Collaborative Structure Learning Framework from Wikipedia Based on Heterogeneous Graph Attention Network

KUI, XIAO

School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China

xiaokui@hubu.edu.cn

WEI, DAI

School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China

1037530806@qq.com

HONGYAN, Li

School of Information and Communication Engineering, Hubei University of Economics, Wuhan 430205, China

hongyanli2000@126.com

YAMIN, Li*

School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China

yamin.li@hubu.edu.cn

The development of internet technology has brought great opportunities to online education. The vast learning resources and complex relationships between these concepts on the internet have brought great difficulties to learners. The prerequisite relation between these concepts has great guiding significance for learners' learning order. Hence, automatic prerequisite relation learning has become a hot research topic. The rise of graph neural networks has driven the entire artificial intelligence field and also brought development to prerequisite relation learning. Therefore, this paper proposes a weak supervised learning method collaborative structured learning framework. This framework takes the advantage of a graph attention neural network and node2vec to learn a heterogeneous graph composed of concepts and learning objects and a concept graph based on Wikipedia. Then, it obtains the prerequisite relation between concepts through a classification network. Our experiments on the W-ML dataset of MOOC and the University Course dataset show that the proposed collaborative structured learning framework has better performance than the current baseline models.

CCS CONCEPTS • Collaborative learning • Artificial intelligence • Machine learning

Additional Keywords and Phrases: Heterogeneous graph attention, Graph embedding, Siamese network, Education Source

1 INTRODUCTION

In the present information era, the field of education is undergoing continuous innovation and improvement, driven by the rapid development of the domestic economy. Many excellent open educational resources have emerged both domestically and internationally, such as MOOCs [4], Tencent Classroom, New Oriental, and other online open educational platforms. Online education, as a prominent contemporary mode of education, has gained recognition from students and parents. Recent research indicates that the number of students engaged in online learning approached 200

* corresponding author

million in 2020, with around a thousand universities offering approximately twenty thousand online courses. Consequently, numerous researchers are leveraging these educational resources for research and innovation.

Educational learning resources encompass various forms, including learning materials extracted from PowerPoint presentations or online texts, educational resources derived from videos and subtitles, and educational content mined from books and online courses. MOOCCubeX [5], for instance, is a large-scale dataset mined from online platforms, offering not only insights into the source courses of relevant concepts but also delving deeper into the integration of academic behaviors and related courses. Another dataset, LectureBank [6], comprises a vast collection of instructional lecture videos and associated metadata. This dataset serves to support research in education, natural language processing (NLP), and machine learning.

As both domestic and international learning resources proliferate, students are encountering challenges in the realm of online education. Consider a simple example: suppose a student lacks the foundational knowledge from several prerequisite courses before delving into a specific subject. In such a scenario, the student might struggle with the course and eventually falter. To mitigate such issues, the existence of a sequential learning order among online educational resources is postulated, giving rise to the notion of prerequisite relationships. For instance, prior to studying a course on databases, one must grasp fundamental concepts like data, its origins, and its impact on our lives.

Addressing the issue of sequencing courses is undoubtedly pivotal within the realm of education. With a well-defined sequence of courses, the efficiency of online learning can be enhanced. Naturally, specialized terminology relevant to specific fields will appear within courses; for example, the domain of algorithms introduces terms like "linear algorithms." These specialized terms, collectively termed as "concepts," are unearthed from educational resources. The factors determining relationships among these online learning resources are these concepts. In other words, the precedence relationship between two educational resources is determined by these concepts. Consequently, to uncover the hidden prerequisite relationships between concepts, scholars adopt strategies such as defining a concept graph by assigning distinct feature vector values [7], utilizing PREREQ [8] to infer prerequisite relationships between concepts based on the prerequisite relationships of online educational resources, and employing various concept feature matrices [9] like PageRank [2], PMI [10], and RefD to predict prerequisite relationships between online learning resources. CPR-Recover [11] retrieves prerequisite relationships between concepts from the existing prerequisite relationships of educational resources.

This paper incorporates modern graph neural networks, specifically HAN [14] and SiameseNet, into the defined heterogeneous graph framework. Furthermore, an open data repository like Wikipedia is utilized to define feature graphs and carry out final feature fusion. The research concludes by showcasing promising results on two datasets.

2 RELATED WORK

Mining the priority relationships among learning resources has become a prominent topic in recent years. In the field of education, Wikipedia stands as a substantial open platform containing interconnected relationships between topic terms, concepts, and various terminologies. These interconnections aid in inferring prerequisite relationships between concepts, shedding light on whether one term is a prerequisite for another. To determine whether a concept serves as a prerequisite for another concept, we can define their feature vectors and approach learning from various angles, such as TF-IDF and PMI.

However, defining feature matrices using manual features can be cumbersome and may overlook contextual and semantic relationships. To address this, graph neural networks (GNNs) have emerged as a means for feature learning. With the continuous evolution of graph neural networks, including GNN and GCN[12] architectures, these networks

have found active application in traditional machine learning paradigms. For instance, [15] proposed utilizing Bert [16] as a pre-trained model input to RGCN [21] for learning. [17] noted the relatively favorable outcomes achieved in relevant domains through unsupervised learning with VGAE [18]. Furthermore, researchers [19] employed Gate-GNN to enhance contexts for fine-tuning Bert in downstream tasks. However, many trainable graphs fail to account for edge direction as a pertinent factor. The RefD [1] graph addresses this issue by amalgamating edge directions with positive and negative values.

