# Multi-view Transformer-Based Network for Prerequisite Learning in Concept Graphs

Zhichun Wang[1,2(✉)], Yifeng Shao[1], Boci Peng[3], Bangui Li[4], Yun Li[5], Qianren Wang[5], and Nijun Li[5]

[1] School of Artificial Intelligence, Beijing Normal University, Beijing, China
zcwang@bnu.edu.cn
[2] Engineering Research Center of Intelligent Technology and Educational Application, Ministry of Education, Beijing, China
[3] School of Intelligence Science and Technology, Peking University, Beijing, China
[4] School of Computer Science and Engineering, Beihang University, Beijing, China
[5] Advanced Cognitive AI Lab, Shanghai Huawei Technologies, Shanghai, China

**Abstract.** Prerequisite learning is the task of identifying prerequisite relations among concepts, which is important for many AI-based educational applications. Previous approaches explore different kinds of learning resources to obtain useful features for predicting prerequisites. Early approaches use handcrafted features while recent approaches use neural networks to encode information of concepts. To further improve the results of prerequisite learning, we build concept graphs to include comprehensive information about concepts from open data and learning resources, and propose a **M**ulti-view **T**ransformer-based **N**etwork (MTN) to encode multi-view features of concepts in the graph. Multi-view features are fused to make accurate predictions of concept prerequisites. We evaluate our approach on four public datasets and compare it with recently published approaches. The results show that our approach can achieve state-of-the-art results in the task of prerequisite learning. The source code and data are available at https://github.com/kg-bnu/MTN.

**Keywords:** prerequisite learning · concept embedding · concept graph

## 1 Introduction

Prerequisite relations identify the underlying dependencies among concepts, which provide a reasonable learning order of concepts. If a concept $A$ is a prerequisite of $B$, it means that it is difficult to learn $B$ without the knowledge of $A$. For example, in Calculus, the concept *Limit* is a prerequisite of concept *Derivative*. To understand *Derivative*, one is required to learn the knowledge of *Limit* first. Concept prerequisites are very important for many educational applications, including intelligent tutoring [5], curriculum planning [1], learning materials generation [7], and recommendation [4]. Prerequisites among courses are usually given in university curricula or by Massive Open Online Courses (MOOCs) platforms, but concept prerequisites are often not provided.

Prerequisite learning is the task of automatically discovering concept prerequisites from different kinds of data, and many approaches have been proposed in recent years.


G. Demartini et al. (Eds.): ISWC 2024, LNCS 15231, pp. 67–86, 2025.
https://doi.org/10.1007/978-3-031-77844-5_4

Typically, prerequisite learning is solved as a binary classification of concept pairs. Features of concept pairs can be obtained from Wikipedia [16,17], MOOCs [2,23], university course curriculums [18], textbooks [34], scientific corpora [8], or lecture files [13,15]. Early approaches extracted manually defined features from learning resources for prerequisite learning. For example, the work in [17] designed 15 graph-based features and 17 text-based features from Wikipedia, [23] defined 7 features for finding prerequisites in MOOCs. For these approaches, the results of prerequisite learning highly depend on the used features. The pre-defined features cannot be easily adapted to new types of resources. Recently, several neural-based approaches for prerequisite learning have been proposed, including PREREQ [25], VGAE [14], CPRL [10], ConLearn [29], and MHAVGAE [39], etc. These approaches first obtain initial concept features by using topic models or language models, and then utilize neural networks to predict prerequisites. Deep models can effectively model useful information of concepts, which achieve promising results without hand-crafted features.

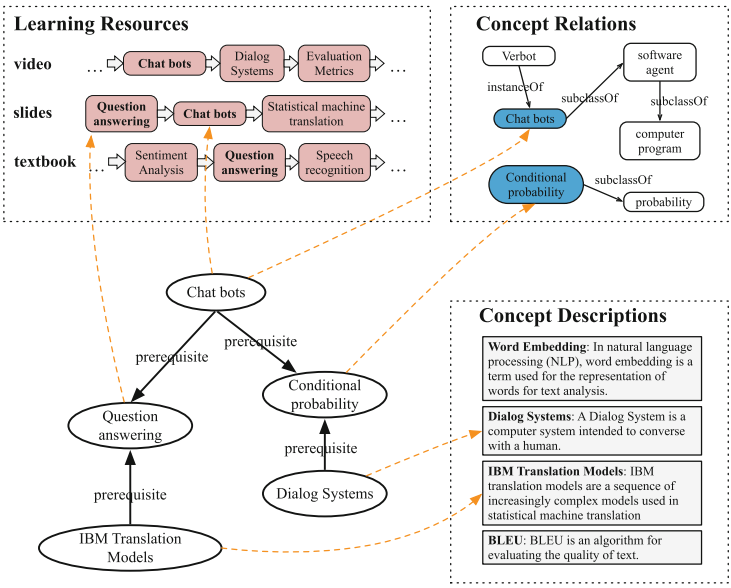


**Fig. 1.** Prerequisite learning based on learning resources, concept relations and concept descriptions.

Although continuous progress has been made by previous studies, we find that there are still challenging problems when using neural models for prerequisite learning. First, there lacks a unified framework for representing useful features of different types of learning resources. Previous approaches extracted texts from lecture slides [13] or MOOC transcripts [25], but the lengths and styles of sentences from these resources are quite different, which were handled with different techniques. For example, GAE [14] employs Doc2Vec model [11] to obtain concept representations from documents, while

PREREQ [25] uses the pairwise-link LDA [22] model for concept representation. Second, background knowledge of concepts is not fully utilized in the existing approaches. There are open knowledge bases containing descriptions and rich relations of concepts, such as ConceptNet [28], Wikidata [33], and Wikipedia[1]. Both structural and textual knowledge of concepts can be obtained from these open resources, but such knowledge is not well utilized in previous neural models. Previous approaches [14,25] only treated known prerequisites as the background knowledge of concepts, without considering concept relations like *subclassOf*, *partOf*, etc. It is important to study whether such background knowledge is useful and how to incorporate the knowledge into prerequisite learning models.

To solve the above challenges, we propose a **M**ulti-view **T**ransformer-based **N**etwork (MTN) for prerequisite learning. We first gather different kinds of information to build a multi-relational concept graph (as shown in Fig. 1), which contains learning resources, concept relations, and concept descriptions. MTN encodes multi-view features of concepts in the concept graph to predict prerequisite relations among them. Specifically, the main contributions of this paper include:

- We propose a method to build a multi-relational concept graph from learning resources and open knowledge bases. Concept prerequisites are predicted based on the multi-view features of the concept graph, including *resource-view*, *relation-view*, and *text-view*.
- We propose a multi-view transformer-based network (MTN) for prerequisite learning. MTN uses three transformer encoders to obtain concept features from different views, and fuses these features with an attention-based aggregation method to accurately predict concept prerequisites.
- We evaluate our approach on four public datasets, our approach overall outperforms the compared approaches and achieves the state-of-the-art results.

