Miaschi et al. [17] utilized linguistic features extracted from Wikipedia articles to measure concept prerequisite relations. To the best of our knowledge, it is the first work to use LSTM-based neural networks for concept prerequisite learning. It is a good inspiration for our work. Liang et al. [18] created 15 graph-based features and 17 text-based features for Wikipedia concept pairs and then employed four widely used binary classifiers: Naïve Bayes, Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF), to discover prerequisite relations between concepts. The UNIGE_SE team [19] developed a neural network classifier that exploited features extracted both from raw text and the structure of the Wikipedia pages to achieving automatic prerequisite learning for Italian concepts. Most of the above methods will be used as the baselines in our experiments.

In addition to Wikipedia, other types of learning resources have also been studied to measure prerequisite relations between concepts, such as university courses [20], textbooks [6], and MOOCs [21]. These works provide the foundation for a wide range of education fields, including learning resources sequencing, learning resource recommendation, and curriculum planning.

3. Our Approach

To identify precedence relations between videos in a MOOC, we need to extract the main concepts from videos, and then measure prerequisite relations between the main concepts. Based on concept prerequisite relations, we can identify the precedence relations between videos in a MOOC. The detailed steps are as follows:

1. Extracting captions from lecture videos in a MOOC
2. Selecting main concepts from the captions of every lecture video
3. Inferring concept prerequisite relations via a LSTM-based neural network model
4. Mining precedence relations between lecture videos based on the concept prerequisite relations

It should be noted that only existing Wikipedia concepts were considered as the main concepts of a video in our work. A Wikipedia concept is the title of a Wikipedia article. To the best of our knowledge, Wikipedia is the largest multilingual online encyclopedia on the Internet, and its content can provide services for MOOCs in different disciplines.

4. Concept Prerequisite Extraction

In this section, we will introduce how to measure prerequisite relations among concepts. Before introducing the details of concept prerequisite learning, we first define some relevant elements. The details are shown in Table 1.
In this paper, concept prerequisite learning is a binary classification task. Given a pair of concepts \((A, B)\), the classification result (equation (1)) is whether \(B\) is a prerequisite of \(A\) or not.

\[
\text{Preq}(A, B) = \begin{cases} 
1, & B \text{ is a prerequisite of } A, \\
0, & \text{else.} 
\end{cases}
\]

Many concepts prerequisite learning algorithms require using artificial design features, such as the method proposed by Liang et al. [18]. In recent years, deep learning technology has been widely used in image recognition, voice processing, natural language processing, and other fields. Deep learning is essentially a feature learning method, which transforms the original data into higher-level feature representation through a nonlinear model, to obtain more abstract expression. In this paper, we employ an LSTM model to extract higher-level features of concept pairs from the raw input.

According to [15], a concept could be represented by its related concepts in Wikipedia’s concept space. Therefore, when measuring prerequisite relations of \((A, B)\), we should not only analyze the two concepts themselves but also analyze their related concepts.

The specific classification process is as follows: First of all, the two concepts \(A\) and \(B\), as well as their sets \(f^{(A)}_L, f^{(B)}_L\), were respectively converted into four 128-dimensional vectors, i.e., \(V_A, V_B, V_{f^{(A)}_L}, V_{f^{(B)}_L}\). Secondly, these vectors were input into four identical LSTM-based subnetworks with 32 units, and the four LSTM outputs were then concatenated \((O_A \oplus O_B \oplus O_{f^{(A)}_L} \oplus O_{f^{(B)}_L})\) and passed to a last Dense Layer. And we use Sigmoid as the activation function. Finally, we got the classification results of concept prerequisite learning. The architecture of the classifier is shown in Figure 1.

For a concept \(A\), there must be a corresponding Wikipedia article. To get the vector \(V_A\), each of the first 400 words in the article was converted into a 128-dimensional vector, and the vectors were built with word2vec [22]. We generated the vectors using the ukWac corpora [23]. We also removed the stop words in the article in advance. At last, \(V_A\) was equal to the mean of the 400 vectors.

Similarly, for each \(R_A\) of concept \(A\), we also calculated its vector according to the above method. Then, vector \(V_{f^{(A)}_L}\) was equal to the mean of all \(R_A\) vectors.

### 5. Precedence Relations Identification

In this section, we use concept prerequisites to help identify precedence relations between lecture videos in a MOOC. First, the lecture videos we used were all from the MOOC provider Coursera. For every lecture video, we got video captions with the toolkit coursera-dl (https://github.com/coursera-dl/coursera-dl). After that, we utilized another toolkit TextRazor to extract the main concepts from video captions. As mentioned above, the main concepts are all Wikipedia concepts.

