Contextualized Non-Local Neural Networks for Sequence Learning

Pengfei Liu,[†‡] Shuaichen Chang, Xuanjing Huang,[†] Jian Tang,[‡] Jackie Chi Kit Cheung[‡♯]
[†]School of Computer Science, Fudan University, Shanghai Insitute of Intelligent Electronics & Systems
[‡]MILA & [♯]McGill University & The Ohio State University
{pfliu14,xjhuang}@fudan.edu.cn, chang.1692@osu.edu,jian.tang@hec.ca,jcheung@cs.mcgill.ca

Abstract

Recently, a large number of neural mechanisms and models have been proposed for sequence learning, of which self-attention, as exemplified by the Transformer model, and graph neural networks (GNNs) have attracted much attention. In this paper, we propose an approach that combines and draws on the complementary strengths of these two methods. Specifically, we propose contextualized non-local neural networks (CN$^{\mathbf{3}}$), which can both dynamically construct a task-specific structure of a sentence and leverage rich local dependencies within a particular neighbourhood.

Experimental results on ten NLP tasks in text classification, semantic matching, and sequence labelling show that our proposed model outperforms competitive baselines and discovers task-specific dependency structures, thus providing better interpretability to users.

Introduction

Learning the representation of sequences is a fundamental task, which requires deep understanding of both the **complex structure** of sentences (Biber, Conrad, and Reppen 1998) and the **contextualized representation** of words (Peters et al. 2018). Recent successful approaches replace the classical compositional functions in neural networks (i.e. CNNs or RNNs) with mechanisms based on self-attention, of which the most effective model is the **Transformer**, having achieved state-of-the-art performance on machine translation (Vaswani et al. 2017) and parsing (Kitaev and Klein 2018). The success of Transformer can be attributed to its non-local structure bias, in which dependencies between any pair of words can be modelled (Baerman 2015, Wang et al. 2017). This property allows it to dynamically learn both the syntactic and semantic structures of sentences (Vaswani et al. 2017). Despite its success, the lack of local bias (local dependencies that exist over adjacent words (Baerman 2015, Futrell 2017)) limits its capacity for learning contextualized representations of words[1].

In contrast to Transformer, another line of work aims to model sequences with **graph neural networks** (GNNs). While GNNs are capable of learning local contextual information flexibly by encoding attribute features (Battaglia et al. 2018), it is less clear how to effectively utilize GNNs for sequence learning, since there is no single structural representation of a sentence that is well suited to all tasks.

Challenges	**Bias**	**G.**	**T.**	**Ours**
Com-Structure	Non-local	×	✓	✓
Con-Representation	Local	✓	×	✓

Table 1: Com-Structure denotes "**complicated structures of sentences**" while Con-Representation denotes "**contextualized representations of words**". G. and T. represent graph neural network and Transformer respectively. The first two columns shows main challenges for sentence learning and corresponding required bias: **Non-local bias**: dependencies can be built with any neighbor (Baerman 2015, Wang et al. 2017). **Local bias**: local dependencies exist over adjacent word (Baerman 2015, Futrell 2017). The last three columns list comparison of typical sequence learning approaches for incorporating local and non-local bias.

A common choice is to use the syntactic dependencies between words in a sentence to determine the sentential graph structure, which reduces the model to a tree-structured neural network (Tai, Socher, and Manning 2015). Inspired by Transformer (Vaswani et al. 2017), we aim to improve upon this fixed structure by learning a task-dependent graphical representation, which should better capture the dependencies that matter for the end task.

In this paper, we draw from both lines of research and propose a model which can benefit from their complementary strengths. On the one hand, the success of Transformer motivates us to explore dynamic "GRAPH" construction. Rather than defining a hard-coded graph for the sentence, we incorporate non-local bias into GNNs, learning the sentence structure dynamically on different tasks. On the other hand, the advantage of the graph neural network framework itself is to provide rich local information by encoding node or edge attributes, which can make up for the deficiency in Transformer.

Consequently, we propose a contextualized non-local networks by extending self-attention (Transformer) to GNNs, which support highly flexible representations in two ways. First, the representations of attributes (feature encodings of

[1] Our experiments also show that Transformer performs worse than the models with local bias, especially on sequence labeling tasks.

nodes and edges); second, the structure of the graph itself (dynamic learning of task-dependent structures). Notably, these two advantages of the proposed model enable us to better learn the representation of sequences in terms of contextualized representation of words and complicated structures of sentences.

