Heterogeneous Graph Neural Networks for Concept Prerequisite Relation Learning in Educational Data

Chenghao Jia, Yongliang Shen, Yechun Tang, Lu Sun, Weiming Lu*
College of Computer Science and Technology, Zhejiang University, Hangzhou, China
{chjia, syl, tangyechun, sunlu98, luwm}@zju.edu.cn

Abstract

Prerequisite relations among concepts are crucial for educational applications, such as curriculum planning and intelligent tutoring. In this paper, we propose a novel concept prerequisite relation learning approach, named CPRL, which combines both concept representation learned from a heterogeneous graph and concept pairwise features. Furthermore, we extend CPRL under weakly supervised settings to make our method more practical, including learning prerequisite relations from learning object dependencies and generating training data with data programming. Our experiments on four datasets show that the proposed approach achieves the state-of-the-art results comparing with existing methods.

1 Introduction

With the increasing availability of learning resources and the requirement of self-regulated learning, there is a rising need to organize knowledge in a reasonable order. Concept prerequisite relations are essentially considered as the dependency among concepts, and they are crucial for people to learn, organize, apply and generate knowledge (Margolis and Laurence, 1999). For example, if someone wants to learn the knowledge about *Conditional Random Fields*, the knowledge about *Hidden Markov Model* should be learned first. Consequently, the concept *Hidden Markov Model* is a prerequisite concept of the concept *Conditional Random Fields*. Nowadays, prerequisite relations among concepts have played a crucial role in educational applications, such as curriculum planning (Liu et al., 2016) and intelligent tutoring (Wang and Liu, 2016; Chen et al., 2018).

Recently, several attempts have been made to extract prerequisite relations among concepts from textbooks (Wang et al., 2016; Liang et al., 2018), MOOCs (Massive Open Online Courses) (Pan et al., 2017), courses (Liang et al., 2015a; Liu et al., 2016; Liang et al., 2017; Li et al., 2019a; Roy et al., 2019) and scientific papers (Gordon et al., 2016). They either proposed a local statistical information, such as *reference distance* (Liang et al., 2015a) and *cross-entropy* (Gordon et al., 2016) to measure the prerequisite relations between concepts, or proposed handcrafted features to learn a prerequisite relation classifier (Pan et al., 2017). Liang et al. (2017) proposed CPR-Recover to recover concept prerequisite relations from course dependencies. More recently, Li et al. (2019a) applied variational graph autoencoders to learn concept prerequisite relations from courses. While Roy et al. (2019) developed a supervised learning approach called PREREQ.

However, there are still several challenges to learn the prerequisite relations among concepts. Firstly, there are multiple and complex relations among concepts and learning resources, but they were not fully utilized before. Secondly, labeling training data is enormously expensive and time consuming, especially when domain expertise is required for concept prerequisite relation judgement.

In order to address these challenges, we propose a novel concept prerequisite relation learning approach, named CPRL, which firstly learns concept representation via a relational graph convolutional network (R-GCN) (Schlichtkrull et al., 2018) on a heterogeneous graph, and predicts the concept prerequisite relations with a Siamese network. Then, it is optimized with the learning object dependencies and handcrafted features.

Moreover, we extend CPRL under the weakly-supervised settings to make our approach more practical, including learning prerequisite relation from learning object dependencies and generating training data with data programming paradigm.

Our contributions can be summarized as follows:

- We propose a heterogeneous concept-learning object graph (HCLoG), which can model the

* corresponding author

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2036–2047
June 6–11, 2021.

multiple and complex relations among concepts and learning resources to learn concept representation.

- We propose a novel concept prerequisite relation learning approach, named CPRL, which combines evidences from concept representations via R-GCN on HCLoG, learning object dependencies, and concept pairwise features.
- We extend CPRL under weakly supervised settings to avoid costly training data labeling.
- We conduct extensive experiments on four real-world datasets with different domains: *Textbook*, *MOOC*, *LectureBank* and *University Course*, and our approach achieves new state-of-the-art performance.

2 Problem Formulation

The educational data can be a textbook or a course, which can be modeled as a sequential learning objects (denoted as LO for short), such as book chapters, MOOC videos and lectures. There are concepts in an educational data, and we would like to extract the prerequisite relation among these concepts, as shown in Figure 1.

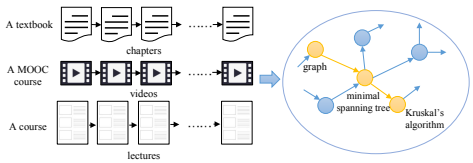

Figure 1: An example of prerequisite relation learning for concepts in educational data.

For convenience, we will use the following notations:

- $D = \{o_1, o_2, ..., o_M\}$ is an educational data, where o_i denotes the i-th learning object in D and is represented as a document. The document can be the text from a book chapter, or the speech script from a MOOC video.
- $C = \{c_1, c_2, ..., c_N\}$ is a set of concepts in D.

Therefore, the problem could be formally defined as: given an educational data D and its corresponding concepts C, the goal is to learn a function $F_\theta : C \times C \rightarrow \{0, 1\}$, which can predict whether c_i is a prerequisite concept of c_j by mapping the concept pair $\langle c_i, c_j \rangle$ to a binary class.

3 The CPRL Framework

The overview of our proposed CPRL is shown in Figure 2.

We firstly build a heterogeneous concept-learning object graph from the educational data, and then use a relational graph convolutional network (R-GCN) (Schlichtkrull et al., 2018) to represent the concepts and learning objects. Then, pairwise features for concepts are extracted according to their textual and structural information. Finally, all features are combined to learn the concept prerequisite relations.

It should be noted that the dependencies among learning objects can be viewed as a signal of weak supervision, which are also used to train the model.

3.1 Heterogeneous Concept-Learning Object Graph

We build a heterogeneous concept-learning object graph from an educational data, which contains concepts and learning objects, so the concept co-occurrence and the learning object-concept relations can be explicitly modeled.

