Taxonomy Construction of Unseen Domains via Graph-based Cross-Domain Knowledge Transfer

Chao Shang[1], Sarthak Dash[2], Md Faisal Mahbub Chowdhury[2], Nandana Mihindukulasooriya[2], Alfio Gliozzo[2]
[1]University of Connecticut, Storrs, CT, USA
[2]IBM Research AI, Yorktown Heights, NY, USA
chao.shang@uconn.edu, sdash@us.ibm.com
mchowdh@us.ibm.com, nandana.m@ibm.com, gliozzo@us.ibm.com

Abstract

Extracting lexico-semantic relations as graph-structured taxonomies, also known as taxonomy construction, has been beneficial in a variety of NLP applications. Recently Graph Neural Network (GNN) has shown to be powerful in successfully tackling many tasks. However, there has been no attempt to exploit GNN to create taxonomies. In this paper, we propose *Graph2Taxo*, a GNN-based cross-domain transfer framework for the taxonomy construction task. Our main contribution is to learn the latent features of taxonomy construction from existing domains to guide the structure learning of an unseen domain. We also propose a novel method of directed acyclic graph (DAG) generation for taxonomy construction. Specifically, our proposed *Graph2Taxo* uses a noisy graph constructed from automatically extracted noisy hyponym-hypernym candidate pairs, and a set of taxonomies for some known domains for training. The learned model is then used to generate taxonomy for a new unknown domain given a set of terms for that domain. Experiments on benchmark datasets from science and environment domains show that our approach attains significant improvements correspondingly over the state of the art.

1 Introduction

Taxonomy has been exploited in many Natural Language Processing (NLP) applications, such as question answering (Harabagiu et al., 2003), query understanding (Hua et al., 2017), recommendation systems (Friedrich and Zanker, 2011), etc. Automatic taxonomy construction is highly challenging as it involves the ability to recognize – (i) a set of types (i.e. hypernyms) from a text corpus, (ii) instances (i.e. hyponyms) of each type, and (iii) *is-a* (i.e. hypernymy) hierarchy between types.

Existing taxonomies (e.g., WordNet (Miller et al., 1990)) are far from being complete. Taxonomies specific to many domains are either entirely absent or missing. In this paper, we focus on construction of taxonomies for such *unseen domains*[1]. Since taxonomies are expressed as *directed acyclic graphs* (DAGs) (Suchanek et al., 2008), taxonomy construction can be formulated as a DAG generation problem.

There has been considerable research on Graph Neural Networks (GNN) (Sperduti and Starita, 1997; Gori et al., 2005) over the years; particularly inspired by the convolutional GNN (Bruna et al., 2014) where graph convolution operations were defined in the Fourier domain. In a similar spirit to convolutional neural networks (CNNs), GNN methods aggregate neighboring information based on the connectivity of the graph to create node embeddings. GNN has been applied successfully in many tasks such as matrix completion (van den Berg et al., 2017), manifold analysis (Monti et al., 2017), predictions of community (Bruna et al., 2014), knowledge graph completion (Shang et al., 2019), and representations of network nodes (Hamilton et al., 2017; Kipf and Welling, 2017).

To the best of our knowledge, there has been no attempt to exploit GNN for taxonomy construction. Our proposed framework, *Graph2Taxo*, is the first to show that a GNN-based model using a cross-domain noisy graph can substantially improve the taxonomy construction of unseen domains (e.g., Environment) by exploiting taxonomy of one or more seen domains (e.g., Food). (The task is described in detail in Section 3.1.)

Another novelty of our approach is we are the first to apply the acyclicity constraint-based DAG structure learning model (Zheng et al., 2018; Yu et al., 2019) for taxonomy generation task.

The input of *Graph2Taxo* is a *cross-domain*

[1]By **unseen domain**, we refer to a domain for which taxonomy is not available to the system.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2198–2208

noisy graph constructed by connecting noisy candidate *is-a* pairs, which are extracted from a large corpus using standard linguistic pattern-based approaches (Hearst, 1992). It is `noisy` because pattern-based approaches are prone to poor coverage as well as wrong extractions. In addition, it is `cross-domain` because the noisy *is-a* pairs are extracted from a large-scale corpus which contains a collection of text from multiple domains.

Our proposed neural model directly encodes the structural information from a noisy graph into the embedding space. Since the links between domains are also used in our model, it has not only structural information of multiple domains but also cross-domain information.

We demonstrate effectiveness of our proposed method on *science* and *environment* datasets (Bordea et al., 2016), and show significant improvements on F-score over the state of the art.

2 Related Work

Taxonomy construction (also known as taxonomy induction) is a well-studied problem. Most of the existing works follow two sequential steps to construct taxonomies from text corpora (Wang et al., 2017). First, *is-a* pairs are extracted using pattern-based or distributional methods. Then, a taxonomy is constructed from these *is-a* pairs.

The pattern-based methods, pioneered by Hearst (1992), detect *is-a* relation of a term pair (x, y) using the appearance of x and y in the same sentence through some lexical patterns or linguistic rules (Ritter et al., 2009; Luu et al., 2014). Snow et al. (2004) represented each (x, y) term-pair as the multiset of dependency paths connecting their co-occurrences in a corpus, which is also regarded as a path-based method.

An alternative approach for detecting *is-a* relation is the distributional methods (Baroni et al., 2012; Roller et al., 2014), using the distributional representation of terms to directly predict relations.

