Efficient Variational Graph Autoencoders for Unsupervised Cross-domain Prerequisite Chains

Irene Li, Vanessa Yan, Dragomir Radev
Department of Computer Science
Yale University
{irene.li,vanessa.yan,dragomir.radev}@yale.edu

Abstract

Prerequisite chain learning helps people acquire new knowledge efficiently. While people may quickly determine learning paths over concepts in a domain, finding such paths in other domains can be challenging. We introduce Domain-Adversarial Variational Graph Autoencoders (DAVGAE) to solve this cross-domain prerequisite chain learning task efficiently. Our novel model consists of a variational graph autoencoder (VGAE) and a domain discriminator. The VGAE is trained to predict concept relations through link prediction, while the domain discriminator takes both source and target domain data as input and is trained to predict target domain labels. Most importantly, this method only needs simple homogeneous graphs as input, unlike the current state-of-the-art model which requires the construction of heterogeneous graphs. We evaluate our model on the LectureBankCD dataset, and results show that our model outperforms recent graph-based benchmarks while using only 1/10 of graph scale and 1/3 of computation time.

1 Introduction

A prerequisite is defined as a concept which must be learned prior to another concept. Knowing prerequisite relationships between concepts helps determine learning paths for students who wish to acquire new knowledge. [1, 2, 3]. Most existing work on prerequisite chain learning is limited to a single domain. [4, 5, 6]. More recently, [7] introduced the task of cross-domain prerequisite chain learning. This task is useful for development of educational resources, intelligent search engine rankings, and other services for people who may have an excellent background in one domain, such as natural language processing (NLP), and wish to learn concepts in a new domain, such as Bioinformatics (BIO). These two domains share common fundamental concepts: machine learning basics, time-series data, statistics, etc. Given the prerequisite chains in a source domain, it is possible to transfer the knowledge to learn prerequisite chains in a target domain. [7] proposed the CD-VGAE (cross-domain variational graph autoencoder) model to apply domain transfer and infer target concept relations.

However, CD-VGAE was trained on a complex graph that contains many resource nodes and concept nodes from both the source and target domains, making it limited in scalability. A known challenge of applying graph neural networks in practice is the difficulty of scaling these models to large graphs. [8, 9]. We seek to develop a model that can be trained on a much smaller graph than the CD-VGAE model, so that the model can be more practical in real-world applications. Specifically, our model is trained on graphs with concept nodes only.

Adversarial methods [10] have been applied to NLP tasks that involve multilingual or multi-domain scenarios [11, 12, 13]. Such methods typically introduce a domain loss to a neural network in order to learn domain-invariant features for unsupervised domain adaptation. However, there has

Efficient Natural Language and Speech Processing (ENLSP) Workshop , 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Domain	# Files	# Concepts	# Pos. Relations
NLP	1,717	322	1,551
CV	1,041	201	871
BIO	148	100	234

Table 1: Statistics of the three domains from LectureBankCD [7]: Files (resource files: lecture slides); Pos. Relations (positive prerequisite relations).

been limited research in training adversarial networks on graphs. The only existing work is the adversarially regularized variational graph autoencoder (ARVGA) model [14] , which learns robust graph embeddings by reconstructing graph structure. We introduce a variant adversarial framework to solve cross-domain prerequisite chain learning.

Our contributions are two-fold. First, we propose domain-adversarial variational graph autoencoders (DAVGAE) to perform unsupervised cross-domain prerequisite chain learning. Second, we offer two ways to construct the concept graph: cross-domain and single-domain. The single-domain method further reduces the scale of the training graph and improves performance. We conduct comprehensive evaluations and show that our model surpasses the state-of-the-art (SOTA) performance while saving space complexity by up to 10 times and training time by up to 3 times. Our code will be made public.

2 Dataset and Task Definition

The LectureBankCD [7] dataset consists of concepts, resources (lecture slides from top universities), and manually annotated prerequisite relations between concepts, in three domains: NLP, BIO and CV (computer vision). We show the statistics of the dataset in Table 1. We follow the same experimental setting as [7], treating NLP as the source domain and BIO and CV as target domains.