In this paper, we introduce the CSL framework, leveraging node2vec [13] to learn the structure of the G_r graph and utilizing the HAN [14] network to learn the heterogeneous G_h graph. Through this framework, we capture hidden relationships between learning entities and concepts acquired from the heterogeneous graph, all while considering directed weighted graphs with values. The paper constructs an undirected anisotropic graph using E_{co}, E_{cc}, and E_{oo} to better aggregate features of neighboring nodes. Furthermore, HAN performs vector aggregation to ensure the optimal concatenation of vectors received by each node.

3 PROBLEM DEFINITION

In this section, we provide notational definitions and propose solutions to the problems addressed in this paper. The educational data we consider can be categorized as either videos or course texts, which can be seen as continuous learning objects (referred to as LOs). Examples of LOs include books, MOOC videos, lectures, and more. In this paper, we define learning objects as LOs and acknowledge the presence of potential precondition relations among them, indicating the suggested order in which they should be studied. While course videos can also be treated as learning resources, we focus on extracting subtitles or course PowerPoint presentations from these videos.

Educational resources are defined as $o = \{o_1, o_2, \ldots\ldots, o_n\}$ representing the spatial vectors of the entire learning resources. The set of concepts is denoted as $c = \{c_1, c_2, \ldots\ldots c_m\}$, encompassing the entire collection of concepts. $P = \{ < c_i, c_j > | \ i -> j \ \}$ represents a set of concept pairs indicating potential precondition relations between concepts. Additionally, we define the weighted edge between concepts as E_{cc}, reflecting the strength or weight of the relationship between them.

To effectively define the entire dataset, we consider both courses and video subtitles as learning resources. Then, we define an undirected heterogeneous graph G_h and a directed RefD graph G_r. These two graphs, G_h and G_r, capture different aspects of the data. The edges in the heterogeneous graph G_h are represented by three types of edge sets: E_{oo}, E_{co}, and E_{cc}. In the G_r graph, we introduce RefD directed edges to train the weakly supervised part of the dataset and obtain the corresponding vectors.

For classification prediction, we utilize the Siamese Network [22]. The Siamese Network architecture will be described in more detail later in the paper, along with the other components and methodologies employed in our proposed solution.

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

4 METHOD

4.1 The CSL Framework

Currently, many of the eigenvalues currently computed between learning objects and concepts are based on undirected graphs, such as TF-IDF and PMI, which can ignore the problem of orientation of the entire graph. We propose a way to

use the RefD (Reference Dependence) graph structure combined with node2vec for learning, so that we make full use of the chain-in and chain-out relationships between concepts in Wikipedia to form a directed graph $G_r(C_r, E_r)$, and utilize the valuable learning encyclopedia of Wikipedia. By doing so, we can leverage the chain-out and chain-in relationships specific to Wikipedia while also considering the orientation of the graphs.

The generated directed graph G_r is then fed into node2vec to obtain the embedding of each vertex in the directed graph G_r. Finally, the cosine similarity of the embeddings of each vertex is used to derive the feature vector for each concept pair, which we refer to as graph structured learning.

Another component of our concept graph learning model, known as heterogeneous graph learning, involves initially inputting the learned heterogeneous graph $G_h(C_h, O_h, E_h)$ into the HAN (Heterogeneous Attention Network) network. We then employ the Siamese Network for prediction by classifying the learned vectors. Intuitively, in this graph, where the concepts and the learned objects are mapped into the same space, we align the learned graph features with the ones we desire.

Ultimately, we stitch together the feature vectors of the concept pairs learned from the G_r graph and the concept pairs learned from the G_h graph to obtain the final loss value. This process is depicted in Figure 1.

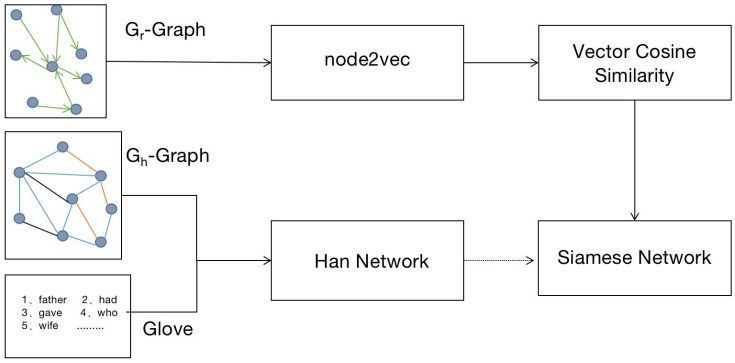

Figure 1: The entire CSL Model

4.2 Graph Embedding

Graph Construction We have carefully examined the role of Wikipedia within the context of the entire graph. [23] used Wikipedia clickstream data and related concept sets to discover the pre-relationships between Wikipedia concepts. Considering its significance, we have opted to utilize the RefD weights with linked relations as the foundation of our learning graph, drawing insights from the vast knowledge base of Wikipedia. Presented below is our proposed RefD concept graph, depicted in Figure 2. Our objective is to ascertain the RefD values that characterize the relationships between concepts, subsequently enabling us to construct a concept graph $G_r(C_r, E_r)$ by incorporating the calculated RefD weights.