As for the step of taxonomy construction using the extracted *is-a* pairs, most of the approaches do it by incrementally attaching new terms (Snow et al., 2006; Shen et al., 2012; Alfarone and Davis, 2015; Wu et al., 2012). Mao et al. (2018) is the first to present a reinforcement learning based approach, named *TaxoRL*, for this task. For each term pair, its representation in TaxoRL is obtained by the path LSTM encoder, the word embeddings of both terms, and the embeddings of features.

Recently, Dash et al. (2020) argued that *strict partial orders*[2] correspond more directly to *DAGs*. They proposed a neural network architecture, called Strict Partial Order Network (SPON), that enforces asymmetry and transitive properties as soft constraints. Empirically, they showed that such a network produces better results for detecting hyponym-hypernym pairs on a number of datasets for different languages and domains in both supervised and unsupervised settings.

Many graph-based methods such as Kozareva and Hovy (2010) and Luu et al. (2014) regard the task of hypernymy organization as a hypernymy detection problem followed by a graph pruning problem. For the graph pruning task, various graph-theoretic approaches such as *optimal branching algorithm* (Velardi et al., 2013), *Edmond's algorithm* (Karp, 1971) and *Tarjan's algorithm* (Tarjan, 1972) have been used over the years. In addition to these, Wang et al. (2017) mentions several other graph-based taxonomy induction approaches. In contrast, our approach formulates the taxonomy construction task as a DAG generation problem instead of an incremental taxonomy learning (Mao et al., 2018), which differentiates it when compared with the existing methods. In addition, our approach uses the knowledge from existing domains (Bansal et al., 2014; Gan et al., 2016) to build the taxonomies of missing domains.

3 The Graph2Taxo Framework

In this section, we first formulate the problem statement and then introduce our proposed *Graph2Taxo* framework as a solution. We describe the individual components of this framework in detail, along with justifications of *how* and *why* these components come together as a solution.

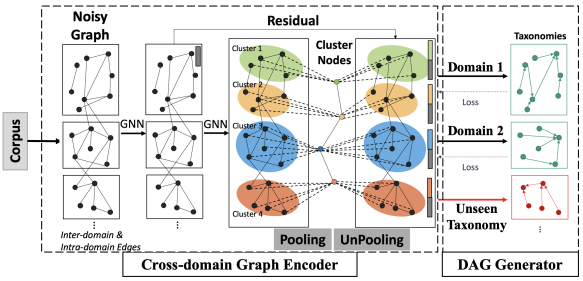

Figure 1: An illustration of our GNN-based cross-domain transfer framework for taxonomy construction.

[2] A binary relation that is transitive, irreflexive and asymmetric.

3.1 Problem Definition

The problem addressed in this paper is, given a list of domain-specific terms from a target *unseen* (aka missing) domain as input, how to construct a taxonomy for that target *unseen* domain. In other words, the problem addressed in this paper is how to organize these terms into a taxonomy.

This problem can be further abstracted out as follows: Given a large *input corpus* and a set of gold taxonomies G_{gold} from some known domains (different from the *target* domain), our task is to learn a model (trained using the corpus and taxonomies of known domains) to construct multiple taxonomies for the *target* unseen domains.

As a solution to the aforementioned problem, we propose a GNN-based cross-domain transfer framework for taxonomy construction (see Figure 1), called *Graph2Taxo* which consists of a cross-domain graph encoder and a DAG generator.

The first step in our proposed approach is to build a *cross-domain noisy graph* as an input to our *Graph2Taxo* model. In this step, we extract candidate *is-a* pairs from a large collection of input corpora that spans multiple domains. To do so, we used the output of Panchenko et al. (2016), which is a combination of standard substring matching and pattern-based approaches. Since such pattern-based approaches are too rigid, the corresponding output not only suffers from recall (i.e., missing *is-a* pairs) but also contains incorrect (i.e., noisy) pairs due to the ambiguity of language and richness in syntactic expression and structure in the input corpora. For example, consider the phrase *"... animals other than dogs such as cats ..."*. As (Wu et al., 2012) noted, pattern-based approaches will extract *(cat is-a dog)* rather than *(cat is-a animal)*.

Based on the noisy *is-a* pairs, we construct a directed graph $G_{input} = (V_{input}, E_{input})$, which is a *cross-domain noisy graph*. Here, V_{input} denotes a set of terms, and $(v_i, v_j) \in E_{input}$ if and only if (v_i, v_j) belongs to the list of extracted noisy *is-a* pairs. The input document collection spans multiple domains, therefore E_{input} not only has intra-domain edges, but also has cross-domain edges (see Figure 1).

Graph2Taxo is a subgraph generation model which uses the large cross-domain noisy graph as the input. Given a list of terms for a target *unseen* domain, it aims to learn a taxonomy structure for the corresponding domain as a DAG. *Graph2Taxo* takes advantage of additional knowledge in the form of previously known gold taxonomies $\{G_{gold,i}, 1 \leq i \leq N_{known}\}$ to train a learning model. During *inference* phase, the model receives a list of terms from the target *unseen* domain and aims to build a taxonomy by using the input terms. Here, N_{known} denotes the number of previously known taxonomies used during the training phase.

This problem of distilling directed acyclic substructures (taxonomies of many domains) using a large cross-domain noisy graph is challenging, because of relatively lower overlap between noisy edges in E_{input} and true edges in the available taxonomies in hand.