We define cross-domain prerequisite chain learning as a binary classification problem. Given a source domain and a target domain, there are a number of concept pairs (p, q) in each domain. The label for the concept pair y is 1 if concept p is a prerequisite of concept q and 0 otherwise. We focus on the unsupervised transfer learning setting, in which the labels of the source domain y_{src} are known, but those of the target domain y_{tgt} are unknown.

3 Methodology

We propose the Domain Adversarial Variational Graph Autoencoders (DAVGAE) for unsupervised cross-domain prerequisite chain learning. The model architecture is shown in Figure 1.

Concept Graph Construction We define a concept graph $G = (X, A)$ as the input to the model. X is the set of node features and A is the adjacency matrix which indicates whether prerequisite relations exist between concept pairs. If $p \rightarrow q$, we define $A_{p,q} = 1$ and $A_{q,p} = 1$. To obtain X, we follow the same approach from [7] to train Phrase2Vec (P2V) [15] node embeddings: we extract free text from LectureBankCD lecture slides then train a P2V model to encode concepts.

We propose two ways to build the concept graph: **cross-domain** and **single-domain**. In the cross-domain one, all concept nodes are modeled in a single graph, and X consists of concepts from the source and target domain. We build the adjacency matrix A using two information sources: relations between source domain concept nodes given in LectureBankCD during training, plus additional relations from cosine similarity or pairwise mutual information (PMI) of node embeddings between concept pairs. We calculate cosine similarity between all possible concept pairs, but only PMIs between source concepts and target concepts.

In the single-domain method, we train on two single-domain concept graphs to further reduce the space complexity during training: source graph G_{src} and target graph G_{tgt}, in which X of the two graphs only contains concept node features from the source and target domains respectively. In G_{src}, A values consists of two parts: labeled relations between concept node pairs, and additional relations computed using cosine similarity. In contrast, the initial A in G_{tgt} only comes from cosine similarity.

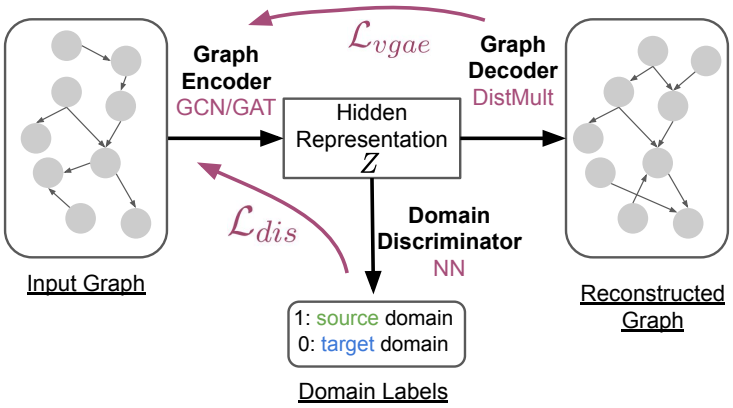

Figure 1: DAVGAE model.

DAVGAE The VGAE model [16] contains a graph neural network (GCN) encoder [17] and an inner product decoder. The loss of VGAE is defined as:

$$\mathcal{L}_{vgae} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}[\log p(\mathbf{A} \mid \mathbf{Z})] - \mathrm{KL}[q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) \| p(\mathbf{Z})], \tag{1}$$

where the first term indicates the reconstruction loss, and the second term represents the KL Divergence between the hidden layer representation Z and a normal distribution. It is possible to replace the GCN encoder with other architectures, such as graph attention networks (GAT) [18].

Domain-adversarial training is an established approach to learn representations for domain adaptation [19], but it has rarely been applied to graphs before to the best of our knowledge. To force the VGAE encoder to learn domain-invariant features of concept nodes, we add a *domain discriminator* module to predict which domain each node in the hidden layer representation Z belongs to. We use a two-layer neural network (NN) to predict domain labels: 1 if the node comes from the source domain and 0 otherwise. Thus, the discriminator loss $\mathcal{L}_{dis}$ is defined as a cross-entropy loss for domain prediction. The total loss of the DAVGAE model is:

$$\mathcal{L} = \mathcal{L}_{vgae} + \mathcal{L}_{dis} \tag{2}$$

We train DAVGAE at the graph level. In each epoch, we feed it with either the cross-domain graph or one of the single-domain concept graphs.

