mathematics

MDPI

Article

Continual Pre-Training of Language Models for Concept Prerequisite Learning with Graph Neural Networks

Xin Tang [1], Kunjia Liu [2], Hao Xu [2,*], Weidong Xiao [1] and Zhen Tan [1]

[1] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China; tangxin20@nudt.edu.cn (X.T.)
[2] Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China; kunjia_liu@nudt.edu.cn
* Correspondence: xuhao@nudt.edu.cn

Abstract: Prerequisite chains are crucial to acquiring new knowledge efficiently. Many studies have been devoted to automatically identifying the prerequisite relationships between concepts from educational data. Though effective to some extent, these methods have neglected two key factors: most works have failed to utilize domain-related knowledge to enhance pre-trained language models, thus making the textual representation of concepts less effective; they also ignore the fusion of semantic information and structural information formed by existing prerequisites. We propose a two-stage concept prerequisite learning model (TCPL), to integrate the above factors. In the first stage, we designed two continual pre-training tasks for domain-adaptive and task-specific enhancement, to obtain better textual representation. In the second stage, to leverage the complementary effects of the semantic and structural information, we optimized the encoder of the resource–concept graph and the pre-trained language model simultaneously, with hinge loss as an auxiliary training objective. Extensive experiments conducted on three public datasets demonstrated the effectiveness of the proposed approach. Our proposed model improved by 7.9%, 6.7%, 5.6%, and 8.4% on ACC, F1, AP, and AUC on average, compared to the state-of-the-art methods.

Keywords: concept prerequisite relationships; pre-trained language model; relational graph convolutional networks; contrastive learning

MSC: 68T50

Citation: 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* **2023**, *11*, 2780. https://doi.org/10.3390/math11122780

Academic Editors: Sławomir Nowaczyk, Rita P. Ribeiro and Grzegorz Nalepa

Received: 19 May 2023
Revised: 13 June 2023
Accepted: 19 June 2023
Published: 20 June 2023

1. Introduction

With the popularity of online education platforms, accessing learning resources has become increasingly convenient; however, the problem of effectively and systematically learning from vast learning resources has become an issue of concern. Concepts are the smallest unit of learning for learners, and constructing the order of concept learning and organization is crucial for learning new knowledge. The prerequisite relationships between concepts can be used to help learners generate reliable learning paths [1], and for some downstream tasks in the education field, such as knowledge tracing [2] and cognitive diagnosis [3].

'Concept prerequisite' refers to the idea that some basic concepts must be understood before tackling more complex or advanced topics: for instance, to comprehend the concept *BERT* in natural language processing, one should first master the concept *Transformer*; similarly, understanding *multi-head attention* and *feed-forward network* is necessary before mastering the concept *Transformer*. Concept prerequisite learning aims to establish a coherent learning sequence between concepts in resources from various sources: specifically, this involves identifying whether two concepts have a prerequisite relationship.

This task has attracted the interest of many researchers. Previous works [4–6] have proposed handcrafted rules and features for learning concept prerequisites from knowledge

graphs, the scientific corpus, and learner behavioral data respectively, and the literature [7] utilizes link information to mine concept prerequisites from Wikipedia. By contrast, recent works have utilized machine learning methods to predict concept prerequisites. These approaches can be divided into two categories: classification-based methods and link-prediction-based methods. Classification-based methods, such as [8,9], mostly follow the text-matching framework, focusing on constructing feature vectors representing the matching relationship between two sentences, and using Siamese networks for prediction. Li et al. [10] found that the BERT [11] model's performance in identifying concept prerequisite relationships was inferior to that of traditional pre-trained language models, and thus, few works have employed the BERT model to obtain concept embedding. Link-prediction-based methods, such as [10,12], typically construct resource–concept heterogeneous graphs, and apply variational graph autoencoder (VGAE) [13] for prediction.

However, existing research has neglected two crucial factors, and thus, learning concept prerequisites remains challenging. Firstly, textual representation is obtained by a traditional pre-trained language model, which is highly reliant on the training corpus: it requires concepts to appear at certain times in training corpora, to obtain effective concept representations, and the representation is fixed according to the statistics of the training corpus. Secondly, the complementary effects of textual and structural information should be further explored: most existing approaches either use textual representations to initialize inputs for graph-based models, or structural representations as inputs for classifiers, which is not an effective way to fuse the two types of information.

This paper, therefore, proposes a two-stage concept prerequisite learning model $(TCPL)$ that combines the strength of two perspectives to solve the challenges above. Firstly, we incorporate domain-specific knowledge, to enhance the pre-trained language model in the continual pre-training stage, so as to obtain better textual representation: specifically, we design the relationship discrimination task to distinguish whether the course contains the given concept, and the pre-trained language model is continually trained on a domain-related corpus with a masked language model for domain-adaptive enhancement, and relationship discrimination tasks for task-specific enhancement. Then, in the joint learning stage, we aim to obtain a better structural representation, and to integrate it with textual features: specifically, Relational Graph Convolutional Networks (R-GCN) are used to obtain the structural representation of concepts in the resource–concept heterogeneous graph. To leverage the complementary effects of textual and structural information, we simultaneously optimize the parameters of the whole model, including the text encoder and the graph encoder, with hinge loss as an auxiliary training objective. The main contributions of this paper are summarized as follows:

- A two-stage framework for concept prerequisite learning is proposed. The pre-trained language model is enhanced by two continual pre-training tasks in the first stage, to obtain better textual representation, the textual and structural information is fused in the second stage, and the prerequisite relationships between concepts are predicted end-to-end;
- A joint optimization approach of R-GCN and pre-trained language models is proposed, with hinge loss as an auxiliary training objective, instead of using them separately as feature extractors, allowing the two models to gradually generate concept representations more suitable for concept prerequisite prediction tasks;
- Extensive experiments were conducted on three real datasets, to evaluate the proposed model. The experimental results demonstrated the effectiveness of the proposed model, compared to all baseline models.

2. Related Works

2.1. Concept Prerequisite Prediction as Text Matching

Concept prerequisite prediction based on the classification perspective refers to classifying and determining the relationship between two concepts, similar to the text-matching task. Early research mainly relied on designing features and rules. Liang et al. [7] proposed

the reference distance model, based on the possibility of a prerequisite relationship between concepts measured by their link density. Liu et al. [14] proposed classification, learning to rank, and the nearest-neighbor search method, to infer prerequisite relationships with a directed graph. Pan et al. [15] first used representation learning to obtain hidden representations for concepts, and proposed seven features based on these representations, to infer relationships. Roy et al. [9] proposed a supervised method, PREREQ, which used a neural network for concept prerequisite relationship recognition, using topic modeling and the paired latent Dirichlet allocation model, to obtain the latent representation of concepts, and prediction based on Siamese networks. Jia et al. [8] considered the relationship between concepts and resources based on PREREQ, and also considered auxiliary tasks, extending this method to the weakly supervised learning setting. Li et al. [16] applied a pre-trained language model to encode text, and utilized link information from web pages between concepts using a graph model.