The *heterogeneous concept-learning object graph* is defined as a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ consists of two types of nodes: concept nodes $V_c = \{c_1, c_2, ..., c_N\}$ and learning object nodes $V_o = \{o_1, o_2, ..., o_M\}$, and $\mathcal{E}$ represents the relations among them.

Specifically, we define the following three types of edges in G.

1. an edge between a concept and a learning object, and the weight is the term frequency-inverse document frequency (tfidf) of the concept in the document, where the term frequency is the number of times the concept appears in the document, while the inverse document frequency is the logarithmically scaled inverse fraction of the number of documents that contain the concept. E.g., e_{co} in Figure 2.

2. an edge between two concepts which co-occur in a fixed size sliding window in documents. Point-wise mutual information (PMI) is used to calculate the weight. Formally, $pmi(i, j) = \log \frac{p(i,j)}{p(i) \cdot p(j)}$, $p(i, j) = \frac{\#W(i,j)}{\#W}$ and $p(i) = \frac{\#W(i)}{\#W}$, where $\#W(i, j)$ is the number of sliding windows that contain both c_i and c_j, $\#W(i)$ is the number of sliding windows that only contain c_i, and $\#W$ is the

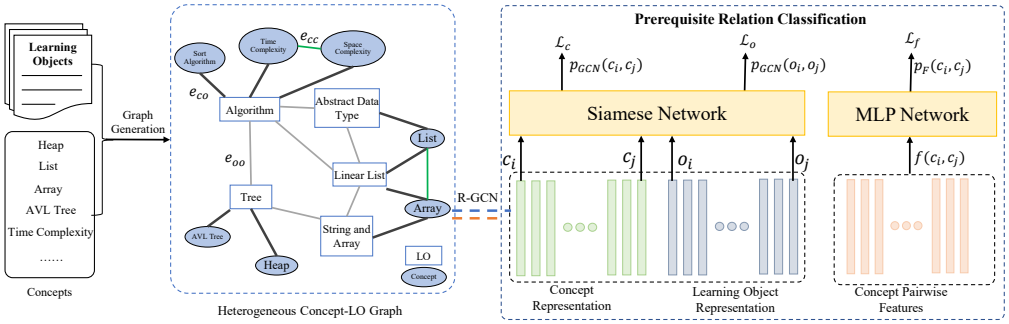

Figure 2: The overview of our proposed CPRL framework.

number of sliding windows in D. E.g., e_{cc} in Figure 2.

3. an edge between two learning objects, and the weight is the normalized distance between these two learning objects in the educational data. Formally, $dis(i,j) = \frac{|j-i|}{M}$. E.g., e_{oo} in Figure 2.

Thus, the adjacency matrix $A \in \mathbb{R}^{(M+N)\times(M+N)}$ of the graph G is defined as:

$$A_{ij} = \begin{cases} pmi(i,j) & i \text{ and } j \text{ are concepts} \\ tfidf(i,j) & i \text{ is a concept and } j \text{ is a LO} \\ dis(i,j) & i \text{ and } j \text{ are LOs} \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

3.2 Concept Representation via R-GCN

Since there are different types of relations among the nodes in the heterogeneous concept-learning object graph, we employ R-GCN to learn the representations of concepts and LOs.

We first use pretrained word embeddings GLoVE (Pennington et al., 2014) to represent each concept node in G. To represent the learning object, we calculate the average word embeddings of concepts in that learning object. Then, we update the node representation with R-GCN by aggregating messages from its direct neighbors as follows:

$$h_i^{l+1} = \sigma(W_0^l h_i^l + \sum_{r\in R}\sum_{j\in N_i^r} \frac{1}{c_{i,r}} W_r^l A_{ij} h_j^l)$$

where N_i^r is the neighbors of node i of relation $r \in R$, $W_r^l \in \mathbb{R}^{d\times d}$ is a relation-specific weight matrix, $W_0^l \in \mathbb{R}^{d\times d}$ is a general weight matrix, h_i^l is the hidden state of node i at l-th layer, σ is the ReLU function, and $c_{i,r} = \sum_{j\in N_i^r} A_{ij}$ is a normalization constant.

We stack the networks for L layers, and the concepts and learning objects can be represented by the hidden state of nodes in the L-th layer.

3.3 Prerequisite Relation Classification

After representing concepts via R-GCN, a Siamese network is used to predict whether the concept c_i is prerequisite of c_j.

We firstly take the concept representation of c_i and c_j as the input of a Siamese network, as shown in Figure 3, to calculate the likelihood of c_i being a prerequisite concept of c_j. Formally, $\vec{c}_i = ReLU(W_s \cdot h_{c_i}^L + b_s)$, where $h_{c_i}^L$ is the output of the R-GCN for concept c_i in L-th layer. Then, the likelihood $p_{GCN}(c_i, c_j)$ is calculated as $\sigma(W^T[\vec{c}_i; \vec{c}_j; \vec{c}_i - \vec{c}_j; \vec{c}_i \otimes \vec{c}_j] + b)$, where σ is the sigmoid function, $\otimes$ and $-$ are the element-wise multiplication and subtraction operators, and $[\cdot;\cdot]$ means the concatenation of vectors.

Finally, we use the cross-entropy as the loss function: $\mathcal{L}_c = \frac{1}{|T|}\sum_{(c_i,c_j,y_{ij})\in T} -[y_{ij} \cdot \log(p_{GCN}(c_i,c_j)) + (1-y_{ij}) \cdot \log(1 - p_{GCN}(c_i,c_j))]$, where T is the training dataset, and $y_{ij} \in \{0,1\}$ is the ground truth of (c_i, c_j).

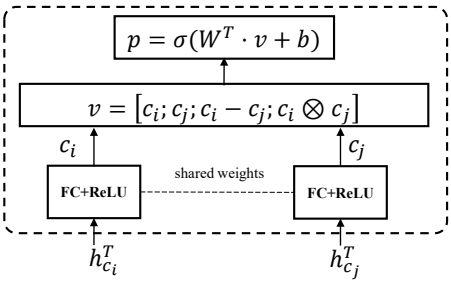

Figure 3: The Siamese network

3.4 Optimized with LO Dependencies

Intuitively, the dependencies among learning objects can reflect the prerequisite relations among concepts, but how can we utilize the learning object dependencies to enhance our model?