Theoretical Analysis The composition of the G_r diagram composed between concepts, the G_r diagram obtained by provides us with valuable insights. Through the mathematical axiom RefD calculation, we can obtain a G_r diagram that captures the presence of potential precondition relationships between concepts, and the formula for the RefD calculation is as follow. The RefD calculation formula is presented as follows:

$$RefD\ (A,B) = \frac{\sum_{i=1}^{k} r\ (c_i,B) \cdot w(c_i,A)}{\sum_{i=1}^{k} w\ (c_i,A)} - \frac{\sum_{i=1}^{k} r\ (c_i,A) \cdot w(c_i,B)}{\sum_{i=1}^{k} w\ (c_i,B)}$$

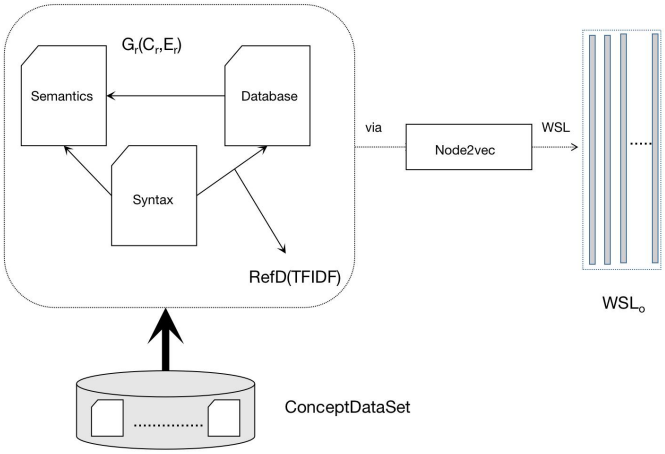

Figure 2: G_r Graph

where $C = \{c_1, ..., c_k\}$ is the concept space; $w(c_i, A)$ weights the importance of c_i to A; $r(c_i, A)$ is an indicator showing whether c_i refers to A, possibly a link in Wikipedia, a mention in a book, a citation in a paper, etc.

Wikipedia is a vast knowledge base from which RefD formulas can be accessed and used in this study. In order to capture the relationship with the corresponding terms in Wikipedia, our RefD is divided into two calculations, EQUAL and TF-IDF. The RefD calculated with TF-IDF is considered more reasonable in comparison, and all subsequent experimental RefDs use TF-IDF as the experimental parameter. Specifically, we define the calculation of *w(c, A)* for the Wikipedia-based RefD as follows:

EQUAL: A is represented by the concepts linked from it (L(A)) with equal weights.

$$w\ (c,A) = \begin{cases} 1 & if\ c \in\ L(A) \\ 0 & if\ \ c \notin\ L(A) \end{cases}$$

TF-IDF: A is represented by the concepts linked from it with TF-IDF weights.

$$w\ (c,A) = \begin{cases} tf(c,A) * log\ \frac{N}{df(c)} & if\ c \in\ L(A) \\ 0 & if\ \ c \notin\ L(A) \end{cases}$$

where *tf*(*c*, *A*) is the number of times c is linked from *A*; *N* is the total number of Wikipedia articles; and *df*(*c*) is the number of Wikipedia articles in which appears.

Get WSL_o Embedding There may be specific relationships between concepts that are implicit in the vector groups obtained through node2vec learning, in order to dig deeper to find such relationships.[24] argues that node2vec is a scalable feature learning method that is more suitable for learning feature-edge relationships between vertices. We used node2vec to learn G_r graphs. The set of WSL_o feature vectors was derived using the vector cosine function. The vector

cosine function, which we call WSL (Wikipedia Structure Learning), is better suited to learning feature-edge relationships between vertices.

4.3 Heterogeneous Concept Graph Construction

In modern machine learning, most features are extracted manually from articles, which are very logical, but also lacks overall framework circulation, for which we were inspired by [20] and designed these three edges for heterogeneous graphs to perform learning.

Formally, we define the whole heterogeneous graph as $G_h = (C_h, O_h, E_h)$, where C_h represents the set of concept vertices $C_h = \{C_1, C_2, C_3, ..., C_m\}$. Learning resources can be defined as $O = \{O_1, O_2, ..., O_n\}$.In order to better explore the implicit relationships between concepts through learning objects, three types of edge relationships are set up. The edge connecting concepts to learning objects and the edge between learning objects can be considered as $E_h = \{E_{co}, E_{oo}, E_{cc}\}$. The following are the definitions of the three kinds of edges.

Here is the formula for the edge we have defined:

$$\mathrm{A(i,j)} = \begin{cases} tfidf(i,j) & if\ \ i \in C_h, j \in O \\ pmi(i,j) & if\ \ i \in C_h, j \in C_h \\ dis(i,j) & if\ \ i \in O, j \in O \end{cases}$$

1. The edge between a concept and a learning object represents their interconnectedness. It is characterized by the term frequency-inverse document frequency (TF-IDF) of the concept. The TF-IDF value, which encompasses both the edge and its weight, signifies the significance of the concept in relation to the document. It is calculated based on the term frequency-inverse document frequency, which quantifies the number of occurrences of the concept within the document. The term frequency refers to the frequency of the concept's occurrences within the document. It represents the number of times the concept appears in the document. On the other hand, the inverse document frequency is a logarithmically scaled inverse proportion that accounts for the prevalence of the concept across multiple documents. It denotes the logarithmic ratio of the number of documents containing the concept. Thus, the logarithmic proportion of the number of documents containing the concept is taken into consideration in determining the weight of the edge. E_{co} has been labeled in Figure 3.