The rest of this paper is organized as follows: Sect. 2 formalizes the problem of prerequisite learning, Sect. 3 describes the proposed multi-view transformer-based network (MTN), Sect. 4 introduces the multi-view features aggregation method, Sect. 5 presents the evaluation results, Sect. 6 discusses some related work, Sect. 7 is the conclusion, and discusses limitations of this work.

## 2 Problem Formulation

In this work, we formalize the prerequisite learning as the task of link prediction in a multi-relational concept graph, which aims to identify missing prerequisites among entities already in the concept graph. We include rich relations and concept descriptions in the concept graph, which provides comprehensive information about concepts. The concept graph in our approach is built using three kinds of inputs:

- **Learning Resources**. Learning resources are those materials used in teaching and learning activities, including MOOCs, textbooks, scientific papers, etc. Concepts are

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

usually defined or introduced in learning resources, which contain important clues for prerequisite learning. Learning resources may be in different types and forms, we consider the order of appearance of concepts in them as the most important information, which contains clues of concept relatedness and dependencies. In this work, we identify concepts appearing in learning resources, and record the occurrence order of concepts in each learning resource. Concepts and learning resources are represented as nodes in the concept graph, there will be a $hasConcept$ link from a learning resource to a concept if the concept appears in the learning resource. The order of concepts appearing in a learning resource is recorded as a node attribute of the learning resource.

- **Concept Relations**. To predict prerequisite relationships among concepts, we believe that having knowledge of other types of concept relationships is helpful, such as $subClassOf$, $partOf$, and $instanceOf$. The previous prerequisite learning approaches built concept graphs containing only known prerequisite relations, while in this work we include more relation types in the graph by integrating concept knowledge from the open knowledge base Wikidata. Let $C_{core} = \{c_i\}_{i=1}^{K}$ denote a set of core concepts among which prerequisite relations are to be predicted, where $K$ is the number of concepts and $c_i$ is the $i$-th concept in $C_{core}$. We locate them in Wikidata and extract their one-hop neighbors with the associated relations to populate the concept graph in our approach. A set of new extended concepts $C_{ext}$ will be introduced in the population process. To control the size of the concept graph and exclude irrelevant concepts, we only keep new concepts linked with at least two concepts in $C_{core}$. In this way, the concept graph will include rich links among concepts while retaining a modest size.
- **Concept Descriptions**. Concept descriptions can usually be found in textbooks or online wikis, which formally define concepts with important information. In this work, we take Wikipedia as the source of concept descriptions, because it covers a huge number of concepts and provides generally high-qualified descriptions of concepts. Given the set of core concepts $C_{core} = \{c_i\}_{i=1}^{K}$, we locate each concept in Wikipedia and extract the first sentence in its wiki page as the concept description.

To learn concept prerequisites in a certain domain, we first build a multi-relational concept graph $\mathcal{G} = \{C, E, R, T, D\}$ by using the above inputs, where $C = C_{core} \cup C_{ext}$ consists of core concepts with partially annotated prerequisites and extended concepts from Wikidata, $E$ is a set of learning resources, $R$ is a set of relations, $T$ is a set of triples representing relations among concepts and learning resources, $D$ is the set of descriptions of concepts in $C$. Given the concept graph $\mathcal{G}$ and a set of known prerequisites among concepts in $C_{core}$, the problem of prerequisite learning is to predict missing prerequisite relations between concepts, i.e. mapping any concept pair $(c, c')$, $c, c' \in C_{core}$, to a binary class indicating whether $c$ is a prerequisite of $c'$.

## 3 Multi-view Transformer-Based Network

To accurately predict prerequisites in a multi-relational concept graph, we propose a **M**ulti-view **T**ransformer-based **N**etwork (MTN). Figure 2 shows the framework of

MTN. Given inputs from the resources-view, relation-view, and text-view of the concept graph, MTN first uses three Transformer models to get important features of concept pairs; it then aggregates features from different views and determines whether they have prerequisite relations or not.

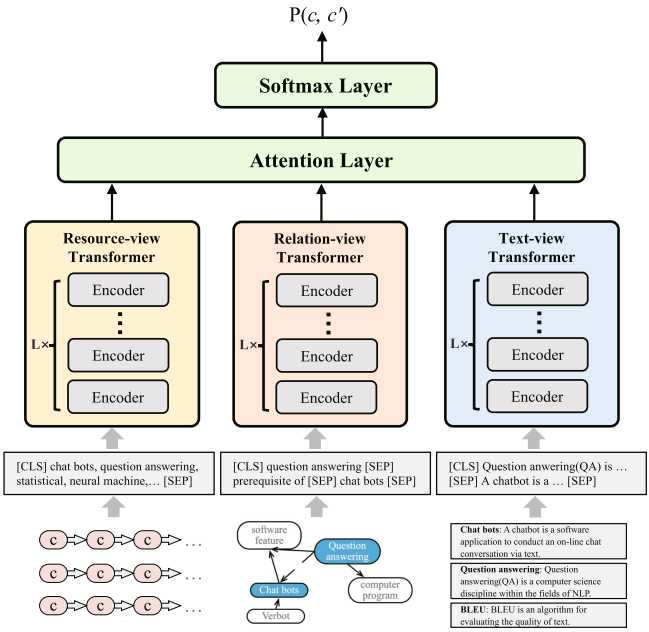


**Fig. 2.** Framework of MTN.

### 3.1 Resource-View Transformer

Resource-view Transformer aims to encode concept features based on the relationships between concepts and learning resources in a concept graph. The underlying assumption is that concepts linked to the same learning resources are related, and the order in which concepts appear may indicate their prerequisite relations. To capture the relatedness and dependencies among concepts, we train a Resource-view Transformer following the scheme of Masked Language Modeling in BERT [6].

Let $\mathcal{G}res = \{C, E, T\}$ denote the resource-view of the original concept graph $\mathcal{G}$. We first obtain a set of concept sequences $\{(c_{i1}, c_{i2}, ..., c_{il})\}_{i=1}^{|E|}$ from $\mathcal{G}$, where $c_{ij} \in C$, $(e_i, hasConcept, c_{ij}) \in T$, the subscript $j$ is the index of concepts appearing in learning resource $e_i \in E$. Concept names in each sequence are separated by tabs. During training, a certain percentage of the input concept tokens are randomly masked. These masked tokens are then predicted based on the context provided by the surrounding sequence. Specifically, for each masked token, a masking probability is applied to

determine whether it should be replaced with a special [MASK] token, a wrong concept token, or remain unchanged. The objective is to maximize the likelihood of predicting the original concept names given the contextual information.