Previous works based on the classification perspective have mainly referred to the text matching task, emphasizing the semantic information of concept text; however, the concept prerequisite relationship is directional, and has transitivity. These works did not utilize structural information formed by prerequisite relationships already well-established.

2.2. Concept Prerequisite Prediction as Link Prediction

Works that take concept prerequisite prediction as link prediction focus on predicting implicit relationships, by constructing a graph based on existing prerequisite relationships. Li et al. [17] constructed a dataset called LectureBank, and proposed constructing a concept map with each concept in the dataset as a node, for the first time. They then applied a VGAE to learn concept prerequisites from a link prediction perspective; however, inferring solely from existing prerequisites is very limited: in most works, it mainly refers to prerequisites between concepts. Li et al. [10] expanded the concept map into a resource–concept heterogeneous graph, and proposed an R-VGAE model, to consider the multiple relationships between two types of nodes: resource and concept. Li et al. [18] further explored cross-domain concept prerequisite chain learning, using an optimized variational graph autoencoder. However, these models did not distinguish the importance of different nodes, when aggregating neighbor node information. Based on the resource–concept heterogeneous graph, Zhang et al. [12] employed a multi-head attention mechanism and a gated fusion mechanism, to enhance the representation of concepts, and, finally, used a variational graph autoencoder to predict the premise relationships between concepts.

Research based on the link prediction perspective has mainly focused on modeling structures, thereby neglecting the textual semantic information of concepts. While [10,12] used pre-training models to obtain textual representations of concepts as the initial input of the graph model, they were all based on traditional pre-trained models, where the representation of each concept was fixed according to the training corpus.

2.3. Continual Pre-Training of Language Models

Most publicly available pre-trained language models are trained on general domain corpora (such as Wikipedia), resulting in poor performance when applied in specific domains or tasks. Recently, some studies have proposed pre-training language models on professional corpora. MathBERT [19] created a mathematical vocabulary and continual pre-training on a large amount of mathematical text. OAG-BERT [20] is pre-trained continually, based on the Open Academic Graph, and integrates heterogeneous entity knowledge. COMUS [21] continually pre-trains language models for math problem understanding, with a syntax-aware memory network.

In addition to pre-training language models for specific fields, some works have also attempted to design task-oriented pre-training tasks for target applications, such as SentiLR [22] for sentiment analysis, CALM [23] for commonsense reasoning, and DAPO [24] for dialog adaption. In order to tackle challenges such as the inability of pre-trained language models to connect with real-world situations, some works have proposed implic-

itly introducing knowledge, by designing pre-training tasks with knowledge constraints. ERNIE1.0 [25] extended the basic unit of MLM from characters to word segments, and proposed two masking strategies: phrase-level and entity-level. SenseBERT [26] introduced a semantic-level language model, which required the model to predict the hypernym corresponding to the masked word. ERICA [27] designed two contrastive learning tasks, to improve the model's understanding of document-level entities and relationships.

3. Preliminaries

3.1. Resource–Concept Graph

Resources in educational data refer to learning resources that have higher granularity, or are more detailed than the concept. These can be a course or a lecture file, such as a *machine learning* course for the *logistic regression* concept. Typically, these resources contain richer textual information. It is a common saying that "some core concepts should be mastered in this course", so between concepts and courses, there exists an inclusive or a non-inclusive relationship, and a course can also be the prerequisite for another course: for example, *basic probability and statistics* is the prerequisite for *machine learning*. In order to reflect the relationship between courses and concepts in educational data, we constructed a resource–concept heterogeneous graph, as shown in Figure 1.

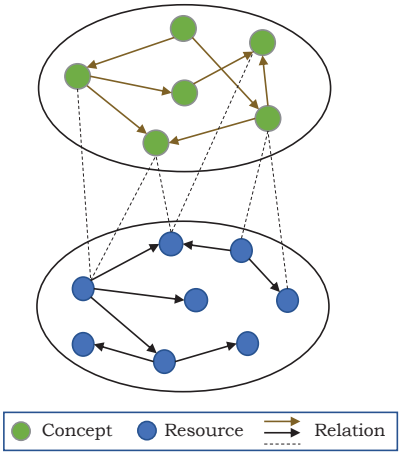

Figure 1. Resource–concept graph, with two types of nodes, and three types of edges. The brown arrow refers to the prerequisites between the concepts, the dark blue arrow refers to the related relationship between the resources, and the dotted line refers to the containment relationship between the concepts and the resources.

The resource–concept graph was denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ was the set of nodes, including two different types, concept and resource, and $\mathcal{E}$ was the set of edges, including three different types: (1) the edge between the resource nodes $\mathcal{E}_{rr}$ representing the related relationship between the resources; (2) the containment relationship between the resources and the concept nodes $\mathcal{E}_{rc}$, representing the resource containing the concept; (3) the existing prerequisites between the concept nodes $\mathcal{E}_{cc}$.

3.2. Task Formulation

We denoted the set of concepts as $C = \{c_1, c_2, \ldots, c_n\}$. Given an unannotated concept pair $\{c_i, c_j\}$, $\mathcal{E}_{rr}, \mathcal{E}_{rc}, \mathcal{E}_{cc}$ in the resource–concept graph, and their text, denoted as $C_{Text} = \left\{x_1^i, \ldots, x_m^i, x_1^j, \ldots, x_n^j\right\}$, where m was the number of words in concept c_i, and n was the length of concept c_j, our goal was to learn a function $\mathcal{F}_\theta$,

$$\mathcal{F}_\theta(C_{Text}, \mathcal{E}_{rr}, \mathcal{E}_{rc}, \mathcal{E}_{cc}) \rightarrow \{0, 1\}, \tag{1}$$

which could predict whether concept c_i was the prerequisite of concept c_j, by mapping the concept pair to a binary class based on the text of concepts and the relationships in the resource–concept graph, where 0 meant there was no prerequisite relationship, while 1 indicated the existence of a relationship.

4. Method

The overall architecture of the model, as shown in Figure 2, was divided into the pre-training and joint optimization stages. The continual pre-training stage aimed to incorporate domain-related knowledge, to enhance the pre-trained language model, while the joint learning stage aimed to leverage the complementary effects of textual and structural information. Specifically, in the continual pre-training stage, concept-related knowledge was injected into the pre-trained language model via training with a masked language model, and relationship discrimination tasks in the domain-related corpus. In the joint learning stage, concept pairs were inputted, to enhance the BERT model, so as to obtain textual representation; meanwhile, the corresponding resource–concept heterogeneous graph was inputted to the R-GCN model, to obtain structure representations of resource and concept nodes. Structural representation of concepts was concatenated with the textual representation, and then fed to the classification layer. Finally, the BERT model, the R-GCN model, and the classification layer were simultaneously optimized with training objective $\mathcal{L}$.