We conduct extensive experiments on ten sequence learning tasks, from text classification and semantic matching, to sequence labelling. Experimental results show that our proposed approach outperforms competitive baselines or achieves comparable results in all tasks. Additionally, we are able to discover task-dependent structures among words and provide better interpretablility to users.

We summarize our contributions as below:

1. We analyze two challenges for sequence learning (structure of sentences and contextualized word representations) from the perspective of models' locality biases, which encourages us to find the complementarity between Transformer and graph neural networks.
2. We draw on the complementary strengths of Transformer and GNNs, proposing a contextualized non-local neural network ($\text{CN}^{\mathbf{3}}$), in which structures of sentences can be learned on the fly and words in sentence can be contextualized flexibly.
3. We perform a comprehensive analysis of existing sequence learning approaches in a unified way and conduct extensive experiments on a number of natural language understanding tasks. Experimental results show that our proposed models achieve comparable results in these tasks as well as offer better interpretability to users.

Related Work

In this paper, we propose to utilize the **locality bias** to systematically analyze existing sequence learning models in a unified way. Here, the locality bias is one kind of inductive bias that local dependencies exist over adjacent words, which is first well established in phonology (van der Lely 2005) and then also be explored in natural language (Baerman 2015, Futrell 2017). Typically, local and non-local bias are two common locality biases implicitly existing in different models for sequence learning.

Next, we will elaborate on explanations following related lines and as shown in Table 2, we present a summary of existing models by highlighting differences among interaction methods for different words, locality bias, as well as the eligible Tasks that corresponding methods can handle.

Neural network-based Sequence Modelling. Neural networks provide an effective way to model the dependencies between different words via parameterized composition functions. Some examples of composition functions involve recurrent neural networks with long short-term memory (LSTM) (Liu et al. 2015), convolutional neural networks (CNN) (Kalchbrenner, Grefenstette, and Blunsom 2014), and tree neural networks (Tai, Socher, and Manning 2015, Zhu, Sobhani, and Guo 2015). There are two major differences between these methods: one is the scope in which words can interact with each other. The other is the compositional functions they used. For example, LSTMs can explicitly model the dependencies between the current word and previous ones, and words can interact with others within the same window. More examples can be found in Table 2.

Attention-based Sequence Modelling. Several attention-based mechanisms have been introduced to expand the interaction scope, enabling more words to interact with each together. For example, Yang et al., Lin et al. [2016, 2017] utilize a learnable query vector to aggregate the weighted information of each word, which can be used to get sentence-level representations for classification tasks. Cheng, Dong, and Lapata [2016] augment neural networks with a re-reading ability while processing each word. Vaswani et al. [2017] proposes to model the dependencies between words based entirely on self attention without any recurrent or convolutional layers. The so-called Transformer (SelfAtt3) has achieved state-of-the-art results on a machine translation task. In contrast with previous work, we introduce different locality biases to learn complex structure of sentences and leverage rich local dependencies within a particular neighbourhood.

In computer vision community, Wang et al. [2017] also incorporate non-local bias into neural network for image representation. However, in this work, we focus on sequence learning, which is different from image processing and requires rich contextual information.

Graph Convolutional Neural Network. Graph convolutional neural networks (GCNN) have been used to learn the representations of graph structure data. Gilmer et al. [2017] propose a general neural message passing algorithm to predict the properties of molecule structures. Velickovic et al. [2017] proposes graph attention networks to model graph-structure data, such as protein and citation networks. While "GRAPH" has been well-studied in above areas, it's still less clear how to build a graph for a sentence. Besides, our work is different from (Velickovic et al. 2017) lying in the following aspects: 1) We propose to incorporate non-local bias into GNNs while Velickovic et al. [2017] aims to combine attention with GNNs. 2) We focus on sentence understanding, which also calls for contextualized word representations. Some existing works Marcheggiani and Titov [2017] try to construct a graph from a sequence based on pre-defined structures, such as syntactic dependencies, which are not fit to the complexity of sentences' structures since in a specific task, the true dependency structures among the words in a sentence can significantly differ from the given input structures (Vaswani et al. 2017). In this paper, we integrate non-local bias into GNNs aim to learn the structures of sentences on the fly based on different tasks.