However, in order to reduce the amount of calculation, we only extracted a few concepts from the subtitles of each video. TextRazor associates two attributes with each entity, called relevance_score and Wikipedia_link (https://www.textrazor.com/docs/python#Entity), which are used to estimate the “relevance” of Wikipedia entities to the input text. Here, the input text is lecture video captions, and the parameter relevance_score represents the relevance of the entity to the input text. For the extracted concepts, we sort them in reverse order according to the relevance_score and select the top \(k\) concepts. We will also set different values for the parameter \(k\) to study the impact of this parameter on the prediction of video dependencies.

A MOOC often consists of a series of lecture videos, as \(C = \{v_1, v_2, \ldots, v_n\}\), given two lecture videos \(v_i\) and \(v_j\). Normally, if \(v_j\) depends on \(v_i\), \(v_j\) must be ranked before \(v_i\). There are three possibilities for the relation between the two videos: (1) \(v_i\) depends on \(v_j\); it means \(v_j\) contains some background knowledge for learning \(v_i\); (2) the two lecture videos are related but have no precedence relation; and (3) they are unrelated.

The precedence relation between two lecture videos can be explained in terms of prerequisites among the concepts that they address. If \(v_j\) contains some prerequisite concepts that are needed to learn \(v_i\), then it can be said that \(v_j\) depends on \(v_i\). In fact, the prerequisite concepts are the background knowledge needed to learn \(v_i\). On the other hand, if there is no prerequisite relation among the concepts of the two videos, then there will be no precedence between the two videos. Note that, if two videos contain too many duplicate concepts, it is not easy to judge the precedence between them. So, we defined a parameter \(\theta\), if the number of

---

**Table 1: Elements used in the approach.**

<table>
<thead>
<tr>
<th>Elements</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>A Wikipedia concept</td>
</tr>
<tr>
<td>(R_A)</td>
<td>A related concept of (A). It means that there is a hyperlink in the content of (A)'s article, users can access the content of (R_A) by clicking the hyperlink.</td>
</tr>
<tr>
<td>(f^{(A)}_L)</td>
<td>Set of (A)'s related concepts</td>
</tr>
<tr>
<td>(V_A)</td>
<td>Vector corresponding to the concept (A)</td>
</tr>
<tr>
<td>(V_{f^{(A)}_L})</td>
<td>Vector corresponding to the set (f^{(A)}_L)</td>
</tr>
</tbody>
</table>

---

\(\theta\) is a parameter used to measure the relevance of \(v_j\) to \(v_i\). The relevance score is calculated as follows:

\[
\text{relevance}(v_i, v_j) = \frac{|\text{set}(v_i) \cap \text{set}(v_j)|}{|\text{set}(v_i)|} \times |\text{set}(v_j)|
\]

where \(\text{set}(v)\) refers to the set of concepts in video \(v\).
duplicate concepts of a video pair is greater than or equal to \( \theta \), the video pair will not be considered in our work.

Given a concept-based representation for each lecture video \( v_i \) \( \in \{A_1, A_2, \ldots, A_n\} \), \( v_j \) \( \in \{B_1, B_2, \ldots, B_m\} \), we calculate precedence score between two videos as

\[
Scr(v_i, v_j) = \frac{1}{\left| P_e(v_i, v_j) \right|} \sum_{(A,B) \in P_e(v_i, v_j)} \text{Preq}(A,B),
\]

s.t. \( P_e(v_i, v_j) = \{(A,B) | A \in v_i \land B \in v_j, A \neq B\}, |v_i \cap v_j| < \theta \).

(2)

Here, \( Scr(v_i, v_j) \) represents the degree of precedence of video \( v_i \) on \( v_j \). Ultimately, whether \( v_i \) depends on \( v_j \) will be determined by

\[
\text{Dep}(v_i, v_j) = \begin{cases} 
1, & \text{if } Scr(v_i, v_j) > Scr(v_j, v_i), \\
0, & \text{else}.
\end{cases}
\]

(3)

6. Experiments

In this section, we tested our approach from two aspects, one is the concept prerequisite learning method, and another one is the precedence relations mining method.