Link Prediction Since the prerequisite prediction should be asymmetric, it is not suitable to use an inner product decoder like the original VGAE did. Instead, we use DistMult [20] to predict the link between a concept pair (p, q) using a hidden layer representation (Z_p, Z_q). Specifically, we reconstruct the adjacency matrix $\hat{A}$ by learning a trainable weight matrix R, such that $\hat{A} = Z^\intercal RZ$. Finally, we apply a Sigmoid function to determine positive/negative label on the value of $\hat{A}_{p,q}$.

4 Evaluation

We apply the same split on the data as two previous works [2, 7]. Positive relations are divided into 85% training, 5% validation, and 10% testing. Negative relations are sampled randomly to ensure balance between positive and negative relations. In Table 2, we report average scores over five randomly seeded splits.

Unsupervised Baseline Models We establish unsupervised baselines using both machine learning classifiers and graph embedding methods. For each method, we experiment with P2V [15] and BERT concept embeddings pretrained on our corpus. `CLS + BERT/P2V`: We adapt the Machine Learning baselines from an existing work [7], concatenating paired concept embeddings and training a classifier. `GraphSAGE+BERT/P2V`: We adapt GraphSAGE [21] to generate node embeddings which

Method	NLP→CV			NLP→BIO		
	F1	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
Unsupervised Baseline Models						
CLS + BERT	0.4277	0.5743	0.3419	0.3930	0.7481	0.2727
CLS + P2V	0.4881	0.6106	0.4070	0.2222	0.6000	0.1364
GraphSAGE + P2V [21]	0.5342	0.5085	0.5515	0.5283	0.5177	0.5287
GraphSAGE + BERT [21]	0.5102	0.3611	0.5105	0.4736	0.4065	0.5180
VGAE + BERT [2]	0.5885	0.5398	0.6488	0.6011	0.6185	0.5909
VGAE + P2V [2]	0.6202	0.5368	0.7349	0.6177	0.6521	0.6091
Baseline with Extra Resource Nodes						
CD-VGAE + BERT [7]	0.6391	0.5441	0.7884	0.6289	0.6425	0.6364
CD-VGAE + P2V [7]	0.6754	0.5468	0.8837	0.6512	0.6667	0.6364
Cross-domain Concept Graph						
GAT [18]	0.6064	0.5281	0.7172	0.6257	0.5969	0.6609
GAT + cos	0.6276	0.5276	0.7793	0.6336	0.5644	0.7304
GAT + cos + DAVGAE (**ours**)	0.6251	0.5613	0.7218	0.6396	0.6557	0.6348
GCN [17]	0.5951	0.5361	0.6713	0.6319	0.6109	0.6609
GCN + cos	0.6318	0.5379	0.7655	0.6174	0.5991	0.6435
*GCN + cos + DAVGAE (**ours**)	**0.6321**	0.5661	0.7195	**0.6421**	0.5932	0.7130
Single-domain Concept Graph						
GAT [18]	0.5573	0.4897	0.7609	0.5756	0.5588	0.6348
GAT + cos	0.6287	0.5213	0.8023	0.5587	0.5248	0.6261
GAT + cos + DAVGAE (**ours**)	0.6356	0.5782	0.7149	0.6545	0.6024	0.7217
GCN [17]	0.5888	0.5169	0.6920	0.5304	0.5218	0.6348
GCN + cos	0.6232	0.5455	0.7287	0.6117	0.5599	0.6783
*GCN + cos + DAVGAE (**ours**)	**0.6771**	0.5734	0.8322	**0.6738**	0.6559	0.6957

Table 2: Evaluation results on two target domains. Underlined scores are the best among the baseline models.

