Structurally Comparative Hinge Loss for Dependency-Based Neural Text Representation

KEXIN WANG, National Laboratory of Pattern Recognition, Institute of Automation, CAS and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, P. R. China
YU ZHOU, National Laboratory of Pattern Recognition, Institute of Automation, CAS, School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, P. R. China, and Beijing Fanyu Technology Co., Ltd
JIAJUN ZHANG, SHAONAN WANG, and CHENGQING ZONG, National Laboratory of Pattern Recognition, Institute of Automation, CAS and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, P. R. China

Dependency-based graph convolutional networks (DepGCNs) are proven helpful for text representation to handle many natural language tasks. Almost all previous models are trained with cross-entropy (CE) loss, which maximizes the posterior likelihood directly. However, the contribution of dependency structures is not well considered by CE loss. As a result, the performance improvement gained by using the structure information can be narrow due to the failure in learning to rely on this structure information. To face the challenge, we propose the novel structurally comparative hinge (SCH) loss function for DepGCNs. SCH loss aims at enlarging the margin gained by structural representations over non-structural ones. From the perspective of information theory, this is equivalent to improving the conditional mutual information of model decision and structure information given text. Our experimental results on both English and Chinese datasets show that by substituting SCH loss for CE loss on various tasks, for both induced structures and structures from an external parser, performance is improved without additional learnable parameters. Furthermore, the extent to which certain types of examples rely on the dependency structure can be measured directly by the learned margin, which results in better interpretability. In addition, through detailed analysis, we show that this structure margin has a positive correlation with task performance and structure induction of DepGCNs, and SCH loss can help model focus more on the shortest dependency path between entities. We achieve the new state-of-the-art results on TACRED, IMDB, and Zh. Literature datasets, even compared with ensemble and BERT baselines.

CCS Concepts: • **Information systems → Document representation**; • **Computing methodologies → Natural language processing**;

Additional Key Words and Phrases: Text representation, graph convolutional networks, loss function

ACM Reference format:
Kexin Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2020. Structurally Comparative Hinge Loss for Dependency-Based Neural Text Representation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19, 4, Article 58 (May 2020), 19 pages.
https://doi.org/10.1145/3387633

Authors' addresses: K. Wang, Y. Zhou, J. Zhang, S. Wang, and C. Zong, Intelligence Building, No. 95, Zhongguancun East Road, Haidian District, Beijing 100190; emails: {kexin.wang, yzhou, jjzhang, shaonan.wang, cqzong}@nlpr.ia.ac.cn.

2375-4699/2020/05-ART58 $15.00
https://doi.org/10.1145/3387633

1 INTRODUCTION

Text representations, which are derived by mapping text into dense real-valued vectors that represent their semantics, have received much attention, playing a critical role in many applications such as relation extraction (RE) [13], sentiment analysis [43], and sentence similarity [5].

Four categories of models for constructing text representations are predominant. First, recurrent neural networks (RNNs) encode texts word by word in sequential order [17, 35]. Second, convolutional neural networks (CNNs) generate text representation by applying convolution operation on receptive fields from different levels [11, 21]. Third, recursive neural networks (RecNNs) embed a sentence recursively along its parsing tree [46, 60] in a bottom-up fashion. Fourth, graph convolutional networks (GCNs), which we focus on in this work, map sentences into their representations by propagating information from base representations generated by RNN or CNN along edges in a graph structure, which is usually a dependency parsing tree [2, 32, 58]. Dependency-based graph convolutional networks (DepGCNs) can construct high-quality text representations efficiently. Further concerns about acquiring dependency structures of sentences at low cost pave the way for structure induction [1, 3].

Among almost all previous studies, cross-entropy (CE) loss is adopted unquestionably to train DepGCNs, which is equivalent to maximizing the posterior likelihood over training data. However, CE loss does not take the contribution from structure information into consideration, and thus the model is not aware of learning to rely on the structure information for prediction in some cases. As a result, DepGCNs are likely to overfit some superficial cues. For instance, Figure 1 shows the examples of dependency-edge saliency for DepGCNs trained with different loss functions, illustrating how much each dependency edge contributes to the model decision [24]. As shown at the top of Figure 1, it is easy for DepGCNs trained with CE loss to recognize the examples from relation type *per:parent* wrongly as type *per:children*, due to overfitting the word *mother* (marked by red dashed lines). Note that the correct understanding should be the parent of SUBJ-PERSON is OBJ-PERSON but the reverse relation predicted by the model.

In this article, we propose a novel structurally comparative hinge (SCH) loss function to solve the problem caused by CE loss. SCH loss is defined by adding gained margin from the structural representation over the non-structural one to the commonly used CE loss, which aims at improving the correlation between model decision and structure information the given text (i.e., conditional mutual information). Trained with SCH loss, DepGCNs are forced to rely on the dependency structures for making predictions. This lead to better generalization on hard cases shown in Figure 1. More specifically, DepGCNs trained with SCH loss focus on the shortest dependency path (marked by the green dashed lines in Figure 1) between entities, as well as the key words *his* and *mother*, indicating that the model considers the dependency between the key elements rather than makes a decision once seeing an attractor word. SCH loss can also provide a measure indicating structure awareness (i.e., to what extent the structure information is needed) for certain examples by checking the margin gained. We conduct experiments on three tasks consisting of RE, document classification (DC), and paraphrase identification (PI) for both English and Chinese, with DepGCNs whose dependency structures are derived from induction or an external parser. Experimental results show that noticeable improvements are obtained by using SCH loss instead of CE loss, especially in RE. And further analysis is carried out on the structure-awareness measure and how SCH loss helps to model text.

Our contributions are twofold:

- We propose a novel SCH loss function as a substitute for CE loss. We show that the SCH loss can consistently improve the quality of dependency-based text representations for two sources of dependency structures.

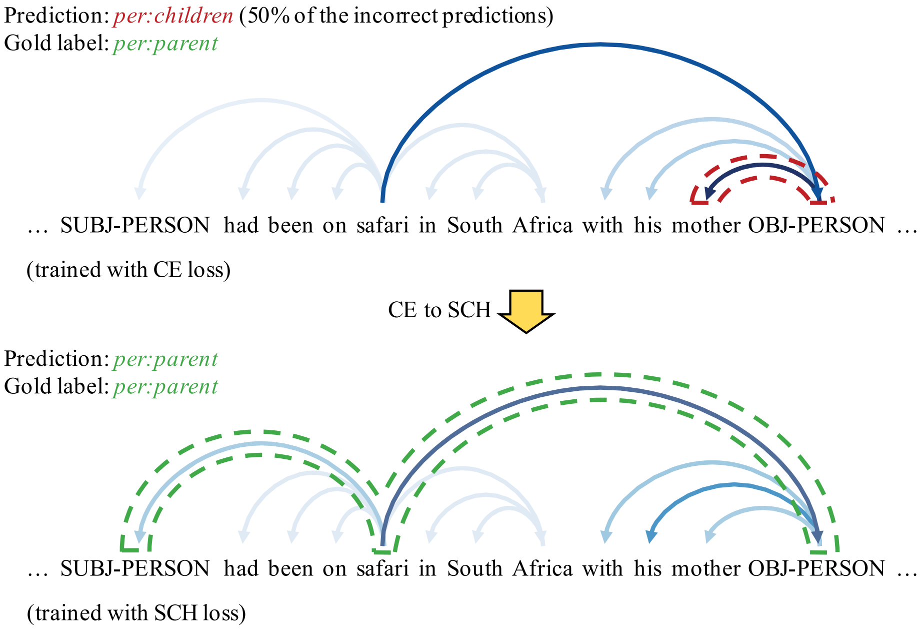

Prediction: per:children (50% of the incorrect predictions)
Gold label: per:parent
… SUBJ-PERSON had been on safari in South Africa with his mother OBJ-PERSON …
(trained with CE loss)
CE to SCH
Prediction: per:parent
Gold label: per:parent
… SUBJ-PERSON had been on safari in South Africa with his mother OBJ-PERSON …
(trained with SCH loss)

Fig. 1. Examples of DepGCNs trained with CE loss and proposed SCH loss. The examples are from the RE dataset TACRED. SUBJ-PERSON and OBJ-PERSON represent subject and object, respectively. The gold relation indicates that the parent of SUBJ-PERSON is OBJ-PERSON. The directed lines show the unlabeled dependency structures. The color of a certain edge indicates to what extent the edge's existence influences the decision, calculated by the derivative of the posterior with respect to that edge.