6.1. Concept Prerequisite Learning Experiments. First of all, we evaluate the performance of our concept prerequisite learning method on the AL-CPL dataset [24]. The AL-CPL dataset consists of binary-labeled concept pairs from four different domains, including data mining, geometry, physics, and precalculus. To perform concept prerequisite relation predicting experiments, we trained and tested the classifiers on concept pairs belonging to the same domain. The evaluation is performed using 5-fold cross validation. The performance was quantified with Accuracy (A), Precision (P), Recall (R), F1-score (F1), and Area under the ROC Curve (AUC).

Besides, we compare our model with five state-of-the-art models:

(1) M1. The model was proposed by Miaschi et al. [17], the authors tested a neural network model (M1) which learned to classify the binary labels using two LSTM-based subnetworks. The embedding vectors of concepts \( A \) and \( B \), i.e., \( V_A \) and \( V_B \), were inputted into the two subnetworks, respectively. The two LSTM outputs were then concatenated \( (O_A \oplus O_B) \) and passed to a last dense layer (with Sigmoid activation).

(2) NN. To verify whether the LSTM part is useful, we also omitted the LSTM part and directly concatenated the \( A \)'s and \( B \)'s embedding vectors \( (V_A \oplus V_B) \) and passed them to a last dense layer (with Sigmoid activation).

(3) RefD. This baseline was proposed by Liang et al. [15]. In terms of the weight of a referred article, we chose the TFIDF weight to calculate the RefD of two concepts, which is better than the EQUAL weight.

(4) BERT. This model, developed by Angel [25], corresponds to a single-layer neural network using concept pairs and its Wikipedia description.
extracted from the Italian-BERT model. Likewise, we obtain two 786-dimensional embeddings for a concept pair and the single-layer neural network maps the embeddings to an output space of dimension 2 as a precondition for the concepts.

(5) GE. The concept prerequisite relations of video can be considered as a graph structure. Currently, in the deep graph learning areas, many models were proposed, which can use node, edge, and neighbor information together to generate features embedding. In our experiments, we also used a Graph embedding (GE) method, PyTorch-BigGraph (PBG) [26], as a baseline to compare our model with graph learning methods.

Table 2 shows that in terms of average F1 scores, our method generally performs better than the baseline method, reaching +4.8% for M1, +14.8% for NN, +25.3% for RefD, +1% for BERT, and +3.1% for GE. Also, we see that the performance of BERT is very close to ours. More specifically, the BERT method has higher F1 on the precalculus domain, which indicates that fine-tuned BERT is sufficient to obtain the prerequisite between concepts in this domain.

The proposed method is superior to the M1 method, because the M1 method only considers the text content of Wikipedia concepts but ignores the fact that related concepts can also provide rich background knowledge for understanding concepts. Besides, the proposed method is also better than the NN method, and we will explain the reason for this problem later. In addition, the RefD method also performs poorly, because the RefD method only uses one feature to predict the prerequisite relations of concepts. Furthermore, thanks to the excellent pre-training of the large-scale corpus, both BERT and GE perform well in concept prerequisite relation learning. But compared with our method, these models lack the word embedding of the main concepts, so our method performs better in several domains such as data mining, geometry, and physics.

On the other hand, because we use a LSTM model in our proposed method, we also investigate two questions about the LSTM model:

Q1. Does the LSTM model help predict the prerequisite relations between concepts?

Q2. Is the four-input LSTM model better than the two-input LSTM model?

We use two baseline methods, NN and M1, to investigate these questions. In the NN method, there is not an LSTM model. Given a pair of concepts \((A, B)\), the embedding vectors of \(A\) and \(B\) are directly concatenated and then passed to a last dense layer to predict whether there is a prerequisite relation between the two concepts. In the M1 method [17], the embedding vectors of \(A\) and \(B\) are inputted into two LSTM-based subnetworks. And then the outputs of the two LSTM subnetworks are concatenated \((O_A@O_B)\) and passed to a last dense layer.

Table 2 suggests that both the proposed method and the M1 method perform better than the NN method with respect to F1-score across all four datasets. It shows that the LSTM model does provide some useful features that can be used to predict the prerequisite relations between concepts. Furthermore, the proposed method also outperforms the M1 method on all four datasets concerning F1-score. It means that another two LSTM subnetworks for the embedding vectors of \(f^{L}_{1[A]}\) and \(f^{L}_{1[B]}\), i.e., \(V_{f^{in}[A]}\) and \(V_{f^{in}[B]}\) are also useful for the concept prerequisite learning task.