In the heterogeneous concept learning object graph, concepts and learning objects are both rep-

resented in the same space, so they can be fed to the same Siamese network.

Formally, we feed the representations of learning object o_i and o_j to the same Siamese network mentioned in previous section, and obtain the likelihood of the learning object dependency as $p_{GCN}(o_i, o_j)$. Similarly, we define the loss function as: $\mathcal{L}_o = \frac{1}{|T|}\sum_{(o_i,o_j,y_{ij})\in T} -[y_{ij}\cdot\log(p_{GCN}(o_i,o_j)) + (1-y_{ij})\cdot\log(1-p_{GCN}(o_i,o_j))]$, where T is the training dataset, and $y_{ij}\in\{0,1\}$ is the ground truth of (o_i, o_j).

Predicting the dependencies among learning objects can be considered as an auxiliary task for concept prerequisite relation learning, so the loss function could be: $\mathcal{L} = \mathcal{L}_c + \mu\mathcal{L}_o$.

3.5 Fusing Handcrafted Pairwise Features

In order to fully utilize the information of LOs, we also extract concept pairwise features from their textual and structural information.

Liang et al. (2015a) pointed out that when learning concept *A*, if one needs to refer to concept *B* a lot but not vice versa, then *B* is more likely to be a prerequisite of *A* than *A* of *B*. Inspired by this idea, we propose a new generic metric, namely *learning object reference distance* (LOrd), in a learning object sequence $D = \{o_1, o_2, ..., o_M\}$ to measure prerequisite relations among concepts.

For a concept pair (c_i, c_j), we propose the reference weight (*rw*) to qualify how c_j is referred by LOs which mention concept c_i, defined as:

$$rw(c_i, c_j) = \frac{\sum_{o\in D} f(c_i, o)\cdot r(o, c_j)}{\sum_{o\in D} f(c_i, o)}$$

where $f(c_i, o)$ indicates the frequency of concept c_i appears in the learning object o, and $r(o, c_j)\in\{0,1\}$ denotes whether concept c_j appears in o. Then, the LOrd is defined as: $LOrd(c_i, c_j) = rw(c_j, c_i) - rw(c_i, c_j)$. Obviously, *LOrd* can be easily calculated for textbooks, MOOC courses and university courses.

In addition, for MOOCs, we use features as in (Pan et al., 2017). While for textbooks, we extract several pairwise features as in (Pan et al., 2017), including *Semantic Relatedness*, *Wikipedia reference distance* and *complexity level distance*. The details can be referred in the Appendix.

Moreover, we also extract *head matching feature* and *ToC distance* (Wang et al., 2016) for concept pairs for textbooks. *Head matching feature* represents whether two concepts have a common head or not, which is obtained by suffix matching. Usually, it implies the existence of prerequisite relation, e.g., *tree* and *binary tree*. *ToC distance* measures the distance of concepts in the table of contents in D.

All the pairwise features are concatenated and fed into a forward neural network, which will generate the prediction result $p_F(c_i, c_j)$ for the concept pair (c_i, c_j). The loss function for the pairwise features is: $\mathcal{L}_f = \frac{1}{|T|}\sum_{(c_i,c_j,y_{ij})\in T} -[y_{ij}\cdot\log(p_F(c_i,c_j)) + (1-y_{ij})\cdot\log(1-p_F(c_i,c_j))]$.

Therefore, the overall loss function is: $\mathcal{L} = \mathcal{L}_c + \mu\mathcal{L}_o + \lambda\mathcal{L}_f$, where μ and λ are two hyperparameters.

4 The CPRL with Weak Supervision

In practice, it is expensive to collect massive hand-labeled data for model training. One intuitive way to alleviate the labeling cost is that we can train the model in one domain (e.g. *Calculus*), and then use it to predict the concept prerequisite relations in other domains (e.g. *Data Structure* and *Physics*). However, the idea fails and we will explain it in our experiments.

Therefore, we extend our model under the weak supervision settings in two ways.

We call the first way as *learning prerequisite relations from LO dependencies*. Since concepts and LOs are embedded into the same space through R-GCN in the heterogeneous graph, our model can implicitly infer the prerequisite relationships between concepts by explicitly learning the dependencies between LOs. This procedure is called CPRL$_{lo}$.

Another way is use the *data programming* (Ratner et al., 2016) paradigm to create probabilistic training data. *Data programming* expresses weak supervision strategies or domain heuristics as labeling functions (LFs), and then estimates the label accuracies by fitting a generative model. The process is shown as Figure 4.

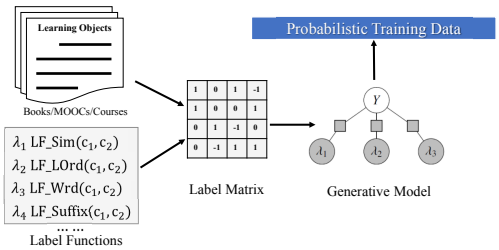

Figure 4: The pipeline of probabilistic label generation.

Here, we express some of the concept pairwise

features extracted before as heuristic labeling functions (LF for short): $\lambda : (c_i, c_j) \rightarrow \{-1, 0, 1\}$, where -1 means the labeling function abstains from providing a label. We define label functions corresponding to the features among concepts, and some examples are shown in Figure 5.

```
def LF_LOrd(c_i, c_j):
    if LOrd(c_i, c_j) < θ^min_LOrd    return 1
    elseif LOrd(c_i, c_j) > θ^max_LOrd    return 0
    else return -1

def LF_Suffix(c_i, c_j):
    if c_i matches c_j's suffix string  return 1  else return -1
```

Figure 5: Two LF examples, where θ^{max}_{LOrd} and θ^{min}_{LOrd} are learned thresholds. Other LFs and the settings of thresholds are listed in the Appendix.