2. The weights of this edge can be determined using point-level mutual information (P-MI). Mathematically, P-MI is calculated as follows: $pmi(i,j) = \log\frac{p(i,j)}{p(i)*p(j)}$, $pmi(i,j) = \frac{\#W(i,j)}{\#W}$ and $p(i) = \frac{\#W(i)}{\#W}$ .where $p(i,j)$ represents the probability of co-occurrence of concepts c_i and c_j in the sliding windows, $p(i)$ represents the probability of concept c_i occurring in the sliding windows, and $p(j)$represents the probability of concept c_j occurring in the sliding windows. To calculate the weights, we determine the probabilities $p(i,j)$, $p(i)$, and $p(j)$ based on the counts of sliding windows. $\#W(i,j)$ represents the number of sliding windows that contain both concepts c_i and c_j, $\#W(i)$ represents the number of sliding windows that contain only concept c_i, and $\#W$ represents the total number of sliding windows.

3. In the context of education data, an edge is established between two learning objects, and the weight of this edge represents the normalized distance between them. Mathematically, this distance is calculated as the absolute difference between the positions of the learning objects, divided by the total number of learning objects (m). This can be denoted as $dis(i,j) = \frac{|i-j|}{m}$, as illustrated in Figure 3 for the E_{oo}.

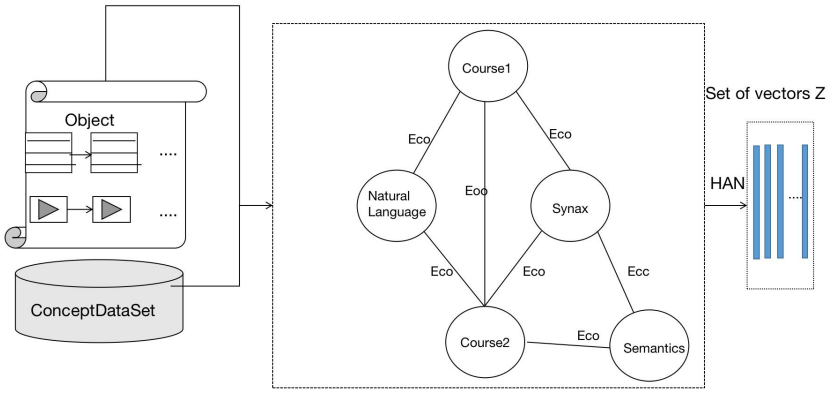

Figure 3: The operation of G_h Graph

4.4 Concept Representation via HAN

To train the graphs representing the relationships between learning objects (LOs) and concepts, we employed the HAN (Graph Attention Mechanism Network) approach.The Han network enables multi-level learning using metapath [26] as the learning path basis, and on the whole graph, we can consider that there are three path relation edges. In our graph G, we initially transformed the text of each concept and LO representation into vectors using GloVe [3] embeddings. These GloVe vectors, readily available from the GloVe library, were used to represent the LOs. We calculated the average word embedding for each concept within its associated LO. Subsequently, we fed the node vectors derived from both our heterogeneous graph G_h and GloVe embeddings into the HAN network. The heterogeneous graph G_h consists of the set of relationships E_h, comprising E_{co} (concept-object), E_{oo} (object-object), and E_{cc} (concept-concept). The overall graph structure is depicted in Figure 3.

Node Level Attention To align the input vertex eigenvector matrices to a common dimension, we utilized the transformation matrix E_{ij} .Specifically, since the HAN network leverages the self-attention mechanism to learn weights, we computed the importance of each concept's corresponding vertices. We obtained the attention values between the *ij* vertices through the Softmax function, yielding the node-level vector for each node:

$$h_i^{'} = M_{Ei} \cdot h_i$$

$$e_{ij}^{E} = \sigma(a_E^T \cdot [h_i^{'} \,||\, h_j^{'}])$$

$$a_{ij}^{E} = softmax(e_{ij}^{E})$$

$$Z_i^E = \sigma[\sum_{j=0}^{N^i} a_{ij}^E h_j^{'}]$$

where σ is the sigmod activation function, $||$ is the splicing operation, a^E is the node-level attention vector, and E is the class of edges, $E \in \{ E_{co}, E_{oo}, E_{cc}\}$.

Semantic Level Attention There are different categories of edges in the heterogeneous graph, and in order to learn the different categories of edges, we use the semantic level function to get the vector q and the embedding of each edge

category after the transformation, and finally the node representation vector group Z for each different type of edge is obtained by the Softmax operation as the next equation:

$$w_E = \frac{1}{|V|}\sum_{i\in V} q^T \cdot tanh\,(W \cdot Z_i^E + b)$$

$$\beta_E = \frac{exp\,(w_{E_i})}{\sum_{i=1}^{p} exp\,(w_{E_i})}$$

$$Z = \sum_{i=1}^{p} \beta_{E_i} \cdot Z_{E_i}$$

where β is the weight of each category and q is the semantic level attention vector.V stands for the number of all nodes.