Besides, we also calculated the AUPRC for the three methods. AUPRC, namely, the area under the precision-recall curve, is a useful performance metric for a problem setting where people care a lot about finding positive examples (https://glassboxmedicine.com/2019/03/02/measuring-performance-auprc/). In Figure 2, we show the AUPRC of the three methods on the four datasets. Still, the proposed method performs best among all three methods on the four datasets. For example, the AUPRC of our method on Data Mining outperforms M1 and NN by 4% and 12%, respectively. We also have the observations that our method and the M1 method also perform very closely on both the Geometry and Precalculus datasets. Looking closely at the two datasets, we find that the related concept sets of the two concepts in a concept pair, i.e., \(f^{L}_{1[A]}\) and \(f^{L}_{1[B]}\), have a higher degree of overlap. As a result, \(V_{f^{in}[A]}\) and \(V_{f^{in}[B]}\) have little effect on generating higher-level features. This is why the four-input LSTM model and two-input LSTM model perform relatively similarly on the two datasets. Table 3 gives two examples of overlapping related concepts.

6.2 Precedence Relations Mining Experiments. We selected four MOOCs from Coursera, including "Food and Health" (https://www.coursera.org/learn/food-and-health), "Algorithms" (https://www.coursera.org/learn/algorithms-part1), "Calculus" (https://www.coursera.org/learn/introduction-to-calculus), and "Global History" (https://www.coursera.org/learn/modern-world-2). The number of lecture videos from these MOOCs is 29, 59, 63, and 51, respectively. Table 4 shows the information of some lecture videos of the MOOC Algorithms. The first column presents the catalog number of the lecture videos in the MOOC, and the second column presents the lecture video titles, while the last column presents some of the main concepts of each lecture video. Moreover, we set the value of parameter \(k\) to 6. In other words, we select the top 6 main concepts from every lecture video.

We generated video pairs for every MOOC like this \((v_1, v_2), (v_1, v_3), \ldots, (v_1, v_p), (v_2, v_3), \ldots, (v_2, v_n), \ldots\). Then, to obtain the ground truth of precedence relations of all video pairs in the MOOCs, we asked students of corresponding majors to vote on the final results. Given a pair of videos \((a, b)\), students had three options: (1) \(b\) depends on \(a\) (“1”); (2) \(b\) does not depend on \(a\) (“0”); (3) do not know...
If there is no option to get a majority vote or option 3 got a majority vote in a video pair, the pair will be discarded.

The datasets have been published on GitHub (https://github.com/Morganbyh/MOOC).

In Table 4, we report the Accuracy (A), Precision (P), Recall (R), and F1-score (F1) obtained for the MOOCs.

From Table 5, we can see that the number of precedence relations in "Food and Health" is the least, and the number of positive samples is less than 10%. In "Calculus," however, the number of positive samples is close to 50%. In another two MOOCs, the number of positive samples is about 20%.

Further observation revealed that "Food and Health" presented only simple common-sense knowledge. It has few precedence relations between lecture videos. This limits our prediction of positive samples. It has the lowest recall of 71.24%. But the accuracy of this domain is generally better than the other three domains. We suppose that this is due to a large number of negative samples and the imbalance in classes. The highest F1 value appeared in "Calculus," with 81.82%. For the other two MOOC courses, their F1 scores are all over 78%. Overall, our method successfully identifies the precedence relations of the MOOC lecture videos in four domains.

On the other hand, we also want to know whether the value of parameter k has any effect on the prediction of video precedence relations. Figure 3 shows the effect of different values of k (1, 3, 6, 10, and 15) on the prediction tasks. For every lecture video, we extracted k main concepts from their captions and used the concept prerequisite learning results to identify the lecture video precedence relations.

We extracted 279, 414, 408, and 611 concepts from four MOOCs, respectively (k = 15). When the parameter k is valued at 1, 3, 6, and 10, we will only use some of these concepts. Looking at four domains, we can see that when the value of k is too large or too small, the performance of precedence relation prediction is not very good. When the parameter k is set to 6, our prediction performances in three MOOCs are the best, including "Food and Health," "Algorithms," and "Calculus." For "Global History," when the parameter k is set to 3, the prediction performance of the precedence relations is the best. But when the parameter k is set to 6, the prediction performance is also better.