- We define a measure of structure awareness by a component of SCH loss. We show that this measure reflects to what extent one example relies on its structure information and has a positive correlation with structure induction.

2 BACKGROUND: DEPGCNS

We now describe GCNs of Kipf and Welling [22], whose input graphs are dependency structures [2, 3], and its commonly used loss function.

2.1 Graph Convolutional Network

A GCN is a multi-layer neural network that operates directly on a graph, encoding information about the neighborhood of a node as a real-valued vector. In each GCN layer, information flows along edges of the graph gathering messages from neighbors. With k layers, a node receives information from neighbors at most k hops away.

Formally, consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of n nodes and $\mathcal{E}$ is a set of edges. In our case, $\mathcal{G}$ is a dependency structure of a sentence, a node represents a word of it, and an edge is a dependency arc. For a given text x containing n words, each word is viewed as a node, and let $\mathbf{h_t}^{(j)} \in \mathbb{R}^d$ $(t = 0, 1, \ldots, n-1)$ be the feature vectors for these words (nodes) at layer j $(j = 0, 1, \ldots, l)$. $\mathbf{h}_t^{(0)}$'s are base input features for GCN, and as the best choice for text representation shown in previous work, they are derived from a bidirectional LSTM (BiLSTM) [18]:

$$\overrightarrow{\mathbf{h}_t} = \mathrm{LSTM_f}(\mathbf{x}_t, \overrightarrow{\mathbf{h}_{t-1}}), \tag{1}$$

$$\overleftarrow{\mathbf{h}_t} = \mathrm{LSTM_b}(\mathbf{x}_t, \overleftarrow{\mathbf{h}_{t+1}}), \tag{2}$$

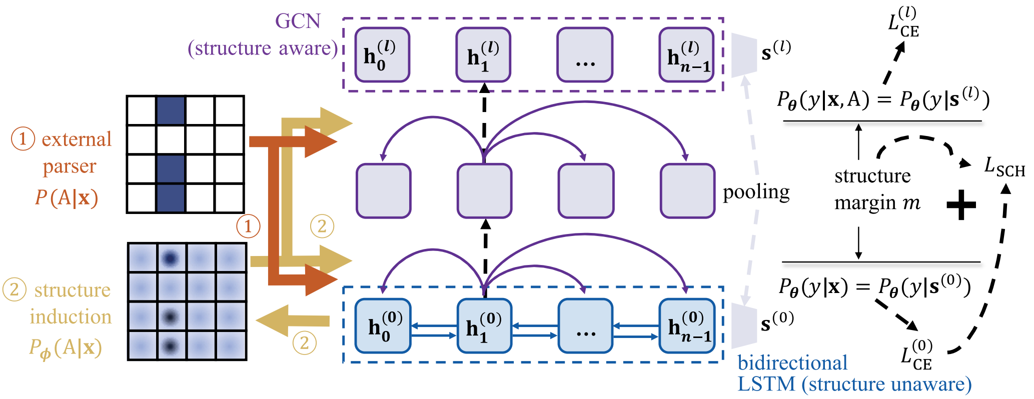


Fig. 2. Dependency-based graph convolutional networks. The left end of the figure shows the two different sources of dependency structures, marked by two different colors. The right end of the figure illustrates the proposed SCH loss, as the combination of structure margin and the CE loss from the non-structural representations.

$$\mathbf{h}_t^{(0)} = \overrightarrow{\mathbf{h}_t} \oplus \overleftarrow{\mathbf{h}_t}, \tag{3}$$

where LSTM_f and LSTM_b are forward and backward LSTM functions, respectively; $\oplus$ is a concatenation operation; and $\mathbf{x}_t$ is the word embedding of t-th word in a sentence x. At the j-th ($j = 1, 2, \ldots, l$) layer of a stacked GCN, the update function of node v's feature is as follows:

$$\mathbf{h}_v^{(j+1)} = \rho\left(\sum_{u \in \mathcal{N}(v)} W_{\text{dir}(u,v)}^{(j)} \mathbf{h}_u^{(j)} + \mathbf{b}_{\text{dir}(u,v)}^{(j)}\right), \tag{4}$$

where $\mathcal{N}(v)$ is the set of adjacent nodes of v; the superscript j is the layer index; ρ is an activation function (e.g., ReLU); dir(u, v) is the directionality of the edge connecting u and v; and $W_{\text{dir}(u,v)}^{(j)} \in \mathbb{R}^{d \times d}$, $\mathbf{b}_{\text{dir}(u,v)}^{(j)} \in \mathbb{R}^d$ are learnable parameters for the j-th-layer GCN. Note that the edge labels are ignored because little improvement can be gained by using them.

To down-weight the contribution of individual edges, a gating mechanism [26] is adopted as follows:

$$g_{u,v}^{(j)} = \sigma\left(\mathbf{h}_u^{(j)} \cdot \hat{\mathbf{w}}_{\text{dir}(u,v)}^{(j)} + \hat{b}_{\text{dir(u,v)}}^{(j)}\right), \tag{5}$$

$$\mathbf{h}_v^{(j+1)} = \rho\left(\sum_{u \in \mathcal{N}(v)} g_{u,v}^{(j)}\left(W_{\text{dir}(u,v)}^{(j)} \mathbf{h}_u^{(j)} + \mathbf{b}_{\text{dir}(u,v)}^{(j)}\right)\right), \tag{6}$$

where the scalar $g_{u,v}$ is the gate for dir(u, v), $\hat{\mathbf{w}}_{\text{dir}(u,v)}^{(j)} \in \mathbb{R}^d$, $\hat{b}_{\text{dir(u,v)}}^{(j)}$ are learnable parameters for the gate, and σ is the sigmoid function.

To derive a text embedding $\mathbf{s}^{(j)} \in \mathbb{R}^d$ at a certain layer j, a max-pooling is applied over the set of node features $\{\mathbf{h}_t^{(j)}\}_{t=0}^{n-1}$.

2.2 Dependency-Based Graph Convolutional Network

In this work, we consider the case where the input graphs of GCNs are dependency structures. We name these kinds of models *DepGCNs*. We note the adjacent matrix form of the graphs as A. The whole architecture of DepGCNs is shown in Figure 2. We next introduce two sources of graph A, namely from an external parser and from induction.

2.2.1 Dependency Structures from an External Parser. As a common choice, one can utilize an external dependency parser $P(A|\mathbf{x})$ to sample the graph matrix A from it. In addition, the dependency structures are directly input to the DepGCNs. We call this case *DepGCN-Ex*. The illustration for this structure source is shown in Figure 2, marked with red color.

2.2.2 Dependency Structures from Induction. One can also parameterize an inner parser $P_{\boldsymbol{\phi}}(A|\mathbf{x})$ to induce dependency structures and provide the GCN with these induced structures. To induce dependency structures, we apply structured self-attention to the representations from the base BiLSTM layer [3, 28]. We call DepGCN with induced structures *DepGCN-In*.