Contextualized Non-local Neural Networks (CN^3)

To better learn the representation of sequences in terms of contextualized representations of words (Peters et al. 2018)

Models	Interaction		Locality Bias	Eligible Tasks
	Scope	Functions		
BOW (Fan et al. 2008)	None	None	Local	T-Sent & T-Word
S-Graph (Mihalcea and Tarau 2004)	All pairs of words	$PMI(\mathbf{w}_i, \mathbf{w}_j)$	Non-Local	T-Sent
CNN (Kalchbrenner, Grefenstette, and Blunsom 2014)	Fixed window	$f_\theta(\mathbf{w}_i) * g_\theta(\mathbf{w}_j)$	Local	T-Sent
SeqRNN (Liu et al. 2015)	Adjacent words	$f_\theta(\mathbf{w}_i, \mathbf{w}_j)$	Local	T-Sent
TreeRNN (Tai, Socher, and Manning 2015, Zhu, Sobhani, and Guo 2015)	Child words	$f_\theta(\mathbf{w}_i, \mathbf{w}_j)$	Local	T-Sent
GraphCNN (Marcheggiani and Titov 2017)	Child words	$f_\theta(\mathbf{w}_i, \mathbf{w}_j)$	Local	T-Word
SelfAtt1 (Yang et al. 2016, Lin et al. 2017)	Adjacent words	$f_\theta(\mathbf{w}_i, \mathbf{w}_j)$	Local	T-Sent
SelfAtt2 (Cheng, Dong, and Lapata 2016)	Preceding words	$\sum_{i\in\Phi} \alpha_i \cdot f_\theta(\mathbf{w}_i)$	Local	T-Sent & T-Word
SelfAtt3 (Vaswani et al. 2017)	All pairs of words	$\sum_{i\in\Phi} \alpha_i \cdot [\mathbf{w}_i, \mathbf{p_i}]$	Non-local	T-Word
Our Work	All pairs of words	$\sum_{i\in\Phi} \alpha_i \cdot f_\theta(\mathbf{w}_i)$	Contextualized Non-local	T-Sent & T-Word

Table 2: A comparison of published approaches for sentence representations. S-Graph denotes statistical graph. $f(\cdot)$ and $g(\cdot)$ are parameterized functions. $\mathbf{w}$ denotes the representation of word and $\mathbf{p}$ is a vector relating to positional information. α_i is a scalar and Φ is a set of indexes for those words within a interaction scope. The **2nd column** shows which part of words can interact with each other while the **3rd column** shows how to compose these words. The **4th column** lists what kinds of knowledge the interaction methods depend on. The **last column** represents which tasks can be processed by corresponding models. "T-Sent" denotes those tasks who require sentence-level representations, such as text classification, while "T-word" represent tasks which rely on word-level representations, such as sequence labelling, machine translation. "-": S-Graph usually calls for complicated ranking algorithm and can not directly obtain a real-valued vector to represent the sentence meanings. "*" SelfAtt3 is proposed for machine translation, rather than sentence level representation learning.

and complex structures (Biber, Conrad, and Reppen 1998), we propose the contextualized non-local network, which incorporates non-local bias into graph neural networks. Generally, CN$^{\mathbf{3}}$ can not only support the encoding of local contextual information, it can also learn task-dependent structures on the fly.

Next, we introduce the contextualized non-local network, which involves two steps. The first is to construct graphs from sentences and introduce contextualized information from attributes. The second is to dynamically update the graph structures for specific tasks. The overall learning process for learning task-dependent graphs is illustrated in Algorithm 1.

Graph Construction

The first part of our framework is to construct meaningful graphs from sentences in order to encode different types of local contextual information.

Nodes Given a sentence $X = x_1, x_2, \cdots, x_n$, we treat each word x_i as a node. For each node i, we define NODE

ATTRIBUTES ($\mathcal{S}_{v_i} = v^1, \cdots, v^{|\mathcal{S}_v|}$) as different types of extra information besides the word identity. In this paper, we list several typical node attributes described as follows:

Position Information The position of each word in a sentence.

Contextual Information Information of nodes can also be enriched by short-term contexts, which can be obtained by convolutional neural network or long-short term memory network. Introducing this information means that we have augmented the model with a local bias: adjacent units are prone to provide useful information (Battaglia et al. 2018, Cheng, Dong, and Lapata 2016).

Tag Information More prior knowledge such as POS tags can be encoded into the nodes.

Edges For each edge between any word pair (x_i, x_j), we define EDGE ATTRIBUTES ($\mathcal{S}_{e_ij} = e^1_{ij}, \cdots, e^{|\mathcal{S}_{e_{ij}}|}_{ij}$) as combined relationships, such as lexical relationships, syntactic dependency obtained from parsing tree, or co-occurrence relationships.