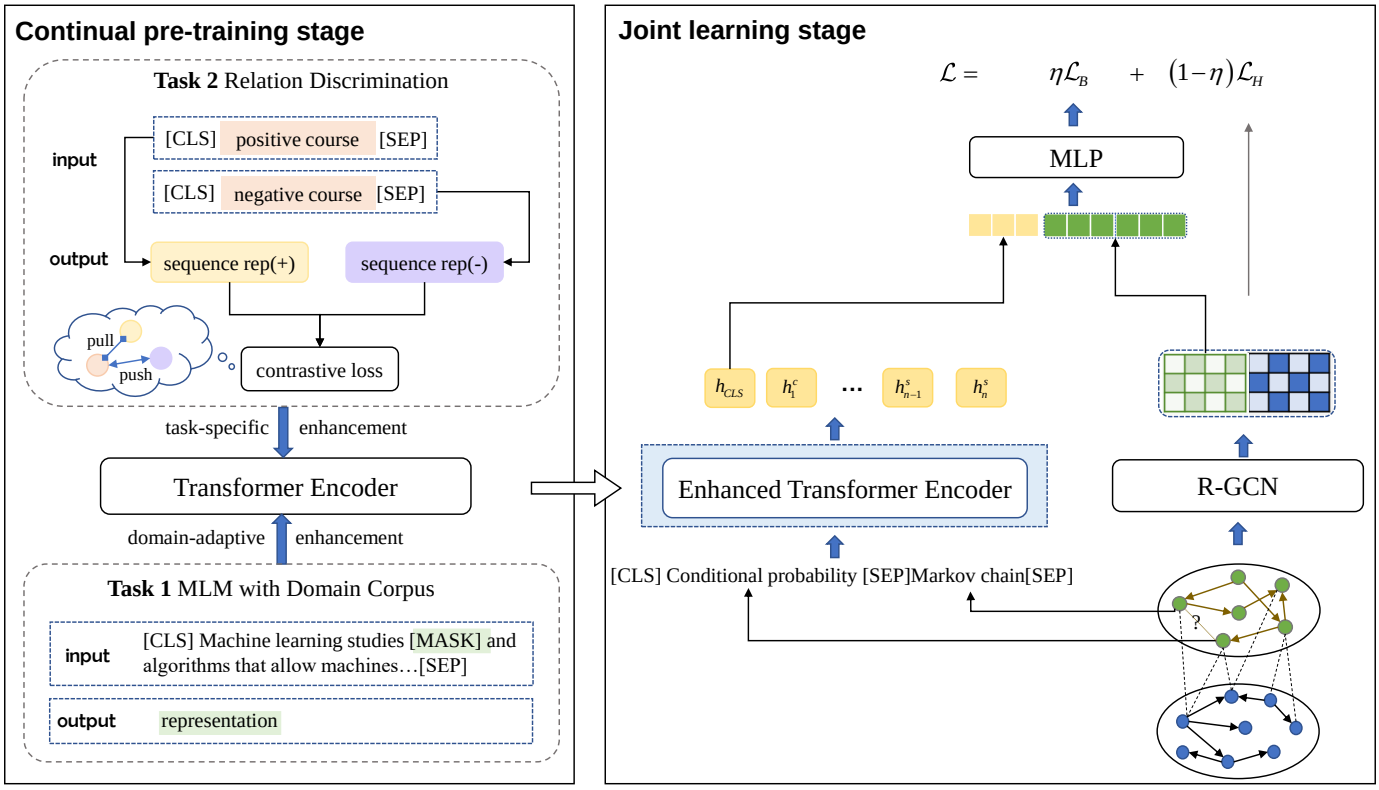

Figure 2. The architecture of our proposed approach. In the continual pre-training stage, the pre-trained language model is enhanced, and fine-tuned with R-GCN in the joint learning stage.

4.1. Continual Pre-Training Stage

4.1.1. Masked Language Model

As the captions of MOOC videos [28,29] contain a number of core concepts about the course, we utilized the masked language model (MLM) task, to achieve domain-adaptive enhancement for a better understanding of the concepts. Specifically, we used the caption text of a MOOC video as a sequence, then randomly selected 15% tokens of

the input sequence to be masked, of which 80% were replaced by a special token [MASK], 10% remained unchanged, and the remaining 10% were replaced by a token randomly selected from the vocabulary. As the [MASK] token did not appear in downstream tasks, the probability selection was meant to alleviate the inconsistency between the pre-training and fine-tuning stages. The objective was to predict the original tokens of the masked ones as

$$L_{MLM} = \sum_{t_i \in \mathcal{V}_{mask}} -\log p(t_i|\tilde{x}), \tag{2}$$

where $\mathcal{V}_{mask}$ denoted the tokens masked in caption text, and $\tilde{x}$ denoted the masked sequence.

4.1.2. Relationship Discrimination

In order to inject knowledge about areas of the concepts into the pre-trained language model, to achieve task-specific enhancement, we designed a relationship discrimination task based on contrastive learning. Contrastive learning aims to acquire effective representation, by bringing semantically similar neighbors closer together, while separating dissimilar non-neighbors [30]: therefore, the relationship discrimination task was proposed, to bring concept and course representations that had relevant relationships close together, and to push apart those that did not. We followed the contrastive framework in [31].

We used BERT as a text encoder to obtain representation, and this is introduced in detail in Section 4.2.1. Constructing positive and negative pairs was a key point in the contrastive framework. We used data from [18,28,29]: specifically, given a concept c_i, we selected a course k_i^+ that contained this concept as positive, and a course k_i^- from another domain, e.g., biology, as negative: for example, given the concept *relational database*, the course *Databases for Informatics* was selected as a positive course, as its description contained the concept, and the course *Basic for Informatics* was selected as a negative one. Moreover, we used in-batch negatives [32]. We let $\mathbf{h}_{c_i}, \mathbf{h}_{k_i^+}, \mathbf{h}_{k_i^-}$ denote the representations of concept c_i, course k_i^+, and course k_i^+, respectively. The training objective InfoNCE [33] was defined by

$$\mathcal{L}_{RD} = -\sum_{i=1}^{N} \log \frac{e^{sim(\mathbf{h}_{c_i}, \mathbf{h}_{k_i^+})/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{h}_{c_i}, \mathbf{h}_{k_j^+})/\tau} + e^{sim(\mathbf{h}_{c_i}, \mathbf{h}_{k_j^-})/\tau}}, \tag{3}$$

where $sim(\mathbf{h}_{c_i}, \mathbf{h}_{k_i}) = \mathbf{h}_{c_i}^\top \mathbf{h}_{k_i} / \|\mathbf{h}_{c_i}\| \|\mathbf{h}_{k_i}\|$ calculated the cosine similarity, N was the batch size, and τ represented the temperature hyperparameter. The role of the temperature hyperparameters was to dig out difficult negative samples, with smaller temperature hyperparameters increasing the distance between difficult negative samples and positive samples.