Let $\mathcal{D} = \{(x_k, M_k)\}_{k=1}^{|E|}$ represent the set of masked sequences and the set of masked tokens in them (i.e. $x_k$ denotes the masked input sequence, and $M_k$ is the set of masked tokens), $|E|$ is the number of concept sequences. The Resource-view Transformer is trained by optimizing the following loss:

$$\mathcal{L}_{\text{res}} = -\frac{1}{|E|}\sum_{k=1}^{|E|}\left(\frac{1}{|M_k|}\sum_{y\in M_k}\log P(y|x_k)\right), \tag{1}$$

where $P(y|x_k)$ is predicted probability of the true token $y$ given the context of the masked input $x_k$.

### 3.2 Relation-View Transformer

Prerequisite learning can be considered as a task of link prediction based on the existing concept relations. In the domain of Knowledge Graph Completion (KGC) [27], there are many knowledge graph embedding models that can be used for this task, such as TransE [3], DistMult [37], and RotatE [30], etc. Typically, a KGC model defines a score function to estimate the plausibility of triples; it learns embeddings of entities and relations by minimizing a global loss function over all the triples in a knowledge graph. The learned embeddings can be used to predict missing links in the knowledge graph. Here we train a Relation-view Transformer following KG-BERT [38] to encode relation-view features of concepts.

Let $\mathcal{G}_{rel} = \{C, R, T\}$ denote the relation-view of the concept graph; for a triple $t = (c, r, c')$, we generate a sequence "[CLS] $c$ [SEP] $r$ [SEP] $c'$ [SEP]" as the input of the Relation-view Transformer. The final hidden state of [CLS] generated by the Relation-view Transformer is taken as the feature of $t$, denoted as $\mathbf{h}_t$, which is used to predicate the score of the triple. The Relation-view Transformer is trained by optimizing the following cross-entropy loss:

$$\mathcal{L}_{rel} = -\sum_{t\in T\cup T^-}\Big(y_t\log(s_t) + (1-y_t)\log(1-s_t)\Big) \tag{2}$$

where $s_t = \sigma\left(\mathbf{w}^\top\mathbf{h}_t + a\right)$ is the score of a triple $t$, $T$ is the set of positive triples, $T^-$ is the set of negative triples generated by replacing the head or tail entities in positive triples; $\mathbf{w}$ and $a$ are the parameters, and $\sigma$ represents the Sigmoid function.

### 3.3 Text-View Transformer

Let $\mathcal{G}_{text} = \{C, D\}$ denote the text-view of the concept graph $\mathcal{G}$, we train a Text-view Transformer to encode useful features from concept descriptions in $\mathcal{G}_{text}$. For a pair of concepts $p_i = (c, c')$, their descriptions $d, d' \in D$ are concatenated to generate a sequence of text tokens. More specifically, the sequence has the form of "[CLS] $d$

[SEP] $d'$ [SEP]". The sequence of concatenated descriptions is fed to the Text-view Transformer model, the hidden state of [CLS] token is used as the text-view embedding of the concept pair $p_i = (c, c')$.

To get discriminative features of concept pairs, we use a set of training data to train the Text-view Transformer. Let $\{(p_i, y_i)\}_{i=1}^{N}$ be a set of $N$ labeled concept pairs, $y_i = 1$ if the concept pair $p_i$ is a positive sample, otherwise $y_i = 0$. The Text-view Transformer model is trained by optimizing the following cross-entropy loss function:

$$\mathcal{L}_{text} = -\sum_{k=1}^{N} \Big( y_i \log f_i + (1 - y_i) \log(1 - f_i) \Big) \tag{3}$$

$f_i$ is computed as:

$$f_i = \sigma\Big(\mathbf{u}^\top \mathbf{h}_i + b\Big) \tag{4}$$

where $\mathbf{u}$ and $b$ are parameters, $\mathbf{h}_i$ is the text-view embedding of concept pair $p_i$, $\sigma$ is the sigmoid function.

## 4 Multi-view Feature Aggregation

The purpose of training multi-view transformers is to get useful features from multiple aspects. Therefore, features from multiple views are aggregated to predict prerequisite relations among concepts. We first pre-train the resource-view, relation-view, and text-view transformers separately. Then the features of concept pairs generated by multiple transformers are aggregated to predict the final prerequisite results.

### 4.1 Attention-Based Feature Aggregation

For a concept pair $p_i = (c, c')$, three sequences are generated as the inputs of resource-view, relation-view, and text-view transformers, respectively. Let $d$ and $d'$ be the descriptions of $c$ and $c'$, these sequences have the following forms:

- Resource-view: $\{[\text{CLS}], c, [\text{SEP}], c', [\text{SEP}]\}$
- Relation-view: $\{[\text{CLS}], c, [\text{SEP}], prerequisite, [\text{SEP}], c', [\text{SEP}]\}$
- Text-view: $\{[\text{CLS}], d, [\text{SEP}], d', [\text{SEP}]\}$

The hidden states of [CLS] generated by three transformers are denoted as $\mathbf{h}_i^{(v)}$, where $v \in \{res, rel, text\}$. We design an attention-based feature aggregator to combine the features of different views for each concept pair. The idea of using an attention mechanism is to distinguish important features during the aggregation of features. For the embedding of concept pair $p_i = (c, c')$ in each view $v$, an attention coefficient $\alpha^{(v)}$ is first computed:

$$\alpha^{(v)} = \text{LeakyReLU}\left(\mathbf{v}^\top \mathbf{h}_i^{(v)} + d\right) \tag{5}$$

where $\mathbf{v}$ and $d$ are parameters to be learned during model training. The coefficients of the three views are then normalized by the softmax function:

$$\beta^{(v)} = \frac{\exp\left(\alpha^{(v)}\right)}{\sum_{j \in \{res, rel, text\}} \exp\left(\alpha^{(j)}\right)} \tag{6}$$

Multi-view features of concepts are aggregated based on their attention:

$$\mathbf{h}_i = \sum_{v \in \{res, rel, text\}} \beta^{(v)} \mathbf{h}_i^{(v)} \tag{7}$$

The aggregated feature $\mathbf{h}_i$ will be used to predict prerequisite relation between $c$ and $c'$.