4.5 Prerequisite Relation Classification

Using the conceptual feature vectors obtained from the HAN, we employed the Siamese Network to make predictions regarding the precondition relation between Z_i and Z_j. The Siamese Network allowed us to determine whether Z_i acts as a precondition for Z_j ,as shown in Figure 4.First we pass the sum of the concept vectors through the HAN network through a fully connected hierarchy and the ReLU layer of the activation function, and then into the Siamese Network.$\vec{Z}_i = ReLU(W \cdot Z_i + b)$,formally we obtain $v = \sigma(W^H[\vec{Z}_i \oplus \vec{Z}_j \oplus \vec{Z}_i - \vec{Z}_j \oplus \vec{Z}_i \otimes \vec{Z}_j] + b^H)$, The vectors are obtained by a series of intermediate vector splicing operations as follows, and finally the probability p is obtained by an activation function.where σ is activation function.Finally, we use the cross-entropy as the loss function : $L_h = \frac{1}{|T|}\sum_{(z_i,z_j,y_{ij})\in T} -[y_{ij}\,log\,(p(z_i, z_j) + (1-y_{ij}) \cdot log\,(1-p(z_i, z_j)]$, where T is the training dataset, and $y_{ij} \in \{0,1\}$ is the ground truth of (z_i, z_j).

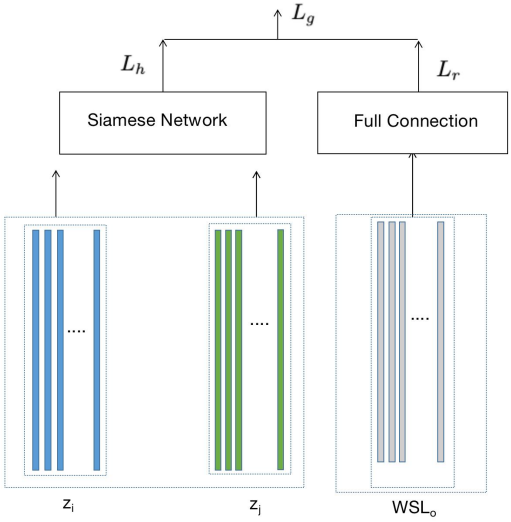

Figure 4: The Global Loss

Optimistic With Global Loss In order to efficiently connect the Loss loss function obtained through the WSL_o model to the above L_h, we propose a hyperparameter θ, finally optimises L_h and L_r jointly by means of a hyperparameter,

we define $L_g = L_h + \theta \cdot L_r$.For L_r, the set of vectors of WSL_o corresponding to all concepts is obtained by putting the set of concepts in the form of concept pairs into a fully connected layer for learning $p_r(z_i, z_j)$,and $L_r = \frac{1}{|T|} \sum_{(z_i,z_j,y_{ij}) \in T} -[y_{ij} \log(p_r(z_i, z_j) + (1 - y_{ij}) \cdot \log(1 - p_r(z_i, z_j)]$.After experiments, we found that θ acts as a bridge throughout the Loss function.

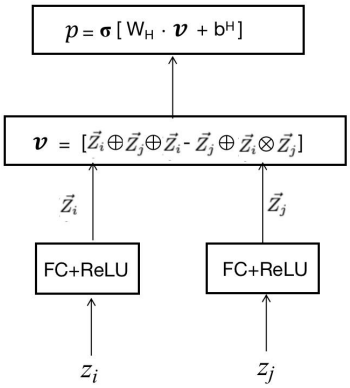

Figure 5: Siamese network

5 EXPERIMENTS

5.1 Datasets and Experimental setup

To validate the performance of our model, we chose two datasets in different domains for this experiment in Table 1：

Table 1: Dataset Performance

Dataset	$\lvert D \rvert$	$\lvert P_c \rvert$	$\lvert E_c \rvert$	$\lvert C \rvert$
University Course	654	90	1008	365
W-ML	548	50	486	120

University Course:This dataset [8] has 654 courses with 861 course prerequisites from various US universities and manually annotated 1008 pairs of concepts with prerequisite relation.

W-ML: For the W-ML dataset, we utilized the MOOC data mentioned in [4]. This dataset covers two areas: Wikipedia Data Structures and Algorithms (**W-DSA**) and Wikipedia Machine Learning (**W-ML**), for W-ML there are 548 course videos. W-ML was extracted from 240 courses, and it consists of 120 concepts and corresponding videos that were extracted from 240 lessons using information from Wikipedia.

In all datasets, only the concepts were labeled. To address the imbalance between positive and negative samples, we divided the datasets into test and training sets. To make the experiment more accurate, we divided it into 10 parts $\lvert P_c \rvert$ for multiple tests. And we allocated 1/3 of the datasets as the test set and 2/3 as the training set. In order to handle the class imbalance, we applied oversampling techniques. Specifically, we oversampled the positive samples 3.5 times in all datasets and 1.5 times in the test set.