4.2. Joint Learning Stage

4.2.1. Text Encoder BERT

After the continual pre-training stage, enhanced BERT was used to encode concept text. Given the text of a concept pair, along with a special token [CLS] at the first position, and [SEP] to separate the text of different concepts $\{[\text{CLS}], x_1^1, \ldots, x_m^1, [\text{SEP}], x_1^2, \ldots, x_n^2\}$, then a conversion to a sequence of BERT embeddings could be made, by summing the position embedding, segment embedding, and token embedding.

BERT is composed of L Transformer [34] encoder layers. Specifically, it consists of stacks of multi-head self-attention layers (denoted by MHAttn(·)) and point-wise feed-forward networks (denoted by FFN(·)). With the output of the $(l-1)$-th layer represented as $\mathbf{C}^{l-1}$, the input and the update process can be formalized as follows:

$$\tilde{\mathbf{C}}^l = \text{LayerNorm}\left(\text{MHAttn}\left(\mathbf{C}^{l-1}\right) + \mathbf{C}^{l-1}\right); \tag{4}$$

$$\mathbf{C}^l = \text{LayerNorm}\left(\tilde{\mathbf{C}}^l + \text{FFN}\left(\tilde{\mathbf{C}}^l\right)\right), \tag{5}$$

where $\mathbf{C}^l = \left\{\mathbf{h}_{CLS}, \mathbf{h}_1^1, \ldots, \mathbf{h}_n^2\right\}$. We adopted $\mathbf{h}_{CLS}$, the final hidden embedding of special token [CLS] as textual representation for the concept pair.

4.2.2. Graph Encoder R-GCN

After constructing the resource–concept graph, R-GCN were used to encode this heterogeneous graph. In R-GCN, the representation of node v_i was updated as follows:

$$\mathbf{v}_i^{l+1} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}\mathbf{W}_r^{(l)}\mathbf{h}_j^{(l)} + \mathbf{W}_i^{(l)}\mathbf{h}_i^{(l)}\right), \tag{6}$$

where $\mathcal{N}_i^r$ was the set of neighbor nodes, given node v_i and relationship $r \in \mathcal{R}$, and where $c_{i,r}$ was a normalization constant, $\mathbf{W}_r^{(l)}$ was the linear transformation parameter matrix of layer l, and $\mathbf{h}_i^{(l)}$ was the hidden representation of node v_i in the l-th layer.

As Equation (6) expresses, R-GCN update the features of a node by aggregating information from its neighboring nodes, and incorporating their feature representations into its own; furthermore, this allows for incorporating relationship-specific parameters when performing feature aggregation. Thus, the importance of different relationships in the resource–concept graph is taken into consideration when learning the structural representation of concepts.

4.2.3. Joint Learning Layer

After obtaining textual representations $\mathbf{h} \in \mathbb{R}^H$ of concept pair(c_i, c_j) from the text encoder, and nodes features $\mathbf{V}$ from the graph encoder, we extracted corresponding concept structure representation $\mathbf{v}_i$, $\mathbf{v}_j \in \mathbb{R}^{N_d}$ from $\mathbf{V}$. H was the dimension of textual representation, and N^d was the dimension of structural representation. Then, we concatenated textual and structural representations describing features from different aspects: $\mathbf{X} = [\mathbf{h}, \mathbf{v}_i, \mathbf{v}_j]$. The probability that concept pairs have prerequisite relationships was calculated as follows:

$$p = \text{Sigmoid}(\mathbf{W}_B\mathbf{X}) \in \mathbb{R}^2, \tag{7}$$

where $\mathbf{X} \in \mathbb{R}^{2N^d+H}$, $\mathbf{W}_B \in \mathbb{R}^{2\times\left(2N^d+H\right)}$ was the parameter matrix in the classification layer. For concept prerequisite prediction, we used binary cross-entropy (BCE) loss as a training objective:

$$\mathcal{L}_B = -\sum_{k=1}^{n}\left(y_k \log p_k + (1-y_k)\log(1-p_k)\right), \tag{8}$$

where $y_k \in \{0,1\}$ was the true label of the relationship between concepts, p_k denoted the possibility for concept pair k to have a prerequisite relationship, and n denoted the batch size.

In order to enable both models to gradually generate more effective representations during training, especially for obtaining better structural representation, we used hinge loss as an auxiliary training objective. Hinge loss aims to achieve a higher score for valid triplets, and is formulated as follows:

$$\mathcal{L}_H = \sum_{t_{ij}^k\in\mathcal{G}} max\left(\gamma + d_{t_{ij}'^k} - d_{t_{ij}^k}, 0\right), \tag{9}$$

where $\gamma > 0$ is a margin hyperparameter, $d_{t_{ij}^k} = \|\mathbf{e}_i + \mathbf{r}_k - \mathbf{e}_j\|_1$ is the translation distance [35], and $t_{ij}^{'k}$ is the corrupted triplet, issued as

$$t_{ij}^{'k} = \left\{ < e_i^{'}, r_k, e_j > | e_i^{'} \in \mathcal{E} \backslash e_i \cup < e_i, r_k, e_j^{'} > | e_j^{'} \in \mathcal{E} \backslash e_j \right\}. \tag{10}$$

The overall training objective was formulated as follows:

$$\mathcal{L} = \eta \mathcal{L}_B + (1 - \eta)\mathcal{L}_H, \tag{11}$$

where η was the hyperparameter that traded off the impact of the auxiliary loss. The parameter of the text encoder and graph encoder would be updated during training.

5. Experiments

5.1. Experimental Setup

5.1.1. Datasets and Evaluation Metrics

We conducted experiments on three public datasets, to evaluate our proposed model and baselines. The statistics of the datasets are provided in Table 1. The number of concepts was denoted by $|C|$, while $|R|$ was the number of resources. The number of concept prerequisite edges, the number of edges between resource nodes, and the number of edges between resource and concept nodes were represented by $\left|C_{edge}\right|$, $\left|R_{edge}\right|$, and $\left|T_{edge}\right|$, respectively. The three public datasets were:

- **The University Course dataset** (UCD) [29], which includes 654 computer science courses from universities in the USA, and 407 concepts. There are also prerequisite relationships of courses and concepts, respectively, in this dataset. For edges between courses and concepts, we assumed that a relationship existed if a concept appeared in the course captions;
- **The LectureBank dataset** (LBD) [17], which includes lecture files and topics from five domains: artificial intelligence; machine learning; natural language processing; deep learning; and information retrieval. We considered lecture files as resources, topics as concepts in this dataset, and hierarchy relationships between topics. For resource edge construction, we computed the cosine similarity of lecture file embedding, and set the threshold as 0.9;
- **The MOOC dataset** (MD) [9], which contains 382 MOOC video texts of computer science courses, and the same topic and number of concept prerequisite relationship pairs as in the University Course dataset. The construction of edges between resources and concepts was the same as the UCD dataset.