We apply m such LFs to the unlabeled concept pairs $\{(c_{t_i}, c_{t_j})^n_{t=1}\}$ to generate a label matrix $\Lambda \in \{-1, 0, 1\}^{n \times m}$. Then, we use the weak supervision framework Snorkel (Ratner et al., 2019a) to train a probabilistic model. The probabilistic model takes the label matrix Λ as input, and generates the probabilistic training labels $\tilde{Y} = p(Y|\Lambda)$ for each concept pair. The generated labels could be used to train our model.

With the probabilistic training data, $\mathcal{L}_c$ and $\mathcal{L}_f$ are changed to the *noise-aware* variants: $\mathcal{L}_c = \sum_{(c_i,c_j)\in T} \mathbb{E}_{y_{ij}\sim\tilde{Y}}[-[y_{ij} \cdot \log(p_{GCN}(c_i, c_j)) + (1 - y_{ij}) \cdot \log(1 - p_{GCN}(c_i, c_j))]]$ and $\mathcal{L}_f = \sum_{(c_i,c_j)\in T} \mathbb{E}_{y_{ij}\sim\tilde{Y}}[-[y_{ij} \cdot \log(p_F(c_i, c_j)) + (1 - y_{ij}) \cdot \log(1 - p_F(c_i, c_j))]]$. This procedure is called CPRL$_{dp}$.

5 Experiments

5.1 Datasets

In order to validate the efficiency of our model, we conducted experiments on four datasets with different domains.

- *Textbook*: we selected six Chinese textbooks in each of the three domains: *Calculus*, *Data Structure*, and *Physics*, and then extracted 89, 84 and 139 concepts, and labeled 449, 439 and 623 prerequisite relations for each domain respectively. The datasets will be publicly available later.

- *MOOC*: we used MOOC data[1] mentioned in (Pan et al., 2017), which involves two domains: *Data Structure and Algorithms (DSA)* and *Machine Learning (ML)*.

- *LectureBank*: This dataset[2] (Li et al., 2019a) contains 1,352 English lecture files collected from university courses, and the annotations of prerequisite relations on 208 concepts.

- *University Course*: This dataset[3] (Liang et al., 2017) has 654 courses with 861 course prerequisite edges from various universities in USA, and 1008 pairs of concepts with prerequisite relations are manually annotated.

The set of concepts and prerequisite relations among them was annotated by experts, and released with the datasets. The statistics of the datasets are listed in the appendix.

5.2 Baselines

We used the following state-of-the-art approaches as baselines.

Binary classifiers: We compared our model with the binary classifiers as in (Pan et al., 2017), including Naïve Bayes classifier (NB), Support vector machine (SVM), Logistic Regression (LR) and Random Forest classifier (RF).

RefD: RefD (Liang et al., 2015b) is a simple link-based metric for measuring the prerequisite relations among concepts.

GAE: GAE denotes graph autoencoder, which encodes a graph with GCN, and predicts links through the adjacency matrix reconstruction. Li et al. (2019a) used GAE for concept prerequisite relation learning.

VGAE: VGAE is an extension to GAE, which was also used in (Li et al., 2019a) for concept prerequisite relation learning.

PREREQ: PREREQ (Roy et al., 2019) obtains latent representations of concepts through the pairwise-link LDA model, and identifies concept prerequisite relations through a Siamese network.

We also compared our weakly-supervised variants with **CPR-Recover** (Liang et al., 2017), which is an unsupervised approach, and can recover concept prerequisite relations from course dependencies.

Consistent with many methods, we mainly used F-score(F_1) to evaluate the performance of CPRL with all the baselines. We also compared precision(P) and recall(R) against other methods.

[1] http://keg.cs.tsinghua.edu.cn/jietang/software/acl17-prerequisite-relation.rar

[2] https://github.com/Yale-LILY/LectureBank

[3] https://github.com/suderoy/PREREQ-IAAI-19/tree/master/datasets/University%20Course%20Dataset

5.3 Implementation Details

In all datasets, only concept prerequisite pairs are manually annotated, and we split the positive samples into train and test sets. In order to fairly compare with the previous researches, 90% samples of *LectureBank* were used for training while the rest 10% for testing. For other datasets, the proportions changed to 70% and 30%. Then, we generated negative samples by sampling random unrelated pairs of concepts from the vocabulary in addition to the reverse pair of original positive samples. In order to address the imbalance problem, we oversampled 3.5 and 1.5 times the number of the positive examples in the training and testing sets for *Textbook* dataset and other datasets respectively. The results are averaged over 5 train-test splits.

The parameters were initialized randomly from a Gaussian distribution with zero mean and standard deviation $\sigma = 0.3$. The initial learning rate γ is 0.5 for *Textbook* and 0.1 for other datasets. Besides, the learning rate annealed every 50 epochs by 0.99γ. We trained CPRL using the Stochastic Gradient Descent method and stopped training if the train loss did not decrease for 30 consecutive epochs. For baseline models, we used default parameter settings as in their original implementations, and also used 300-dimensional GloVE as the pre-trained word embeddings.

For R-GCN, we set the number of R-GCN layers $L = 2$ and set the embedding size of the first convolution layer as 256 and the second convolution layer as the number of concepts in each dataset. We experimented with other settings and found that small changes did not influence the result much. In addition, we set $\lambda = 0.2$ and $\mu = 0.1$, since they made the best performance. The influence of parameters L, λ and μ can be referred to the Appendix.

5.4 Performance Comparison

Table 1 shows the precision, recall and F-score on four datasets with different domains.

From the table, we find that (1) CPRL achieves the best performance with F-score against all baselines on all datasets, except for DSA domain of the MOOC dataset. (2) CPRL performs best in LectureBank and University even without pairwise features and dependencies among learning objects. It tells that HCLoG can effectively model the multiple and complex relations among concepts and learning resources to learn better concept representation. (3) RefD can indeed measure the prerequisite relations among concepts, and obtains a higher precision, but a lower recall. (4) GAE and VGAE utilize GCN for adjacency matrix reconstruction, but they perform worse than CPRL. The reason is that CPRL utilizes the heterogeneous concept learning object graph to learn the concept representation, which can fully utilize the complex relationships among concepts and learning objects, while GAE and VGAE only use the graph among concepts.