### 4.2 Prerequisite Prediction

The aggregated feature $\mathbf{h}_i$ of a concept pair $p_i = (c, c')$ is passed to a feed-forward layer followed by a softmax classification layer, which predicts the probability distribution of being prerequisite or not:

$$g_i = \text{softmax}\left(\mathbf{u}^\top \mathbf{h}_i + z\right) \tag{8}$$

where $\mathbf{u}$ and $z$ are parameters. Given a set of labeled concept pairs $\{(p_i, y_i)\}_{i=1}^N$ as training data, the prerequisite prediction model is trained by optimizing the cross-entropy loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} \Big( y_i \log g_i + (1 - y_i) \log \Big(1 - g_i\Big) \Big) \tag{9}$$

## 5 Experiments

### 5.1 Datasets

We evaluate our approach on four datasets, which are built from different types of learning resources.

- **LectureBank**. It was built by Li et al. [14], which contains 3,119 lecture files in five different domains, including NLP, machine learning, artificial intelligence, deep learning and information retrieval. There are 208 concepts and 913 prerequisites in this dataset.
- **MOOC-DSA**. It was built by Pan et al. [23], which covers concepts about Data Structure and Algorithm (DSA). It contains video transcripts obtained from Coursera[2], course concepts in the transcripts are manually labeled. There are 200 concepts and 479 prerequisites in MOOC DSA.
- **MOOC-ML**. It was also created by Pan et al. [23], which covers concepts about Machine Learning (ML). There are 244 concepts and 1,735 prerequisites in MOOC ML.
- **University Course**. It was built by Chen et al. [18], which contains descriptions of 654 university courses, and covers 407 concepts in the domain of computer science. There are 1,008 prerequisites in this dataset (Table 1).

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

**Table 1.** Dataset Statistics.

| Dataset | #Resources | #Concepts | #Prerequisites |
|---|---|---|---|
| LectureBank | 3,119 | 208 | 913 |
| MOOC-DSA | 449 | 200 | 479 |
| MOOC-ML | 548 | 244 | 1,735 |
| University Course | 654 | 407 | 1,008 |

### 5.2 Concept Graphs

We build a concept graph for each dataset following the method introduced in Sect. 2. Firstly, concepts in the learning resources are identified by finding mentions of concept names in the texts in learning resources, which establishes links between concepts and resources. Because domain concepts are mostly mentioned using their formal names, this simple method can obtain desirable results for our approach. Secondly, concept descriptions are obtained by extracting introductory texts (i.e. the first sentence) in concepts' Wikipedia pages. To find the corresponding page of a concept in Wikipedia, we send a query of the concept's name to Wikipedia's search engine; if the top ranked page in the returned results has the same title as the concept, we take it as the corresponding page of the concept. Thirdly, concept relations are obtained from Wikidata, which has high coverage of concepts and relations. Most Wikipedia pages have links to their corresponding Wikidata pages, we can use these links to locate the equivalent items of the core concepts in Wikidata; then the one-hop neighbors of the core concepts can be further obtained.

Table 2 shows the statistics of concept graphs built for four datasets, which outlines the numbers of extended concepts, edges, concept descriptions and the average length of descriptions. It should be noted that we only extract descriptions for core concepts, there is a small number of them having no descriptions because their Wikipedia pages are not found; we use their names as the descriptions in such cases. Table 3 outlines all the relations in the concept graph. There are 9 relations in the concept graphs. *prerequisiteOf* relations between concepts are from annotations in the datasets; *hasConcept* and *hasDescription* relations are established by using our methods introduced above. The rest 6 relations are obtained from Wikidata.

**Table 2.** Statistics of Concept Graphs

| Dataset | #Ext. Concepts | #Edges | #Descriptions(Avg. Length) |
|---|---|---|---|
| LectureBank | 1,598 | 3,762 | 178 (27.0) |
| MOOC-DSA | 971 | 2,315 | 185 (28.1) |
| MOOC-ML | 1,925 | 4,239 | 216 (28.8) |
| University Course | 9,167 | 22,278 | 372 (28.0) |

**Table 3.** Relations in Concept Graphs

| Relation | Explanation |
|---|---|
| *prerequisiteOf* | It states that one concept is a prerequisite of another; |
| *hasConcept* | It states that one resource contains a concept; |
| *hasDescription* | It states that one concept has a textual description; |
| *subClassOf* | It states that one concept is a subconcept of another; |
| *partOf* | It states that one concept is a part of another; |
| *hasPart* | It is an inverse relation of partOf; |
| *instanceOf* | It states that one concept is an instance of another; |
| *facetOf* | It states that one concept is an aspect of another; |
| *hasQuality* | It states that one concept has an inherent characteristic. |

### 5.3 Baselines

To validate the performance and effectiveness of our approach, MTN, we compare it with six state-of-the-art prerequisite learning approaches and one highly relevant knowledge graph completion model. The compared prerequisite learning approaches are from formally published works, which provide source codes or have results in the same experimental setting as ours.

- **GAE** [14]. It trains a Doc2Vec [11] model to get concept embeddings, and uses graph auto-encoders to predict prerequisites.
- **VGAE** [14]. It is an extension of GAE, which uses variational auto-encoders to predict concept prerequisites.
- **PREREQ** [25]. It uses pairwise-link LDA to obtain concept representations and predicts prerequisites using a Siamese network.
- **CPRL** [10]. It uses R-GCN to learn concept embeddings, and predicts the prerequisite relations using a Siamese network.
- **MHAVGAE** [39]. It uses a multi-head attention variational graph auto-encoder for learning prerequisites in the concept-resource graph.
- **ConLearn** [29]. It is a contextual-knowledge-aware concept prerequisite relation learning approach.
- **KG-BERT** [38]. It is a knowledge graph completion model based on transformers. We include it in the baselines because the relation-view transformer in our approach is trained following the method in KG-BERT.

### 5.4 Experiment Settings

We conducted experiments in two settings: the *conventional setting*, used in previous works [10, 29], allowing us to compare our results with those reported in the established works; and the *hard setting*, where we constructed a more challenging evaluation setting using unified cross-validations and more negative samples.

**Conventional Setting.** In the conventional setting, the proportions of the training, validation, and testing sets are 8:1:1 for the LectureBank dataset and 6:1:3 for the MOOC-DSA, MOOC-ML, and University Course datasets. Negative samples are generated by reversing each positive prerequisite pair and randomly sampling unrelated concept pairs. Reversing a positive sample $(c, c')$ means exchanging the positions of the two concepts to generate a negative pair $(c', c)$. Positive samples are oversampled to balance the number of positive and negative samples. In the conventional setting, the Precision, Recall, and F1 scores on the testing data are reported for all the compared methods; the results are averaged over five train-test splits.