Table 1. The statistics of the datasets.

Dataset	$\|C\|$	$\|R\|$	$\left\|C_{edge}\right\|$	$\left\|R_{edge}\right\|$	$\left\|T_{edge}\right\|$
UCD	407	654	1008	861	580
LBD	307	250	471	995	265
MD	406	382	1004	1404	3634

We adopted widely used evaluation metrics, including Accuracy, F1 score, Average Precision, and Area Under the ROC Curve (AUC). Accuracy represented the proportion of all classified samples that consisted of correctly predicted samples. F1 score was the weighted average of precision and recall. Precision was the proportion of positive samples that were correctly predicted as positive among all the samples predicted as positive. AP was the average precision obtained by taking the mean of the precision values on the PR curve. AUC represented the probability that the model scores positive examples would be

higher than the negative ones. All the evaluation metrics had larger values, indicating the better performance of the model.

5.1.2. Baseline Models

We compared our proposed method to the following baseline models:

- **PREREQ** [9]: we used the Pairwise Latent Dirichlet Allocation model to obtain the latent representation of concepts, and fed them into a Siamese network, to infer the relationships between concepts;
- **BERT-base** [11]: we fine-tuned on datasets, and used the [CLS] vector of concept pairs for prediction;
- **DAPT-BERT** [36]: continual pretraining BERT on domain-related corpus. We used the captions of computer science courses [28], with the masked language model task for implementation;
- **VGAE** [13]: the AU unsupervised model combines the ideas of autoencoder and variational inference; it samples the latent variables from a multidimensional Gaussian distribution, and the decoder predicts the label, based on the latent variables;
- **R-GCN** [37]: the core idea of R-GCN is to process tasks by learning the embedding vectors of nodes and relationships; it has strong scalability in dealing with multiple types of relationships;
- **R-GCN(BERT)**: using the textual representation of concepts from BERT as the initialization of R-GCN;
- **R-VGAE** [10]: a model that combines the advantages of R-GCN and VGAE; it updates the original GCN encoder of VGAE to R-GCN that consider relationships and use DistMult as the training objective for reconstruction; it initializes node features with two types of features, TF-IDF and Phrase2vec [38];
- **MHAVGAE** [12]: this model constructs a resource–concept heterogeneous graph, initializing node feature with word2vec [39], and then uses multi-head attention and gating mechanisms to enhance concept representation; finally, it uses a variational graph autoencoder to predict the relationships between concepts.

5.1.3. Implementation Details

The experiments were conducted in Python, with the deep learning toolkit PyTorch, and we trained our model on NVIDIA GeForce RTX 3090. For performance comparison of all models, we divided the dataset into a training set, a validation set, and a test set, at a rate of 8:1:1. For the baseline models, the results of R-GCN https://github.com/tkipf/relational-gcn (accessed on 12 June 2023), VGAE https://github.com/zfjsail/gae-pytorch (accessed on 12 June 2023) R-VGAE https://github.com/Yale-LILY/LectureBank/tree/master/LB-Paper/LectureBank2 (accessed on 12 June 2023), and MHAVGAE https://github.com/zhang-juntao/MHAVGAE (accessed on 12 June 2023) were re-evaluated by us, using the released code. All the hyperparameters were set, following the suggestions from the original papers, to obtain optimal results. For BERT-related models, we implemented them based on HuggingFace Transformers [40].

For our model, in the pre-training stage, we initialized the weights of BERT with BERT-base https://huggingface.co/bert-base-uncased (accessed on 12 June 2023), then pre-trained on a corpus with a total of 32 batch sizes for 150 steps. The max length of input sequences was set to 512, τ was set to 0.05 for the relationship discrimination task, using the AdamW [41] optimizer, and the learning rate was set to 2×10^{-4}. In the joint learning stage, we trained our model using the AdamW optimizer for 20 epochs at most, and the batch size was set to 64. For fairness, we deleted the corresponding concept edge in the test set in the resource–concept heterogeneous graph, when training. As the parameters of R-GCN and fully connected layers were randomly initialized, and the parameters of the BERT model were updated, based on a large amount of the pre-training corpus, the learning rates were set differently, according to the different networks. The learning rates of the R-GCN and fully connected layers were set to 0.01, and the learning rate of the BERT model was set

to 10×10^{-6}. The hyperparameter that balanced the two kinds of losses was set to 0.5, 0.7, and 0.7 on the three datasets, respectively. The margin hyperparameter γ was set as 1. The number of R-GCN layers was set as 2, and the embedding size of the first layer was set to 16, 32, and 128 on the three datasets, respectively. The second layer contained 32 hidden units.

5.2. Main Results

As mentioned in Related Works, the baselines could also be divided into two groups, where the first group consisted of classification models, and the second group interpreted concept prerequisite learning as link prediction based on graph-network-based models. The experiment results of five runs of all methods are reported in Table 2, and the optimal results are marked in bold.

Table 2. Overall performance of the UCD, LBD, and MOOC datasets. The optimal results are marked in bold, and the sub-optimal results are underlined.

Dataset	Method	ACC	F1	AP	AUC
UCD	PREREQ ◇	0.5433	0.5866	0.5309	0.6702
	BERT-base	0.6916	0.6635	0.6412	0.7433
	DAPT BERT	0.7173	0.7085	0.7497	0.7944
	R-GCN	0.6450	0.5989	0.6333	0.6548
	VGAE	0.6700	0.6413	0.7534	0.6972
	R-GCN (BERT)	0.6100	0.5244	0.5964	0.6200
	R-VGAE	0.6950	0.6772	0.8073	0.7661
	MHAVGAE	<u>0.7450</u>	<u>0.7330</u>	<u>0.8201</u>	<u>0.7797</u>
	TCPL (ours)	**0.8088**	**0.7900**	**0.8434**	**0.8668**
LBD	PREREQ ◇	0.4875	0.5130	0.5032	0.5557
	BERT-base	0.6526	0.6207	0.6143	0.6516
	DAPT BERT	0.6526	0.6374	<u>0.7677</u>	0.7176
	R-GCN	0.5394	0.5921	0.5870	0.5840
	VGAE	0.5904	0.5792	0.5733	0.6053
	R-GCN (BERT)	0.5120	0.5239	0.5357	0.5536
	R-VGAE	0.6538	0.6764	0.6467	0.6338
	MHAVGAE	<u>0.6774</u>	<u>0.6899</u>	0.7608	<u>0.7256</u>
	TCPL (ours)	**0.7737**	**0.7774**	**0.8380**	**0.8393**
MOOC	PREREQ ◇	0.5429	0.5746	0.5286	0.6248
	BERT-base	<u>0.7645</u>	0.7776	0.8313	0.8461
	DAPT BERT	0.7628	<u>0.7851</u>	0.8428	0.8564
	R-GCN	0.6500	0.5532	0.5742	0.6208
	VGAE	0.6550	0.5818	0.7371	0.7045
	R-GCN (BERT)	0.6120	0.5346	0.5333	0.6037
	R-VGAE	0.7050	0.7204	0.7978	0.7544
	MHAVGAE	0.7485	0.7653	<u>0.8832</u>	<u>0.8789</u>
	TCPL (ours)	**0.8400**	**0.8411**	**0.9115**	**0.9076**

◇ Results are from [12].

