Efficient Variational Graph Autoencoders for Unsupervised Cross-domain Prerequisite Chains

Yale

Irene Li, Vanessa Yan and Dragomir Radev

Yale University, USA

LILY Lab

Introduction

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.

Unsupervised Cross-domain Prerequisite chain learning: transfer knowledge from a source domain to a target domain.

Efficient Modeling using Graph Convolutional Neural Networks: utilizing domain adversarial training.

Efficiency: only 1/10 of graph scale and 1/3 of computation time compared with the previous sota.

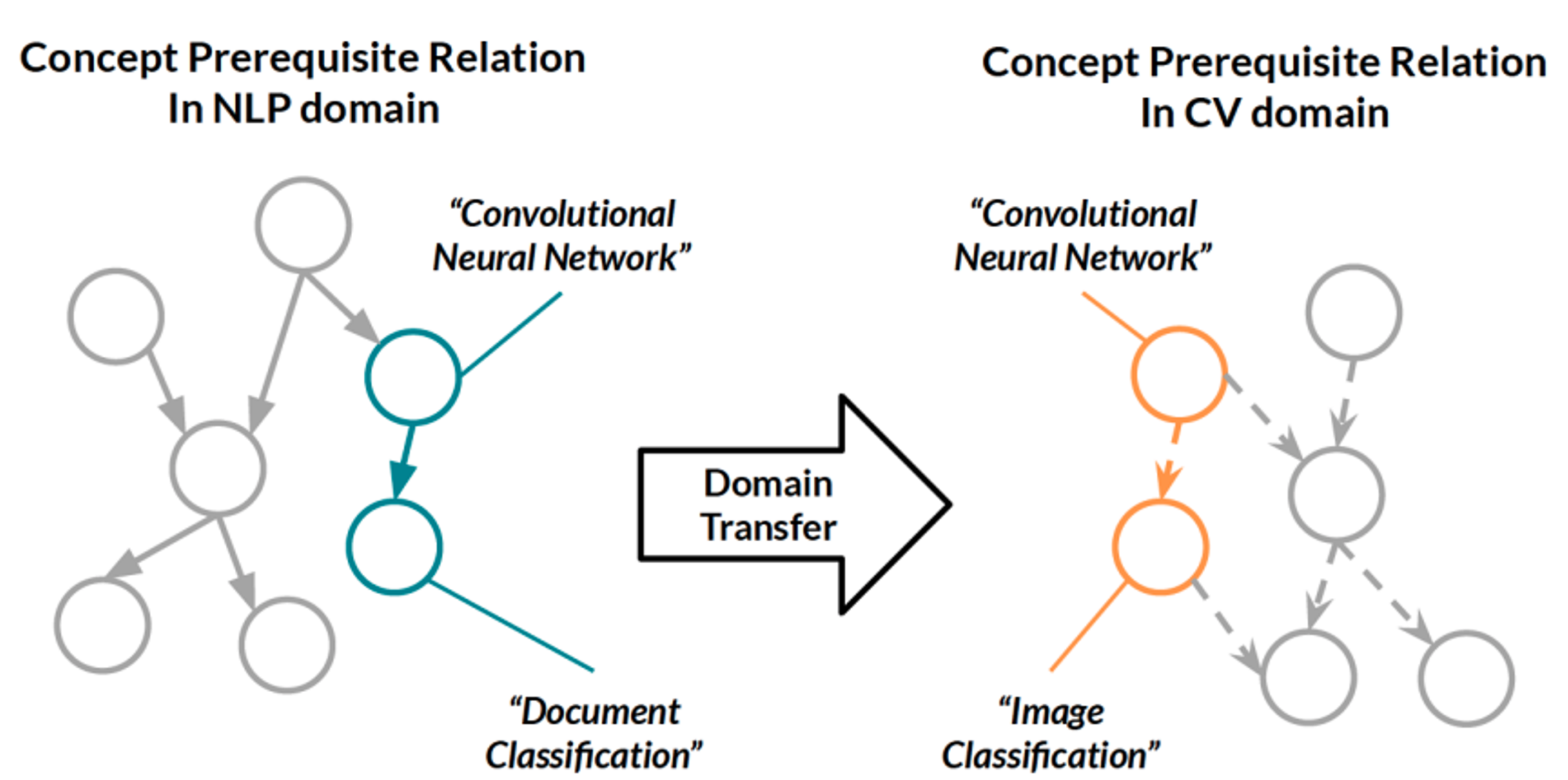

Figure 1: Cross-domain Prerequisite Chain Learning.

Dataset

LectureBankCD (Li et al., 2021): 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).

Transfer Settings:

NLP→ CV

NLP→ BIO

Domain	Files	Pages	Tks/pg	Con.	PosRel
NLP	1,717	65,028	47	322	1,551
CV	1,041	58,32	43	201	871
BIO	148	7,13	135	100	234

Table1: statistics of the three domains from LectureBankCD.
Files (resource files: lecture slides);Pos. Relations (positive prerequisite relations).

