Learning Latent Forests for Medical Relation Extraction

Zhijiang Guo[1*], **Guoshun Nan**[1*], **Wei Lu**[1] and **Shay B. Cohen**[2]
[1]StatNLP Research Group, Singapore University of Technology and Design
[2]ILCC, School of Informatics, University of Edinburgh
zhijiang_guo@mymail.sutd.edu.sg, guoshun_nan@sutd.edu.sg, luwei@sutd.edu.sg, scohen@inf.ed.ac.uk

Abstract

The goal of medical relation extraction is to detect relations among entities, such as genes, mutations and drugs in medical texts. Dependency tree structures have been proven useful for this task. Existing approaches to such relation extraction leverage off-the-shelf dependency parsers to obtain a syntactic tree or forest for the text. However, for the medical domain, low parsing accuracy may lead to error propagation downstream the relation extraction pipeline. In this work, we propose a novel model which treats the dependency structure as a latent variable and induces it from the unstructured text in an end-to-end fashion. Our model can be understood as composing task-specific dependency forests that capture non-local interactions for better relation extraction. Extensive results on four datasets show that our model is able to significantly outperform state-of-the-art systems without relying on any direct tree supervision or pre-training.

1 Introduction

With a significant growth in the medical literature, researchers in the area are still required to track it mostly manually. This is an opportunity to automate some of this tracking process, and indeed Natural Language Processing (NLP) techniques have been used to automatically extract knowledge from medical articles. Among these techniques, relation extraction plays a significant role as it facilitates the detection of relations among entities in the medical literature [Peng *et al.*, 2017; Song *et al.*, 2019]. For example in Figure 1, the sub-clause "human ***phenylalanine hydroxylase*** catalytic domain with bound ***catechol*** inhibitors" drawn from the CPR dataset [Krallinger *et al.*, 2017] contains two entities, namely ***phenylalanine hydroxylase*** and ***catechol***. There is a "down regulator" relation between these two entities, denoted as "CPR:4".

Dependency structures are often used for relation extraction as they are able to capture non-local syntactic relations that are only implicit in the surface form alone [Zhang *et al.*, 2018]. In the medical domain, early efforts leverage graph LSTM [Peng *et al.*, 2017] or graph neural networks [Song *et al.*, 2018; Guo *et al.*, 2019a] to encode the 1-best dependency tree. However, dependency parsing accuracy is relatively low in the medical domain. Figure 1 shows the 1-best dependency tree obtained by the Stanford CoreNLP [Manning *et al.*, 2014], where the dependency tree contains an error. In particular, the entity phrase ***phenylalanine hydroxylase*** is broken since the word ***hydroxylase*** is mistakenly considered as the main verb. In order to mitigate the error propagation when incorporating the dependency structure, Song *et al.* [2019] construct a dependency forest by adding additional edges with high confidence scores given by a dependency parser trained on the news domain or merging the K-bests trees [Eisner, 1996] by combining identical dependency edges. Lifeng *et al.* [2020] jointly train a pre-trained dependency parser [Dozat and Manning, 2017] and a relation extraction model. The dependency forest generated by the parser is a 3-dimensional tensor, with each position representing the conditional probability of one word modifying another word with a relation, which encodes all possible dependency relations with the confidence scores.

Unlike previous research efforts that rely on dependency parsers trained on newswire text, our proposed model treats the dependency parse as a latent variable and induces it in an end-to-end fashion. We build our model based on the mechanism of structured attention [Kim *et al.*, 2017;

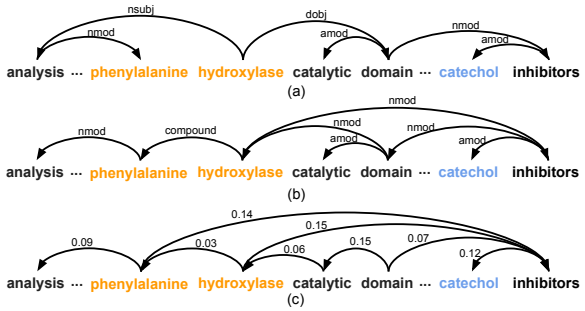

Figure 1: (a) 1-best dependency tree; (b) manually labeled gold tree; (c) a dependency forest generated by the LF-GCN model, where the number for each arc indicates the weight of the edge in the forest. We omit some edges for simplicity. Phrase ***phenylalanine hydroxylase*** and ***catecho*** are gene and drug entity, respectively.