More specifically, the raw directed correlation between each pair of words is first modeled by self-attention scores:

$$\alpha_{ij} = \frac{1}{\sqrt{d}} \cdot \left(W_q \mathbf{h}_i^{(0)}\right)^{\mathrm{T}} \left(W_k \mathbf{h}_j^{(0)}\right). \tag{7}$$

Then, this directed correlation is summed among a candidate dependency tree A and normalized to parameterize the distribution of dependency trees given a sentence x:

$$P_{\boldsymbol{\phi}}(A|\mathbf{x}) = P_{\boldsymbol{\phi}}(A_0, A_1, \ldots, A_{n-1}|\mathbf{x}) = \frac{\exp(\sum_i \alpha_{i,A_i})}{Z(A|\mathbf{x})}, \tag{8}$$

where the random variables A_j's indicate the head of word j and $Z(A|\mathbf{x})$ is the partition function over all possible head combinations. As the last step of inferring a dependency tree from this distribution, the joint distribution is marginalized by the matrix-tree theorem [23] and the inference is done by independent samplings:

$$A_j \sim P_{\boldsymbol{\phi}}(A_j|\mathbf{x}) = \sum_{A_j, j \neq i} P_{\boldsymbol{\phi}}(A_0, A_1, \ldots, A_{n-1}|\mathbf{x}). \tag{9}$$

The illustration for this structure source is shown in Figure 2, marked with dark yellow. The adjacent matrix is plotted as a heat map, as the induced adjacent matrix is in the form of soft distributions over head positions.

2.3 Loss Function

Among almost all previous studies, CE loss is adopted to train DepGCNs. For a certain task dataset $D = \{(x_i, y_i)\}_{i=0}^{N-1}$ and the text representations $\mathbf{s}^{(j)}$ at layer j, we note the gold label as y and $P_{\boldsymbol{\theta}}(y|x) = P_{\boldsymbol{\theta}}(y|\mathbf{s}^{(j)})$ as the prediction distribution. Then the CE loss at layer j is

$$L_{\mathrm{CE}}^{(j)}(\boldsymbol{\theta}) = -\mathbb{E}_{(x,y)\sim D}[\log P_{\boldsymbol{\theta}}(y|x)]. \tag{10}$$

By minimizing CE loss, the posterior likelihood over the dataset is maximized.

However, CE loss does not emphasize the contribution from mining structure information (i.e., structure awareness). As a consequence, when trained with CE loss, the advantage of this structure information can be small, and the model is at the risk of overfitting to some superficial cues.

3 SCH LOSS

To measure structure awareness and further encourage the DepGCNs to fully utilize the structure information, we propose SCH loss as a substitute for the commonly used CE loss.

3.1 Derivation of SCH Loss

Formally, note $P_{\theta}(y|x, A)$ and $P_{\theta}(y|x)$ as the task-specific posterior predicted by the model given text x with and without structure information A, respectively. In the case of DepGCNs, $P_{\theta}(y|x, A) = P_{\theta}(y|\mathbf{s}^{(l)})$ and $P_{\theta}(y|x) = P_{\theta}(y|\mathbf{s}^{(0)})$, as only GCN layers above the base BiLSTM layer explicitly model the structure information.

We start from the mutual information between the structure information A and the model decision y given the text x:

$$\begin{aligned} I(y; A|x) &= \mathbb{E}_{(x,y)\sim D, A\sim P(A|x)} \left[\log \frac{P(y, A|x)}{P(A|x)P(y|x)}\right] \\ &= \mathbb{E}_{(x,y)\sim D, A\sim P(A|x)} [\log P(y|x, A) - \log P(y|x)]. \end{aligned}$$

Improving this conditional mutual information is actually forcing the model to correlate its decision to the structure information, thus being structure aware. We define the difference between the log-likelihood of the two posteriors as the measure of structure awareness m, named *structure margin*:

$$\begin{aligned} m &= \log P_{\theta}(y|x, A) - \log P_{\theta}(y|x) \\ &= \log P_{\theta}(y|\mathbf{s}^{(l)}) - \log P_{\theta}(y|\mathbf{s}^{(0)}). \end{aligned}$$

To maximize the conditional mutual information, we can encourage the model to enlarge the structure margin by minimizing the hinge loss over m[1]:

$$L_{\text{hinge}}(\boldsymbol{\theta}) = \mathbb{E}_{(x,y)\sim D}[\max(0, \delta - m)], \tag{11}$$

where $\delta > 0$ is a threshold value. Due to the definition of m, this is equivalent to requiring the conditional mutual information $I(y; A|x)$ to be large enough.

3.2 Practical Issue

In practical usage, the model can cheat to minimize L_{hinge} by degrading $\log P_{\theta}(y|x)$, which results in intractable training procedure.[2] To deal with this issue, we add the CE loss over $P_{\theta}(y|x)$ to prevent this from happening:

$$L_{\text{SCH}}(\boldsymbol{\theta}) = L_{\text{CE}}^{(0)}(\boldsymbol{\theta}) + L_{\text{hinge}}(\boldsymbol{\theta}). \tag{12}$$

We call L_{SCH} *SCH loss*.

By extending the SCH loss formula, we can derive the following:

$$\begin{aligned} L_{\text{SCH}}(\boldsymbol{\theta}) &= L_{\text{CE}}^{(0)}(\boldsymbol{\theta}) + L_{\text{hinge}}(\boldsymbol{\theta}) \\ &= -\mathbb{E}_{(x,y)\sim D}[-\log P_{\theta}(y|x) + \max(0, \delta - m)] \\ &= -\begin{cases} -\mathbb{E}[\log P_{\theta}(y|x)], \tilde{}m > \delta \\ \delta - \mathbb{E}[\log P_{\theta}(y|x, A)], \tilde{}m \leq \delta \end{cases}. \end{aligned}$$

This means that if the structure margin m is large enough, the referred baseline $P_{\theta}(y|x)$ is improved; otherwise, the structural representation is not powerful enough, and $P_{\theta}(y|x, A)$ should be improved until it is larger than the non-structural baseline $P_{\theta}(y|x)$.

By minimizing SCH loss for DepGCNs, the structure margin is enlarged to be large enough, and harder cases like the one in Figure 1 that highly rely on the structure information can be well dealt with. In addition, as a natural choice, by comparing the structure margin one can examine which examples from certain tasks need structure information the most. Similar ideas appear in

[1] For simplicity, we omit the notation of the variables inside m and the sampling procedure of A.

[2] We observe that the loss does not decrease at all when simply minimizing L_{hinge}.

Table 1. Experimental Settings

Task	Dataset	d_e	d_h	Embedding	Fixed	Optimizer	Drop.	Lr.	δ
RE	TACRED	300	200	glove.840B.300d	False	SGD	0.5	1.0	1.4
	Zh. Literature	300	200	sgns.literature.char	False	SGD	0.5	1.0	1.0
DC	IMDB	300	100	glove.840B.300d	True	Adadelta	0.5	1.0	1.0
	IFeng	100	100	—	True	Adam	0.5	0.001	0.3
PI	QQP	300	100	glove.840B.300d	False	Adam	0.1	0.001	0.4
	LCQMC	300	100	sgns.zhihu.word	True	Adam	0.1	0.001	0.5

d_e, d_h, sizes of word embedding and hidden states, respectively; Fixed, whether the embedding weight is fixed without tuning; Drop., dropout ratio; Lr., learning rate; δ, hyper-parameter for SCH.

document-level neural machine translation [20], where context-based representations at different levels are encouraged to gain a large margin over representations constructed without modeling context. In fact, besides DepGCNs, SCH loss can be applied to other models that rely on structure information such as TreeLSTM [46].

4 EXPERIMENT

We evaluate the proposed SCH loss function for DepGCNs on three tasks: RE, DC, and PI. Since the effectiveness of using dependency structures for RE has been reported in many previous studies (e.g., [29, 54]), RE is a highly suitable evaluation task for our proposed method. Considering that structure information can provide gradient shortcuts for the training process of neural network models and then help the model capture the long-term dependencies [28, 40], we are interested in evaluating the models on the DC task to check whether SCH loss can help model long documents better. As a sanity check, we also experiment on PI datasets built on short user queries to show that using SCH loss would not lead to a performance drop when the structure information is shallow.

For each task, four models are compared:

- *DepGCN-Ex*: DepGCN whose dependency structures are from an external parser.
- *DepGCN-In*: DepGCN whose dependency structures are from induction.
- *DepGCN-En + SCH*: DepGCN-Ex with SCH loss.
- *DepGCN-In + SCH*: DepGCN-In with SCH loss.
- *BiLSTM*: Bidirectional LSTM model.