5.5 Ablation Study

In order to prove the effects of pairwise features and LO dependencies, we conducted ablation experiments on *Textbook* and *MOOC* datasets. The results are shown in table 2

Dataset		Metric	CPRL	CPRL$_f$	CPRL$_c$
Textbook	DS	P	0.795	0.793	**0.811**
		R	**0.809**	0.802	0.749
		F$_1$	**0.802**	0.797	0.779
	PHY	P	0.778	**0.779**	0.778
		R	0.798	**0.799**	0.716
		F$_1$	0.788	**0.789**	0.746
	CAL	P	0.770	**0.772**	0.769
		R	**0.825**	0.809	0.755
		F$_1$	**0.797**	0.790	0.762
MOOC	DSA	P	0.640	**0.659**	0.562
		R	**0.619**	0.615	0.565
		F$_1$	0.630	**0.636**	0.563
	ML	P	**0.800**	0.788	0.767
		R	**0.642**	0.628	0.598
		F$_1$	**0.712**	0.699	0.672

Table 2: Ablation Study on CPRL. Row-wise best results are in bold. CPRL$_f$ and CPRL$_c$ are the models which minimize $\mathcal{L}_c + \lambda\mathcal{L}_f$ and $\mathcal{L}_c$ respectively.

As shown in Table 2, CPRL performs better than CPRL$_f$ and CPRL$_c$ on most of the datasets, so pairwise features and learning object dependencies can both contribute to the performance. Besides, even CPRL$_c$ obtains a better performance than the baselines in Table 1, which proves the effectiveness of the heterogeneous graph.

5.6 Effectiveness of Weak Supervision

In order to evaluate our weakly supervised prerequisite relation learning approaches, we compared our two variants CPRL$_{lo}$ and CPRL$_{dp}$ with CPR-Recover (Liang et al., 2017) in *Textbook* dataset, and the results are shown in Table 3.

From the table, we find that CPRL$_{lo}$ and CPRL$_{dp}$ outperform CPR-Recover in all metrics, and CPRL$_{dp}$ achieves the best performance. It proves that the knowledge of learning object dependencies can be transferred to learn the concept prerequisite relations through the concept learning object graph. In addition, the data programming with our designed label functions can generate help-

Dataset		Metric	SVM	LR	RF	NB	RefD	GAE	VGAE	PREREQ	CPRL
Textbook	DS	P	0.818	0.852	0.755	0.481	**0.920**	0.446	0.434	0.226	0.795
		R	0.632	0.590	0.685	0.897	0.244	**0.900**	0.570	0.369	0.809
		F_1	0.713	0.697	0.718	0.626	0.385	0.597	0.493	0.280	**0.802**
	PHY	P	0.806	0.863	0.748	0.399	**0.900**	0.505	0.460	0.432	0.770
		R	0.655	0.588	0.752	0.922	0.409	**0.943**	0.649	0.423	0.825
		F_1	0.723	0.699	0.750	0.557	0.562	0.657	0.538	0.427	**0.797**
	CAL	P	0.839	0.860	0.746	0.404	**0.950**	0.436	0.414	0.391	0.778
		R	0.637	0.570	0.715	**0.995**	0.302	0.900	0.558	0.506	0.798
		F_1	0.724	0.686	0.730	0.574	0.458	0.587	0.475	0.441	**0.788**
MOOC	DSA	P	0.705	0.808	0.344	0.613	**0.920**	0.294	0.269	0.492	0.641
		R	0.624	0.168	**0.715**	0.696	0.252	**0.715**	0.657	0.462	0.619
		F_1	**0.662**	0.278	0.464	0.652	0.396	0.417	0.382	0.476	0.630
	ML	P	0.668	0.748	0.375	0.577	0.784	0.293	0.266	0.448	**0.800**
		R	0.577	0.27	0.669	0.623	0.188	**0.733**	0.647	0.592	0.642
		F_1	0.619	0.397	0.481	0.599	0.303	0.419	0.377	0.510	**0.712**
LectureBank		P	0.857	0.744	0.855	0.670	0.666	0.462	0.417	0.590	**0.861**
		R	0.692	0.744	0.681	0.640	0.228	0.811	0.575	0.502	**0.858**
		F_1	0.766	0.744	0.758	0.655	0.339	0.589	0.484	0.543	**0.860**
University Course		P	0.796	0.595	0.739	0.478	**0.919**	0.450	0.470	0.468	0.689
		R	0.635	0.546	0.480	0.649	0.415	0.886	0.694	**0.916**	0.760
		F_1	0.707	0.569	0.582	0.550	0.572	0.597	0.560	0.597	**0.723**

Table 1: The performance of CPRL on four datasets with different domains. Row-wise best results are in bold. CPRL is the model which minimizes $\mathcal{L}_c + \mu\mathcal{L}_o + \lambda\mathcal{L}_f$, while the models for *LectureBank* and *University Course* only minimize $\mathcal{L}_c$ since the LOs have no prerequisite relations in them and we cannot extract structural features.

Dataset		Approach	P	R	F_1
Textbook	DS	CPR-Recover	0.317	0.577	0.409
		$CPRL_{lo}$	0.425	0.504	0.461
		$CPRL_{dp}$	**0.570**	**0.926**	**0.706**
	PHY	CPR-Recover	0.291	0.609	0.394
		$CPRL_{lo}$	0.427	0.487	0.455
		$CPRL_{dp}$	**0.503**	**0.856**	**0.634**
	CAL	CPR-Recover	0.447	0.624	0.521
		$CPRL_{lo}$	0.470	0.659	0.551
		$CPRL_{dp}$	**0.517**	**0.803**	**0.629**

Table 3: Comparison our weakly supervised prerequisite relation learning variants with CPR-Recover.

ful training data, and achieve comparable performance with the supervised CPRL.