The classification group included three typical methods based on deep learning: PREREQ; BERT-base; and DAPT BERT. The results indicated that continual pre-training on a domain-related corpus helps improve model performance. Compared to the graph-network-based group based on these results, we found that, while the classification group performed close to or even better on Accuracy and F1 scores than the graph-network-based group, the performance on AP tended to be poorer, compared to graph-based methods. For a binary classification task, AP reflects the classifier's ability to correctly identify positive samples while incorrectly categorizing negative samples when the probability threshold is varied. Unlike graph-based methods that predict the existence of prerequisite relationships by performing inner products on node representations and specifying thresholds, classification models often choose the label with the highest predicted probability as the

final prediction, which is equivalent to a constant threshold of 0.5; therefore, graph-based methods tend to score higher on this metric. We found that the graph-based methods, R-VGAE and MHAVGAE, performed better than the BERT model. Moreover, the overall performances of the models on the LBD dataset were generally worse than on the UCD dataset, mainly due to the lower edge density in the LBD dataset.

As for the graph-based models, MHAVGAE—which utilizes multi-head attention and gating mechanisms—performed better than the other methods. The unsupervised VGAE method performed better than R-GCN. However, using BERT to obtain concept representations as the initial node inputs for R-GCN resulted in decreased performance, as the node representations in the graph neural networks were often of low dimensionality. The high-dimensional concept representations obtained by BERT could cause overfitting, and were not in the same representation space as those obtained by graph neural networks. Despite this, the comprehensive performance of these models was not as good as our proposed model. The reason was that, except for the BERT+RGCN baseline model, the models mentioned above all used traditional pre-trained language models as initialization for concept representation, which had two limitations: firstly, it required text corpora that included concepts with a certain frequency of occurrence, to obtain concept representations; secondly, using concept representations as the initial input for the graph model did not guarantee that the model made full use of the text features of the concepts when predicting, unlike the joint optimization method proposed in this paper, which was shown to be effective in the experimental results.

The experimental results show that our proposed model significantly improved performance on all metrics, compared to the baseline model. The proposed model achieved an improvement of 7.9%, 6.7%, 5.6%, and 8.4%, on average, on ACC, F1, AP, and AUC, respectively, compared to the baseline model with a suboptimal F1 value, demonstrating the proposed model's effectiveness. Our proposed model was also from the classification perspective, but combined the strength of link prediction, and ultimately outperformed all baselines, for three main reasons: (1) design of a continual pre-training stage to inject domain-related knowledge into the pre-trained language model; moreover, use of our enhanced BERT to encode the text of concepts, which not only eliminated the restriction mentioned above, as in traditional pre-trained language models such as word2vec and phrase2vec, but also enabled the obtaining of better textual features based on the rich knowledge encoded in the model; (2) for the constructed resource–concept heterogeneous graph, using R-GCN to obtain resource-enhanced concept representations allowed the incorporation of information related to other concept-related resources in the final concept structural features used for prediction; (3) jointly optimizing the text encoder and graph encoder, rather than using them merely as feature extractors, fully leveraged the complementary effects of the features obtained by both encoders.

5.3. Ablation Study

Our proposed approach contains several complementary modules: thus, we conducted an ablation study, to prove the effectiveness and contribution of these modules. Specifically, we removed the module of BERT, the R-GCN, the optimization goal BCE loss and hinge loss, or the continual pre-training, respectively. The results are provided in Table 3, where w/o CP means without continual pre-training. We implemented it by replacing the text encoder in the joint learning stage with BERT-base.

Table 3. The ablation study performance comparison.

Method	UCD				LBD				MD			
	ACC	**F1**	**AP**	**AUC**	**ACC**	**F1**	**AP**	**AUC**	**ACC**	**F1**	**AP**	**AUC**
TCPL	**0.8088**	**0.79**	**0.8434**	**0.8668**	**0.7737**	**0.7774**	**0.838**	**0.8393**	**0.8400**	**0.8411**	**0.9115**	**0.9076**
-w/o *BERT*	0.6450	0.5989	0.6333	0.6548	0.5394	0.5921	0.587	0.584	0.6500	0.5532	0.5742	0.6408
-w/o *R-GCN*	0.6916	0.6635	0.6412	0.7433	0.6526	0.6207	0.6143	0.6516	0.7645	0.7776	0.8313	0.8461
-w/o *hinge loss*	0.8062	0.7877	0.7768	0.8416	0.7333	0.7039	0.8176	0.7958	0.7463	0.7584	0.7026	0.7555
-w/o *BCE loss*	0.4493	0.6177	0.4828	0.5372	0.5395	0.5270	0.5494	0.5528	0.5249	0.6056	0.5907	0.5841
-w/o *CP*	0.8017	0.7798	0.8366	0.8608	0.7605	0.7598	0.8163	0.8135	0.8192	0.8256	0.8812	0.8870

Table 3 shows the effectiveness of these modules or continual pre-training in our proposed model. When each encoder was implemented separately, the performance decreased significantly in all metrics, with a minimum drop of 12% compared to our proposed approach. This demonstrated that the proposed model effectively leveraged the complementary effects of the different aspects of information learned by the two encoders. When removing one training goal, our model performance also became poorer. In particular, the model performance decreased sharply when we removed the BCE loss, which implied that it was essential to train our model effectively with BCE loss. Training without hinge loss had a more significant impact on the LBD dataset than on the UCD dataset: it weakened the model's effectiveness mainly on AP on the UCD dataset, but reduced the overall performance on the LBD and MD datasets. In addition, removing continual pre-training also resulted in reduced efficacy on all datasets. However, continual pre-training did not significantly improve the model's performance, especially on the UCD dataset: this may have been due to the relatively small size of the training corpus, and the max length of input in our experiment.

We also tried the gated mechanism, to fuse textual and structural representation, but the results indicated that it hurt the model performance. We concluded that the reason was the different dimensions of textual and structural representation, and we will explore more effective ways of fusion in the future.

5.4. Model Analysis

In our model, the hyperparameter η and embedding size are vital parameters. Here, we report the performance of our model, based on the above different hyperparameters.