We use the dependency parser supported by spaCy [19] for English and Stanford CoreNLP [31] for Chinese if without supervision. δ is tuned based on the validation performance, and other hyper-parameters for our models are set based on previous state-of-the-art models. The detailed experimental settings are shown in Table 1. Glove word embeddings [36] and SGNS word embeddings [25] are adopted for English and Chinese datasets, respectively. For all datasets, two-layer GCNs are adopted with a residual connection [15], and edge dropout is applied with drop ratio equal to 0.2. Since the size of some datasets is small, the validation results can deviate from the test ones. We thus apply model averaging [52] over five best models on the validation data to obtain stable results.

4.1 Relation Extraction

RE is a task of predicting the relation type between the subject and the object entities in a sentence. We evaluate our SCH loss on the TACRED dataset [59] and the Zh. Literature dataset [53], whose statistics are shown in Table 2. Examples in TACRED are built over newswire and web text, and ones in Zh. Literature are obtained over Chinese literature articles from the web. Part-of-speech

Table 2. Dataset Overview

Task	Eval.	Dataset	Lang.	#class	#train	#valid.	#test	Ave. Len.
RE	F_1	TACRED	EN	42	68,124	226,31	155,09	36.4
		Zh. Literature	ZH	9	13,462	1,347	1,675	52.0
DC	Acc.	IMDB	EN	10	67,426	8,381	9,112	379.5
		IFeng	ZH	5	719,995	79,999	50,000	49.5
PI	Acc.	QQP	EN	2	384,348	10,000	10,000	12.7
		LCQMC	ZH	2	238,766	8,802	12,500	6.8

Eval., evaluation metric; Lang., dataset language; Ave. Len., averaged length of the text with respect to words.

Table 3. Results (%) on the TACRED Dataset

Model	Precision	Recall	F_1
Tree-LSTM (Tai et al. [46])	66.0	59.2	62.4
C-GCN (Zhang et al. [58])	69.9	63.3	66.4
BERT-LSTM-base (Shi and Lin [41])	**73.3**	63.10	67.8
C-CGN ensemble (Zhang et al. [58])	71.3	**65.4**	**68.2**
BiLSTM	**72.6**	62.5	67.2
DepGCN-In	67.5	66.3	66.9
DepGCN-In + SCH	70.1(+2.6)	66.4	68.3(+1.4)
DepGCN-Ex	67.1	**67.5**	67.3
DepGCN-Ex + SCH	69.3(+2.2)	**67.5**	**68.4(+1.1)**

The top set shows the performance reported in the previous work, including a state-of-the-art result from an ensemble model. The bottom set shows our results. Bold marks the highest number among each model set. Parentheses indicate the performance gain by using SCH loss over CE loss.

(POS) tags, named entity (NE) types, and dependency structures are provided within TACRED, and we use the Stanford CoreNLP toolkit to derive these features for Zh. Literature.

Following Zhang et al. [58], 30-D embeddings for POS tags and NE types are appended to the word embeddings. After constructing the text embedding $\mathbf{s}^{(j)}$'s, subject and object representations $\mathbf{s}^{(j)}_{subj}$ and $\mathbf{s}^{(j)}_{obj}$ at layer j are extracted from $\{\mathbf{h}^{(j)}_t\}^{n-1}_{t=0}$ among the given positions indicating the entities. In addition, the posterior for an input example is given by a Softmax layer over the concatenated representation $\mathbf{s}^{(j)}_{subj} \oplus \mathbf{s}^{(j)} \oplus \mathbf{s}^{(j)}_{obj}$:

$$P_\theta\left(y|\mathbf{s}^{(j)}\right) = \mathrm{Softmax}\left(W_{RE}(\mathbf{s}^{(j)}_{subj} \oplus \mathbf{s}^{(j)} \oplus \mathbf{s}^{(j)}_{obj}) + \mathbf{b}_{RE}\right). \tag{13}$$

For the Chinese dataset Zh. Literature, both character and word representations are used, with word embeddings appended to the character ones. Characters forming the same word share dependency edges, POS tags, and NE types.

The result for TACRED is shown in Table 3. We observe that our proposed SCH loss can improve performance of DepGCNs by 1.4 F_1 at most and further make DepGCN-Ex become the new state-of-the-art model. To our surprise, DepGCN-Ex trained with our proposed SCH loss can achieve better F_1, even compared with an ensemble model and the model fine tuned on BERT [8], which is 5.6 times the size of DepGCN-Ex in terms of the number of parameters. Comparing DepGCN-In + SCH with DepGCN-Ex, we can find that SCH loss helps narrow the gap caused by lack of strong structure supervision. DepGCNs trained with CE loss, however, have little advantage over the BiLSTM baseline and are even worse when the dependency structure is from induction. The result for Zh. Literature is given in Table 4. Similar observations can be found for this dataset,

Table 4. Results (%) on the Zh. Literature Dataset

Model	Precision	Recall	F_1
DepNN (Liu et al. [29])	—	—	55.2
BRCNN (Cai et al. [4])	—	—	55.6
SR-BRCNN (Wen et al. [53])	—	—	**65.9**
BiLSTM	70.2	67.2	68.3
DepGCN-In	72.7	69.7	70.4
DepGCN-In + SCH	72.9(+0.2)	70.7(+1.0)	71.5(+1.1)
DepGCN-Ex	71.6	69.4	69.7
DepGCN-Ex + SCH	**74.2(+2.6)**	**70.8(+1.4)**	**71.7(+2.0)**

Table 5. Results (%) on the DC Task

Model	IMDB	Ifeng
LSTM gated RNN (Tang et al. [47])	45.3	—
Structured Attention (Liu and Lapata [28])	49.2	—
Hierarchical Attention (Yang et al. [55])	**49.4**	—
fastText (Sun et al. [44])	—	83.7
S.C. (Sun et al. [44])	—	**84.4**
BiLSTM	45.6	**84.7**
DepGCN-In	46.9	84.5
DepGCN-In + SCH	47.1(+0.2)	84.6(+0.1)
DepGCN-Ex	50.1	84.6
DepGCN-Ex + SCH	**51.4(+1.2)**	**84.7(+0.1)**

where SCH loss can both remarkably increase the F_1 scores of DepGCN-In and DepGCN-Ex. New state-of-the-art performance is also achieved.

4.2 Document Classification

In the DC task, a document consisting of multiple sentences is input to the model, and a corresponding class label is predicted. For this task, we conduct experiments on the IMDB review dataset [9] and Ifeng news dataset [57]. Examples from the IMDB dataset are randomly crawled movie reviews, with the rating scores to be the class labels. The Ifeng news dataset is constructed by crawling all news from the year 2006 to the year 2016 from the Chinese news website ifeng.com.[3] The class labels of Ifeng news are five topic classes, including mainland China politics, international news, Taiwan-Hong Kong-Macau politics, military news, and society news. The statistics of these two datasets are shown in Table 2.

In our experiments, a document is viewed as a sequence without sentence splitters. For both datasets, only word tokenization is adopted. After generating text representation $\mathbf{s}^{(j)}$ by the models, this representation is input to a multi-layer perceptron to decode labels. The experimental results of the two datasets are shown in Table 5 in terms of accuracy. Note that all of the compared baselines use sentence-document hierarchical architecture. On the IMDB dataset, we find that our proposed SCH loss can improve the performance of DepGCNs by 1.2 at most in terms of accuracy, which is the new state-of-the-art result outperforming the previous best model by 2.0

[3] www.ifeng.com.

Table 6. Results (%) on the PI Task

Model	QQP	LCQCM
L.D.C (Wang et al. [51])	85.6	—
BiMPM (Wang et al. [50])	88.2	**83.4**
DIIN (Gong et al. [12])	**89.1**	—
BiLSTM	87.8	81.7
DepGCN-In	88.1	82.6
DepGCN-In + SCH	**88.4(+0.3)**	82.9(+0.3)
DepGCN-Ex	87.9	83.0
DepGCN-Ex + SCH	88.0(+0.1)	**83.1(+0.1)**

accuracy. Compared with the (structured) attention models, the dependency parses from an external parser are much more effective, indicating the crucial role of modeling dependency structures for document-level text representations. On the Ifeng news dataset, the improvement is relatively small. Since the topics of the Ifeng news dataset are closely related to some key words, the structure information is not crucial. Table 5 shows that applying CNN models S.C. on transformed images from the texts can also achieve similar performance of our BiLSTM model. This comparison can also support the idea that this Ifeng news dataset relies more on bag-of-words information.