The following sections describe our proposed Cross-domain Graph Encoder and the DAG Generator in further detail.

3.2 Cross-domain Graph Encoder

This subsection describes the *Cross-domain Graph Encoder* in Figure 1 for embedding generation. This embedding generation algorithm uses two strategies, namely *Neighborhood aggregation* and *Semantic clustering aggregation*.

3.2.1 Neighborhood Aggregation

This is the first of the two strategies used for embedding generation. Let $A \in \mathbb{R}^{n \times n}$ be the adjacency matrix of the noisy graph G_{input}, where n is the size of V_{input}. Let h_i^l represent the feature representation for the node v_i in the l-th layer and thus $H^l \in \mathbb{R}^{n \times d_l}$ denotes the intermediate representation matrix. The initial matrix H^0 is randomly initialized from a standard normal distribution.

We use the adjacency matrix A and the node representation matrix H^l to iteratively update the representation of a particular node by aggregating representations of its neighbors. This is done by using a GNN. Formally, a GNN layer (Gilmer et al., 2017; Hamilton et al., 2017; Xu et al., 2019) employs the general message-passing architecture which consists of a message propagation function M to get messages from neighbors and a vertex update function U. The message passing works via the following equations,

$$m_v^{l+1} = M(h_u^l) \quad \forall u \in \mathcal{N}(v)$$
$$h_v^{l+1} = U(h_v^l, m_v^{l+1})$$

where $\mathcal{N}(v)$ denotes the neighbors of node v and m is the message. In addition, we use the following

definitions for M and U functions,

$$M(h_u^l) = \sum_{u \in \mathcal{N}(v)} A_{vu} h_u^l, \forall u \in \mathcal{N}(v)$$
$$U(h_v^l, m_v^{l+1}) = \sigma(m_v^{l+1}\Theta^l + h_v^l \Theta^l)$$

where $\Theta^l \in \mathbb{R}^{d_l \times d_{l+1}}$ denotes trainable parameters for layer l and σ represents an activation function.

Let $\tilde{A} = A + I$, here I is the identity matrix, the information aggregation strategy described above can be abstracted out as,

$$H^{l+1} = GNN_l(A, H^l) = \sigma(\tilde{A} H^l \Theta^l)$$

3.2.2 Semantic Clustering Aggregation

This is the second of the two strategies used for embedding generation, which *operates* on the output of the previous step. The learned representations from the previous step are highly likely not to be uniformly distributed in the Euclidean Space, but rather form a bunch of clusters. In this regard, we propose a soft clustering-based pooling-unpooling step, that uses semantic clustering aggregating for learning better model representations. In essence, this step shares the similarity information for any pair of terms in the vocabulary.

Analogous to an auto-encoder, the pooling layer adaptively creates a smaller *cluster* graph comprising of a set of cluster nodes, whose representations are learned based on a trainable cluster assignment matrix. This idea of using an assignment matrix was first proposed by the *DiffPool* (Ying et al., 2018) approach. On the other hand, the unpooling layer decodes the *cluster* graph into the original graph using the same cluster assignment matrix learned in the pooling layer. The learned semantic cluster nodes can be thought of as "bridges" between nodes from the same or different clusters to pass messages.

Mathematically speaking, we learn a soft cluster assignment matrix $S^l \in \mathbb{R}^{n \times n_c}$ at layer l using the GNN model, where n_c is the number of clusters. Each row in S^l corresponds to one of n nodes in layer l and each column corresponds to one of the n_c clusters. As a first step, the pooling layer uses the adjacency matrix A and the node feature matrix H^l to generate a soft cluster assignment matrix as,

$$S^l = softmax(GNN_{l,cluster}(A, H^l)) \quad (1)$$

where the $softmax$ is a row-wise softmax function, $\Theta^l_{cluster} \in \mathbb{R}^{d_l \times n_c}$ denotes all trainable parameters in $GNN_{l,cluster}$.

Since the matrix S^l is calculated based on node embeddings, nodes with similar features and local structure will have similar cluster assignment.

As the final step, the pooling layer generates an adjacency matrix A_c for the *cluster* graph and a new embedding matrix containing cluster node representations H_c^l as follows,

$$H_c^l = (S^l)^T H^l \in \mathbb{R}^{n_c \times d_l}$$
$$A_c = (S^l)^T A S^l \in \mathbb{R}^{n_c \times n_c}$$

A GNN operation is used within the small cluster graph,

$$H_c^{l+1} = GNN_l(A_c, H_c^l) \in \mathbb{R}^{n_c \times d_{l+1}}$$

to further propagate messages from the neighboring clusters. The trainable parameters in GNN_l are $\Theta^l \in \mathbb{R}^{d_l \times d_{l+1}}$.

For passing clustering information to the original graph, the unpooling layer restores the original graph using cluster assignment matrix, as follows,

$$\tilde{H}^{l+1} = S^l H_c^{l+1} \in \mathbb{R}^{n \times d_{l+1}}$$

The output of the pooling-unpooling layer results in the node representations possessing latent cluster information. Finally, we combine the neighborhood aggregation and semantic clustering aggregation strategies via a residual connection, as,

$$H^{l+1} = concate(\tilde{H}^{l+1}, H^l)$$

where $concate$ means concatenate the two matrices. H^{l+1} is the output of this pooling-unpooling step.