5.4.1. Hyperparameter η

We analyzed the effects of different hyperparameters η. We set the learning rate at 0.01, the embedding dimension at 16, and we varied η as $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. The experimental results are reported in Figure 3. Based on the results, we found that on the University Course dataset, with η increasing from 0.3 to 0.5, the model performed better, while increasing from 0.5 to 0.7 resulted in decreasing model efficacy. When η was 0.5, our model performed at its best. In the LectureBank dataset, the situation was different. The model performance showed a slight improvement, followed by a subsequent decline, as the parameter increased from 0.3 to 0.6, and the model achieved optimal performance when η was 0.7: this was because this hyperparameter controlled the ratio of two types of loss, and the BCE focused more on classification, while the hinge loss aimed at learning better structural representation. The statistics of the datasets show that the LectureBank dataset had fewer edges between concepts: in that circumstance, textual information was more helpful for model performance. On the MOOC dataset, our proposed model's performance was relatively stable, as the hyperparameter changed. The model performed best when η was 0.3 or 0.7, illustrating that this ratio was effective on the MOOC dataset.

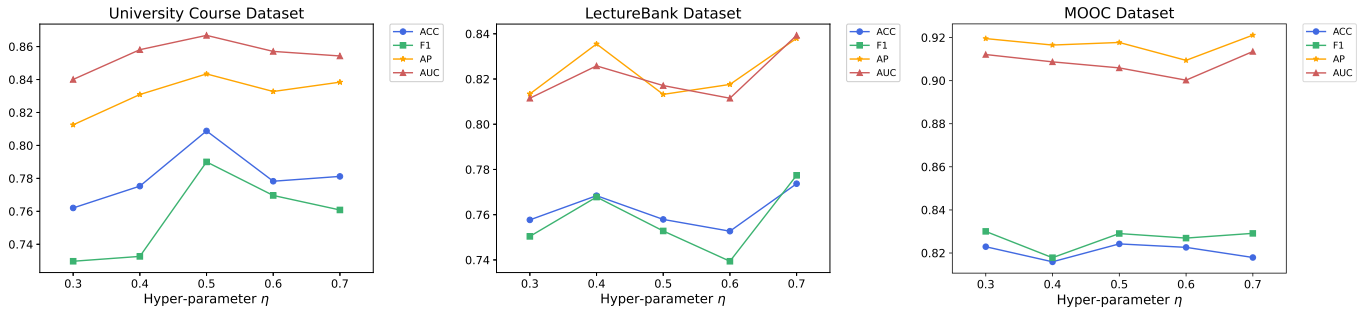

Figure 3. Performance comparison with different hyperparameter η.

5.4.2. Embedding Dimension

We set the embedding size of the R-GCN to η in $\{16, 32, 64, 128, 256\}$, to investigate the effect on model performance when the dimension of textual representation from BERT was fixed at 768, and the hyperparameter η was set to 0.5, 0.7, and 0.7, respectively. The experimental results are provided in Figure 4. Based on the results, we found that our model achieved the best performance when the embedding size was 16 on the University Course dataset. Furthermore, as the embedding size increased, the model performance initially decreased at 32 dimensions, then improved at 64 dimensions, and remained relatively stable, with only minor fluctuations in performance observed up to 256 dimensions. On the LectureBank dataset, our model performed best when the embedding size was 32, and the model's performance fluctuated more as the embedding size changed. On the MOOC dataset, the model performed best when the embedding size was 128 or 256.

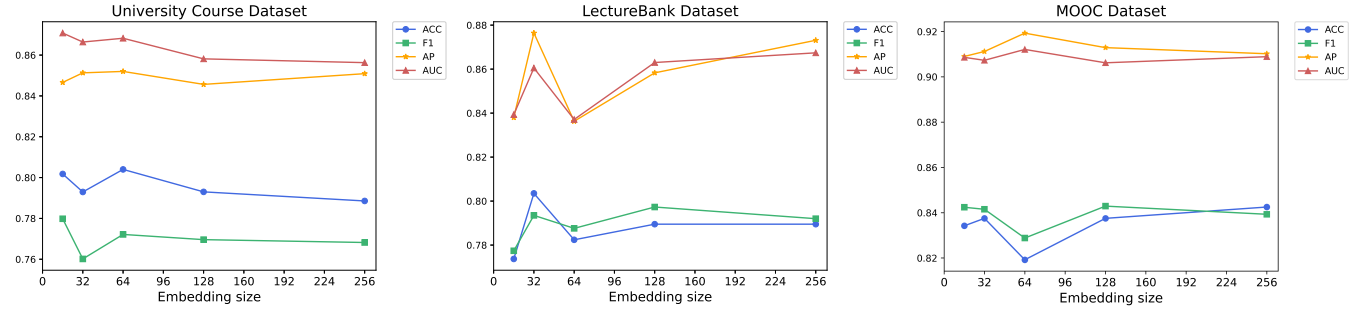

Figure 4. Performance comparison with different embedding size.

6. Conclusions

In this paper, we have proposed TCPL, a two-stage framework for concept prerequisite learning. In the continual pre-training stage, we designed the relationship discrimination task, together with a masked language model to enhance the pre-trained language model, so as to obtain better textual representations of the concepts. In the joint learning stage, we leveraged the complementary effects of the semantic and structural information. Specifically, we constructed a resource–concept graph, and utilized hinge loss with BCE loss, to simultaneously optimize the pre-trained language model and graph encoder R-GCN. Our approach outperformed all competitive baselines in experimental studies on three public datasets.

Nonetheless, this study demonstrated three major limitations: firstly, our proposed model cannot utilize information from other modalities, such as vision or speech; secondly, we mainly evaluated our model on computer science subjects, which cannot guarantee the best performance when applied to datasets from other areas; thirdly, there are many relationship between concepts, but our proposed model was only trained to predict prerequisite relationships.

In the future, we will explore a more practical approach to modeling heterogeneous graphs, and an effective way to merge different feature types. Additionally, we plan to consider multiple types of relationships beyond merely the prerequisites between concepts.