The experimental results were averaged over 200 iterations. We employed a learning rate decay of 0.95% every 10 iterations. The hyperparameters were set as follows: 1 and 0.4 for the University Course and W-ML datasets respectively.The normalization parameter was set to 0.3.For the node2vec algorithm, we set the output dimension (d) to

128. The return parameters (p and q) were set to 0.25 and 4 respectively. The walk length (l) was set to 80, and the number of walks per node (r) was set to 10.Regarding the HAN (Heterogenous Attention Network), we optimized the model using three heads and two layers. The number of hidden neurons was set to 256 and 300 for the University Course and W-ML datasets respectively. The output dimension was determined by the number of concepts. We utilized pre-trained GloVe embeddings with 300 dimensions.We trained all the datasets as depicted in Figure 6. We observed stable results when the number of training iterations reached 100.

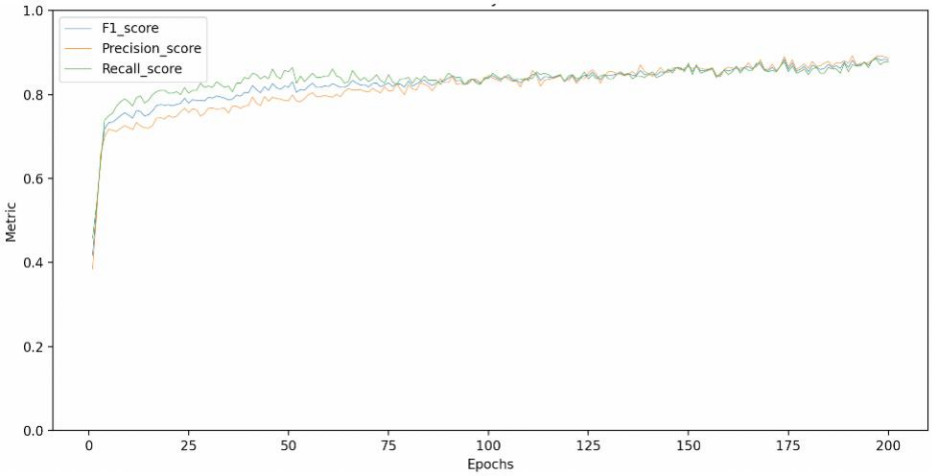

Figure 6: The Chart of Experimental results

5.2 Baselines

In our experimental approach, we explored several existing baselines to compare their performance on the dataset. These baselines include:

SVM: SVM [25] is a machine learning model. We employed SVM for classification prediction by inputting models of our concept pairs.

RefD: RefD [1] is a link-based metric that measures prior relationships between concepts. It provides a simple approach to quantify the connections between concepts.

GAE: GAE [18] is an unsupervised model that utilizes graph self-encoding. We input concept graphs into GAE for classification prediction.

PREREQ: PREREQ [8] leverages a pairwise linked LDA (Latent Dirichlet Allocation) model to obtain potential representations of concepts. It further identifies concept prior relationships through Siamese networks.

By conducting experiments with these existing baselines on our dataset, we obtained corresponding experimental results. These results serve as a basis for evaluating the performance and effectiveness of our proposed approach.

5.3 Experimental Result

We evaluate our model in terms of Precision (*P*), Recall (*R*), and F1-score (*F1*). From Table 2, it is clear that the results clearly demonstrate the F1-score and Recall metricsat in the University Course dataset far, our method outperforms the RefD method by 0.17 and 0.20, and significantly, achieving a higher F1-score and Recall by 0.17 and 0.20, respectively. It is worth noting that our method has the best performance on the University Course dataset.

Table 2: Evaluation Results

Dataset	Metric	SVM	RefD	GAE	PREREQ	CSL
University Course	P	0.698	0.631	0.450	0.468	**0.844**
	R	0.667	0.667	0.776	0.791	**0.862**
	F1	0.682	0.649	0.597	0.597	**0.852**
W-ML	P	**0.979**	0.357	0.300	0.563	0.733
	R	0.324	0.475	0.691	0.716	**0.730**
	F1	0.487	0.408	0.401	0.630	**0.721**

In Figure 7, the SVM method has a very high SVM achieves high scores in the W-ML dataset, except for the Precision metric, which exceeds the surpasses other baseline methods. However, when considering terms of the F1-score composite, the CSL method is more stable, so I think our method is successful composite F1-score, the CSL method demonstrates more stability. Therefore, we conclude that our method is successful for both datasets.

These results highlight the superior performance of our proposed method in terms of Precision, Recall, and F1-score, surpassing the RefD method in the University Course dataset. Although the SVM method excels in the W-ML dataset, the CSL method exhibits greater stability in terms of the F1-score. Thus, our method proves to be effective for both datasets.

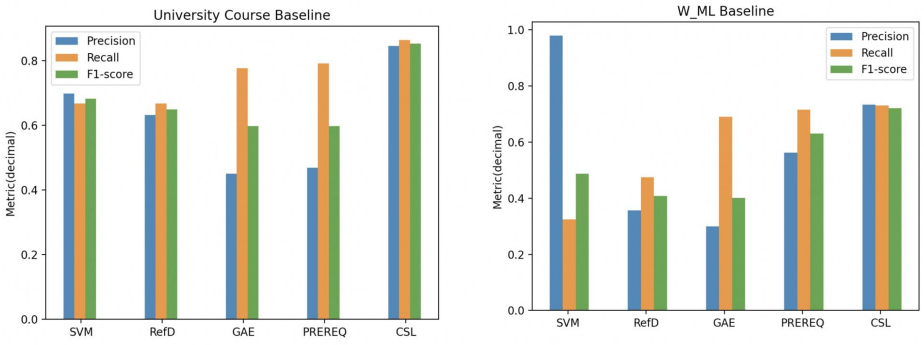

Figure 7: The ensemble results of Datasets

5.4 Ablation Experiments

To assess the impact of different experimental parameters on the overall performance of our model, we conducted ablation experiments on the CSL model, divided into three distinct parts. The first part focused on evaluating the effectiveness of Wikipedia Structure Learning WSL.