**Hard Setting.** In the conventional setting, the train-test splits are not unified across all datasets, and the 1:1 ratio between positive and negative samples is overly simplistic compared to real-world problems. To enable more objective comparisons, we designed a harder evaluation setting with unified training and testing splits and a greater number of negative samples. In this setting, we conducted 5-fold cross-validation for all experiments. In each fold, 60%, 20%, and 20% of prerequisites are used for training, validation, and testing, respectively. Negative samples are generated by corrupting and reversing the positive ones. Corrupting a positive sample $(c, c')$ involves replacing $c$ or $c'$ with randomly selected concepts. We generate eight negative samples for each positive sample, including seven corrupted concept pairs and one reversed concept pair. All compared approaches are run on the same splits of datasets, and the average Precision, Recall, F1, and AUC (Area Under the ROC Curve) over five train-test splits are reported.

**Training Details.** We implemented our approach using PyTorch[3] and conducted experiments on a workstation with an Intel Xeon CPU @ 2.50 GHz, 128 GB RAM, and a GPU with 24 GB VRAM. We used the pre-trained BERT model[4] to initialize the transformers in our approach. During training, we employed the Adam optimizer to train the three transformers with a batch size of 8. We considered the masking probability of the Resource-view Transformer among $\{0.05, 0.10, 0.15, 0.20, 0.25\}$ and the learning rates for the three transformers among $\{10^{-6}, 10^{-5}, 10^{-4}\}$. The best configurations of these hyper-parameters were selected based on the F1 score. For the baseline approaches, we followed their respective papers and tuned their parameters according to the F1 scores.

### 5.5 Experimental Results

**Results in Conventional Setting.** Table 4 presents the results of the compared approaches in the conventional setting. The best and second-best results in each row are highlighted in boldface and with an underline, respectively. The results of GAE, VGAE, PREREQ, CPRL, and ConLearn are cited from [29], as we use the same datasets and experimental setting. The results of MHAVGAE and KG-BERT in this conventional setting are not available in previous works, so their results are obtained by running their source codes. Our approach, MTN, achieves the highest F1 scores on three datasets and the second-highest F1 score on one dataset. Among all the compared approaches,

[3] https://pytorch.org.
[4] https://huggingface.co/bert-base-uncased.

**Table 4.** Results of compared approaches in the conventional setting.

| Dataset | Metric | GAE | VGAE | PREREQ | CPRL | MHAVGAE | ConLearn | KG-BERT | MTN |
|---|---|---|---|---|---|---|---|---|---|
| LectureBank | P | 0.462 | 0.417 | 0.590 | 0.861 | 0.421 | 0.831 | 0.551 | **0.915** |
| | R | 0.811 | 0.575 | 0.502 | 0.858 | 0.654 | **0.960** | 0.899 | 0.871 |
| | F1 | 0.589 | 0.484 | 0.543 | 0.860 | 0.512 | 0.891 | 0.683 | **0.893** |
| MOOC-DSA | P | 0.294 | 0.269 | 0.492 | 0.641 | 0.392 | 0.823 | 0.558 | **0.853** |
| | R | 0.715 | 0.657 | 0.462 | 0.619 | 0.626 | **0.816** | 0.801 | 0.778 |
| | F1 | 0.417 | 0.382 | 0.476 | 0.630 | 0.482 | **0.819** | 0.658 | 0.814 |
| MOOC-ML | P | 0.293 | 0.266 | 0.448 | 0.800 | 0.364 | **0.895** | 0.643 | 0.894 |
| | R | 0.733 | 0.647 | 0.592 | 0.642 | 0.713 | 0.850 | 0.820 | **0.905** |
| | F1 | 0.419 | 0.377 | 0.510 | 0.712 | 0.482 | 0.872 | 0.721 | **0.900** |
| University Course | P | 0.450 | 0.470 | 0.468 | 0.689 | 0.449 | 0.611 | 0.558 | **0.874** |
| | R | 0.886 | 0.694 | 0.916 | 0.760 | 0.822 | **0.966** | 0.947 | 0.851 |
| | F1 | 0.597 | 0.560 | 0.597 | 0.723 | 0.581 | 0.749 | 0.703 | **0.862** |

ConLearn is the most competitive one. In terms of F1, ConLearn gets the best F1 on MOOC-DSA, with a 0.5% improvement over MTN. However, our approach outperforms ConLearn by 0.2%, 2.8% and 11.3% on LectureBank, MOOC-ML, and University Course datasets.

The performance of GAE-based approaches is suboptimal compared to ConLearn and MTN. We believe this is due to their decoding methods' inability to effectively distinguish between positive concept pairs and their reversed pairs. For instance, VGAE and MHAVGAE predict prerequisite relationships between two concepts based on the inner product of their latent vectors. Reversing the order of the two concepts does not affect the results. Since a large portion of negative samples in the training and testing datasets are reversals of positive ones, it is difficult for GAE-based approaches to achieve high precision.

The results of KG-BERT demonstrate that prerequisite relationships between concepts can be predicted based on known triples in the concept graph. KG-BERT outperforms GAE-based approaches, highlighting its effectiveness in prerequisite learning tasks. However, its performance falls short of MTN, which incorporates additional information from resources and concept descriptions.

**Results in Hard Setting.** Table 5 presents the results of the compared approaches in the hard setting. As in Table 4, the best and second best result in each row are highlighted in boldface and with an underline, respectively. In the hard setting, the results of all the compared approaches are produced with their source codes. Experiments are conducted on the same 5-fold cross-validations. Because CPRL does not provide source code, we are not able to compare it in the hard setting. According to the results, our approach MTN gets the best F1 and AUC scores on all of the four datasets. ConLearn is still the most competitive baseline, however, our approach gets 1.0%, 17.8%, 8.6% and 8.6% improvements of F1 over ConLearn on four datasets, respectively. In terms of AUC, our approach gets 2.7%, 7.9%, 5.1% and 2.5% improvements over ConLearn.