4.3 Paraphrase Identification

The PI task requires models to classify whether a pair of sentences are similar. We experiment on the dataset QQP[4] and LCQCM [27]. Example pairs from QQP are collected from user queries on the Quora website. In addition, LCQCM is constructed using user queries in different domains from Baidu Knows. Information about these two datasets is given in Table 2.

We simply use the sentence embedding $\mathbf{s}$ to represent a sentence. Denoting the sentence embeddings of a given pair as $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{R}^d$, the features of a sentence pair is computed as $\mathbf{s}_1 \oplus \mathbf{s}_2 \oplus |\mathbf{s}_1 - \mathbf{s}_2| \oplus (\mathbf{s}_1 \otimes \mathbf{s}_2) \in \mathbb{R}^{4d}$. These features are then used to predict the class label via a Softmax layer. The results are shown in Table 6. We find for both datasets that SCH loss can slightly improve the performance of DepGCNs. Note that the baselines apply intensive matching between token-pair representations, which is expensive in terms of computation. Since the examples from this dataset are simple search queries with short sentence length (12.7 and 6.8, respectively), the advantage of the dependency structure is not obvious. We can also find that SCH loss can help DepGCN-In more than DepGCN-Ex. Since the user queries are formed freely without strict syntax, the more flexible way of dependency induction is more suitable for this task.

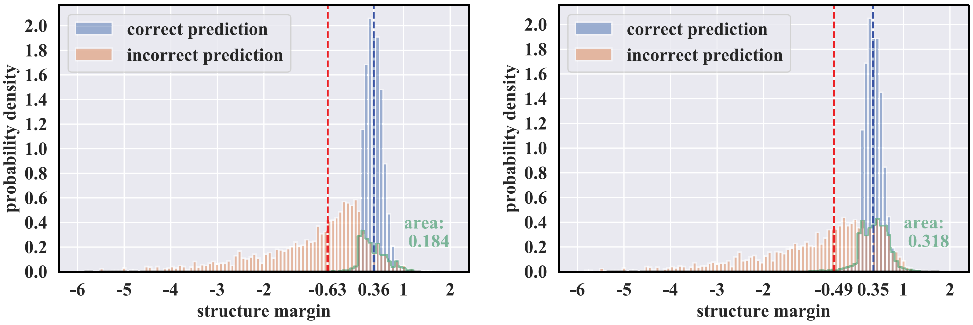
5 ANALYSIS

In this section, we are interested in three questions: (1) What is the relation between structure margin (defined in Section 3) and task performance? (2) How does SCH loss help modeling text? and (3) What is the relation between structure margin and dependency induction? Since great performance is achieved and the gold parses are provided in TACRED, we carry out analysis on its test set.

5.1 Structure Margin vs. Task Performance

To answer the first question, we visualize the distribution of structure margin conditioned on the correct or incorrect predictions for DepGCN-Ex. The visualization is shown in Figure 3(a).

[4] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.



(a) Trained with SCH loss

(b) Trained with CE loss

Fig. 3. Distributions of structure margin conditioned on correct or incorrect predictions from the models trained with SCH loss (a) and CE loss (b). Blue and red dot lines indicate the median of the structure margin for the correct and incorrect predictions, respectively. The text in green indicates the overlapping area.

We can observe that the distribution conditioned on the correct predictions is in the shape of concentrated Gaussian and that the shape of the one conditioned on the incorrect predictions is skew-Gaussian. Thus, the median value of the structure margin is used as the general indicator. The median conditioned on the correct predictions is larger than that conditioned on incorrect ones, which implies that structure margin has a positive correlation with task performance. To gain more insight into the role of SCH loss for enlarging the structure margin, we plot the distribution of structure margin[5] for DepGCN trained with CE loss in Figure 3(b). We can find that for both cases in Figure 3(a) and (b), the shapes of the distributions are similar, especially for the ones conditioned on the correct predictions. Obvious distinctions appear when the distributions conditioned on the incorrect predictions are compared: for Figure 3(a), the distribution is sharper and has less area of distribution overlapping than that in Figure 3(b). This implies that the structure margin is a good indicator for separating easy examples from hard ones, and SCH loss helps the model to deal with the hard cases by fully utilizing the structure information. In addition, the model trained with CE loss has missed many chances for this kind of situation.

Besides the preceding general analysis, we are interested in investigating the influence on specific classes. Thus, we also plot the box plots of the structure margins from correct predictions versus model performance gaps over different class sets. The result is shown in Figure 4. In this figure, three models are compared: the base model BiLSTM, DepGCN-Ex with CE loss, and DepGCN-Ex with SCH loss. We can find that in most cases, indicated by the red and orange x-ticks, large structure margins (i.e., nearly equal or larger than the mean level) correspond to large performance gain for DepGCN-Ex + SCH compared with non-structural representation model BiLSTM. In turn, when the performance margin gained by DepGCN-Ex + SCH over the other two models is small or none at all, the structure margin is usually below the mean level, indicated by green x-ticks. These findings also support that structure margin is positively related to task performance, which is as expected since minimizing SCH loss is in theory equivalent to maximizing the conditional mutual information of structure information and model decision. This can be intuitively explained as follows: hard cases that strongly rely on the structure information are assigned larger structure margins and can be better handled by DepGCNs trained with SCH loss, and in reverse, for the examples that do not rely on the structure information, the structure margins are low and the advantage of SCH loss is not obvious.

[5] For fair comparison, we assign the same examples the same structure margin.

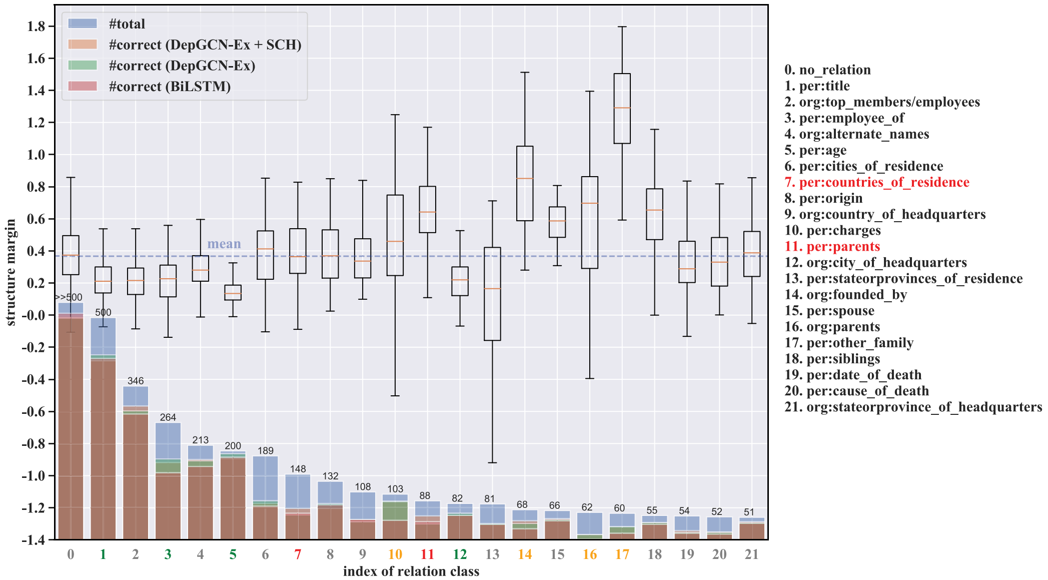
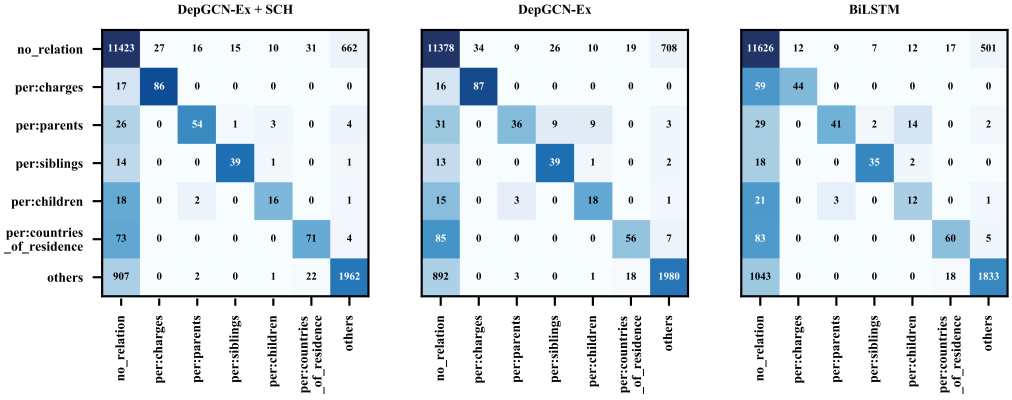