5.7 Verification of Domain Transfer Ability

In order to explore the transfer ability of our model between different domains, we conducted an experiment on *Textbook* dataset.

Specifically, for CPRL, we firstly trained the model in one domain, and then used the model to predict prerequisite relations between concepts in another domain. While for $CPRL_{dp}$, we obtained the best thresholds such as θ^{max}_{LOrd} and θ^{min}_{LOrd} in LFs in one domain and then used them to other domains. The results are shown in Table 4

	CPRL			$CPRL_{dp}$		
	DS	PHY	CAL	DS	PHY	CAL
DS	0.802	0.393	0.219	0.706	0.621	0.587
PHY	0.640	0.797	0.430	0.692	0.634	0.616
CAL	0.520	0.438	0.788	0.658	0.633	0.629

Table 4: Domain transfer ability verification experiments for CPRL and $CPRL_{dp}$, where each row and column represent the source and target domain respectively, and the values in the cells are F_1-scores.

We observe that (1) F-scores drop severely in CPRL, so we cannot simply transfer the model across domains due to the difference among concepts and LOs. (2) $CPRL_{dp}$ is more stable and can be used in practice since we only need to label a small amount of training data in one domain.

5.8 Effectiveness of Ensemble

Our approach can learn the concept prerequisite relations from one learning object sequence, such as a textbook. While the concepts in textbooks in the same domain are basically the same, so the prerequisite relations among them can be aggregated.

Here, we used a simple majority voting strategy for aggregation, and the results are shown in Figure 6. From the table, we see a significant improvement for the ensemble results.

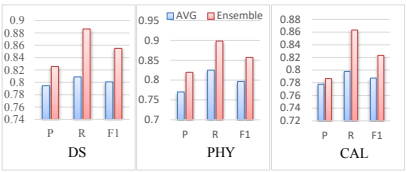

Figure 6: The ensemble results which are aggregated from six textbooks in each domain.

6 Related Work

6.1 Prerequisite Relation Learning

Learning prerequisite relations between concepts has attracted much recent work, and can be classified into three categories: local statistical information based approaches, recovery based approaches and learning based approaches.

As local statistical information, *reference distance* (Liang et al., 2015a) and *cross-entropy* (Gordon et al., 2016) were proposed to measure the concept prerequisite relations. CPR-Recover (Liang et al., 2017) is a recovery based approach, which recovers prerequisite relations from course dependencies. The learning based approaches are the most popular. For example, Pan et al. (2017) proposed contextual, structural and semantic features for concept prerequisite relation classification. Roy et al. (2019) applied the pairwise-link LDA model to represent concept, and trained a Siamese network to identify prerequisite relations. Li et al. (2019a) trained variational graph autoencoders to predict concept prerequisite relations. However, these approaches didn't model the mutiple and complex relations among concepts and learning resources. Meanwhile, they also need a large set of training data, which is costly to obtain.In order to reduce the amount of training data required, active learning was investigated in (Liang et al., 2018) and (Liang et al., 2019) for concept prerequisite learning.

6.2 Weakly Supervised Learning

One of the most significant bottlenecks for machine learning is the need for a big training data set. Nowadays, it is very promising to use weakly supervised learning techniques to reduce the amount of human intervention needed. For example, *distant supervision* can produce noisy training data by aligning unlabeled data with an external knowledge base, e.g. relation extraction in (Smirnova and Cudré-Mauroux, 2018). *Crowdsourcing* (Yuen et al., 2011) and *heuristic rules* (Sa et al., 2016) can also generate noisy training data.

However, these weakly supervised data is incomplete, inexact and inaccurate, so it is important to integrate multiple noisy labeling data to produce more accuracy data. *Data programming* (Ratner et al., 2016) provides a simple and unifying framework for the creation of training sets, which expresses weak supervision strategies as labeling functions, and then uses a generative model to denoise the labeling data. Snorkel[4] (Ratner et al., 2019a) is a system built around the data programming paradigm for rapidly creating, modeling, and managing training data. Several works have been explored to use *data programming* for training data creation. For example, SwellShark (Fries et al., 2017) was proposed for quickly building biomedical named entity recognition taggers using lexicons, heuristics, and other forms of weak supervision instead of hand-labeled data. GWASkb with thousands of genotype-phenotype associations was created by using Snorkel in (Kuleshov et al., 2019). Snorkel was also used for chemical reaction relationship extraction (Mallory et al., 2020), discourse structure learning (Badene et al., 2019) and medical entity classification (Fries et al., 2020).

In addition, *data programming* was further improved under different situations. For example, MeTaL (Ratner et al., 2019b) was proposed for modeling and integrating weak supervision sources with different unknown accuracies, correlations, and granularities. Cross-modal data programming was proposed in (Dunnmon et al., 2020). FlyingSquid (Fu et al., 2020) speeded up weak supervision with triplet methods.

7 Conclusion

In this paper, we propose a novel concept prerequisite relation learning approach, named CPRL, which combines both concept representation learned from a heterogeneous graph and concept pairwise features. Furthermore, we extend CPRL under weakly supervised settings to make our method more practical. The experiments on four datasets show that our method achieves state-of-the-art performance. In addition, we also prove the effectiveness of our weakly supervised prerequisite relation learning variants.

In future, we plan to design more effective label functions or employ more reliable weakly supervised learning approaches (Li et al., 2019b; Guo et al., 2019) to further improve the performance. Moreover, we will also introduce concept prerequisite relations into curriculum planning and intelligent tutoring applications, e.g. organizing learning resources into a reasonable order and incorporating prerequisite relations into knowledge tracing technologies.

Acknowledgments

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900), the Zhejiang Provincial Natural Science Foundation of China (No. LY17F020015), the Chinese Knowledge Center of Engineering Science and Technology (CKCEST) and MOE Engineering Research Center of Digital Library.

[4]https://www.snorkel.org/

References

Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019. Data programming for learning discourse structure. In *Association for Computational LInguistics (ACL)*.