In fact, when the parameter k is set to 1, we may miss some of the main concepts of a lecture video. When the parameter k is valued at 15, we may consider some unimportant words as the main concept of the video. The relevance scores of the words may be very low. Take "Algorithms" as an example, when the k is 15, there will be some general concepts in every lecture video, such as "Data structures," "Algorithms," "Classes," "Sorting," and "Arrays." These general concepts are not helpful for precedence relation prediction.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Measure</th>
<th>Data mining</th>
<th>Geometry</th>
<th>Physics</th>
<th>Precalculus</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>A</td>
<td>83.5</td>
<td>91.9</td>
<td>85.7</td>
<td>89.5</td>
<td>87.7</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>77.7</td>
<td>88.2</td>
<td>75.2</td>
<td>86.5</td>
<td>81.9</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>74.3</td>
<td>85.7</td>
<td>63.3</td>
<td>82.0</td>
<td>76.3</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>75.8</td>
<td>86.9</td>
<td>68.5</td>
<td>84.1</td>
<td>78.8</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>90.2</td>
<td>96.4</td>
<td>91.5</td>
<td>95.3</td>
<td>93.4</td>
</tr>
<tr>
<td>NN</td>
<td>A</td>
<td>74.3</td>
<td>82.1</td>
<td>87.4</td>
<td>90.1</td>
<td>83.5</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>61.6</td>
<td>69.3</td>
<td>53.5</td>
<td>71.8</td>
<td>64.1</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>73.8</td>
<td>77.5</td>
<td>74.0</td>
<td>76.9</td>
<td>75.6</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>67.0</td>
<td>73.0</td>
<td>61.6</td>
<td>73.6</td>
<td>68.8</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>84.3</td>
<td>90.1</td>
<td>86.2</td>
<td>90.8</td>
<td>87.9</td>
</tr>
<tr>
<td>RefD</td>
<td>A</td>
<td>66.2</td>
<td>61.9</td>
<td>74.9</td>
<td>81.9</td>
<td>71.2</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>51.7</td>
<td>42.4</td>
<td>49.9</td>
<td>75.1</td>
<td>54.8</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>76.2</td>
<td>62.3</td>
<td>49.6</td>
<td>69.4</td>
<td>64.4</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>61.4</td>
<td>50.4</td>
<td>49.4</td>
<td>72.1</td>
<td>58.3</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>69.5</td>
<td>62.4</td>
<td>67.7</td>
<td>79.2</td>
<td>69.7</td>
</tr>
<tr>
<td>BERT</td>
<td>A</td>
<td>86.0</td>
<td>92.8</td>
<td>87.4</td>
<td>91.1</td>
<td>89.3</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>79.7</td>
<td>88.1</td>
<td>75.6</td>
<td>86.5</td>
<td>82.5</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>81.1</td>
<td>89.3</td>
<td>73.9</td>
<td>87.9</td>
<td>83.1</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>80.3</td>
<td>88.6</td>
<td>74.4</td>
<td>87.1</td>
<td>82.6</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>92.2</td>
<td>96.9</td>
<td>92.3</td>
<td>96.6</td>
<td>94.5</td>
</tr>
<tr>
<td>GE</td>
<td>A</td>
<td>85.5</td>
<td>91.2</td>
<td>82.4</td>
<td>90.1</td>
<td>87.3</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>76.5</td>
<td>86.5</td>
<td>74.4</td>
<td>83.4</td>
<td>80.2</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>81.5</td>
<td>84.1</td>
<td>75.9</td>
<td>81.9</td>
<td>80.9</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>78.9</td>
<td>85.3</td>
<td>74.9</td>
<td>82.6</td>
<td>80.4</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>91.7</td>
<td>97.0</td>
<td>93.3</td>
<td>95.4</td>
<td>94.4</td>
</tr>
<tr>
<td>Proposed</td>
<td>A</td>
<td>87.0</td>
<td>93.6</td>
<td>88.2</td>
<td>90.9</td>
<td>89.9</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>83.7</td>
<td>91.3</td>
<td>75.6</td>
<td>86.7</td>
<td>84.3</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>78.8</td>
<td>88.7</td>
<td>78.0</td>
<td>86.6</td>
<td>83.0</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>81.1</td>
<td>89.8</td>
<td>76.7</td>
<td>86.6</td>
<td>83.6</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>92.8</td>
<td>97.4</td>
<td>93.0</td>
<td>96.0</td>
<td>94.8</td>
</tr>
</tbody>
</table>

Bold fonts represent the maximum value of a measure on a dataset.
Figure 2: Comparison of LSTM models. (a) Data mining. (b) Geometry. (c) Physics. (d) Precalculus.