**Equally Contributed. Work done while Zhijiang at University of Edinburgh.

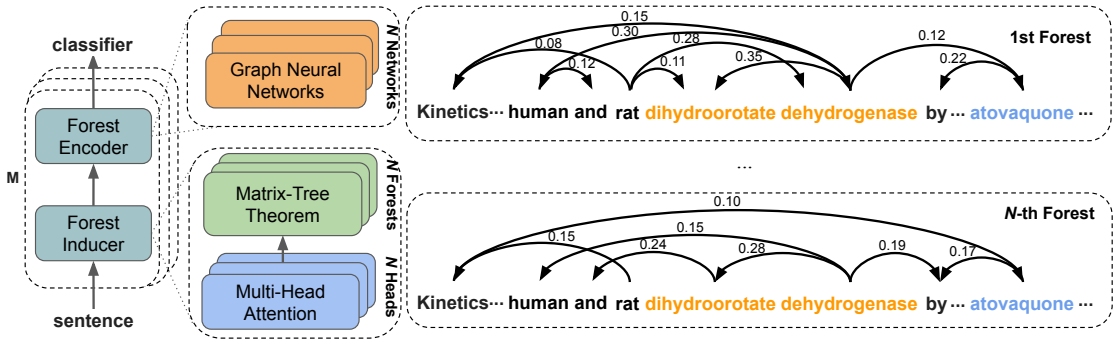

Figure 2: Overview of our proposed LF-GCN model. It is composed of M identical blocks and each block has two components—forest inducer and forest encoder. The forest inducer consists of two sub-modules, where the first sub-module computes N attention matrices based on the multi-head attention, and the second sub-module takes the N attention matrices as inputs to obtain N dependency forests based on the Matrix-Tree Theorem. Then, the forest encoder uses graph neural networks to encode the induced forests.

Liu and Lapata, 2018]. Using a variant of the Matrix-Tree Theorem [Tutte, 1984; Koo *et al.*, 2007], our model is able to generate task-specific non-projective dependency structures for capturing non-local interactions between entities without recourse to any pre-trained parsers or tree supervision. We further construct multiple forests by projecting the representations of nodes to different representation subspaces, allowing an induction of a more informative latent structure for better relation prediction. We name our proposed model as LF-GCN, where LF is the abbreviation of latent forests.

Experiments show that our LF-GCN model is able to achieve better performance on various relation extraction tasks. For the sentence-level tasks, our model surpasses the current stat-of-the-art models on the CPR dataset [Krallinger *et al.*, 2017] and the PGR dataset [Sousa *et al.*, 2019]) by 3.2% and 2.6% in terms of $F1$ score, respectively. For the cross-sentence tasks [Peng *et al.*, 2017], our model is also consistently better than others, showing its effectiveness on long medical text. We release our code at https://github.com/Cartus/Latent-Forests.

2 Model

Here we present the proposed model as shown in Figure 2.

2.1 Forest Inducer

Existing approaches leverage a dependency parser trained on newswire text [Song *et al.*, 2019] or a fine-tuned parser for the medical domain [Lifeng *et al.*, 2020] to generate a dependency forest, which is a fully-connected weighted graph. Unlike previous efforts, we treat the forest as a latent variable, which can be learned from a targeting dataset in an end-to-end manner. Inspired by Kim *et al.* [2017] and Liu and Lapata [2018], we use a variant of Kirchhoff's Matrix-Tree Theorem (MTT) [Tutte, 1984; Koo *et al.*, 2007; Smith and Smith, 2007] to induce the latent structure of an input sentence. Such latent structure can be viewed as multiple full dependency forests, which efficiently represent all possible dependency trees within a compact and dense structure.

Given a sentence $\mathbf{s} = w_1, ..., w_n$, where w_i represents the i-th word, we define a graph $\mathbf{G}$ on n nodes, where each node refers to the word in the $\mathbf{s}$, and the edge (i, j) refers to the dependency between the i-th word (head) to the j-th node (modifier). We denote the contextual output of the sentence $\mathbf{h} \in \mathbb{R}^{n\times d}$ as $\mathbf{h} = \mathbf{h}_1, ..., \mathbf{h}_n$, where $\mathbf{h}_i \in \mathbb{R}^d$ represents the hidden state of the i-th word with a d dimension. We use the bidirectional LSTM to obtain contextual representations of the sentence.

For the graph $\mathbf{G}$, MTT takes the edge scores and root scores as inputs then generates a latent forest by computing the marginal probabilities for each edge. Given the input $\mathbf{h}$ and a weight vector $\boldsymbol{\theta}^k \in \mathbb{R}^m$ of dependencies, where $m \in \mathbb{R}$ represents the number of dependencies for the k-th ($k \in [1, N]$) latent structure, inducing the k-th latent forest for the input $\mathbf{h}$ amounts to finding the latent variables $z_{ij}^k(\mathbf{h}, \boldsymbol{\theta}^k)$ for all edges that satisfy $i \neq j$ and root node whose index equals to 0.