Author Contributions: Conceptualization, X.T.; methodology, X.T., Z.T.; investigation, H.X.; validation, H.X. writing—original draft preparation, X.T.; writing—review and editing, K.L., X.T.; visualization, X.T., K.L.; supervision, Z.T.; project administration, W.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study are openly available in the literature [10,28,29].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Changuel, S.; Labroche, N.; Bouchon-Meunier, B. Resources Sequencing Using Automatic Prerequisite-Outcome Annotation. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 1–30. [CrossRef]
2. Lu, Y.; Chen, P.; Pian, Y.; Zheng, V.W. CMKT: Concept Map Driven Knowledge Tracing. *IEEE Trans. Learn. Technol.* **2022**, *15*, 467–480. [CrossRef]
3. Gao, W.; Liu, Q.; Huang, Z.; Yin, Y.; Bi, H.; Wang, M.; Ma, J.; Wang, S.; Su, Y. RCD: Relation Map Driven Cognitive Diagnosis for Intelligent Education Systems. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 501–510.
4. Manrique, R.; Nunes, B.P.; Mariño, O.; Cardozo, N.; Siqueira, S.W.M. Towards the Identification of Concept Prerequisites Via Knowledge Graphs. In Proceedings of the 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 15–18 July 2019; pp. 332–336.
5. 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, Berlin, Germany, 7–12 August 2016.
6. Chen, W.; Lan, A.S.; Cao, D.; Brinton, C.G.; Chiang, M. Behavioral Analysis at Scale: Learning Course Prerequisite Structures from Learner Clickstreams. In Proceedings of the International Conference on Educational Data Mining, Raleigh, NC, USA, 16–20 July 2018.
7. 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, the Association for Computational Linguistics, Lisbon, Portugal, 17–21 September 2015; pp. 1668–1674.
8. 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, Online, 6–11 June 2021; pp. 2036–2047.
9. 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, Waikiki, HI, USA, 27 January–1 February 2019; AAAI Press: Washington, DC, USA, 2019; pp. 9589–9594.
10. Li, I.; Fabbri, A.R.; Hingmire, S.; Radev, D.R. R-VGAE: Relational-variational Graph Autoencoder for Unsupervised Prerequisite Chain Learning. In Proceedings of the COLING, International Committee on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 1147–1157.
11. Devlin, J.; Chang, M.; 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, Minneapolis, MO, USA, 2–7 June 2019; pp. 4171–4186.
12. Zhang, J.; Lin, N.; Zhang, X.; Song, W.; Yang, X.; Peng, Z. Learning Concept Prerequisite Relations from Educational Data via Multi-Head Attention Variational Graph Auto-Encoders. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual, 21–25 February 2022; pp. 1377–1385.
13. Kipf, T.N.; Welling, M. Variational graph auto-encoders. *arXiv* **2016**, arXiv:1611.07308.
14. Liu, H.; Ma, W.; Yang, Y.; Carbonell, J.G. Learning Concept Graphs from Online Educational Data. *J. Artif. Intell. Res.* **2016**, *55*, 1059–1090. [CrossRef]
15. 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, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1447–1456.
16. Li, B.; Peng, B.; Shao, Y.; Wang, Z. Prerequisite Learning with Pre-trained Language and Graph Embedding Models. In *Natural Language Processing and Chinese Computing, Proceedings of the 10th CCF International Conference, NLPCC 2021, Qingdao, China, 13–17 October 2021, Proceedings, Part II 10*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; Volume 13029, pp. 98–108.

17. 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, Waikiki, HI, USA, 27 January–1 February 2019; AAAI Press: Washington, DC, USA, 2019; pp. 6674–6681.
18. Li, I.; Yan, V.; Li, T.; Qu, R.; Radev, D.R. Unsupervised Cross-Domain Prerequisite Chain Learning using Variational Graph Autoencoders. *arXiv* **2021**, arXiv:2105.03505.
19. Shen, J.T.; Yamashita, M.; Prihar, E.; Heffernan, N.T.; Wu, X.; Lee, D. MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education. *arXiv* **2021**, arXiv:2106.07340.
20. Liu, X.; Yin, D.; Zheng, J.; Zhang, X.; Zhang, P.; Yang, H.; Dong, Y.; Tang, J. OAG-BERT: Towards a Unified Backbone Language Model for Academic Knowledge Services. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 3418–3428.
21. Gong, Z.; Zhou, K.; Zhao, X.; Sha, J.; Wang, S.; Wen, J. Continual Pre-training of Language Models for Math Problem Understanding with Syntax-Aware Memory Network. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 5923–5933.
22. Ke, P.; Ji, H.; Liu, S.; Zhu, X.; Huang, M. SentiLR: Linguistic Knowledge Enhanced Language Representation for Sentiment Analysis. *arXiv* **2019**, arXiv:1911.02493.
23. Zhou, W.; Lee, D.; Selvam, R.K.; Lee, S.; Ren, X. Pre-training Text-to-Text Transformers for Concept-centric Common Sense. *arXiv* **2020**, arXiv:2011.07956.
24. Li, J.; Zhang, Z.; Zhao, H.; Zhou, X.; Zhou, X. Task-specific Objectives of Pre-trained Language Models for Dialogue Adaptation. *arXiv* **2020**, arXiv:2009.04984.
25. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv* **2019**, arXiv:1904.09223.
26. Levine, Y.; Lenz, B.; Dagan, O.; Ram, O.; Padnos, D.; Sharir, O.; Shalev-Shwartz, S.; Shashua, A.; Shoham, Y. SenseBERT: Driving Some Sense into BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 4656–4667.
27. Qin, Y.; Lin, Y.; Takanobu, R.; Liu, Z.; Li, P.; Ji, H.; Huang, M.; Sun, M.; Zhou, J. ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; pp. 3350–3363.
28. Pan, L.; Wang, X.; Li, C.; Li, J.; Tang, J. Course Concept Extraction in MOOCs via Embedding-Based Graph Propagation. In Proceedings of the Eighth International Joint Conference on Natural Language Processing, Taipei, Taiwan, 27 November–1 December 2017; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2017; pp. 875–884.
29. Liang, C.; Ye, J.; Wu, Z.; Pursel, B.; Giles, C.L. Recovering Concept Prerequisite Relations from University Course Dependencies. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; AAAI Press: Washington, DC, USA, 2017; pp. 4786–4791.
30. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality Reduction by Learning an Invariant Mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 1735–1742.
31. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G.E. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the International Conference on Machine Learning, Proceedings of Machine Learning Research, Virtual, 13–18 July 2020; Volume 119, pp. 1597–1607.
32. Chen, T.; Sun, Y.; Shi, Y.; Hong, L. On Sampling Strategies for Neural Network-based Collaborative Filtering. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 767–776.
33. Gutmann, M.; Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; Volume 9, pp. 297–304.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
35. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2013; Volume 26.
36. Gururangan, S.; Marasovic, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8342–8360.
37. Schlichtkrull, M.S.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the 15th International Conference, ESWC 2018, Heraklion, Greece, 3–7 June 2018; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10843, pp. 593–607.
38. Wu, Y.; Zhao, S.; Li, W. Phrase2Vec: Phrase embedding based on parsing. *Inf. Sci.* **2020**, *517*, 100–127. [CrossRef]

39. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the ICLR (Workshop Poster), Scottsdale, AZ, USA, 2–4 May 2013.
40. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 conference on Empirical Methods in Natural Language Processing: System Demonstrations (Demos), Association for Computational Linguistics, Online, 16–20 November 2020; pp. 38–45.
41. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the ICLR (Poster), New Orleans, LA, USA, 6–9 May 2019.