Domain Adversarial Variational Graph Autoencoders (DAVGAE)

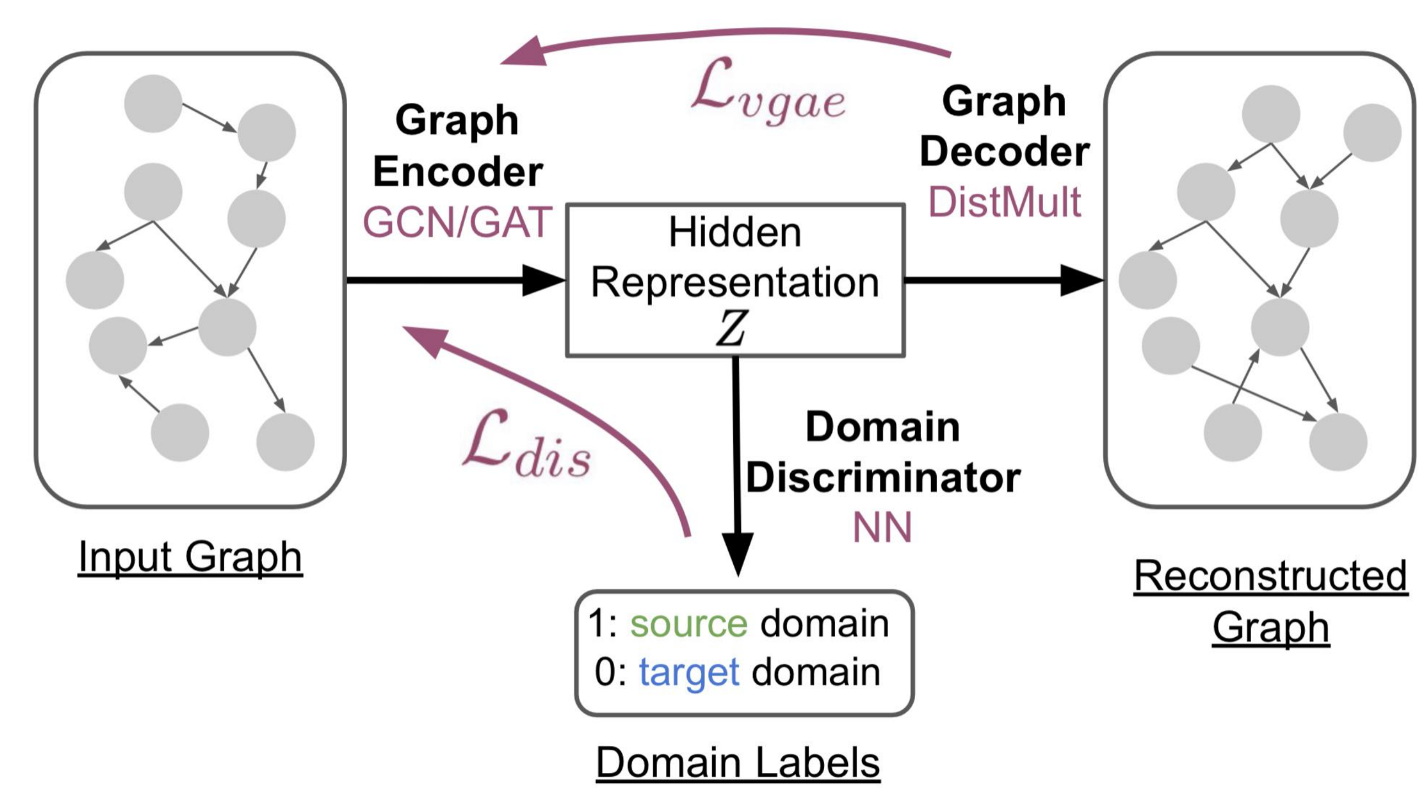

Figure 2: Model Illustration.

Graph Construction: all concept nodes from both source and target domain; shared concepts will be the *bridge* between the two domains. Pretrained node embeddings X by BERT, Phrase2Vec.

Graph Encoder: two-layer GCN or GAT.

$$f_{GCN}\left(H^{(l)}, A\right) = \phi\left(\widetilde{D}^{\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$$f_{GAT}\left(H^{(l)}, A\right) = \phi(\sum \boldsymbol{\alpha} W^{(l-1)}H^{(l-1)}),\quad \boldsymbol{\alpha} = \mathrm{Attention}(H^{(l-1)})$$

Decoder: DistMult given a concept pair, given hidden features Z, we learn a R matrix:

$$\hat{A} = Z^{\intercal}RZ$$

VGAE as the main link prediction framework: variational loss and edge reconstruction loss.

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

Domain Discriminator: a simple neural network (NN), domain classification. Then the final loss becomes:

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

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.

Method	NLP→CV F1	NLP→CV Precision	NLP→CV Recall	NLP→BIO F1	NLP→BIO Precision	NLP→BIO 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]	<u>0.6202</u>	0.5368	0.7349	<u>0.6177</u>	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]	<u>0.6754</u>	0.5468	0.8837	<u>0.6512</u>	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: Main Results.

Evaluation

Main Results in Table 2:

CLS: binary classifiers; **GraphSAGE:** topic embeddings

CD-VGAE: Baseline with Extra Resource Nodes. concept and resource graph, strong performance but graphs are very large.

Our best setting: GCN (encoder) + cosine (edge) + DAVGAE framework

Graph scale and computational time comparison in Table 3:

Significantly reduced the graph scale in both domains, as well as much less training time, compared to the previous sota.

Case Studies:

The path of a given concept pair (a blue and an orange concept), in the BIO case, our model predicts a longer path; but in the CV case, our model predicts a much shorter path with many concepts skipped.

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*.

Conclusion

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