The k-th latent forest induced by MTT contains many non-projective dependency trees, which are denoted by $\mathbf{T}^k$. Let $P(\mathbf{y}|\mathbf{h}; \boldsymbol{\theta}^k)$ denote the conditional probability of a tree $\mathbf{y}$ over $\mathbf{T}^k$. Following the formulation by Koo *et al.* [2007], the marginal probability of a dependency edge from i-th word to j-th word for the k-th forest can be expressed as :

$$P(z_{ij}^k = 1) = \sum_{\mathbf{y}\in\mathbf{T}^k:(i,j)\in\mathbf{y}} P(\mathbf{y}|\mathbf{h}; \boldsymbol{\theta}^k) \quad (1)$$

We derive two steps to obtain the marginal probabilities expressed in Equation (1).

Computing Attention Scores

Inspired by Vaswani *et al.* [2017], we calculate the edge scores by the multi-head attention mechanism, which allows the model to jointly attend to information from different representation subspaces. N attention matrices will be fed into the MTT to obtain N latent forests in order to capture different dependencies in different representation subspaces. The attention matrix for the k-th head is calculated by a function of the query $\mathbf{Q}$ with the corresponding key $\mathbf{K}$. Here $\mathbf{Q}$ and $\mathbf{K}$ are both equal to the contextual representation $\mathbf{h}$. We project $\mathbf{Q}$ and $\mathbf{K}$ to different representation subspaces in order to generate N attention matrices for calculating N latent forests. Formally, the k-th forest $\mathbf{S}^k$ is given by:

$$\mathbf{S}^k = \text{softmax}(\frac{\mathbf{Q}\mathbf{W}^Q \times (\mathbf{K}\mathbf{W}^K)^T}{\sqrt{d}}) \quad (2)$$

where $\mathbf{W}^Q \in \mathbb{R}^{d\times d}$ and $\mathbf{W}^K \in \mathbb{R}^{d\times d}$ are parameters for projections. $\mathbf{S}^k_{ij}$ denotes the normalized attention score between the i-th token and the j-th token with $\mathbf{h}_i$ and $\mathbf{h}_j$. Then, we compute the root score $\mathbf{r}^k_i$, which represents the normalized probability of the i-th node to be selected as the root node of the k-th forest:

$$\mathbf{r}^k_i = \mathbf{W}^k_r \mathbf{h}_i \tag{3}$$

where $\mathbf{W}^k_r \in \mathbb{R}^{1\times d}$ is the weight for the projection.

Imposing Structural Constraint

Following Koo *et al.* [2007] and Smith and Smith [2007], we calculate the marginal probability of each dependency edge of the k-th latent forest, by injecting a structural bias on $\mathbf{S}^k$. We assign non-negative weights $\mathbf{P}^k \in \mathbb{R}^{n\times n}$ to the edges as:

$$\mathbf{P}^k_{ij} = \begin{cases} 0 & \text{if } i = j \\ \exp\left(\mathbf{S}^k_{ij}\right) & \text{otherwise} \end{cases} \tag{4}$$

where $\mathbf{P}^k_{ij}$ is the weight of the edge between the i-th and the j-th node. We define a Laplacian matrix $\mathbf{L}^k \in \mathbb{R}^{n\times n}$ of $\mathbf{G}$ in Equation (5), and its variant $\hat{\mathbf{L}}^k \in \mathbb{R}^{n\times n}$ in Equation (6).

$$\mathbf{L}^k_{ij} = \begin{cases} \sum_{i'=1}^{n} \mathbf{P}^k_{i'j} & \text{if } i = j \\ -\mathbf{P}^k_{ij} & \text{otherwise} \end{cases} \tag{5}$$

$$\hat{\mathbf{L}}^k_{ij} = \begin{cases} \exp(\mathbf{r}^k_i) & \text{if } i = 1 \\ \mathbf{L}^k_{ij} & \text{if } i > 1 \end{cases} \tag{6}$$

We use $\mathbf{A}^k_{ij}$ to denote the marginal probability of the dependency edge between the i-th node and the j-th node. Then, $\mathbf{A}^k_{ij}$ can be derived based on:

$$\begin{aligned} \mathbf{A}^k(z_{ij} = 1) = (1 - \delta_{1,j})\mathbf{P}^k_{ij}[(\hat{\mathbf{L}}^k)^{-1}]_{ij} \\ -(1 - \delta_{i,1})\mathbf{P}^k_{ij}[(\hat{\mathbf{L}}^k)^{-1}]_{ji} \end{aligned} \tag{7}$$

where δ is Kronecker delta and $\mathbf{A}^k \in \mathbb{R}^{n\times n}$ can be interpreted as a weighted adjacency matrix for the k-th forest. Now we can feed $\mathbf{A} \in \mathbb{R}^{N\times n\times n}$ into the forest encoder to update the representations of nodes in the latent structure.