Fig. 4. Box plots for structure margins of correct predictions from different classes. Bars below each box plot indicate the number of examples from its class. The mapping between relation class and its index is listed at the right side of the figure. The blue dashed line shows the average of structure margins from the correct predictions by DepGCN-Ex + SCH.



DepGCN-Ex + SCH

	no_relation	per:charges	per:parents	per:siblings	per:children	per:countries_of_residence	others
no_relation	11423	27	16	15	10	31	662
per:charges	17	86	0	0	0	0	0
per:parents	26	0	54	1	3	0	4
per:siblings	14	0	0	39	1	0	1
per:children	18	0	2	0	16	0	1
per:countries_of_residence	73	0	0	0	0	71	4
others	907	0	2	0	1	22	1962

DepGCN-Ex

	no_relation	per:charges	per:parents	per:siblings	per:children	per:countries_of_residence	others
no_relation	11378	34	9	26	10	19	708
per:charges	16	87	0	0	0	0	0
per:parents	31	0	36	9	9	0	3
per:siblings	13	0	0	39	1	0	2
per:children	15	0	3	0	18	0	1
per:countries_of_residence	85	0	0	0	0	56	7
others	892	0	3	0	1	18	1980

BiLSTM

	no_relation	per:charges	per:parents	per:siblings	per:children	per:countries_of_residence	others
no_relation	11626	12	9	7	12	17	501
per:charges	59	44	0	0	0	0	0
per:parents	29	0	41	2	14	0	2
per:siblings	18	0	0	35	2	0	0
per:children	21	0	3	0	12	0	1
per:countries_of_residence	83	0	0	0	0	60	5
others	1043	0	0	0	0	18	1833

Fig. 5. Confusion matrices for the three models.

5.2 How Does SCH Loss Help Modeling Text?

We next investigate how SCH helps in understanding text. Since significant performance gain appears in relation 7 *per:countries_of_residence* and 11 *per:parents* and the structure margin is also large (indicated by red x-ticks in Figure 4), we start from these two relation types. We plot in Figure 5 the confusion matrices of DepGCN-Ex + SCH, DepGCN-Ex, and BiLSTM. As shown in Figure 5, DepGCN-Ex and BiLSTM tend to classify the examples from *per:parents* as *per:siblings* or *per:children*, and it is harder to recognize the relation *per:countries_of_residence* for them. To gain more insight, four examples that are predicted correctly by DepGCN-Ex + SCH are shown in Table 7. The examples for relation *per:countries_of_residence* shows that this relation is expressed

Table 7. Four Examples That Are Correctly Predicted by DepGCN-Ex + SCH on TACRED

Relation	DepGCN-Ex	BiLSTM	Text
per:countries_ of_residence	*no_relation*	*no_relation*	… discuss the fate of SUBJ-PERSON SUBJ-PERSON jailed in OBJ-COUNTRY for nearly 11 months on suspicion of spying …
	no_relation	*no_relation*	SUBJ-PERSON SUBJ-PERSON was sentenced to hang in OBJ-COUNTRY 's central province of Punjab … after being found guilty of insulting the Prophet Mohammed.
per:parents	*per:siblings*	*per:children*	SUBJ-PERSON SUBJ-PERSON SUBJ-PERSON had been on safari in South Africa with his brother Enzo, 11, mother OBJ-PERSON …
	no_relation	*per:children*	… SUBJ-PERSON had been on safari in South Africa with his mother Trudy, 41, father OBJ-PERSON …

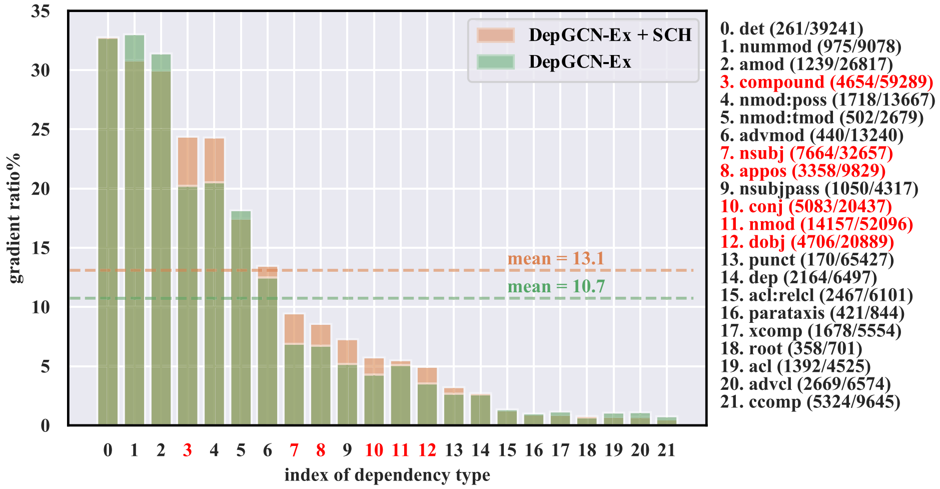
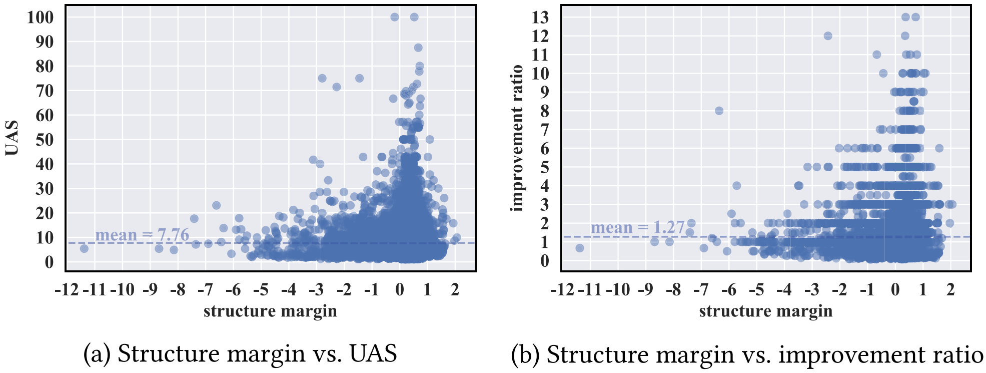


Fig. 6. Averaged gradient ratio of the posterior allocated on certain dependency types of edges on the shortest dependency path between entities. Dashed lines indicate the mean value of the gradient ratio for each model. On the right side is the mapping between the dependency type and its index, and the corresponding frequency of the dependency type on the shortest dependency path/in the whole test set.

implictly, and reasoning along the dependency is needed. For relation *per:parents*, topic words such as *brother* and *mother* show up in high frequency (0.43 and 0.18, respectively), and models without sufficient use of dependency structures are easily distracted by these words.