Contextualized Node Representations Each word and attribute will be first converted to be a real-valued vector through a look-up table, and if only the word identity is used, the node representations are simply as word representations. The information of word then will be enriched by fusing its attributes therefore constructing contextualized node representations.

Dynamic Graph Structure Updating

Non-local bias in CN**3** is realized as pairwise interactions between any two words, which makes it possible to dynamically learn sentence structures that are geared towards specific tasks. Specifically, The model first encodes each node in a low-dimensional vector. Afterwards, different nodes communicate with each other via messages, which updates the graph structures and the node representations. After the nodes are represented by low-dimensional vectors, we create multiple message passing layers $l = 1, 2, \ldots, L$, which either update the graph structure or the node representations.

Graph Structure Updating The graph structures are updated by promoting communication between different nodes. Specifically, let α_{ik} denote the strength of the relationship between nodes i and k. We update α_{ik} according to the following formula:

$$\begin{aligned} s_{ik} &= f(\mathbf{h}^l_k, \mathbf{h}^l_i, \mathbf{v}_i, \mathbf{v}_k, e_{ik}) && (1)\\ &= \mathbf{u}^T \tanh(\mathbf{W}[\mathbf{h}^l_k, \mathbf{h}^l_i, \mathbf{v}_i, \mathbf{v}_k, e_{ik}]) && (2)\\ \alpha_{ik} &= \mathrm{Softmax}(s_{ik}), && (3) \end{aligned}$$

where $\mathbf{v}$ denotes the node attributes. $e_{i,k}$ represents the edge attributes between node i and k, $\mathbf{h}^l_i$ is the representation of node i in layer l. $\mathbf{u}$ and $\mathbf{W}$ are parameters that can be learned by backpropagation. $\mathbf{h}^0$ denotes the word embedding.

Algorithm 1 Learning Processes of Contextualized Non-local Neural Networks for Sequences

```
Require: Tag sequence $Y = \{y_1, y_2, \ldots, y_T\}$ and text sequence $X = \{x_1, x_2, \ldots, x_T\}$ from a specific task.
Require: A set of node attributes $\mathcal{S}_v = v^1, \cdots, v^{|\mathcal{S}_v|}$ and edge attributes $\mathcal{S}_e = e^1, \cdots, e^{|\mathcal{S}_e|}$
for $l \in \{1 \cdots L\}$ do
    for $i \in \{1 \cdots N\}$ do                          ▷ Graph Construction
        // Contextualized representations
        $\mathbf{v_i}$ = Concat-RealValue($\mathcal{S}_v$)
        for $j \in \{1 \cdots N\}$ do
            $\mathbf{e_{ij}}$ = Concat-RealValue($\mathcal{S}_e$)
        end for
    end for
    for $i \in \{1 \cdots N\}$ do                          ▷ Graph Learning
        for $j \in \{1 \cdots N\}$ do
            // Dynamic Structure Updating
            ${\alpha_{ij}}^l \leftarrow$Edge-Update($\mathbf{h}^l_i, \mathbf{h}^l_j, \mathbf{v_i}, \mathbf{v_j}, \mathbf{e_{ij}}, \theta_\mathbf{e}$)
        end for
        $\mathbf{h}^{l+1}_i \leftarrow$ Node-Update($\mathcal{H}^l, \boldsymbol{\alpha}^l_i$)
    end for
                                                           ▷ Application Layer
    if Node-Level then
        $p(Y|X) = \mathrm{CRF}(\mathbf{h}^L_i, \theta^{(c)})$
    else if Graph-Level then
        $\mathbf{h}^L = \frac{1}{N}\sum_i \mathbf{h}^L_i,$
        $p(Y|X) = \mathrm{Softmax}(\mathbf{W}\mathbf{h}^L + \mathbf{b})$
    end if
end for
```

Node Updating Once the graph structure α_{ik} is updated, for each node k, it first aggregates information from its neighbors. Specifically, let $\tilde{\mathbf{h}}^l_k$ denote the information collected from neighbors for node k in the l-th graph layer, which can be simply calculated as:

$$\tilde{\mathbf{h}}^l_k = \boldsymbol{\alpha_i}\mathcal{H}^l = \sum_i^m \alpha_{ik}\mathbf{h}^l_i \qquad (4)$$

Afterwards, we update the node representations based on the current node representations $\mathbf{h}^l_k$ and the information aggregated from its neighbors $\tilde{\mathbf{h}}^l_k$. Here, we choose a gating-based updating methods.

$$\mathbf{h}^{l+1}_k = \mathbf{g} \odot \tilde{\mathbf{h}}^l_k + (1 - \mathbf{g}) \odot \mathbf{h}^l_k \qquad (5)$$

where $\odot$ denotes element-wise multiplication, and

$$\mathbf{g} = \sigma(\mathbf{W}\mathbf{h}^l_k + \mathbf{b}), \qquad (6)$$

where σ represents the logistic function. $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters.