Adaptive Pruning Strategy

Prior dependency-based model [Zhang *et al.*, 2018] also proposes rule-based method to prune a dependency tree to further improve relation classification performance. However, the weighted adjacency matrix $\mathbf{A}$ is derived based on a continuous relaxation, and such induced structures are not discrete, so the existing rule-based pruning methods are not applicable. Instead, we use α-entmax [Blondel *et al.*, 2018; Correia *et al.*, 2019] to remove irrelevant information by imposing the sparsity constraints on the adjacency matrix. α-entmax is able to assign exactly zero weights. Therefore, an unnecessary path in the induced latent forests will not be considered by the latent forest encoder. The expression of our soft pruning strategy is described as:

$$\mathbf{A}^k = \alpha\text{-entmax}(\mathbf{A}^k) \tag{8}$$

where α is a parameter to control the sparsity of each adjacency matrix. When α=2, the entmax recovers the sparsemax mapping [Martins and Astudillo, 2016]. When α=1, it recovers the softmax mapping. Correia *et al.* [2019] propose an exact algorithm to learn α automatically. Here we apply k α-entmax to k latent forests, which enables the model to develop different pruning strategies for different latent forest.

2.2 Forest Encoder

Given N latent forests generated by the forest inducer, we encode them by using densely-connected graph convolutional networks [Kipf and Welling, 2017; Guo *et al.*, 2019b]. Formally, given the k-th latent forest, which is represented by the adjacency matrix $\mathbf{A}^k$, the convolution computation for the i-th node at the l-th layer, which takes the representation $\mathbf{h}^{l-1}_i$ from previous layer as input and outputs the updated representations $\mathbf{h}^l_i$, can be defined as:

$$\mathbf{h}^l_{k_i} = \sigma(\sum_{j=1}^{n} \mathbf{A}^k_{ij} \mathbf{W}^l_k \mathbf{h}^{l-1}_i + \mathbf{b}^l_k) \tag{9}$$

where $\mathbf{W}^l_k$ and $\mathbf{b}^l_k$ are the weight matrix and bias vector for the k-th latent forest in the l-th layer, respectively. σ is an activation function. $\mathbf{h}^0_i \in \mathbb{R}^d$ is the initial contextual representation of the i-th node. Then a linear combination layer is used to integrate representations of the N latent forests:

$$\mathbf{h}_{comb} = \mathbf{W}_{comb}\mathbf{h}_{out} + \mathbf{b}_{comb} \tag{10}$$

where $\mathbf{h}_{out}$ is the output by concatenating outputs from N separated convolutional layers, *i.e.,* $\mathbf{h}_{out} = [\mathbf{h}^{(1)}; ...; \mathbf{h}^{(N)}] \in \mathbb{R}^{d\times N}$. $\mathbf{W}_{comb} \in \mathbb{R}^{(d\times N)\times d}$ is a weight matrix and $\mathbf{b}_{comb}$ is a bias vector for the linear transformation.

3 Experiments

3.1 Data

We evaluate our LF-GCN model with four datasets on two tasks, namely cross-sentence n-ary relation extraction and sentence-level relation extraction.

For cross-sentence n-ary relation extraction, we use two datasets generated by Peng *et al.* [2017], which has 6,987 ternary relation instances and 6,087 binary relation instances extracted from PubMed. The relation label contains five categories, *e.g.,* "sensitivity", "resistance" and "none". Following Song *et al.* [2018], we define two sub-tasks for a more detailed evaluation, *i.e.,* binary-class n-ary relation extraction and multi-class n-ary relation extraction. For binary-class extraction, we cluster the four relation classes as "Yes" and treat the label "None" as "No".

For sentence-level relation extraction, we follow the experimental settings by Lifeng *et al.* [2020] on BioCreative Vi CPR (CPR) [Krallinger *et al.*, 2017] and Phenotype-Gene relation (PGR) [Sousa *et al.*, 2019]. The CPR dataset contains the relations between chemical components and human proteins. It has 16,107 training, 10,030 development and 14,269 testing instances, with five regular relations, such as "CPR:3", "CPR:9" and "None" relation. PGR introduces the relations between human phenotypes with human genes, and it contains 11,780 training instances and 219 test instances, with binary class "Yes" and "No" on relation labels.

Syntax Type	Model	Binary-class				Multi-class	
		Ternary		Binary		Ternary	Binary
		Single	Cross	Single	Cross	Cross	Cross
Full Tree	DAG LSTM [Peng *et al.*, 2017]	77.9	80.7	74.3	76.5	-	-
	GRN [Song *et al.*, 2018]	80.3	83.2	83.5	83.6	71.7	71.7
	GCN [Zhang *et al.*, 2018]	84.3	84.8	84.2	83.6	77.5	74.3
Pruned Tree	GCN [Zhang *et al.*, 2018]	85.8	85.8	83.8	83.7	78.1	73.6
Forest	AGGCN [Guo *et al.*, 2019a]	87.1	87.0	85.2	85.6	79.7	77.4
	LF-GCN (Ours)	**88.0**	**88.4**	**86.7**	**87.1**	**81.5**	**79.3**

Table 1: Average test accuracies on the [Peng *et al.*, 2017] dataset for binary-class n-ary relation extraction and multi-class n-ary relation extraction. "Ternary" and "Binary" denote drug-gene-mutation tuple and drug-mutation pair, respectively. `Single` and `Cross` indicate that the entities of relations reside in single sentence or multiple sentences, respectively.