P. Chen, Yu Lu, V. Zheng, and Yang Pian. 2018. Prerequisite-driven deep knowledge tracing. *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48.

Jared A Dunnmon, Alexander J Ratner, Khaled Saab, Nishith Khandwala, Matthew Markert, Hersh Sagreiya, Roger Goldman, Christopher Lee-Messer, Matthew P Lungren, Daniel L Rubin, et al. 2020. Cross-modal data programming enables rapid medical machine learning. *Patterns*, page 100019.

Jason A Fries, Ethan Steinberg, Saelig Khattar, Scott L Fleming, Jose Posada, Alison Callahan, and Nigam H Shah. 2020. Trove: Ontology-driven weak supervision for medical entity classification. *arXiv preprint arXiv:2008.01972*.

Jason Alan Fries, Sen Wu, A. Ratner, and Christopher Ré. 2017. Swellshark: A generative model for biomedical named entity recognition without labeled data. *ArXiv*, abs/1704.06360.

Daniel Y. Fu, M. F. Chen, F. Sala, Sarah Hooper, K. Fatahalian, and Christopher Ré. 2020. Fast and threerious: Speeding up weak supervision with triplet methods. *ArXiv*, abs/2002.11955.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 866–875.

Lan-Zhe Guo, Yu-Feng Li, Ming Li, Jinfeng Yi, Bowen Zhou, and Z. Zhou. 2019. Reliable weakly supervised learning: Maximize gain and maintain safeness. *ArXiv*, abs/1904.09743.

V. Kuleshov, J. Ding, C. Vo, B. Hancock, A. Ratner, Yang I. Li, C. Ré, S. Batzoglou, and Michael T. Snyder. 2019. A machine-compiled database of genome-wide association studies. *Nature Communications*, 10.

Irene Li, Alexander R. Fabbri, Robert R. Tung, and Dragomir R. Radev. 2019a. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning. In *AAAI*.

Y. Li, Lan-Zhe Guo, and Z. Zhou. 2019b. Towards safe weakly supervised learning. *IEEE transactions on pattern analysis and machine intelligence*.

Chen Liang, Z. Wu, W. Huang, and C. Lee Giles. 2015a. Measuring prerequisite relations among concepts. In *EMNLP*.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. 2015b. Measuring prerequisite relations among concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Chen Liang, Jianbo Ye, Shuting Wang, B. Pursel, and C. Lee Giles. 2018. Investigating active learning for concept prerequisite learning. In *AAAI*.

Chen Liang, Jianbo Ye, Z. Wu, B. Pursel, and C. Lee Giles. 2017. Recovering concept prerequisite relations from university course dependencies. In *AAAI*.

Chen Liang, Jianbo Ye, H. Zhao, B. Pursel, and C. Lee Giles. 2019. Active learning of strict partial orders: A case study on concept prerequisite relations. *ArXiv*, abs/1801.06481.

Hanxiao Liu, Wanli Ma, Yiming Yang, and J. Carbonell. 2016. Learning concept graphs from online educational data. *J. Artif. Intell. Res.*, 55:1059–1090.

Emily K Mallory, Matthieu de Rochemonteix, Alex Ratner, Ambika Acharya, Chris Re, Roselie A Bright, and Russ B Altman. 2020. Extracting chemical reactions from text using snorkel. *BMC bioinformatics*, 21(1):1–15.

Eric Margolis and Stephen Laurence. 1999. *Concepts: core readings*. Mit Press.

Liangming Pan, C. Li, Juan-Zi Li, and J. Tang. 2017. Prerequisite relation learning for concepts in moocs. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

A. Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and C. Ré. 2019a. Snorkel: rapid training data creation with weak supervision. *The Vldb Journal*, 29:709 – 730.

A. Ratner, B. Hancock, Jared Dunnmon, F. Sala, Shreyash Pandey, and C. Ré. 2019b. Training complex models with multi-task weak supervision. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 33:4763–4771.

Alexander J. Ratner, C. D. Sa, Sen Wu, Daniel Selsam, and C. Ré. 2016. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29:3567–3575.

S. Roy, Meghana Madhyastha, Sheril Lawrence, and V. Rajan. 2019. Inferring concept prerequisite relations from online educational resources. In *AAAI*.

C. D. Sa, A. Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. 2016. Deepdive: Declarative knowledge base construction. *SIGMOD record*, 45 1:60–67.

M. Schlichtkrull, Thomas Kipf, P. Bloem, R. V. Berg, Ivan Titov, and M. Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*.

Alisa Smirnova and Philippe Cudré-Mauroux. 2018. Relation extraction using distant supervision. *ACM Computing Surveys (CSUR)*, 51:1 – 35.

Shuting Wang and Lei Liu. 2016. Prerequisite concept maps extraction for automaticassessment. In *WWW*.

Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 2016. Using prerequisites to extract concept maps from textbooks. In *Proceedings of the 25th acm international on conference on information and knowledge management*, pages 317–326. ACM.

M. Yuen, Irwin King, and Kwong-Sak Leung. 2011. A survey of crowdsourcing systems. *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*, pages 766–773.

A Appendix

A.1 Datasets Statistics

We conducted experiments on four datasets with different domains and the statistics are detailed in Table 5.

A.2 Concept Pairwise Features

For MOOC dataset, we used the features as in (Pan et al., 2017). While for *Textbook* dataset, we also extract several pairwise features as in (Pan et al., 2017), including *Learning object reference distance*, *Semantic Relatedness*, *Wikipedia reference distance (Wrd)* and *complexity level distance (Cld)*.

Semantic Relatedness measures the relatedness of two concepts in the semantic space, which can be calculated by: $sr(c_i, c_j) = \frac{1}{2}(1 + \frac{v_{c_i} \cdot v_{c_j}}{||v_{c_i}|| \cdot ||v_{c_j}||})$, where v_{c_i} is the semantic vector of concept c_i.