Application Layers

After multiple steps of graph structure updating ($l = 1, 2, \ldots, L$), we can obtain node-level representations of the final layer L : $\mathbf{h}^L_1, ..., \mathbf{h}^L_m$, which can be used for different

NLP tasks when followed by different output layers. Here, we show how to utilize these representations at the node level for POS, Chunking, NER tasks, and graph-level for text classification, semantic matching tasks.

Node-level Representations A direct application in this scenario is sequence labelling, which aims to assign a tag sequence $Y = \{y_1, y_2, \dots, y_T\}$ to a text sequence $X = \{x_1, x_2, \dots, x_T\}$. The output at each time step $\mathbf{h}_i^L$ can be regarded as the representation of the preceding subsequence, which is then fed into a CRF layer that calculates scores for corresponding tag categories. Then, in the CRF layer, the conditional probability $p(Y|X)$ is formalized as:

$$p(Y|X) = \mathrm{CRF}(\mathbf{h}_i^L, \theta^{(c)}) \tag{7}$$

where CRF represents the CRF layer and $\theta^{(c)}$ is learnable parameters. For detailed formulation of the CRF layer, see additional resources.

Graph-level Representations In this scenario, we should compute a representation for the whole graph. The simplest way is to take the average of all the node representations:$\mathbf{h}^L = \frac{1}{m}\sum_i \mathbf{h}_i^L$

Then, the representation $\mathbf{h}^L$ can be further fed into a softmax function to yield a probability distribution over the output labels for text classification or semantic matching tasks.

Experiments

In this section, we evaluate the effectiveness of our proposed method on ten tasks. Next, we will give brief descriptions of the tasks and datasets. For more details of the above dataset descriptions and training hyper-parameters, please see the Additional Resources section.

Tasks and Datasets

We evaluate our models on five text classification tasks, two semantic matching tasks and three sequence labelling tasks.

- **Text Classification**: QC (Question Classification); SST2 (the Stanford Sentiment Treebank); MR (The movie reviews with two classes (Pang and Lee 2005)); IMDB (Long text movie reviews.)
- **Semantic Matching**: In this task, given two sentences A and B, the model is used to determine the semantic relationship between these two sentences. We choose two typical datasets SICK (Marelli et al. 2014) and SNLI (Bowman et al. 2015) for this tasks.
- **Sequence Labelling**: We choose POS, Chunking and NER as evaluation tasks on Penn Treebank, CoNLL 2000 and CoNLL 2003 respectively.

We parse the sentences in the datasets with Stanford NLP toolkit (Manning et al. 2014) to obtain dependency relations and Part-of-Speech tags for our models and several competitor models.

Settings

To minimize the objective, we use stochastic gradient descent with the diagonal variant of AdaDelta (Zeiler 2012). The word embeddings for all of the models are initialized with GloVe vectors (Pennington, Socher, and Manning 2014). The other parameters are initialized by randomly sampling from a uniform distribution in $[-0.1, 0.1]$. For each task, we take the hyperparameters which achieve the best performance on the development set via grid search. Other detailed settings of our models can be seen in the Additional Resources section.

Quantitative Evaluation

The proposed models support highly flexible graph representations in two ways: first, in terms of the representation of the attributes (feature encoding of nodes and edges); and second, in terms of the structure of the graph itself (dynamic learning of task-dependent structure). Next, we will elaborate on these evaluations.

Evaluation on Dynamic Structure Learning Table 3 shows the performances of different models on 10 different tasks. We have following observations:

- For graph-level task, CN$^{\mathbf{3}}{}_{LSTM}$ consistently outperforms neural networks with different structural biases (sequential, tree, pre-defined graph) and attention mechanisms, indicating the effectiveness of non-local bias and dynamic learning nature for sentence structures. Particularly, compared with PDGraph, CN$^{\mathbf{3}}{}_{LSTM}$ achieves better performance and doesn't rely on external syntactic tree, with the ability to handle more longer texts, such as IMDB. Compared with LSTM, the improvement of CN$^{\mathbf{3}}{}_{LSTM}$ also indicates the functions of local and non-local biases are complementary.
- For node-level tasks (sequence labelling), CN$^{\mathbf{3}}{}_{LSTM}$ achieves comparable results as opposed to CNN (which utilizes mulit-task learning framework) and LSTM (which additionally introduces many external features: gazetteer features and spelling features).