Experiment	Model	# Graph node	Computational time
NLP→CV	CD-VGAE	3,281	127.5s
	Ours	**322**	**47.1s**
NLP→BIO	CD-VGAE	2,287	71.6s
	Ours	**322**	**30.2s**

Table 3: Comparison of graph scale and computation time. Computation time includes 200 epochs of training and one inference run. Ours: `GCN+cos+DAVGAE`.

are passed into DistMult. Model inputs include the BERT/ P2V embeddings of the source and target domain concepts, as well as an adjacency matrix constructed from annotations of source domain prerequisite relations and cosine similarities of target domain concept embeddings. `VGAE+BERT/P2V`: We use a VGAE model [2] to predict concept pair relations. All baseline models are trained on the NLP domain and applied directly on the target domains, so we call them unsupervised baselines.

Baseline with Extra Resource Nodes The recent CD-VGAE model from [7] constructs a cross-domain concept-resource graph to predict target domain prerequisite relations via optimized VGAEs [16].

Cross-domain, Single-domain Concept Graph In both groups, we experiment with GAT and GCN (in light gray shaded color) graph encoders, as well as cosine similarity for additional edge values when building the input graph. GAT works better in some settings, but the best performing model is `GCN+cos+DAVGAE` in terms of F1 score, across both CV and BIO domains. The results suggest that DAVGAE does not require a complex graph encoder like GAT. A simple GCN is enough to train domain-invariant features. The same trend is observed when DAVGAE is trained on the single-domain concept graph setting. No matter whether we train on a cross-domain or single-domain concept graph, DAVGAE consistently improves upon comparable unsupervised baselines. Furthermore, a

Domain	Graph	Path
CV	Ground Truth	object recognition, robotics, artificial intelligence,..., image processing, feature extraction, autonomous driving
	DAVGAE	object recognition, video classification, autonomous driving
BIO	Ground Truth	DNA, motif discovery
	DAVGAE	DNA, dynamic programming, RNA secondary structure, energy minimization, decision trees, sampling, motif discovery

Table 4: Case studies of concept paths.

DAVGAE trained on a single-domain concept graph not only achieves better performance compared to one trained on a cross-domain concept graph, but it also outperforms CD-VGAE. Overall, our best performing model is `GCN+cos+DAVGAE` (marked with *).

CD-VGAE is a competitive baseline, but it shows limited scalability as it trains on a larger graph and requires longer time for training. We list detailed numbers in Table 3. For example, in the NLP→CV experiment, our best model is trained on a graph with 322 nodes while CD-VGAE constructs a large graph of 3,281 nodes. In the best case, DAVGAE requires only 10% of the graph size and one-third of the training time as CD-VGAE does.

5 Analysis

We provide quantitative analysis and case studies on selected domains to shed light on the edges predicted by our model.

5.1 Quantitative Analysis

We compare our best model with the ground truth and another baseline model (`CLS+P2V`). We first recover the concept graph of the CV domain and review the degree of each concept node. Our model predicts 1,151 positive edges while the base model predicts 527. There are 871 in the ground truth. In general, our model has higher recall than the selected baseline. Higher recall is beneficial because we'd rather students learn extra concepts than miss important concepts.

5.2 Case Studies

In the concept graph recovered by DAVGAE, we observe that there are a few concept pairs which are connected by more than one path. The same is true for the ground truth graph. When there are cycles in the graph, finding all possible prerequisite paths becomes especially challenging. With this in mind, we conduct case studies on randomly selected paths.

In the CV domain, with random selection, there are usually 5-10 concepts in each path within the ground truth graph. The concept graph recovered by our model tends to have more and longer paths because more positive edges are predicted. In Table 4, we compare paths from the ground truth concept graph and our recovered graph. Both start with *object recognition* (colored blue) and end with *autonomous driving* (colored orange), with concepts linked from top to bottom. There exists a long path in the ground truth; however, but our model predicts a shorter one, indicating another possible learning path. In BIO, we show paths from *DNA* → *motif discovery*. There are 8 paths found in the ground truth with an average path length of 4; in comparison, the 8 paths found in our model prediction has an average length of 10.63. We show the shortest path between the selected concepts in the ground truth and DVGAE concept graphs. Our model this time predicts more concepts along the path than the ground truth.