**Table 5.** Results of compared approaches in the hard setting.

| Dataset | Metric | GAE | VGAE | PREREQ | MHAVGAE | ConLearn | KG-BERT | MTN |
|---|---|---|---|---|---|---|---|---|
| LectureBank | P | 0.365 | 0.366 | 0.403 | 0.371 | **0.811** | 0.255 | 0.758 |
| | R | 0.604 | 0.612 | 0.450 | 0.648 | 0.671 | 0.552 | **0.732** |
| | F1 | 0.455 | 0.458 | 0.425 | 0.472 | 0.734 | 0.348 | **0.744** |
| | AUC | 0.779 | 0.789 | 0.684 | 0.803 | 0.909 | 0.740 | **0.936** |
| MOOC-DSA | P | 0.334 | 0.335 | 0.240 | 0.351 | 0.541 | 0.262 | **0.678** |
| | R | 0.585 | 0.577 | 0.399 | 0.553 | 0.408 | 0.514 | **0.611** |
| | F1 | 0.425 | 0.424 | 0.299 | 0.430 | 0.465 | 0.347 | **0.643** |
| | AUC | 0.788 | 0.785 | 0.556 | 0.798 | 0.813 | 0.756 | **0.892** |
| MOOC-ML | P | 0.301 | 0.301 | 0.318 | 0.307 | **0.698** | 0.249 | 0.696 |
| | R | 0.557 | 0.561 | 0.606 | 0.601 | 0.566 | 0.694 | **0.726** |
| | F1 | 0.391 | 0.392 | 0.417 | 0.406 | 0.625 | 0.367 | **0.711** |
| | AUC | 0.777 | 0.770 | 0.728 | 0.792 | 0.897 | 0.782 | **0.948** |
| University Course | P | 0.307 | 0.307 | 0.485 | 0.346 | 0.620 | 0.252 | **0.672** |
| | R | 0.436 | 0.435 | 0.533 | 0.536 | 0.557 | **0.813** | 0.675 |
| | F1 | 0.360 | 0.360 | 0.508 | 0.421 | 0.587 | 0.385 | **0.673** |
| | AUC | 0.685 | 0.683 | 0.793 | 0.776 | 0.903 | 0.823 | **0.928** |

In the hard setting, the training and testing datasets contain more negative samples. The GAE-based approaches (GAE, VGAE, and MHAVGAE) still perform poorly. KG-BERT also performs worse in the hard setting, indicating that relying solely on triples in concept graphs does not yield desirable results when there are more negative samples than positive ones.

### 5.6 Contributions of Component Models

To get better insight into the effectiveness of component models, we conduct an ablation study of MTN. Table 6 presents the ablation study of MTN's component models on four datasets in the hard setting. MTN(-w.o. Res.), MTN(-w.o. Rel.), and MTN(-w.o. Text.) are variations of MTN by removing resource-view, relation-view, and text-view embedding models, respectively. The results outlined in Table 6 are F1 values, the numbers in brackets are the differences of F1 values compared with MTN. It is observed that removing any models in MTN leads to decreases in F1 values. According to the F1 decreases, we find that the relation-view embedding model contributes the most to MTN, with a decrease of 0.040 in LectureBank, 0.059 in MOOC-DSA, 0.016 in MOOC-ML, and 0.047 in University Course. This suggests that the relation-view model is a key factor in the success of MTN and has a substantial impact on its ability to perform the task effectively. It shows that transformers of three views all encode useful features for predicting prerequisites. When features from three views are aggregated, better results are obtained.

**Table 6.** Results of Ablation Study of MTN's Component Models (F1).

| Dataset | MTN(-w.o. Res.) | MTN(-w.o. Rel.) | MTN(-w.o. Text.) | MTN |
|---|---|---|---|---|
| LectureBank | 0.736 (−0.008) | 0.704 (−0.040) | 0.737 (−0.007) | 0.744 |
| MOOC DSA | 0.636 (−0.007) | 0.584 (−0.059) | 0.623 (−0.020) | 0.643 |
| MOOC ML | 0.706 (−0.005) | 0.695 (−0.016) | 0.705 (−0.006) | 0.711 |
| University Course | 0.648 (−0.025) | 0.626 (−0.047) | 0.651 (−0.022) | 0.673 |

### 5.7 Effect of Embedding Aggregation Method

To evaluate the effectiveness of the attention-based embedding aggregation method in our approach, we compared it with two simple aggregation methods, embedding addition and concatenation. Embedding addition directly adds concept embeddings from three views, embedding concatenation aggregates concept embeddings by concatenating them. Figure 3 compares the results of different aggregation methods on four datasets in the hard setting. It is observed that embedding addition gets the lowest F1 values on all the datasets. Attention-based embedding aggregation and embedding concatenation have close results, while the attention-based method performs slightly better on the four datasets, with 0.4% to 2.4% improvements of F1. Comparing with embedding concatenation method, our attention-based method can promote the results in most cases.

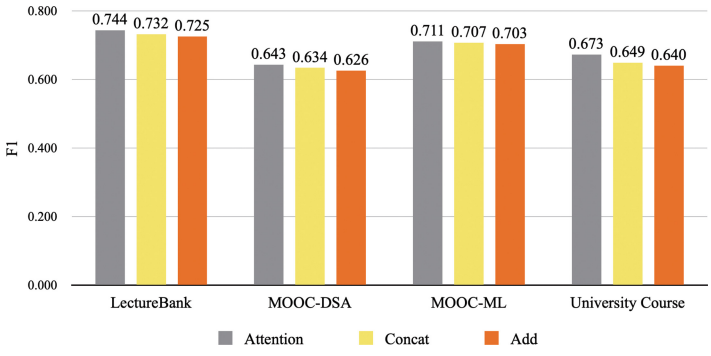


**Fig. 3.** Results comparison of Different Aggregation Methods (F1).

## 6 Related Work

### 6.1 Feature Engineering Approaches

To determine whether one concept is a prerequisite of another concept, researchers have investigated various features of concept pairs. These features are computed based

on different kinds of resources, including Wikipedia, MOOCs, Textbooks, Scientific Corpus, Knowledge Graphs, etc.

Links and textual contents in Wikipedia are commonly used in early prerequisite learning approaches. Talukdar and Cohen [31] investigated reliable features based on hyperlinks, edits, and page contents in Wikipedia for discovering prerequisite relations. Liang et al. [16] proposed a link-based metric called reference distance (RefD), which measures the prerequisite relations between concepts. RefD measures how differently two concepts refer to each other to predict prerequisite relations. Liang et al. [17,19] compared various features in the active learning framework for prerequisite learning, including graph-based features and text-based features. Clickstream data of Wikipedia is also used to generate features of concepts, which are fed to binary classifiers to predict prerequisite relations [26]. Xiao et al. [36] explored more comprehensive features from Wikipedia for prerequisite learning, including link-based, clickstream-based, category-based, content-based and time-based features.