Given the preceding observations and Figure 1, we further make the hypothesis that SCHTo verify this hypothesis, we calculate the dependency-edge saliency for all examples as we do in Figure 1, and we report the averaged gradient ratio of posterior allocated on the shortest dependency paths between entities. The result is shown in Figure 6. We can find that in general, more gradient is allocated on the shortest dependency path by DepGCN trained with SCH loss than CE loss. This supports the idea that SCH loss forces DepGCNs to pay more attention to the key dependency paths and rely on them to make decisions. Specifically, there are more gradients allocated on the dependency types *compound*, *nsubj*, *appos*, *conj*, and *dobj* (indicated by the red x-ticks in Figure 6) for the model trained with SCH than with CE loss, which are high-frequency types on



(a) Structure margin vs. UAS (b) Structure margin vs. improvement ratio

Fig. 7. Scatter plot for structure margin vs. UAS of the induced trees from DepGCN-In+SCH (a) and the improvement ratio of UAS over the case of CE loss (b). The purple dashed lines indicate the mean value.

the shortest dependency paths. In turn, DepGCNs trained with CE loss rely on the key dependency path less and are easy to overfit some superficial features.

We also analyze the cases when all models cannot differentiate relations and find that the wrong predictions concentrate on predicting examples from *org:founded_by* as *org:top_members/ employees* and predicting examples from *per:origin* as *per:countries_of_residence*. The distinctions within these relation pairs are indeed hard to resolve, as these concepts heavily overlap with each other and a much larger range of context is needed for making precise predictions, which is usually not provided by the TACRED dataset. For example, people who found a company can also be its top members or employees, and we cannot verify whether a resident from a country is born in a particular country or not until we have read the mention about the birthplace. As the deficiency of the SCH loss, we find that there are four examples from relation *per:parents* wrongly predicted as *per:other_family* (correctly predicted by the other models), in which low-frequency words *stepmother* and *stepfather* appear instead of *mother* and *father*. However, these sparse word features never appear in the relation class *per:other_family*, indicating that using SCH loss may excessively focus on the structure information and slightly ignore the importance of learning some word representations.

5.3 Structure Margin vs. Dependency Induction

To answer the second question, we record both the structure margin and the corresponding unlabeled attachment score (UAS) of the induced parsing tree for each given text. The scatter plot between UAS and structure margins for DepGCN-In + SCH is shown in Figure 7(a). We find when the structure margin increases, the maximum of possible UAS also increases, but the variance is large. This implies that the structure margin has a weak positive relation with the performance of dependency induction. We also calculate the improvement ratio of UAS for DepGCN-In + SCH over the model trained with CE loss. The scatter plot of the result is shown in Figure 7(b). We can observe that only when the structure margin is large enough can the improvement ratio be high. This observation implies that when the example strongly relies on the structure information, the quality of the induced tree should be high for better understanding of it. The mean value of the improvement ratio of UAS is 1.27, indicating that SCH loss can help tree induction.

We also analyze the accuracy of different dependency types of the induced trees generated by DepGCN-In + SCH and Dep-GCN-In. For better understanding, we also include a random-branching baseline in this analysis. The result is shown in Figure 8. In general, the figure shows that there is bias on dependency types for both models. Specifically, both models perform better

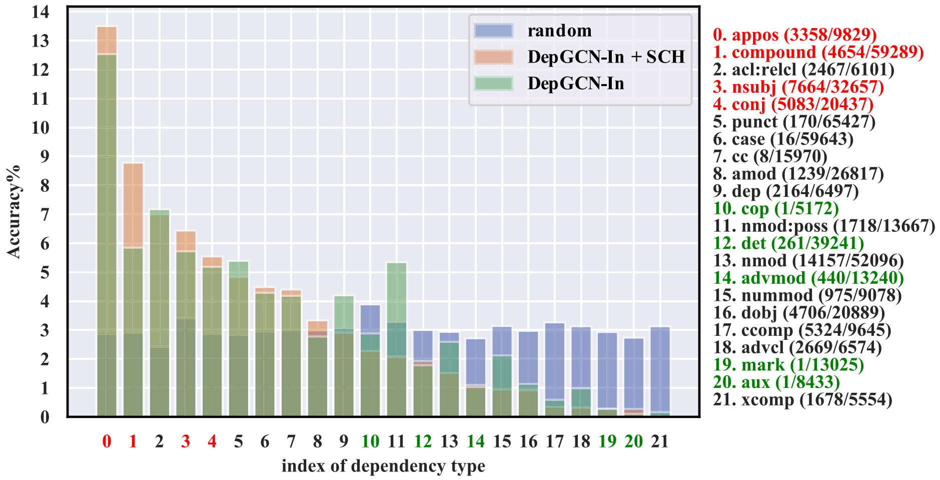


Fig. 8. Accuracy of different dependency types of induced trees generated by DepGCN-In+SCH, DepGCN-In, and the random-branching baseline. On the right side is the mapping between the dependency type and its index, and the corresponding frequency of the dependency type on the shortest dependency path/in the whole test set.

than random branching on some high-frequency dependency types on the shortest dependency paths, such as *appos*, *compound*, and *nsubj* and *conj* (indicated by the red x-ticks). In addition, for some low-frequency dependency types on the shortest dependency paths, such as *cop*, *det*, *advmod*, *mark*, and *aux* (indicated by the green x-ticks in Figure 8), both models perform worse than the random level. For most of the dependency types well captured by both models, the model trained with SCH loss performs better. This observation also supports that SCH loss helps the model to induce trees of higher quality. However, for a few dependency types, such as *dobj* and *nmod*, which are also high-frequency on the shortest dependency paths, the induction performance is relatively low for both models. This observation is also reflected in Figure 6 for the case external parsers, where the same dependency types are not well focused on. We assume that this is a potential obstacle for better task performance, and we leave this for future work.

6 RELATED WORK

Recently, incorporating syntactic structures in neural network models to better encoding text has enjoyed great interest. We here introduce two types of related work: (1) utilizing syntactic structures in different ways and (2) investigating the role of syntactic structures for neural text representation.

6.1 Utilizing Syntactic Structures

One of the most simple ways of incorporating syntactic structures into neural text representations is embedding them together with words as additional input features [39, 49]. Another convenient way is to distill syntactic knowledge into contextualized word representations through multi-task learning, viewing syntactic parsing as an auxiliary task [45, 56]. These two approaches do not consider modeling syntactic connections (i.e., edges in a parsing tree) and thus cannot benefit from gradient shortcuts guided by the syntactic structures [7]. As a result, long-term dependency problem appears.

To model the syntactic connections directly, one of the most popular and attractive approaches is to use tree-structured RecNNs [10, 33, 43, 46] on top of RNN layers to represent text into vectors.

Although this approach matches the intuitions of semantic composition, the recursive construction process is hard to parallelize, resulting in bad efficiency. GCN models [22], however, run on graph structures in a highly parallel way and model the dependency relation much better in many previous works [2, 14, 32, 58].

As the shortcoming of utilizing syntactic structures from external parsers or manual labeling, the cost in terms of time complexity and money is high and the flexibility of the structures is low. Thus, representing text with induced syntactic structures is a hot topic [1, 6, 28, 34]. These models usually involve parameterizing dependency relation with attention scores and composing constituents with discrete optimization methods such as Reinforce and Gumbel-Softmax [30].

6.2 Investigating the Importance of Syntactic Structures

As shown in many previous studies, neural text representations have already been syntax aware, as syntactic knowledge can be probed in these representations [37, 38, 42, 48]. Thus, to what extent explicitly modeling syntactic structures improves the quality of neural text representations remains unclear. However, little attention has been paid to exploring syntactic contribution or structure awareness for these models on NLP tasks. He et al. [16] add noise at different levels to examine how much syntax contributes to dependency-feature-based neural SRL systems. This approach is only used as a diagnostic evaluator, without improving the modeling performance of dependency-based text representations. Instead, our proposed loss method can both interpret sample-level structure awareness and boosting the representation quality.