Wikipedia reference distance is defined based on the structure of Wikipedia. Specifically, for a concept c_i, we can obtain a set of top M most related concepts according to the semantic relatedness between two concepts, denoted as R_{c_i}. Then, the Wikipedia reference weight can be defined as: $Wrw(c_i, c_j) = \frac{\sum_{e \in R_{c_j}} Erw(e,c_j) \cdot sr(e,c_i)}{\sum_{e \in R_{c_i}} sr(e,c_i)}$, where $Erw(e, c_i)$ is a binary indicator, in which $Erw(e, c_i) = 1$ if the Wikipedia article of e refers to any concept in R_{c_i}, and $Erw(e, c_i) = 0$ otherwise. Thus, the Wikipedia reference distance is defined as: $Wrd(c_i, c_j) = Wrw(c_j, c_i) - Wrw(c_i, c_j)$.

Complexity level distance is defined based on the observation that if a concept covers more chapters in a textbook or it survives longer time in the textbook, it is more likely to be a basic concept rather than an advanced one. Besides, the larger the difference between the distribution of two concepts, the more likely they are to have a prerequisite relationship. Therefore, we can measure complexity level distance from two aspects. On the one hand, $Cld_{Frequency}(c_i, c_j) = ava(c_i) - ava(c_j)$, where $ava(c_i) = \frac{|I(D,c_i)|}{|D|}$. On the other hand, $Cld_{Distribution}(c_i, c_j) = D(P(c_i)||P(c_j))$ where $P(c_i)$ means the distribution of concept i in D and $D(P(c_i)||P(c_j))$ represents the *KL Divergence* between the distribution of concept i and concept j.

Moreover, we also extract *head matching feature* and *ToC distance* for concept pairs in *Textbook* dataset. *Head matching feature* represents whether two concepts have a common head or not, which is obtained by suffix matching. Usually, it implies the existence of prerequisite relation, e.g., *tree* and *binary tree*. *ToC distance* measures the distance of concepts in the table of contents in D.

A.3 Labeling Functions

We use the concept pairwise features mentioned before as heuristics labeling functions, and the labeling functions used in *Textbook* dataset are listed in Figure 7.

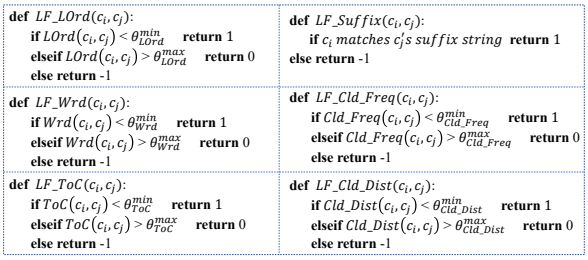

Figure 7: The labeling functions used in *Textbook* dataset.

The optimal thresholds of the labeling functions can be obtained by grid search with a small amount of training data. Some empirical values are given in Table 6 for the *Textbook* dataset.

	Wrd	$LOrd$	ToC	Cld_{Freq}	Cld_{Dist}
$\theta^{max}_{Feature}$	1.3	-0.2	0.2	0.4	0.3
$\theta^{min}_{Feature}$	-1.3	-0.8	-0.2	-0.8	-0.5

Table 6: The thresholds of label functions for the *Textbook* dataset.

A.4 Influence of Parameters

In order to determine the parameters λ and μ in the loss function, we conducted the experiments on Textbook dataset in Physics domain with different λs and μs, and Figure 8 shows the results. Therefore, we chose $\lambda = 0.2$ and $\mu = 0.1$ in our experiments, which made the best performance.

		μ				
		0	0.1	0.2	0.5	1
λ	0	0.778	0.783	0.781	0.792	0.783
	0.1	0.789	0.793	0.789	0.789	0.785
	0.2	0.796	0.797	0.786	0.779	0.782
	0.5	0.790	0.784	0.787	0.790	0.787
	1	0.783	0.786	0.786	0.785	0.784

Figure 8: The F-score of CPRL with different λs and μs on Textbook dataset in Physics domain.

In addition, we also evaluated our approach with different number of GCN layers (L), and the result

Dataset		# Learning Object	# Concepts	# Pairs(+)	# Pairs(-)	# Tokens per Learning Object
Textbook	DS	102	89	449	673	1861
	CAL	134	84	439	658	1608
	PHY	113	139	623	934	3745
MOOC	DSA	148	175	354	1239	5285
	ML	271	216	1446	5061	1893
LectureBank		923	208	913	1369	3240
University Course		654	407	1007	1510	60

Table 5: Statistics of the Datasets. In University Course, each course is described only using its brief introduction, so the average number of tokens in the learning objects is limited.

is shown in Figure 9. From the figure, we can see that the F-score increases gradually and then drops finally. Thus, we chose $L = 2$ in our experiments.

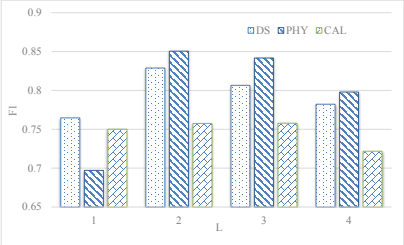

Figure 9: The F-score of CPRL with different Ls on Textbook dataset in different domains.

A.5 Impact of Training Set Size

To compare with the previous research, we used 90% positive samples of *LectureBank* and 70% positive samples of other datasets to train the model.

In order to further explore the ability of our model, we train our model with with different number of training data, and show the result in *Physics* domain in the *Textbook* dataset in Figure 10.

It is shown that, when we use more positive samples to train the model, it can reach a higher F1-Score. Besides, it could outperform the baselines with only about 30% positive samples, which implies our model's ability to fully utilize the training samples.

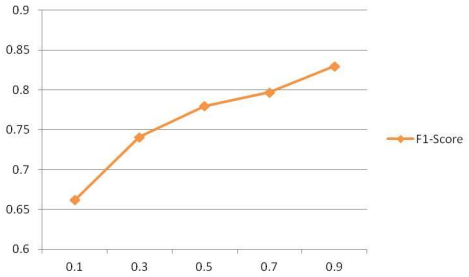

Figure 10: The impact of the size of training set.