Table 3: Examples of related concept overlap of two concepts.

<table>
<thead>
<tr>
<th>Domains</th>
<th>Geometry</th>
<th>Precalculus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept pair ((A, B))</td>
<td>(Slope, secant line)</td>
<td>(Elementary matrix, identity matrix)</td>
</tr>
<tr>
<td>#Related concepts of ((A, B))</td>
<td>(204, 186)</td>
<td>(185, 190)</td>
</tr>
<tr>
<td>#Overlapping related concepts</td>
<td>155</td>
<td>172</td>
</tr>
</tbody>
</table>
Table 4: Lecture videos in the MOOC Algorithms.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Lecture videos</th>
<th>Main concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Course introduction</td>
<td>Sorting, Merge sort, Search algorithm, Algorithm, Spanning tree, Sorting algorithm</td>
</tr>
<tr>
<td>5</td>
<td>Quick union improvements</td>
<td>Path compression, Linear time, Log, Connected component, Node, Constructor</td>
</tr>
<tr>
<td>11</td>
<td>Theory of algorithms</td>
<td>Theory of algorithms, Constant time, Big theta, Big o, Worst case, Binary search</td>
</tr>
<tr>
<td>16</td>
<td>Generics</td>
<td>Generics, Generic, Generic type, Type checking, Data type, Bug</td>
</tr>
<tr>
<td>20</td>
<td>Selection sort</td>
<td>Selection sort, Array, Quadratic time, Sorting, For loop, Pointer</td>
</tr>
<tr>
<td>25</td>
<td>Merge sort</td>
<td>Merge sort, Sort, Stable sort, Recursion, Loop, Sequence</td>
</tr>
<tr>
<td>29</td>
<td>Stability</td>
<td>Insertion sort, Selection sort, Sorting algorithm, Shell Sort, Sort, Merge sort</td>
</tr>
<tr>
<td>32</td>
<td>Duplicate keys</td>
<td>Heap, Sorting algorithms, Heap sort, Heap data structure, Max heap, Binary heap</td>
</tr>
<tr>
<td>48</td>
<td>Line segment intersection</td>
<td>Priority queue, Binary search tree, ( N \log n ), Brute force method, Interval, Binary search</td>
</tr>
</tbody>
</table>

Table 5: Precedence relations predicting results.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>#Video pairs</th>
<th>#Precedence relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food and health</td>
<td>95.40</td>
<td>69.23</td>
<td>71.24</td>
<td>70.22</td>
<td>174</td>
<td>13</td>
</tr>
<tr>
<td>Algorithms</td>
<td>84.34</td>
<td>81.34</td>
<td>76.63</td>
<td>78.91</td>
<td>437</td>
<td>96</td>
</tr>
<tr>
<td>Calculus</td>
<td>87.87</td>
<td>81.32</td>
<td>82.33</td>
<td>81.82</td>
<td>781</td>
<td>378</td>
</tr>
<tr>
<td>Global history</td>
<td>79.51</td>
<td>80.13</td>
<td>76.98</td>
<td>78.52</td>
<td>441</td>
<td>84</td>
</tr>
</tbody>
</table>

Figure 3: Effect of the number of extracted concepts from lecture videos on the identification of precedence relations. (a) Food and health. (b) Algorithms. (c) Calculus. (d) Global history.
7. Conclusions and Future Work

We studied the problem of automatically mining precedence relations between lecture videos in a MOOC. We extracted the main concepts from video captions and analyzed the prerequisite relations between the concepts. After that, the concept prerequisite relations were used to infer the precedence relations between lecture videos. Experiments showed that our concept prerequisite learning method outperforms the existing methods and helps accurately identify the precedence relations between lecture videos in a MOOC.

Promising future direction would be using Wikipedia independent methods to measure prerequisite relations among concepts, so that more accurate main concepts can be selected from video captions for the precedence relation identification. Besides, we can also use the learner behaviour to identify the precedence relations between lecture videos. When users perform personalized learning on a MOOC platform, the order in which they learn videos often implies the precedence relations between the videos.

Data Availability

The CSV data used to support the findings of this study have been deposited in the GitHub repository (https://github.com/Morganbyh/MOOC). These data are the results of user voting for the video precedence relations. There are no restrictions on access to these data.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61977021), the Technology Innovation Special Program of Hubei Province (Nos. 2018ACA133 and 2019ACA144), and the Teaching Research Project of Hubei University (No. 202008).

References