We also use SemEval-2010 Task 8 [Hendrickx *et al.*, 2009] dataset from the news domain to evaluate the generalization capability of our model. It has 10,717 instances with 9 types of relations and a special "Other" relation.

3.2 Settings

We tune the hyper-parameters according to the results on the development sets. For the cross-sentence n-ary relation extraction task, we use the same data splits as Song *et al.* [2018], stochastic gradient descent optimizer with a 0.9 decay rate, and 300-dimensional GloVe. The hidden size of both BiLSTM and GCNs are set as 300.

For cross-sentence task, we report the test accuracy averaged over five cross validation folds [Song *et al.*, 2018] for the cross-sentence n-ary task. For the sentence-level task, we report the $F1$ scores [Lifeng *et al.*, 2020].

3.3 Results on Cross-Sentence n-ary Task

To verify the effectiveness of the model in predicting intersentential relations, we compare LF-GCN against state-of-the-art systems on the cross-sentence n-ary relation extraction task [Peng *et al.*, 2017], as shown in Table 1. Previous systems using the same syntax type are grouped together.

Full Tree: models use the 1-best dependency graph constructed by connecting roots of dependency trees correspond to the input sentences. DAG LSTM encodes the graph by using graph-structure LSTM, while GRN and GCN encode it using graph recurrent networks and graph convolutional networks, respectively.

Pruned Tree: model with pruned trees as inputs, whose dependency nodes and edges are removed based on rules [Zhang *et al.*, 2018]. GCN is used to encode the resulted structure.

Forest: model constructs multiple fully-connected weighted graphs based on the multi-head attention [Vaswani *et al.*, 2017], where the graph can be viewed as a dependency forest.

Models with pruned trees as inputs tend to achieve higher results than models with full trees. Intuitively, longer sentences in the cross-sentence task correspond to more complex dependency structures. Using an out-of-domain parser may introduce more noise to the model. Removing the irrelevant nodes and edges of the parse tree enables the model to perform better prediction. However, a rule-based pruning strategy [Zhang *et al.*, 2018] may not yield optimal performance. In contrast, LF-GCN induces the dependency structure automatically, which can be viewed as a soft pruning strategy learned from the data. Compared to GCN models, our model obtains 2.2 and 2.6 points improvement over the best performing model with pruned trees for the ternary relation extraction. For binary relation extraction, our model achieves accuracy of 86.7 and 87.1 under `Single` and `Cross` settings, respectively, which surpasses the state-of-the-art AGGCN model. We believe that our LF-GCN is able to distill relevant information and filter out noises from the representation for better prediction.

Syntax Type	Model	$F1$
None	Random-DDCNN [Lifeng *et al.*, 2020]	45.4
	Att-GRU [Liu *et al.*, 2017]	49.5
	Bran [Verga *et al.*, 2018]	50.8
Tree	GCN [Zhang *et al.*, 2018]	52.2*
	Tree-DDCNN [Lifeng *et al.*, 2020]	50.3
	Tree-GRN [Lifeng *et al.*, 2020]	51.4
Forest	Edgewise-GRN [Song *et al.*, 2019]	53.4
	KBest-GRN [Song *et al.*, 2019]	52.4
	AGGCN [Guo *et al.*, 2019a]	56.7*
	ForestFT-DDCNN [Lifeng *et al.*, 2020]	55.7
	LF-GCN (Ours)	**58.9**

Table 2: Test results on the CPR dataset. Results on AGGCN and GCN are reproduced based on their released implementation.

3.4 Results on Sentence-Level Task

To examine LF-GCN on sentence-level task, we compare LF-GCN with state-of-the-art models on two medical datasets, *i.e.,* CPR [Krallinger *et al.*, 2017] and PGR [Sousa *et al.*, 2019]. These systems are grouped in three types based on the syntactic structure used as shown in Table 2 and Table 3. Results labeled with "$*$" are obtained based on the retrained models using their released implementations, as we don't have published results for the dataset.

None: models do not use any pre-trained parsers. Random-DDCNN uses a randomly initialized parser [Dozat and Manning, 2017] fine-tuned by the relation prediction loss. Att-GRU stacks a self-attention layer on top of the gated recurrent units and Bran relies on a bi-affine self-attention model to capture the interactions in the sentence. BioBERT is a pre-trained language representation model for biomedical text.

Tree: models use the 1-best dependency tree. Full trees are encoded by GCN, GRN and DDCNN, respectively. BO-LSTM only encodes words on the shortest dependency path.