With the quick growth of MOOCs, lecture transcripts, video playlists and course descriptions in MOOCs are explored for prerequisite learning. Pan et al. [23] defined concept relatedness, contextual features and structure features based on MOOCs, including video references, sentence references and average position distance, etc. ALSaad et al. [2] proposed BEM (Bridge Ensemble Measure) and GDM (Global Direction Measure) based on lecture transcripts of MOOCs for prerequisite prediction. BEM captures concept dependencies based on lecture bridging concepts, sliding windows, and the first lecture indicator; GDM incorporates time directly by analyzing the concept time ordering both globally and within lectures. Xiao et al. [35] designed features based on the concept frequencies and positions in course descriptions for prerequisite learning.

Features from Textbooks, scientific corpus, and university course curriculums are also studied for identifying prerequisite relations. Wang et al. [34] proposed a concept map extraction model that jointly extracts key concepts and identifies prerequisite relations from textbooks. Gordon et al. [8] used a LDA model to identify concepts from scientific papers, and employed cross-entropy and information flow methods to discover prerequisite relations. CPR-Recover [18] recovers concept prerequisite relations from course dependencies by using an optimization based framework. Information and dependencies of courses are collected from different universities, which are consumed by CPR-Recover to get the concept-level prerequisite relations. EMRCM [9] builds a concept map containing multiple relations between concepts including the prerequisite relations. Student question logs, textbooks, and Wikipedia are used by EMRCM to infer prerequisite relations.

Open knowledge graphs contain rich structure information about concepts, which also have been explored for prerequisite learning. Manrique et al. [20,21] proposed an approach which uses the structural knowledge in DBpedia to search prerequisite candidates; features of concept pairs are computed from DBpedia and a document corpus to train a binary classifier for prerequisite learning. To avoid generating too many candidates, they set a maximum length of the paths that link concepts. Our approach also uses a similar filtering method to control the size of concept graphs.

The above studies mainly focus on computing useful features of concept pairs from different types of resources. Features are manually designed and cannot be easily transferred to new resources. Our approach gathers information from various resources into a multi-relational concept graph, and then employs deep models to learn latent features of concepts.

### 6.2 Deep Learning Approaches

Roy et al. [25] proposed PREREQ, which predicts prerequisites from labeled prerequisite relations of concepts, courses, or video playlists. It uses a pairwise-link LDA model to get vector representations of concepts, and uses a Siamese network to predict new concept prerequisites. Li et al. [12] used pre-trained language model and graph embedding model to get concept representations, which are used for predicting prerequisites.

Several approaches solved the prerequisite learning problem as link prediction within a graph, which consists of concept and resource nodes. Li et al. [13] used a relational-variational graph auto-encoder (R-VGAE) to predict prerequisite relations in the concept-resource graph. Following similar framework, Li et al. further proposed an unsupervised cross-domain prerequisite learning model, which learns to transfer concept prerequisite relations from an information-rich domain to an information-poor domain [15]. Jia et al. [10] proposed to use a relational graph convolutional network (R-GCN) to learning representations of concepts and resources. Prerequisites were predicted by a Siamese network which takes the learned concept representations as inputs. Zhang et al. [39] proposed a multi-head attention variational graph auto-encoders for learning prerequisites in a concept-resource graph, which employs a resource prerequisite reference distance metric to generate weak supervision labels. The work in [24] proposed a directed graph neural network based on the Weisfeiler-Leman algorithm for prerequisite learning. ConLearn [29] used gated GNN to generate representations of concepts, and applied Siamese network to predict prerequisites. The work in [40] proposed a novel alternating knowledge distillation approach to take advantage of both content-based and graph-based models for prerequisite learning. TCPL [32] also combines textual and structural features of concepts for prerequisite learning.

## 7 Conclusion

This paper presents a Multi-view Transformer-based Network (MTN) for prerequisite learning. To fully explore various information for discovering prerequisites, our approach learns resource-view, relation-view, and text-view concept representations from a concept graph. Features from multiple views are aggregated as the input for a prediction model, which determines whether given concept pairs have prerequisite relationships. We evaluate our approach on four datasets built from different resources. Compared to recently published approaches, our method achieves state-of-the-art results in both conventional setting and hard setting.

The main limitation of this work is that transformers in different views are trained separately, and the features from each view are frozen before passing them to the feature aggregator. As a result, the component models in MTN are not optimized jointly during

training. Exploring efficient methods for the joint learning of multi-view transformers is worthwhile in the future work, which is likely to enhance the results.

**Acknowledgement.** This work was partially supported by the National Science and Technology Major Project (No. 2021ZD0113004) and National Natural Science Foundation of China (No. 62276026).

**Supplemental Material Statement.** Our source code, experimental data, and instructions for repeating all experiments are available at https://github.com/kg-bnu/MTN.

## References

1. Agrawal, R., Golshan, B., Papalexakis, E.: Toward data-driven design of educational courses: a feasibility study. J. Educ. Data Min. **8**(1), 1–21 (2016). https://doi.org/10.5281/zenodo.3554601
2. ALSaad, F., Boughoula, A., Geigle, C., Sundaram, H., Zhai, C.: Mining MOOC lecture transcripts to construct concept dependency graphs. International Educational Data Mining Society (2018)
3. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13, Red Hook, NY, USA, pp. 2787—2795. Curran Associates Inc. (2013). https://doi.org/10.5555/2999792.2999923
4. Chanaa, A., El Faddouli, N.E.: Prerequisites-based course recommendation: recommending learning objects using concept prerequisites and metadata matching. Smart Learn. Environ. **11**(1), 16 (2024). https://doi.org/10.1186/s40561-024-00301-0
5. Chen, P., Lu, Y., Zheng, V.W., Pian, Y.: Prerequisite-driven deep knowledge tracing. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 39–48 (2018). https://doi.org/10.1109/ICDM.2018.00019
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1423
7. Gordon, J., Aguilar, S., Sheng, E., Burns, G.: Structured generation of technical reading lists. In: Tetreault, J., Burstein, J., Leacock, C., Yannakoudakis, H. (eds.) Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, Copenhagen, Denmark, pp. 261–270. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/W17-5029
8. Gordon, J., Zhu, L., Galstyan, A., Natarajan, P., Burns, G.: Modeling concept dependencies in a scientific corpus. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, pp. 866–875. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/P16-1082
9. Huang, X., et al.: Constructing educational concept maps with multiple relationships from multi-source data. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 1108–1113 (2019). https://doi.org/10.1109/ICDM.2019.00132
10. Jia, C., Shen, Y., Tang, Y., Sun, L., Lu, W.: Heterogeneous graph neural networks for concept prerequisite relation learning in educational data. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human