Evaluation on Attributes The proposed model has the advantage of being able to contextualize words by encoding information of nodes' or edges' attributes. Here we use superscripts and subscripts to introduce nodes' or edges' attributes. Table 3 illustrates:

- Compared to SelfAtt3 (Transformer), CN$^{\mathbf{3}}{}_{LSTM}$ obtained substantial improvements, especially on SST dataset constructed by lots of sentences with complicated sentence patterns. For those node-level tasks, CN$^{\mathbf{3}}{}_{LSTM}$ surpasses SelfAtt3 and we attribute the success to its power in both the ability of learning structures dynamically and encoding short-term contextual information.
- The performances can be enhanced when rich node or edge attributes were taken into accounts. And we observed that effects of different attributions are different. Specifically, CN$^{\mathbf{3}}{}_{LSTM+char}$ achieves best performances

Models	**Text Classification**					**Semantic Matching**		**Sequence Labelling**		
	QC	MR	SST	SUBJ	IMDB	SICK	SNLI	POS	Chunking	NER
NBOW	88.2	77.2	80.5	91.3	87.5	73.4	75.1	96.38	90.51	87.91
NEURAL NETWORK WITH DIFFERENT STRUCTURE BIAS										
CNN	92.2	81.5	**88.1**	93.2	88.5	75.4	77.8	97.20	93.63	88.67
RNN	90.2	77.2	85.0	92.1	87.0	74.9	76.8	97.30	92.56	88.50
LSTM	91.3	77.7	85.8	92.5	88.0	76.3	77.6	97.55	94.46	90.10
TreeLSTM	92.8	78.7	88.0	93.3	×	77.5	78.3	-	-	-
PDGraph	93.2	80.2	86.8	92.5	×	78.8	78.9	-	-	-
ATTENTION-BASED MODELS										
SelfAtt1	93.2	80.3	86.4	92.9	90.3	78.4	77.4	*	*	*
SelfAtt2	93.5	79.8	87.0	93.1	89.2	79.8	79.2	-	-	-
SelfAtt3	90.1	77.4	83.6	92.2	88.5	76.3	76.9	96.42	91.12	87.61
TASK-DEPENDENT GRAPHS										
CN$^{\mathbf{3}}_{LSTM}$	94.2	81.3	87.5	93.5	91.5	80.2	81.3	97.12	93.81	89.33
CN$^{\mathbf{3}}_{LSTM+POS}$	94.8	80.7	87.1	94.3	91.0	81.4	82.1	-	-	-
CN$^{\mathbf{3}}_{LSTM+char}$	94.6	**82.5**	88.0	94.0	**92.5**	81.5	82.7	97.65	94.82	90.51
CN$^{\mathbf{3}Dep}_{LSTM}$	**95.0**	81.0	87.8	**94.6**	*	**81.9**	**83.5**	-	-	-
CN$^{\mathbf{3}}_{LSTM+char+Spell}$	-	-	-	-	-	-	-	**97.78**	**95.13**	**91.10**

Table 3: Performance of the proposed models on all datasets compared to typical baselines. × indicates that corresponding models can not work since the sentences are too long to be processed by parser. $*$ denotes corresponding models can not be used for sequence labelling tasks. $-$ denotes corresponding publications don't evaluate models on related tasks. The superscript of CN$^{\mathbf{3}}$ denotes the edge attributes while subscript represents node attributes. Specifically, $LSTM$ denotes the information of each node is enriched by long-short-term memory unit. And the $Spell$ feature is an indicator that the first letter of a word is capital or small. Dep provides the information that if two node have an edge in syntactic dependency tree. POS denotes the POS taging tags while $char$ represents character information. **CNN**: We use (Kim 2014) for graph-level representation and (Collobert et al. 2011) for node-level representation. **LSTM** We use (Tai, Socher, and Manning 2015) for graph-level representation and (Huang, Xu, and Yu 2015) for node-level representation. **TreeLSTM**: LSTM over tree-structure (Tai, Socher, and Manning 2015). **PDGraph**: Pre-defined Graph Neural network based on syntactic dependency. (Marcheggiani and Titov 2017). **SelfAtt1**: A structured self-attentive sentence (Lin et al. 2017). **SelfAtt2**: Proposed by (Cheng, Dong, and Lapata 2016). **SelfAtt3**: Also known as Transformer, which is proposed by (Vaswani et al. 2017).