Syntax Type	Model	$F1$
None	BioBERT [Lee *et al.*, 2019]	67.2
Tree	BO-LSTM [Lamurias *et al.*, 2019]	52.3
	GCN [Zhang *et al.*, 2018]	81.3*
	Tree-GRN [Lifeng *et al.*, 2020]	78.9
Forest	Edgewise-GRN [Song *et al.*, 2019]	83.6
	KBest-GRN [Song *et al.*, 2019]	85.7
	AGGCN [Guo *et al.*, 2019a]	89.3*
	ForestFT-DDCNN [Lifeng *et al.*, 2020]	89.3
	LF-GCN (Ours)	**91.9**

Table 3: Test results on the PGR dataset. Results on AGGCN and GCN are reproduced based on their released implementations.

Syntax Type	Model	$F1$
Tree	Tree-GRN [Song *et al.*, 2019]	84.6
	GCN [Zhang *et al.*, 2018]	84.8
Forest	ForestFT-DDCNN [Lifeng *et al.*, 2020]	85.5
	AGGCN [Guo *et al.*, 2019a]	85.7
	KBest-GRN [Song *et al.*, 2019]	85.8
	Edgewise-GRN [Song *et al.*, 2019]	**86.3**
	LF-GCN (Ours)	85.7

Table 4: Test results on the SemEval dataset.

Forest: models leverage the dependency forest. Edgewise-GRN constructs a dependency forest by keeping all the edges with scores greater than a pre-defined threshold. KBest-GRN generates a forest by merging K-bests trees. ForestFT-DDCNN builds a forest by a learnable dependency parser. AGGCN computes attention matrices and treats them as the adjacency matrices of forests.

As shown in Table 3, models with full dependency trees or forests as inputs are able to significantly outperform all models that only consider the text sequence including BioBERT, which is trained on a very large-scale medical corpus. These results demonstrate that modeling structure in the input sentence is beneficial to the relation extraction task. Models with dependency forests as inputs yield better performance than those use 1-best dependency trees, which confirms our hypothesis that the error propagation, which is caused by the low parsing accuracy of an out-of-domain parser, can be alleviated by constructing weighted graphs (forests). Compared with models which encode fixed dependency forests that are generated at the data preprocessing stage (Edgewise-GRN and KBest-GRN), models with dynamic forests including AGGCN, ForestFT-DDCNN and LF-GCN achieve higher performance. On the other hand, LF-GCN also makes predictions without recourse to any pre-trained parsers, while it outperforms Random-DDCNN by a large margin, *i.e.,* 14.5. Furthermore, our LF-GCN model achieves 58.9 and 91.9 scores on CPR and PGR datasets, which are consistently better than all forest generation approaches. These results suggest that the induced latent structure is able to capture task-specific information for better relation extraction.

3.5 Results on News Domain

LF-GCN can also be used in other domain. Table 4 gives the results on SemEval [Hendrickx *et al.*, 2009] dataset from the news domain. Using limited training data, LF-GCN outperforms the models with a dependency tree including Tree-GRN and GCN by almost 1 point and is comparable with the models with dependency forests including AGGCN and ForestFT-DDCNN. This demonstrates that LF-GCN is able to learn a comparable expressive structure compared with the structure generated by an in-domain parser. LF-GCN is 0.6 point worse than Edgewise-GRN. The reason is that the parsing performance for newswire is much more accurate than the biomedical domain. We believe that our model is able to achieve higher performance if more training data is available.

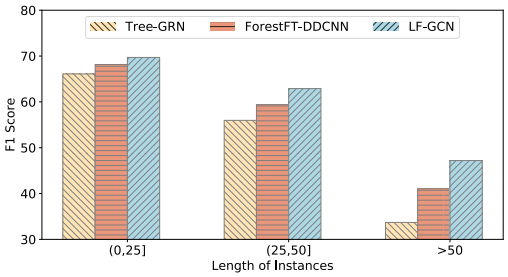

Figure 3: $F1$ scores against sentence length. The results on Tree-GRN and ForestFT-DDCNN come from [Lifeng *et al.*, 2020].

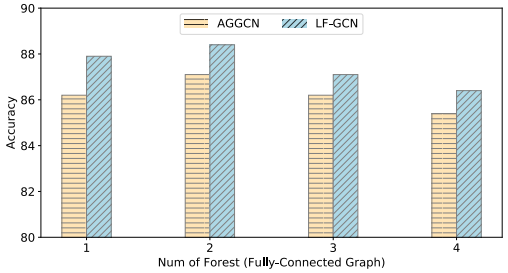

Figure 4: $F1$ scores against the number of forests on the dataset [Peng *et al.*, 2017] under the $\langle$ Binary-class, Ternary, `Cross` $\rangle$ setting shown in Table 1. The results on AGGCN are reproduced based on its released implementation.