Effectiveness of Wikipedia Structure Learning To verify the effectiveness of WSL during the training of the CSL model, we performed a control experiment by comparing the performance of the complete model with and without WSL learning. The Precision (*P*), Recall (*R*), and F1-score (*F1*) metrics for the University Course and W-ML datasets are presented in Table 3.

From the results of the control experiments shown in Table 3, it is evident that the inclusion of WSL improves the F1-score metric by 0.08 on the University Course dataset, corresponding to a 10.4% increase in performance compared to the model without WSL. This clearly demonstrates the positive impact of WSL on the overall performance of the CSL model.

Table 3: Ablation Experiment Results - Effectiveness of Wikipedia Structure Learning

Dataset	Metric	HAN	HAN+WSL
University Course	P	0.814	↑0.844(+0.036)
	R	0.751	↑0.862(+0.147)
	F1	0.772	↑0.852(+0.103)
W-ML	P	0.719	↑0.720(+0.0013)
	R	0.678	↑0.726(+0.0707)
	F1	0.696	↑0.721(+0.0359)

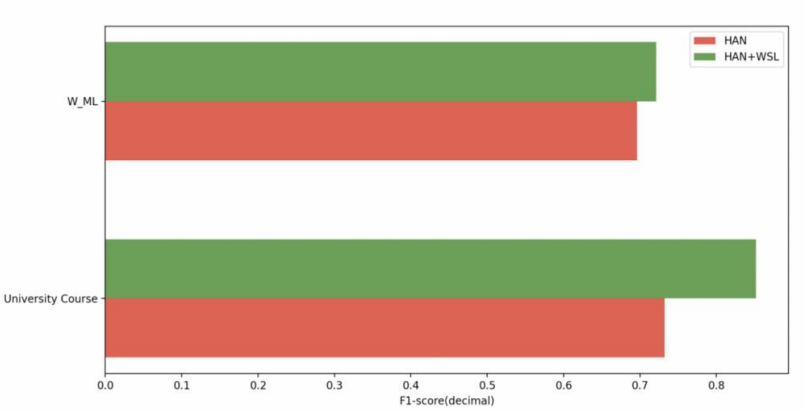

Figure 8: This Figure shows the effect of WSL on the results of the experiment

The results in Table 3 clearly demonstrate that the inclusion of WSL in the CSL model leads to a significant improvement in the F1-score metric for the University Course dataset, highlighting its effectiveness in enhancing the model's performance. The W-ML dataset also exhibited improvement, as indicated by an increase of 0.03 in Figure 8, corresponding to a 3.6% performance gain. The impact of WSL on both datasets is significant. This experiment provides evidence for the effectiveness of incorporating WSL-learned concept feature vectors into HAN networks. By extending the feature vectors through WSL, our model achieves enhanced performance on both datasets.

Effectiveness of Hyper-parameters In this section, we conducted experiments with different hyperparameters of the model, ranging from 0.1 to 1, and recorded the results presented in Table 4. Based on the obtained results, we observed that the University Course dataset achieves the highest F1-score, Recall, and Precision when the hyperparameter is set to 1.0. Conversely, the W-ML dataset achieves the highest F1-score when the hyperparameter is set to 0.4.The WSL module incorporates vector features obtained from node2vec, and for the University Course dataset, the features derived from WSL learning play a crucial role in improving performance. However, for the W-ML dataset, the impact of WSL-derived features is not as significant compared to the Siamese Network. Nevertheless, the WSL function acts as a bridge between the two loss functions, resulting in a more rational and effective framework for the entire model.

Table 4: This table represents the influence of θ on the whole experiment in our CSL approach

Dataset	Metric	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
University Course	P	0.842	0.842	0.839	0.841	0.843	0.842	0.837	0.832	0.833	**0.844**
	R	0.854	0.855	0.860	0.856	0.860	0.850	0.846	0.854	0.848	**0.862**
	F1	0.848	0.848	0.849	0.848	0.852	0.846	0.841	0.842	0.840	**0.852**
W-ML	P	0.726	0.727	0.726	0.715	0.720	0.731	0.732	0.733	0.719	**0.734**
	R	0.708	0.670	0.707	**0.730**	0.704	0.710	0.709	0.708	0.722	0.705
	F1	0.716	0.696	0.716	**0.721**	0.711	0.718	0.719	0.719	0.719	0.720

Effectiveness of Hidden Neural Node In addition to the previous experiments, we also conducted an investigation into the impact of the number of neurons in the hidden layer on the overall performance of the model. The F1-score was used as the primary metric to evaluate the results.

Table 5 presents the results of this experiment, showing the F1-scores obtained for different numbers of hidden neurons for both the University Course and W-ML datasets.