Model	Chunking	NER	POS
Collobert and Weston [2008]	94.32	89.59	97.29
Yang, Salakhutdinov, and Cohen [2016]	**95.41**	90.94	97.55
Peters et al. [2018]	-	**92.22**	-
Yasunaga, Kasai, and Radev [2018]	-	-	97.58
Ma and Hovy [2016]	-	91.21	97.55
Ours	95.13	91.10	**97.78**

Table 4: Performances of our model against state-of-the-art models.

on MR and IMDB datasets while CN$^{\mathbf{3}Dep}_{LSTM}$ has obtained best performances on QC, SUBJ, SICK and SNLI datasets. The reason is that the texts in QC, SUBJ, SICK and SNLI is more formal where higher accuracies on external tools (Parser or tagger) was achieved, while for MR, and IMDB, they contain more informal expressions such as “‘cooool, gooood”’, leading to the failure of external linguistic tools though they can be resolved by character-aware models.

- While introducing the same spelling features with LSTM model, the performances of CN$^{\mathbf{3}}_{LSTM+char+Spell}$ in tagging tasks can be further improved. To make an complete comparison, we have also listed the performances of our models in tagging tasks aginst state-of-the art models in Tab.4[2]. Competitor models in Tab.4 or introduce multi-task learning methods (Collobert and Weston 2008, Yang, Salakhutdinov, and Cohen 2016), or utilize more unsupervised knowledge (Peters et al. 2018, Yasunaga, Kasai, and Radev 2018), or design more handcrafted features (Ma and Hovy 2016). Rather, proposed models utilize less knowledge in terms of data or external features while achieving comparable results.

[2] Since it's hard to integrate this table into Tab.3

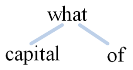
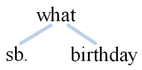
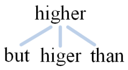
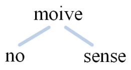
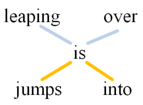
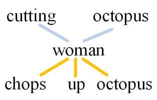
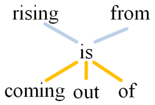
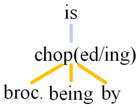
	Interpretable Sub-Structures				Explanations
Semantic					Key sentence patterns for question classification and sentiment analysis tasks
Syntactic					Key Subgraph pairs for semantic matching task. Relating to different types of phrases (verb-adverb, verb-object) and voices (active or passive)

Table 5: Multiple interpretable sub-structures generated by different tasks. For text classification tasks, we show the sub-structure of a sentence, which is a key pattern for final prediction. For semantic matching, since there are two sentences as input, we simultaneously give a pair of sub-structures with different color lines. For example, the subgraph pair in the second row and column denote the structure of "`is leaping over`" is learned in the first sentence and simultaneously "`is jumping into`" is learned in the second sentence.

Qualitative Analysis

In this section, we aim to better understand our models in terms of the following two aspects: 1) How do learned task-dependent structures contribute to different tasks? 2) How do these attributes influence the learned structures? We design a series of experiments to address these questions.

Analysis on Task-dependent Structures Since the proposed models can explicitly learn the relationship between each word pair by computing edge weights, we randomly sample several examples across different tasks (text classification and semantic matching) and visualize their learned structures.

As shown in Table 5, there are multiple interpretable sub-structures. We observe the following points:

- For several simple tasks such as text classification, the words in a sentence are usually organized so as to accurately express certain semantic information.

  For example, for a question classification task, the informative patterns constructed by the question word "`what`" can be easily learned. For a sentiment classification task, the word "`movie`" is prone to connecting to sentimental words, such as "`terrible`", "`no sense`".

- For more complex tasks such as semantic matching, a ground-truth understanding of the syntactic structure is important. In this context, we find that, given a sentence pair, our model is more likely to learn their syntactic information. For example, for the sentence pair "`A man is rising from a swamp/A man is coming out of the water`", our model learns that "`is rising from`" and "`is coming out of`" are two informative patterns in two sentences respectively, which are crucial for accurately predicting the relationship of the sentence pair ("*Entailment*").

Analysis on Attributes As the above results demonstrate, attributes (Part-of-Speech or dependency relation) will influence the learning process of relationships among different nodes. In order to obtain a better intuitive understanding, we randomly pick samples from the dataset (QC), and compare the learned dependencies among words learned by $\mathbf{CN^3}_{LSTM}$ and $\mathbf{CN^3}_{LSTM}^{Dep}$.