3.6 Analysis and Discussion

Performance against Sentence Length

To investigate our LF-GCN performance under different sentence lengths, we split the test set of CPR into three categories $((0, 25], (25, 50], >50)$ based on the lengths. As shown in Figure 3, we compare our LF-GCN with Tree-GRN [Song *et al.*, 2019] and ForestFT-DDCNN [Lifeng *et al.*, 2020]. In general, LF-GCN outperforms Tree-GRN and ForestFT-DDCNN for each group of instances, showing the effectiveness of our model based on a latent structure induction. The performance gap is enlarged when the instance length increases. Intuitively, longer instances are more challenging since the dependency structure is a more sophisticated tree. These results illustrate that the induced structure is able to capture complex non-local interactions for better prediction.

Performance against Number of Forests

Figure 4 shows the performance of LF-GCN and AGGCN with different number of forests, since AGGCN also lever-

ages the multi-head attention mechanism [Vaswani *et al.*, 2017] to generate multiple weighted graphs. Even though AGGCN is initialized with the 1-best dependency tree generated by a pre-trained parser, LF-GCN consistently outperforms it under the same number of forest without relying on any parsers, where the numbers range from 1 to 4. These results demonstrate that our model is able to construct informative structures only based on the input text.

3.7 Case Study

In this section, we use the Chu-Liu-Edmonds [Chu and Liu, 1965] algorithm to extract N non-projective dependency trees from N latent forests, where each forest is expressed by a weighted adjacency matrix in Equation 8. Here N equals to 2. We select an instance from the CPR development set, whose relations can be correctly predicted by our LF-GCN.

Case I: As shown in Figure 5, these two dependency trees are able to capture rich interactions between the entities ***Schisandrin B*** (index 0 and 1) and ***DT-diaphorase*** (index 10, 11 and 12), which have a "up regulator" relation, denoted as "CPR:3". For example, the token "enhancing" (index 9), which shares the similar semantic as the gold relation "up regulator", is selected in the path between these two entities. Furthermore, these two trees show different dependencies between tokens, which confirms our hypothesis that inducing multiple forests can include more useful information.

Case II: However, as shown in Figure 6, many dependency trees induced by structure attention are shallow and do not resemble to a linguistic syntax structure. Figure 6 shows two shallow trees extracted from the latent forests before imposing sparsity constraints. We observe that the constructed dependency trees tend to pick the first token of the sentence as the root, and all other tokens as the children. Interestingly, even though such trees have little to no structure, the model is still able to predict the correct relation label. We also notice that adding the α-entmax helps to induce deeper and more informative structures. We leave the investigation of this phenomenon as future work.

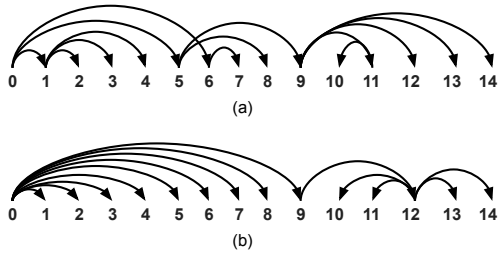

Figure 5: (a) the first and (b) the second non-projective dependency tree, which are extracted from two latent forests induced by LF-GCN. The sentence is "VT recurred with the addition of ***aminophylline***, a competitive ***adenosine A1-receptor*** antagonist."

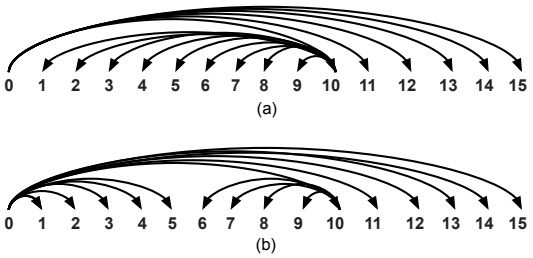

Figure 6: (a) the first and (b) the second non-projective dependency tree, which are extracted from two latent forests induced by LF-GCN. The sentence is "VT recurred with the addition of ***aminophylline***, a competitive ***adenosine A1-receptor*** antagonist."

4 Related Work

Latent Structure Induction: Latent structure models are powerful tools for modeling compositional data and building NLP pipelines [Yogatama *et al.*, 2016; Niculae *et al.*, 2018]. A challenge with structured latent models is that they involve computing an "argmax" (*i.e.,* finding a best scoring discrete structure such as a parse tree) in the middle of a computation graph. Since this operation has null gradients, back propagation cannot be used. There are three main strategies to solve this issue including reinforcement learning, surrogate gradients and continuous relaxation. In this paper, we mainly focus on continuous relaxations, for which the exact gradient can be computed and back propagated [Kim *et al.*, 2017; Liu and Lapata, 2018; Nan *et al.*, 2020].