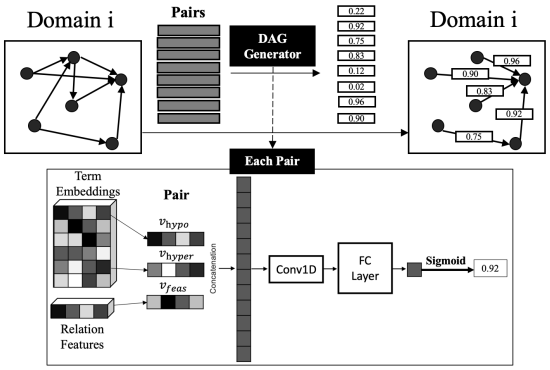

Figure 2: An illustration of DAG generator.

3.3 DAG Generator

The *DAG generator* takes in the noisy graph G_{input} and representations of all the vocabulary terms (output of Section 3.2) as input, encodes *acyclicity* as

a soft-constraint (as described below), and outputs a distribution of edges within G_{input} that encodes the likelihood of true *is-a* relationships. This output distribution is finally used to induce taxonomies, i.e., DAGs of *is-a* relationships.

In each training step, *DAG generator* is applied to one domain (see Figure 2), using a noisy graph G, which is a subgraph from G_{input}, as a training sample and a DAG is generated for that domain. Here let N_t denote the number of $(hypo, hyper)$ pairs belonging to the edge set of G. During the training, we also know label vector $label \in \{0,1\}^{N_t}$ for these N_t pairs, based on whether they belong to the gold known taxonomy.

3.3.1 Edge Prediction

For each edge within the noisy graph G, our DAG generator estimates the probability that the edge represents a valid *hypernymy* relationship. Our model estimates this probability through the use of a convolution operation illustrated in Figure 2.

For each edge $(hypo, hyper)$, in the first step the term embeddings and edge features are concatenated as follows,

$$v_{pair} = concate(v_{hypo}, v_{hyper}, v_{feas})$$

where v_{hypo} and v_{hyper} are the embeddings for $hypo$ and $hyper$ nodes (from Section 3.2) and v_{feas} denotes a feature vector for the edge $(hypo, hyper)$, which includes edge frequency and substring features. The substring features includes *ends with*, *contains*, *prefix match*, *suffix match*, *length of longest common substring (LCS)*, *length difference* and a boolean feature denoting whether *LCS in V_{input} (the set of terms) or not*.

Inspired by ConvE model (Dettmers et al., 2018), a well known convolution based algorithm for link prediction, we apply a 1D convolution operation on v_{pair}. We use a convolution operation since it increases the expressiveness of the DAG Generator through additional interaction between participating embeddings.

For the convolution operation, we make use of C different kernels parameterized by $\{w_c, 1 \leq c \leq C\}$. The 1D convolution operation is then calculated as follows,

$$v_c = [U_c(v_{pair}, 0), ..., U_c(v_{pair}, d_v - 1)] \quad (2)$$

$$U_c(v_{pair}, p) = \sum_{\tau=0}^{K-1} \omega_c(\tau) \hat{v}_{pair}(p + \tau)) \quad (3)$$

where K denotes the kernel width, d_v denotes the size of v_{pair}, p denotes the position to start the kernel operation and the kernel parameters ω_c are trainable. In addition, $\hat{v}_{pair}$ denotes the padded version of v_{pair}, wherein the padding strategy is as follows. If $|K|$ is odd, we pad v_{pair} with $\lfloor K/2 \rfloor$ zeros on both the sides. On the other hand, if $|K|$ is even, we pad $\lfloor K/2 \rfloor - 1$ zeros at the beginning, and $\lfloor K/2 \rfloor$ zeros at the end of v_{pair}. Here, $\lfloor value \rfloor$ returns the floor of $value$.

Each kernel c generates a vector v_c, according to Equation 2. As there are C different kernels, this results in the generation of C different vectors which are then concatenated together to form one vector V_C, i.e. $V_C = concatenate(v_0, v_1, \ldots, v_C)$.

The probability $p_{(hypo,hyper)}$ of a given edge $(hypo, hyper)$ expressing a *hypernymy* relationship can then be estimated using the following scoring function,

$$p_{(hypo,hyper)} = sigmoid(V_C^T W) \quad (4)$$

where W denotes the parameter matrix of a fully connected layer, as illustrated in Figure 2.

Finally, for the loss calculations, we make use of differentiable F1 loss (Huang et al., 2015),

$$\text{Precision} = \frac{\sum_{t=0}^{N_t-1} p_t \times label_t}{\sum_{t=0}^{N_t-1} p_t}$$

$$\text{Recall} = \frac{\sum_{t=0}^{N_t-1} p_t \times label_t}{\sum_{t=0}^{N_t-1} label_t}$$

$$L_{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.3.2 DAG Constraint

The edge prediction step alone does not guarantee that the generated graph is acyclic. Learning DAG from data is an NP-hard problem (Chickering, 1995; Chickering et al., 2004). To this effect, one of the first works that formulate the acyclic structure learning task as a continuous optimization problem was introduced by Zheng et al. (2018).

In that paper, the authors note that the trace of B^k denoted by $tr(B^k)$, for a non-negative adjacency matrix $B \in \mathbb{R}^{n \times n}$ counts the number of length-k cycles in a directed graph. Hence, positive entries within the diagonal of B^k suggests the existence of cycles. Or, in other words, B has no cycle if and only if $\sum_{k=1}^{\infty} \sum_{i=1}^{n} (B^k)_{ii} = 0$.