7 CONCLUSION

This article proposes a margin-based loss function—SCH loss—for improving the structure awareness of dependency-based text representations. Equivalent to maximizing the conditional mutual information of the structure information and the model decision given a text, SCH loss is formulated as enlarging the structure margin between structural representations and non-structural ones. One can also interpret to what extent understanding one example relies on its structure information by this margin value. Experiments on both English and Chinese datasets show that SCH loss can enhance text representation quality of dependency-based GCNs with induced structures or external structures. Detailed analysis is also conducted to show how the structure awareness is correlated with task performance and induced structures, and the models trained with SCH loss tend to focus more on the shortest dependent path between entities. It is our hope that this work inspires more research on evaluating and improving structure awareness of text representation models.

REFERENCES

[1] Joost Bastings, Wilker Aziz, Ivan Titov, and Khalil Sima'an. 2019. Modeling latent sentence structure in neural machine translation. arxiv:1901.06436.

[2] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 1957–1967. https://aclanthology.info/papers/D17-1209/d17-1209.

[3] Yonatan Bisk and Ke Tran. 2018. Inducing grammars with and for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation (NMT@ACL'18)*. 25–35. https://aclanthology.info/papers/W18-2704/w18-2704.

[4] Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16), Volume 1: Long Papers*. 756–765. http://aclweb.org/anthology/P/P16/P16-1072.pdf.

[5] Daniel Cer, Yinfei Yang, Sheng-Yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, et al. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18): System Demonstrations* 169–174. https://aclanthology.info/papers/D18-2029/d18-2029.

[6] Jihun Choi, Kang Min Yoo, and SangGoo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18).* 5094–5101. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16682.

[7] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17): Conference Track Proceedings.* https://openreview.net/forum?id=S1di0sfgl.

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

[9] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14).* 193–202. DOI: https://doi.org/10.1145/2623330.2623758

[10] Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17), Volume 2: Short Papers.* 72–78. DOI: https://doi.org/10.18653/v1/P17-2012

[11] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2017. Learning generic sentence representations using convolutional neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17).* 2390–2400. https://aclanthology.info/papers/D17-1254/d17-1254.

[12] Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18): Conference Track Proceedings.* https://openreview.net/forum?id=r1dHXnH6-.

[13] Matthew R. Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15).* 1774–1784. http://aclweb.org/anthology/D/D15/D15-1205.pdf.

[14] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19).* 241–251. DOI: https://doi.org/10.18653/v1/P19-1024

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16).* 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

[16] Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18), Volume 1: Long Papers.* 2061–2071. https://aclanthology.info/papers/P18-1192/p18-1192.

[17] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'16).* 1367–1377. http://aclweb.org/anthology/N/N16/N16-1162.pdf.

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[19] Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15).* 1373–1378. https://aclweb.org/anthology/D/D15/D15-1162.

[20] Sébastien Jean and Kyunghyun Cho. 2019. Context-aware learning for neural machine translation. arxiv:1903.04715.

[21] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14), Volume 1: Long Papers.* 655–665. http://aclweb.org/anthology/P/P14/P14-1062.pdf.

[22] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17): Conference Track Proceedings.* https://openreview.net/forum?id=SJU4ayYgl.

[23] Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07).* 141–150. http://www.aclweb.org/anthology/D07-1015.

[24] Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'16).* 681–691. http://aclweb.org/anthology/N/N16/N16-1082.pdf.

[25] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18), Volume 2: Short Papers*. 138–143. http://aclweb.org/anthology/P18-2023.

[26] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated graph sequence neural networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR'16): Conference Track Proceedings*. http://arxiv.org/abs/1511.05493

[27] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC: A large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'18)*. 1952–1962. https://aclanthology.info/papers/C18-1166/c18-1166.

[28] Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics* 6 (2018), 63–75. https://transacl.org/ojs/index.php/tacl/article/view/1185.

[29] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL'15), Volume 2: Short Papers*. 285–290. http://aclweb.org/anthology/P/P15/P15-2047.pdf.

[30] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17): Conference Track Proceedings*. https://openreview.net/forum?id=S1jE5L5gl.

[31] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14): System Demonstrations*. 55–60. http://aclweb.org/anthology/P/P14/P14-5010.pdf.

[32] Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 1506–1515. https://aclanthology.info/papers/D17-1159/d17-1159.

[33] Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16), Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1105.pdf.

[34] Vlad Niculae, André F. T. Martins, and Claire Cardie. 2018. Towards dynamic computation graphs via sparse latent structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*. 905–911. https://aclanthology.info/papers/D18-1108/d18-1108.

[35] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 4 (2016), 694–707. DOI: https://doi.org/10.1109/TASLP.2016.2520371

[36] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14), a Meeting of SIGDAT, a Special Interest Group of the ACL*. 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

[37] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18), Volume 1 (Long Papers)*. 2227–2237. https://aclanthology.info/papers/N18-1202/n18-1202

[38] Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP@EMNLP'18)*. 287–297. https://aclanthology.info/papers/W18-5431/w18-5431.

[39] Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the 1st Conference on Machine Translation (WMT'17), Colocated with ACL 2016*. 83–91. http://aclweb.org/anthology/W/W16/W16-2209.pdf.

[40] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proceedings of the 7th International Conference on Learning Representations (ICLR'19)*. https://openreview.net/forum?id=B1l6qiR5F7.

[41] Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. arxiv:1904.05255.

[42] Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 1526–1534. http://aclweb.org/anthology/D/D16/D16-1159.pdf.

[43] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the*

2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13), a Meeting of SIGDAT, a Special Interest Group of the ACL. 1631–1642. https://aclanthology.info/papers/D13-1170/d13-1170.

[44] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. 2018. Super characters: A conversion from sentiment classification to image classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment, and Social Media Analysis (WASSA@EMNLP'18)*. 309–315. https://aclanthology.info/papers/W18-6245/w18-6245.

[45] Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 3772–3782. https://aclanthology.info/papers/D18-1412/d18-1412.

[46] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL'15), Volume 1: Long Papers*. 1556–1566. http://aclweb.org/anthology/P/P15/P15-1150.pdf.

[47] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. 1422–1432. http://aclweb.org/anthology/D/D15/D15-1167.pdf.

[48] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. arxiv:1905.05950.

[49] Yufei Wang, Mark Johnson, Stephen Wan, Yifang Sun, and Wei Wang. 2019. How to best use syntax in semantic role labelling. arxiv:1906.00266.

[50] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 4144–4150. DOI: https://doi.org/10.24963/ijcai.2017/579

[51] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING'16): Technical Papers*. 1340–1349. http://aclweb.org/anthology/C/C16/C16-1127.pdf.

[52] Larry Wasserman. 2000. Bayesian model selection and model averaging. *Journal of Mathematical Psychology* 44, 1 (2000), 92–107.

[53] Ji Wen, Xu Sun, Xuancheng Ren, and Qi Su. 2018. Structure regularized neural network for entity relation classification for Chinese literature text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18), Volume 2 (Short Papers)*. 365–370. https://aclanthology.info/papers/N18-2059/n18-2059.

[54] Yunlun Yang, Yunhai Tong, Shulei Ma, and Zhi-Hong Deng. 2016. A position encoding convolutional neural network based on dependency tree for relation classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 65–74. DOI: https://doi.org/10.18653/v1/D16-1007

[55] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'16)*. 1480–1489. http://aclweb.org/anthology/N/N16/N16-1174.pdf.

[56] Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19), Volume 1 (Long and Short Papers)*. 1151–1161. https://aclweb.org/anthology/papers/N/N19/N19-1118/.

[57] Xiang Zhang and Yann LeCun. 2017. Which encoding is the best for text classification in Chinese, English, Japanese and Korean?arxiv:1708.02657.

[58] Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 2205–2215. https://aclanthology.info/papers/D18-1244/d18-1244.

[59] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 35–45. https://aclanthology.info/papers/D17-1004/d17-1004.

[60] Xiao-Dan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*. 1604–1612. http://jmlr.org/proceedings/papers/v37/zhub15.html.

Received August 2019; revised December 2019; accepted March 2020