Medical Relation Extraction: Early efforts focus on predicting relations between entities by modeling interactions in the 1-best dependency tree [Peng *et al.*, 2017; Song *et al.*, 2018]. Recently, dependency forests were used to alleviate the error cascading caused by an out-of-domain parser. Song *et al.* [2019] build a forest by adding edges and labels that a pre-trained parser is confident about. Lifeng *et al.* [2020] construct full forests represented by a 3-dimensional tensor generated by a pre-trained parser fine-tuned by the relation prediction loss. Instead of using an out-of-domain parser, our model dynamically induces multiple dependency forests solely based on the medical dataset in an end-to end manner.

5 Conclusion

In this paper, we propose a novel model that is able to automatically induce a latent structure for better relation extraction, without recourse to any tree supervisions or pre-training. Extensive results on four medical datasets show that our approach is able to better alleviate the error propagation caused by an out-of-domain dependency parser, giving significantly better results than previous state-of-the-art systems.

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful and constructive comments. This research is supported by Ministry of Education, Singapore, under its Academic Research Fund (AcRF) Tier 2 Programme (MOE AcRF Tier 2 Award No: MOE2017-T2-1-156). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the Ministry of Education, Singapore.

References

[Blondel *et al.*, 2018] Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning classifiers with fenchel-young losses: Generalized entropies, margins, and algorithms. In *AISTATS*, 2018.

[Chu and Liu, 1965] Yoeng-Jin Chu and Tseng-Hong Liu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 1965.

[Correia *et al.*, 2019] Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. In *EMNLP*, 2019.

[Dozat and Manning, 2017] Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *ICLR*, 2017.

[Eisner, 1996] Jason M. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *COLING*, 1996.

[Guo *et al.*, 2019a] Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *ACL*, 2019.

[Guo *et al.*, 2019b] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *TACL*, 2019.

[Hendrickx *et al.*, 2009] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *NAACL-HLT*, 2009.

[Kim *et al.*, 2017] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In *ICLR*, 2017.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Koo *et al.*, 2007] Terry K Koo, Amir Globerson, Xavier Carreras, and Michael Collins. Structured prediction models via the matrix-tree theorem. In *EMNLP*, 2007.

[Krallinger *et al.*, 2017] Martin Krallinger, Obdulia Rabal, Saber A Akhondi, et al. Overview of the biocreative vi chemical-protein interaction track. In *BioCreative challenge evaluation workshop*, 2017.

[Lamurias *et al.*, 2019] Andre Lamurias, Diana Sousa, Luka A Clarke, and Francisco M Couto. Bo-lstm: classifying relations via long short-term memory networks along biomedical ontologies. *BMC bioinformatics*, 2019.

[Lee *et al.*, 2019] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2019.

[Lifeng *et al.*, 2020] Jin Lifeng, Linfeng Song, Yue Zhang, Kun Xu, Weiyun Ma, and Dong Yu. Relation extraction exploiting full dependency forests. In *AAAI*, 2020.

[Liu and Lapata, 2018] Yang Liu and Mirella Lapata. Learning structured text representations. *TACL*, 2018.

[Liu *et al.*, 2017] Sijia Liu, Feichen Shen, Yanshan Wang, Majid Rastegar-Mojarad, Ravikumar Komandur Elayavilli, Vipin Chaundary, and Hongfang Liu. Attention-based neural networks for chemical protein relation extraction. In *Proceedings of the BioCreative VI Workshop*, 2017.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Demo@ACL*, 2014.

[Martins and Astudillo, 2016] André F. T. Martins and Ramón Fernández Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, 2016.

[Nan *et al.*, 2020] Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. Reasoning with latent structure refinement for document-level relation extraction. *arXiv preprint*, 2020.

[Niculae *et al.*, 2018] Vlad Niculae, André F. T. Martins, and Claire Cardie. Towards dynamic computation graphs via sparse latent structure. In *EMNLP*, 2018.

[Peng *et al.*, 2017] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen tau Yih. Cross-sentence n-ary extraction with graph lstms. *TACL*, 2017.

[Smith and Smith, 2007] David A. Smith and Noah A. Smith. Probabilistic models of nonprojective dependency trees. In *EMNLP*, 2007.

[Song *et al.*, 2018] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. N-ary relation extraction using graph state lstm. In *EMNLP*, 2018.

[Song *et al.*, 2019] Linfeng Song, Yue Zhang, Daniel Gildea, Mo Yu, Zhiguo Wang, and Jinsong Su. Leveraging dependency forest for neural medical relation extraction. In *EMNLP*, 2019.

[Sousa *et al.*, 2019] Diana Sousa, André Lamúrias, and Francisco M Couto. A silver standard corpus of human phenotype-gene relations. In *Proc. of NAACL-HLT*, 2019.

[Tutte, 1984] William Thomas Tutte. *Graph theory*. 1984.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Verga *et al.*, 2018] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *NAACL-HLT*, 2018.

[Yogatama *et al.*, 2016] Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. In *ICLR*, 2016.

[Zhang *et al.*, 2018] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*, 2018.