However, calculating B^k for every value of k, i.e. repeated matrix exponentiation, is impractical and can easily exceed machine precision. To solve this

problem, Zheng et al. (2018) makes use of Taylor Series expansion as $e^B = \sum_{k=0}^{\infty} \frac{B^k}{k!}$, and show that a non-negative matrix B is a DAG iff,

$$\sum_{k=1}^{\infty} \sum_{i=1}^{n} \frac{(B^k)_{ii}}{k!} = tr(e^B) - n = 0$$

To make sure this constraint is useful for an arbitrary weighted matrix with both positive and negative values, a Hadamard product $B = A \circ A$ is used, which leads us to the following theorem.

Theorem 1 *(Zheng et al., 2018) A matrix $A \in \mathbb{R}^{n \times n}$ is a DAG if and only if:*

$$tr(e^{A \circ A}) - n = 0$$

where tr represents the trace of a matrix, $\circ$ represents the Hadamard product and e^B equals matrix exponential of B.

Since the matrix exponential may not be available in all deep learning frameworks, (Yu et al., 2019) propose an alternative constraint that is practically convenient as follows.

Lemma 2 *(Yu et al., 2019) Let $\alpha = c/m > 0$ for some c. For any complex λ, since $(1 + \alpha|\lambda|)^m \leq e^{c|\lambda|}$, the DAG constraint from Theorem 1 can be relaxed and stated as follows,*

$$h(A) = tr\big[(I + \alpha A \circ A)^n\big] - n = 0$$

where α is a hyper-parameter.

Finally, using an augmented Lagrangian approach, we propose the combined loss function,

$$L = L_{F1} + \lambda h(A) + \frac{\rho}{2} h(A)^2$$

where λ and ρ are the hyper-parameters. During the backpropagation, the gradients will be passed back to all domains through the intra-domain and cross-domain edges from G_{input} to update all parameters.

4 Experiments

We evaluate *Graph2Taxo* on *Semeval-2016 Task 13: Taxonomy Extraction Evaluation*[3], otherwise known as *TExEval-2 task* (Bordea et al., 2016). All experiments are implemented in PyTorch. Code is publicly available at https://github.com/IBM/gnn-taxo-construction.

[3]Semeval-2016 Task 13: http://alt.qcri.org/semeval2016/task13

Domain	Source	V	E
Science	WordNet	429	452
	Eurovoc	125	124
	Combined	453	465
Environment	Eurovoc	261	261

Table 1: Dataset statistics for TExEval-2 task obtained from Bordea et al. (2016). The *Vertices*(V) and *Edges*(E) columns represent structural measures of taxonomies for *English* language only.

4.1 Benchmark Datasets

For experiments, we used the English *environment* and the *science* taxonomies within the *TExEval-2* benchmark datasets. These datasets do not come with any training data, but a list of terms and the task is to build a meaningful taxonomy using these terms. The *science* domain terms come from *Wordnet*, *Eurovoc* and a manually constructed taxonomy (henceforth referred to as *combined*), whereas the terms for *environment* domain comes from *Eurovoc* taxonomy only. Table 1 shows the dataset statistics.

We chose to evaluate our proposed approach on *environment* and *science* taxonomies only, because we wanted to compare our approach with the existing state-of-the-art system named *TaxoRL* (Mao et al., 2018) as well as with *TAXI*, the winning system in the *TExEval-2* task. Note that we use the same datasets with *TaxoRL* (Mao et al., 2018) for *TExEval-2* task.

In addition, we used the dataset from Bansal et al. (2014) as gold taxonomies (i.e. sources of additional knowledge), $G_{gold} = \{G_{gold,i}, 1 \leq i \leq N_{known}\}$ that are known apriori. This dataset is a set of medium-sized full-domain taxonomies consisting of bottom-out full subtrees sampled from Wordnet, and contains 761 taxonomies in total.

To test our model for taxonomy prediction (and to remove overlap), we removed any taxonomy from G_{gold} which had term overlap with the set of provided terms for *science* and *environment* domains within TExEval-2 task. Because of this, we get 621 non-overlapping taxonomies in total, partitioned by 80-20 ratio to create *training* and *validation* datasets respectively.

4.2 Experimental Settings

We ran our experiments in two different settings. In each of them, we train on a different noisy input graph (and the same gold taxonomies as mentioned before), and evaluate on the *science* and *environ-*

	Science (Combined)			Science (Eurovoc)			Science (WordNet)			Science (Average)			Environment (Eurovoc)		
Model	P_e	R_e	F_e	P_e	R_e	F_e	P_e	R_e	F_e	P_e	R_e	F_e	P_e	R_e	F_e
Baseline	0.63	0.29	0.39	0.62	0.21	0.31	0.69	0.27	0.38	0.65	0.26	0.36	0.50	0.21	0.30
JUNLP	0.14	0.31	0.19	0.13	**0.36**	0.19	0.21	0.31	0.25	0.16	0.33	0.21	0.13	0.23	0.17
USAAR	0.38	0.26	0.31	0.63	0.15	0.25	**0.82**	0.19	0.31	0.61	0.20	0.29	0.81	0.15	0.25
TAXI	0.39	**0.35**	0.37	0.30	0.33	0.31	0.37	**0.38**	0.38	0.35	0.35	0.35	0.34	0.27	0.30
TaxoRL^A	–	–	–	–	–	–	–	–	–	0.57	0.33	0.42	0.38	0.24	0.29
TaxoRL^B	–	–	–	–	–	–	–	–	–	0.38	**0.38**	0.38	0.32	0.32	0.32
Graph2Taxo^1	**0.91**	0.31	0.46	0.78	0.26	0.39	**0.82**	0.32	**0.46**	**0.84**	0.30	0.44	**0.89**	0.24	0.37
Graph2Taxo^2	0.90	0.33	**0.48**	**0.79**	0.33	**0.46**	0.77	0.32	**0.46**	0.82	0.33	**0.47**	0.67	**0.28**	**0.39**