Language Technologies, pp. 2036–2047. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.naacl-main.164
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, Beijing, China, pp. 1188–1196. PMLR (2014). https://proceedings.mlr.press/v32/le14.html
12. Li, B., Peng, B., Shao, Y., Wang, Z.: Prerequisite learning with pre-trained language and graph embedding models. In: Wang, L., Feng, Y., Hong, Yu., He, R. (eds.) NLPCC 2021. LNCS (LNAI), vol. 13029, pp. 98–108. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88483-3_8
13. Li, I., Fabbri, A., Hingmire, S., Radev, D.: R-VGAE: relational-variational graph autoencoder for unsupervised prerequisite chain learning. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1147–1157. International Committee on Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.coling-main.99
14. Li, I., Fabbri, A.R., Tung, R.R., Radev, D.R.: What should I learn first: introducing lecturebank for NLP education and prerequisite chain learning. In: Proceedings of the AAAI Conference on Artificial Intelligence **33**, 6674–6681 (2019). https://doi.org/10.1609/aaai.v33i01.33016674
15. Li, I., Yan, V., Li, T., Qu, R., Radev, D.: Unsupervised cross-domain prerequisite chain learning using variational graph autoencoders. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 1005–1011. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.acl-short.127
16. Liang, C., Wu, Z., Huang, W., Giles, C.L.: Measuring prerequisite relations among concepts. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 1668–1674. Association for Computational Linguistics (2015). https://doi.org/10.18653/v1/D15-1193
17. Liang, C., Ye, J., Wang, S., Pursel, B., Giles, C.L.: Investigating active learning for concept prerequisite learning. In: Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (2018). https://doi.org/10.1609/aaai.v32i1.11396
18. Liang, C., Ye, J., Wu, Z., Pursel, B., Giles, C.: Recovering concept prerequisite relations from university course dependencies. In: Proceedings of the AAAI Conference on Artificial Intelligence **31**(1) (2017). https://doi.org/10.1609/aaai.v31i1.10550
19. Liang, C., Ye, J., Zhao, H., Pursel, B., Giles, C.: Active learning of strict partial orders: a case study on concept prerequisite relations. In: Lynch, C., Merceron, A., Desmarais, M., Nkambou, R. (eds.) EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining. pp. 348–353. EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining, International Educational Data Mining Society (2019)
20. Manrique, R., Pereira, B., Mariño, O.: Exploring knowledge graphs for the identification of concept prerequisites. Smart Learn. Environ. **6**(1), 1–18 (2019). https://doi.org/10.1186/s40561-019-0104-3
21. Manrique, R., Pereira, B., Marino, O., Cardozo, N., Wolfgand, S.: Towards the identification of concept prerequisites via knowledge graphs. In: 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), vol. 2161-377X, pp. 332–336 (2019). https://doi.org/10.1109/ICALT.2019.00101
22. Nallapati, R.M., Ahmed, A., Xing, E.P., Cohen, W.W.: Joint latent topic models for text and citations. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '08, New York, NY, USA, pp. 542–550. Association for Computing Machinery (2008). https://doi.org/10.1145/1401890.1401957

23. Pan, L., Li, C., Li, J., Tang, J.: Prerequisite relation learning for concepts in MOOCs. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, pp. 1447–1456. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/P17-1133
24. Qu, X., Shang, X., Zhang, Y.: Concept prerequisite relation prediction by using permutation-equivariant directed graph neural networks (2024). https://arxiv.org/abs/2312.09802
25. Roy, S., Madhyastha, M., Lawrence, S., Rajan, V.: Inferring concept prerequisite relations from online educational resources. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33(01), pp. 9589–9594 (2019). https://doi.org/10.1609/aaai.v33i01.33019589
26. 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. WWW '19, New York, NY, USA, pp. 1240—1247. Association for Computing Machinery (2019). https://doi.org/10.1145/3308560.3316753
27. Shen, T., Zhang, F., Cheng, J.: A comprehensive overview of knowledge graph completion. Knowl.-Based Syst. **255**, 109597 (2022). https://doi.org/10.1016/j.knosys.2022.109597
28. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI'17, pp. 4444—4451. AAAI Press (2017). https://doi.org/10.5555/3298023.3298212
29. Sun, H., Li, Y., Zhang, Y.: ConLearn: contextual-knowledge-aware concept prerequisite relation learning with graph neural network, pp. 118–126. https://doi.org/10.1137/1.9781611977172.14
30. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2019)
31. Talukdar, P., Cohen, W.: Crowdsourced comprehension: Predicting prerequisite structure in Wikipedia. In: Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, Montréal, Canada, pp. 307–315. Association for Computational Linguistics (2012). https://doi.org/10.5555/2390384.2390423
32. Tang, X., Liu, K., Xu, H., Xiao, W., Tan, Z.: Continual pre-training of language models for concept prerequisite learning with graph neural networks. Mathematics **11**(12) (2023). https://doi.org/10.3390/math11122780
33. Vrandečić, D.: Wikidata: A new platform for collaborative data collection. In: Proceedings of the 21st International Conference on World Wide Web. pp. 1063—1064. WWW '12 Companion, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2187980.2188242
34. Wang, S., Ororbia, A., Wu, Z., Williams, K., Liang, C., Pursel, B., Giles, C.L.: Using prerequisites to extract concept maps fromtextbooks. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. CIKM '16, New York, NY, USA, pp. 317—326. Association for Computing Machinery (2016). https://doi.org/10.1145/2983323.2983725
35. Xiao, K., Bai, Y., Wang, Z.: Extracting prerequisite relations among concepts from the course descriptions. Int. J. Software Eng. Knowl. Eng. **32**(04), 503–523 (2022). https://doi.org/10.1142/S0218194022400034
36. Xiao, K., Fu, Y., Deng, Y., Xia, L.: Identifying prerequisite relations between concepts in Wikipedia. In: 2022 International Conference on Service Science (ICSS), pp. 271–276 (2022). https://doi.org/10.1109/ICSS55994.2022.00049
37. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015 (2015)
38. Yao, L., Mao, C., Luo, Y.: KG-BERT: Bert for knowledge graph completion (2019)

39. Zhang, J., Lan, H., Yang, X., Zhang, S., Song, W., Peng, Z.: Weakly supervised setting for learning concept prerequisite relations using multi-head attention variational graph auto-encoders. Knowl.-Based Syst. **247**, 108689 (2022). https://doi.org/10.1016/j.knosys.2022.108689
40. Zhu, Y., Zamani, H.: Predicting prerequisite relations for unseen concepts. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, pp. 8542–8548. Association for Computational Linguistics (2022). https://aclanthology.org/2022.emnlp-main.585