This experiment highlights the importance of selecting an appropriate number of hidden neurons for different datasets. For the University Course dataset, increasing the number of hidden neurons can lead to improved performance, while for the W-ML dataset, a higher number of hidden neurons can compensate for the limited number of concepts and result in better outcomes.

Table 5: This table represents the influence of the number of hidden neurons on the whole experiment in our CSL approach

Dataset	Metric	32	64	128	256	300
University Course	F1	0.829	0.824	0.824	0.851	**0.852**
W-ML	F1	0.452	0.576	0.698	0.616	**0.721**

6 CONCLUSIONS AND FUTURE WORK

Overall, a weak supervised learning method collaborative structured learning framework is proposed in this paper. This framework takes the advantage of a graph attention neural network and node2vec to learn a heterogeneous graph composed of concepts and learning objects and a concept graph based on Wikipedia. The experiment demonstrates the effectiveness of integrating graph embedding vectors and leveraging the HAN network for concept relationship prediction. This method has positive significance in promoting the development of online education and recommendation systems.

For the future work, the hyperparameters used in our experiments were set based on empirical observations. Further optimization of these hyperparameters through techniques like grid search or Bayesian optimization could potentially improve the model's performance. Extension to other types of relationships: In this experiment, we focused on predicting preconditioned relationships between concepts. Expanding the model to handle other types of relationships, such as co-occurrence or prerequisite relation, could provide a more comprehensive understanding of concept interdependencies.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China [62102136],and this research was supported by The National Natural Science Foundation of China (No.61977021, No.62102136), The Technology

Innovation Special Program of Hubei Province (No.2018ACA139, No.2019ACA144), The Research Project of Hubei Provincial Department of Education (No.D20191002), and this work is supported by the National Natural Science Foundation of China (No. 62377009).

REFERENCES

[1] L. Chen, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[2] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. In Proceedings of the 7th International World Wide Web Conference, pages 161–172. Elsevier, 1999.

[3] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In Conference on Empirical Methods in Natural Language Processing, 2014.

[4] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, W. Feng, J. Luo, C. Wang, L. Hou, and J. Li. Mooccube: A large-scale data repository for nlp applications in moocs. In Meeting of the Association for Computational Linguistics, 2020.

[5] J. Yu, Y. Wang, Q. Zhong, G. Luo, Y. Mao, K. Sun, W. Feng, W. Xu, S. Cao, and K. Zeng. Mooccubex: A large knowledge-centered repository for adaptive learning in moocs. 2021.

[6] I. Li, Alexander R Fabbri, Robert R Tung, and Dragomir R Radev. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning, 2018.

[7] Y. Yang, H. Liu, J. Carbonell, and W. Ma. Concept graph learning from educational data. ACM, 2015.

[8] Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. Inferring concept prerequisite relations from online educational resources. 2018.

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

[10] Kenneth Ward Church and Patrick Hanks. Word associations: Norms, similarity, and frequency of occurrence. In Proceedings of the 28th annual conference on Association for Computational Linguistics, pages 76–83, 1990.

[11] L. Chen, J. Ye, Z. Wu, Bart K Pursel, and Clyde Lee Giles. Recovering concept prerequisite relations from university course dependencies. 2017.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[13] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. ACM, 2016

[14] X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P Yu, and Y. Ye. Heterogeneous graph attention network. 2019.

[15] Y. Xu and J. Yang. Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution. 2019.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[17] I. Li, V. Yan, T. Li, R. Qu, and D. Radev. Unsupervised cross-domain prerequisite chain learning using variational graph autoencoders. 2021.

[18] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In Advances in Neural Information Processing Systems, pages 722–730, 2016.

[19] Hao Sun, Yuntao Li, and Yan Zhang. ConLearn: Contextual-knowledge-aware Concept Prerequisite Relation Learning with Graph Neural Network, pages 118–126.

[20] Chenghao Jia, Yongliang Shen, Yechun Tang, Lu Sun, and Weiming Lu. 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, pages 2036–2047, Online, June 2021. Association for Computational Linguistics.

[21] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks, 2017.

[22] ROOPAK, SHAH, EDUARD, SCKINGER, JAMES, W., BENTZ, ISABELLE, GUYON, and CLIFF. Signature verification using a ″siamese″ time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 07(4):669–669, 1993.

[23] Cheng Hu, Kui Xiao, Zesong Wang, Shihui Wang, and Qifeng Li. Extracting prerequisite relations among wikipedia concepts using the clickstream data. In Han Qiu, Cheng Zhang, Zongming Fei, Meikang Qiu, and Sun-Yuan Kung, editors, Knowledge Science, Engineering and Management, pages 13–26, Cham, 2021. Springer International Publishing.

[24] Haoyu Wen, Xinning Zhu, Moyu Zhang, Chunhong Zhang, and Changchuan Yin. Combining wikipedia to identify prerequisite relations of concepts in moocs. In Teddy Mantoro, Minho Lee, Media Anugerah Ayu, Kok Wai Wong, and Achmad Nizar Hidayanto, editors, Neural Information Processing, pages 739–747, Cham, 2021. Springer International Publishing.

[25] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

[26] Deepti Gaur, Aditya Shastri, Ranjit Biswas and D. Seema Gaur, "Vague Metagraph," International Journal of Computer Theory and Engineering vol. 1, no. 2, pp. 126-130, 2009.