Table 2: Results on TExEval-2 task: Taxonomy Extraction Evaluation (a.k.a TExEval-2). `First four rows` represent participating systems in the TExEval-2 task, whose performances are taken from Bordea et al. (2016). TaxoRL$^{A/B}$ illustrate the performance of a Reinforcement Learning system by Mao et al. (2018) under the *Partial* and *Full* setting respectively. Graph2Taxo$^{1/2}$ represent our proposed algorithm under both the settings as described in Section 4.2. All results reported above are rounded to 2 decimal places.

ment domains, within TExEval-2 task. In the first setting, we used the same input as *TaxoRL* (Mao et al., 2018) for a fair comparison. This input of *TaxoRL* consists of term pairs and associated dependency path information between them, which has been extracted from three public web-based corpora. For *Graph2Taxo*, we only make use of the term pairs to create a noisy input graph.

In the second setting, we used data[4] provided by *TAXI* (Panchenko et al., 2016), which comprises of a list of candidate *is-a* pairs extracted based on substrings and lexico-syntactic patterns. We used these noisy candidate pairs to create a noisy graph.

A *Graph2Taxo* model is then trained on the noisy graph obtained in each of the two settings. In the *test* phase, all candidate term-pairs for which both terms are present in the *test* vocabulary are scored (between 0 and 1) by the trained *Graph2Taxo* model. A threshold of 0.5 is applied, and the candidate pairs scoring beyond this threshold are accumulated together as the predicted taxonomy G_{pred}. Notice that there are different optimal thresholds for different tasks. We get better performance if we tune the thresholds. However, we chose a harder task and proved our model has better performance than others even we simply use 0.5 as the threshold. In addition, We specify the hyper-parameter ranges for our experiments: learning rate $\{0.01, 0.005, 0.001\}$, number of kernels $\{5, 10, 20\}$ and number of clusters $\{10, 30, 50, 100\}$. Finally, Adam optimizer (Kingma and Ba, 2015) is used for all experiments.

Evaluation Metrics. Given a gold taxonomy G_{gold} (as part of the *TExEval-2* benchmark dataset) and a predicted taxonomy G_{pred} (by our proposed *Graph2Taxo* approach), we evaluate G_{pred} using Edge Precision, Edge Recall and F-score measures as defined in Bordea et al. (2016).

[4]Data is available at `http://panchenko.me/data/joint/taxi/res/resources.tgz`

4.3 Hyper-parameters

We use the following hyper-parameter configuration for training the model. We set *dropout* to 0.3, *number of kernels* C to 10, *kernel size* K to 5, *learning rate* to 0.001 and initial *embedding size* to 300. For the loss function, we use the $\lambda = 1.0$ and $\rho = 0.5$. In addition, *number of clusters* n_c is set to 50 for all our experiments. In the scenario wherein the input resource comes from TAXI, only hyponym-hypernym candidate pairs observed more than 10 times are used to create a noisy graph. Also, we use one pooling and one unpooling layer for our experiments.

We use dropouts in two places, one at the end of the cross-domain encoder module, and the other after the Conv1D operation. Our models are trained using NVIDIA Tesla P100 GPUs.

4.4 Results and Discussions

Table 2 shows the results on the *TExEval-2 task* Evaluation on *science* and *environment* domains. The first row represents a string-based baseline method (Bordea et al., 2016), that exploits term compositionality to hierarchically relate terms. For example, it extracts pairs such as (*Statistics Department*, *Department*) from the provided Wikipedia corpus, and utilizes aforementioned technique to construct taxonomy.

The next three rows in Table 2, namely, TAXI, JUNLP and USAAR are some of the top perform-

Model	Science (Combined)			Science (Eurovoc)			Science (WordNet)			Environment (Eurovoc)		
	P_e	R_e	F_e	P_e	R_e	F_e	P_e	R_e	F_e	P_e	R_e	F_e
Graph2Taxo(2GNN+SC+Res)	0.90	0.33	**0.48**	0.79	0.33	**0.46**	0.77	0.32	**0.46**	0.67	0.28	**0.39**
Graph2Taxo(2GNN+Res)	0.92	0.32	0.48	0.83	0.29	0.43	0.80	0.31	0.45	0.73	0.26	0.38
Graph2Taxo(2GNN)	0.90	0.33	0.48	0.81	0.29	0.42	0.81	0.31	0.45	0.74	0.25	0.37
Graph2Taxo(NoConstraint)	0.92	0.32	0.48	0.81	0.28	0.41	0.83	0.31	0.45	0.76	0.25	0.37
Graph2Taxo(Without Feas)	0.82	0.33	0.47	0.73	0.27	0.39	0.70	0.33	0.45	0.61	0.23	0.33
Graph2Taxo(AddEmbeddings)	0.90	0.33	**0.48**	0.80	0.33	**0.47**	0.77	0.32	**0.46**	0.71	0.28	**0.40**