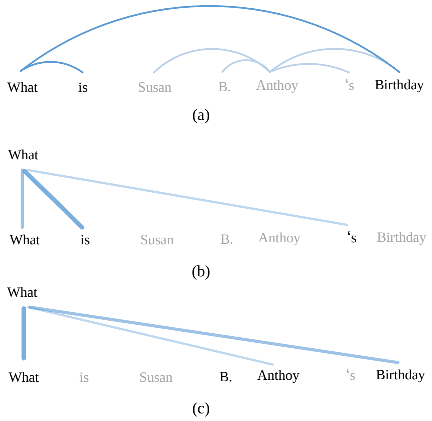


Figure 1: (a) The dependency tree obtained by the Stanford Parser. (b-c) The relationship between "`What`" and other words, which are learned by $\mathbf{CN^3}_{LSTM}$ and $\mathbf{CN^3}_{LSTM}^{Dep}$ models respectively. The correct label of "`What is Susan B. Anthoy 's birthday ?`" is "`Number`", indicating that it's a question asked about "`Number`".

As shown in Figure 1 (b-c), given a sentence, $\mathbf{CN^3}_{LSTM}^{Dep}$ makes a correct prediction about the type of this question while $\mathbf{CN^3}_{LSTM}$ fails. We note the following points.

1) With the help of edge attributes, more useful patterns can be built into the graph structure. For example, "`What`" strongly connects to the words "`What`" and "`birthday`" therefore it captures the latent sentence pattern "`What ... birthday `", which is the key to predicting the type of this question.

2) The relationships between different words are task-specific rather than exactly matching the pre-defined syntactic dependencies. Although our model gets a hint that there is strong connection between "`What`" and "`is`" from the dependency parser as shown in Figure 1 (a), yet $\mathbf{CN^3}_{LSTM}^{Dep}$ regards it as less informative. We think it simply reflects the

Conclusion

In this paper, we first analyze two challenges for sequence learning from the perspective of a model's locality biases, which motivates us to examine the complementary natures of Transformer and graph neural networks. Then, we draw on their complementary strengths to propose a contextualized non-local neural network. Experimental results show that learning task-dependent structures of sentences and contextualized word representations are crucial to many NLP tasks.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61751201, 61672162), STCSM (No.16JC1420401, No.17JC1404100), and Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Baerman, M. 2015. *The Oxford handbook of inflection*. Oxford Handbooks in Linguistic.

Battaglia, P. W.; Hamrick, A.; Raposo; et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Biber, D.; Conrad, S.; and Reppen, R. 1998. *Corpus linguistics: Investigating language structure and use*. Cambridge University Press.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on EMNLP*.

Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on EMNLP*, 551–561.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The JMLR* 12:2493–2537.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *JMLR* 9(Aug):1871–1874.

Futrell, R. L. J. 2017. *Memory and locality in natural language*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *ICML*, 1263–1272.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on EMNLP*, 1746–1751.

Kitaev, N., and Klein, D. 2018. Constituency parsing with a self-attentive encoder. *arXiv preprint arXiv:1805.01052*.

Lin, Z.; Feng, Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, P.; Qiu, X.; Chen, X.; Wu, S.; and Huang, X. 2015. Multi-timescale lstm for modelling sentences and documents. In *Proceedings of the EMNLP*.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of ACL*, volume 1, 1064–1074.

Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd ACL*.

Marcheggiani, D., and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on EMNLP*, 1506–1515.

Marelli, M.; Bentivogli, L.; Baroni, M.; Bernardi, R.; Menini, S.; and Zamparelli, R. 2014. Semeval-2014 task 1: Evaluation of compositional distributional. *SemEval-2014*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of conference on EMNLP*.

Pang, B., and Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 115–124.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. *Proceedings of the EMNLP* 12:1532–1543.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of NAACL*, volume 1, 2227–2237.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd ACL*, 1556–1566.

van der Lely, H. 2005. Domain-specific cognitive systems: Insight from grammatical specific language impairment. *Trends in Cognitive Sciences* 9(2):53–59.

Vaswani, A.; Shazeer, N.; Parmar, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in NIPS*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2017. Non-local neural networks. *arXiv preprint arXiv:1711.07971*.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of NAACL*, 1480–1489.

Yang, Z.; Salakhutdinov, R.; and Cohen, W. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.

Yasunaga, M.; Kasai, J.; and Radev, D. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the NAACL*.

Zeiler, M. D. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhu, X.-D.; Sobhani, P.; and Guo, H. 2015. Long short-term memory over recursive structures. In *ICML*, 1604–1612.