We include two random concept paths from CV and BIO in Table 5. Both are predictions from our best performing model. In the left column, our model predicts many relations accurately in the CV domain, such as *video classification→autonomous driving*, *video and image augmentation→image generation* and *image generation→image to image translation*. However, a few concepts may not be predicted in the correct path, e.g. *gibbs sampling*. Similarly, in the right column, we observe

CV	BIO
optical flow	BLAST
trajectory prediction	hardy-weinberg equilibrium
eye tracking	ChIP-seq
camera localization	gene finding
gibbs sampling	multivariate linear model
shading analysis	graph theory
background modeling and update	transcription
motion detection and tracking	transcription factor
action or gesture recognition	position weight matrix
video classification	yeast 2-hybrid
autonomous driving	energy minimization
remote sensing	markov clustering
crowd counting	
graph rendering	
image processing	
video and image augmentation	
image generation	
image to image translation	

Table 5: A random concept path from CV (left column) and BIO (right column).

correct prerequisite relations in the BIO domain, such as *transcription→transcription factor* and *ChIP-seq→gene finding*.

6 Conclusion

In this paper, we propose the DAVGAE model to solve cross-domain prerequisite chain learning efficiently. DAVGAE outperforms unsupervised baselines trained on concept graphs by a large margin. It also outperforms an unsupervised SOTA model trained on a concept-resource graph, while significantly reducing computation space and time.

References

[1] F. ALSaad, A. Boughoula, C. Geigle, H. Sundaram, and C. Zhai, "Mining mooc lecture transcripts to construct concept dependency graphs.," *International Educational Data Mining Society*, 2018.

[2] I. Li, A. Fabbri, R. Tung, and D. Radev, "What should I learn first: Introducing lecturebank for NLP education and prerequisite chain learning," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6674–6681, AAAI Press, 2019.

[3] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, W. Feng, J. Luo, C. Wang, L. Hou, J. Li, Z. Liu, and J. Tang, "MOOCCube: A large-scale data repository for NLP applications in MOOCs," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 3135–3142, Association for Computational Linguistics, 2020.

[4] Y. Yang, H. Liu, J. G. Carbonell, and W. Ma, "Concept graph learning from educational data," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015* (X. Cheng, H. Li, E. Gabrilovich, and J. Tang, eds.), pp. 159–168, ACM, 2015.

[5] L. Pan, C. Li, J. Li, and J. Tang, "Prerequisite relation learning for concepts in MOOCs," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1447–1456, Association for Computational Linguistics, 2017.

[6] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles, "Investigating active learning for concept prerequisite learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 7913–7919, AAAI Press, 2018.

[7] I. Li, V. Yan, T. Li, R. Qu, and D. Radev, "Unsupervised cross-domain prerequisite chain learning using variational graph autoencoders," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[8] A. Bojchevsi, J. Klicpera, B. Perozzi, A. Kapoor, M. Blais, B. Rózemberczki, M. Lukasik, and S. Günnemann, "Scaling graph neural networks with approximate pagerank," in *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*, (Virtual Event, CA, USA), ACM, 2020.

[9] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, "Sign: Scalable inception graph neural networks," *arxiv*, 2020.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *CoRR*, vol. abs/1406.2661, 2014.

[11] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 557–570, 2018.

[12] X. Chen and C. Cardie, "Multinomial adversarial networks for multi-domain text classification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 1226–1240, Association for Computational Linguistics, 2018.

[13] I. Li, "Detecting bias in transfer learning approaches for text classification," *CoRR*, vol. abs/2102.02114, 2021.

[14] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden* (J. Lang, ed.), pp. 2609–2615, ijcai.org, 2018.

[15] M. Artetxe, G. Labaka, and E. Agirre, "Unsupervised statistical machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 3632–3642, Association for Computational Linguistics, 2018.

[16] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," in *Domain Adaptation in Computer Vision Applications* (G. Csurka, ed.), Advances in Computer Vision and Pattern Recognition, pp. 189–209, Springer, 2017.

[20] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[21] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds.), pp. 1024–1034, 2017.