Table 3: Ablation tests reporting the *Precision*, *Recall* and *F-score*, across Science and Environment domains. The `first block` of values reports results by *ablating* each layer utilized within *Graph2Taxo* model. In the `second block`, we demonstrate that addition of constraint does in fact improve performance. In the `third block`, we illustrate that the importance of features v_{feas} for improving performance. The `final block` uses pretrained fastText embeddings to initialize our *Graph2Taxo* model, and then *fine tunes* based on our training data. All results reported above are rounded off to 2 decimal places.

ing systems that participated in the TExEval-2 task. Furthermore, TaxoRL^{A,B} illustrates the performance of a Reinforcement Learning system by under the *Partial induction* and *Full induction* settings respectively (Mao et al., 2018). Since Mao et al. (2018) has shown that it outperforms other methods such as Gupta et al. (2017); Bansal et al. (2014), we only compare the results of our proposed *Graph2Taxo* approach against the state-of-the-art system TaxoRL.

Finally, Graph2Taxo^1 and Graph2Taxo^2 depict the results of our proposed algorithm under both aforementioned settings, i.e. using the input resources of TaxoRL in the first scenario, and using the resources of TAXI in the second scenario. In each of these settings, we find that the overall precision of our proposed *Graph2Taxo* approach is far better than all the other existing approaches, demonstrating the strong ability of *Graph2Taxo* to find true relations. Meanwhile, the recall of our proposed *Graph2Taxo* approach is comparable to that of the existing state-of-the-art approaches. Combining the precision and recall metrics, we observe that *Graph2Taxo* outperforms existing state-of-the-art approaches on the F-score, by a significant margin. For example, for the Science (Average) domain, Graph2Taxo^2 improves over TaxoRL's F-score by 5%. For the Environment (Eurovoc) domain, our model improves TaxoRL's F-score by 7% on the TExEval-2 task.

Besides, our proposed model has high scalability. For example, the GNN method has been trained for a large graph, including about 1 million nodes (Kipf and Welling, 2017). Besides, the GNN part can be replaced by any improved GNN methods (Hamilton et al., 2017; Gao et al., 2018) designed for large-scale graphs.

Ablation Tests. Table 3 shows the results of proposed *Graph2Taxo* in the second setting for the ablation experiments (divided into *four* blocks), which indicates the contribution of each layer used in our *Graph2Taxo* model. In Table 3, all the experiments are run three times, and the average values of the three runs are reported. Furthermore, in Figure 3, we randomly choose Science (Eurovoc) domain as the one to report the *error-bars* (corresponding to the standard-deviation values) for our experiments.

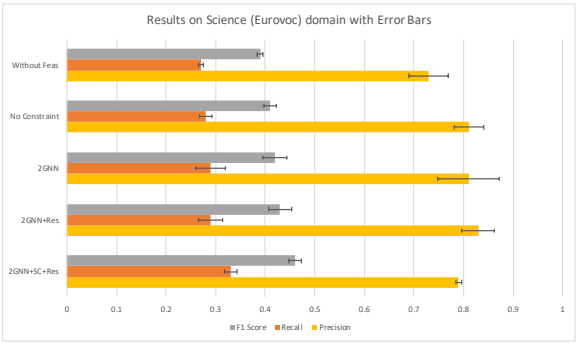

Figure 3: Results on Science (Eurovoc) domain: The average *Precision*, *Recall* and *F-score* values and their standard error values. It is clear that addition of Residual Layer and SC Layer *lowers* the variance of the results.

The first block of values in Table 3 illustrates results by *ablating* layers from within our *Graph2Taxo* model. Comparing the first two rows, it's evident that adding a *Semantic Cluster* (SC) layer improves recall at the cost of precision, however improving the overall F-score. This improve-

ment is clearly seen for the Science (Eurovoc) domain, wherein we have an increase of 3%.

In the second block, we show that the addition of constraints improves performance. Row 4 represents a *Graph2Taxo* i.e. 2GNN+SC+Res setup, but without any constraint. Adding the DAG Constraint (Row 1) to this yields can get a better F-score. Specifically, we observe a major increase of +5% F1 for the Science (Eurovoc) domain.

In the third block, we remove the features v_{feas} as mentioned in section 3.3.1. The results, i.e. row 5 in Table 3 shows that these features are critical in improving the performance of our proposed system on both Science (Eurovoc) and Environment (Eurovoc) domains. Note that these features denoted as v_{feas} are not a novelty of our proposed method, but rather have been used by existing state-of-the-art approaches.

Finally, we study the effect of initializing our model using pre-trained embeddings, rather than initializing at random. Specifically, we initialize the input matrix H_0 of our *Graph2Taxo* model with pre-trained fastText[5] embeddings. Our model using fastText embeddings improves upon Row 1 by a margin of 4% in precision values for the Environment (Eurovoc) domain, but unfortunately has no significant effect on the F-score. Hence, we have not used pre-trained embeddings in reporting the results in Table 2.

We provide an illustration of the output of the *Graph2Taxo* model in Figure 4, for the Environment domain.The generated taxonomy in this example contains multiple trees, which serve the purpose of generating taxonomical classifications. As future work, we plan to figure out different strategies to connect the subtrees into a large graph for better DAG generation.

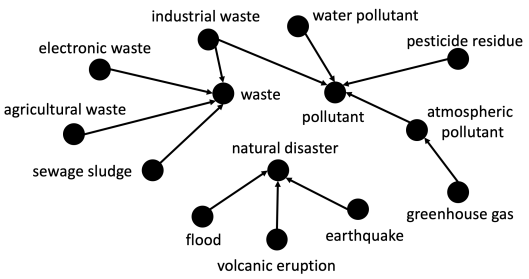

Figure 4: A simple example of the taxonomy generated by *Graph2Taxo* in the environment domain.

[5] https://fasttext.cc

5 Conclusion

We have introduced a GNN-based cross-domain knowledge transfer framework *Graph2Taxo*, which makes use of a cross-domain graph structure, in conjunction with an acyclicity constraint-based DAG learning for taxonomy construction. Furthermore, our proposed model encodes acyclicity as a *soft* constraint and shows that the overall model outperforms state of the art.

In the future, we would like to figure out different strategies to merge individual gains, obtained by separate application of the DAG constraint, into a setup that can take the best of both precision and recall improvements, and put forth a better performing system. We also plan on looking into strategies to improve recall of the constructed taxonomy.

Acknowledgments

The authors would like to thank Dr. Jie Chen from MIT-IBM Watson AI Lab and Prof. Jinbo Bi from the University of Connecticut for in-depth discussions on model construction.

References

Daniele Alfarone and Jesse Davis. 2015. Unsupervised learning of an IS-A taxonomy from a limited domain-specific corpus. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1434–1441. AAAI Press.

Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1041–1051, Baltimore, Maryland. Association for Computational Linguistics.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France. Association for Computational Linguistics.

Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *CoRR*, abs/1706.02263.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego,

California. Association for Computational Linguistics.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

David Maxwell Chickering. 1995. Learning bayesian networks is np-complete. In *Learning from Data - Fifth International Workshop on Artificial Intelligence and Statistics, AISTATS 1995, Key West, Florida, USA, January, 1995. Proceedings*, pages 121–130. Springer.

David Maxwell Chickering, David Heckerman, and Christopher Meek. 2004. Large-sample learning of bayesian networks is np-hard. *J. Mach. Learn. Res.*, 5:1287–1330.

Sarthak Dash, Md Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, and Nicolas Rodolfo Fauceglia. 2020. Hypernym detection using strict partial order networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, USA, February 7 - February 12, 2020*. AAAI Press.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Gerhard Friedrich and Markus Zanker. 2011. A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98.

Chuang Gan, Yi Yang, Linchao Zhu, Deli Zhao, and Yueting Zhuang. 2016. Recognizing an action using its name: A knowledge-based approach. *Int. J. Comput. Vis.*, 120(1):61–77.

Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1416–1424. ACM.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 1263–1272. PMLR.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE.

Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. 2017. Taxonomy induction using hypernym subsequences. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems, NeurIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1024–1034.

Sanda M. Harabagiu, Steven J. Maiorano, and Marius Pasca. 2003. Open-domain textual question answering techniques. *Nat. Lang. Eng.*, 9(3):231–267.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992*, pages 539–545.

Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2017. Understand short texts by harvesting and analyzing semantic knowledge. *IEEE Trans. Knowl. Data Eng.*, 29(3):499–512.

Hao Huang, Haihua Xu, Xianhui Wang, and Wushour Silamu. 2015. Maximum f1-score discriminative training criterion for automatic mispronunciation detection. *IEEE ACM Trans. Audio Speech Lang. Process.*, 23(4):787–797.

Richard M. Karp. 1971. A simple derivation of edmonds' algorithm for optimum branchings. *Networks*, 1(3):265–272.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N. Kipf and Max Welling. 2017. 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.

Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1110–1118, Cambridge, MA. Association for Computational Linguistics.

Anh Tuan Luu, Jung-jae Kim, and See Kiong Ng. 2014. Taxonomy construction using syntactic contextual evidence. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 810–819, Doha, Qatar. Association for Computational Linguistics.

Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-end reinforcement learning for automatic taxonomy induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2462–2472, Melbourne, Australia. Association for Computational Linguistics.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society.

Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. TAXI at SemEval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California. Association for Computational Linguistics.

Alan Ritter, Stephen Soderland, and Oren Etzioni. 2009. What is this, anyway: Automatic hypernym discovery. In *Learning by Reading and Learning to Read, Papers from the 2009 AAAI Spring Symposium, Technical Report SS-09-07, Stanford, California, USA, March 23-25, 2009*, pages 88–93. AAAI.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3060–3067. AAAI Press.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. A graph-based approach for ontology population with named entities. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 345–354. ACM.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia. Association for Computational Linguistics.

Alessandro Sperduti and Antonina Starita. 1997. Supervised neural networks for the classification of structures. *IEEE Trans. Neural Networks*, 8(3):714–735.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. YAGO: A large ontology from wikipedia and wordnet. *J. Web Semant.*

Robert Endre Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 481–492. ACM.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4805–4815.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97, pages 7154–7163. PMLR.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. 2018. Dags with NO TEARS: continuous optimization for structure learning. In *Advances in Neural Information Processing Systems, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 9492–9503.