Knowledge-Based Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Improving tree-based neural machine translation with dynamic lexicalized dependency encoding☆

Baosong Yang [a], Derek F. Wong [a,*], Lidia S. Chao [a], Min Zhang [b]

[a] Natural Language Processing & Portuguese-Chinese Machine Translation Laboratory, Department of Computer and Information Science, University of Macau, Macau, China
[b] Institute of Artificial Intelligence, Soochow University, Suzhou, China

ARTICLE INFO

Article history:
Received 7 May 2019
Received in revised form 12 September 2019
Accepted 12 September 2019
Available online xxxx

Keywords:
Syntactic modeling
Dynamic parameters
Tree-RNN
Neural machine translation (NMT)

ABSTRACT

Tree-to-sequence neural machine translation models have proven to be effective in learning the semantic representations from the exploited syntactic structure. Despite their success, tree-to-sequence models have two major issues: (1) the embeddings of constituents at the higher tree levels tend to contribute less in translation; and (2) using a single set of model parameters is difficult to fully capture the syntactic and semantic richness of linguistic phrases. To address the first problem, we proposed a lexicalized dependency model, in which the source-side lexical representations are learned in a head-dependent fashion following a dependency graph. Since the number of dependents is variable, we proposed a variant recurrent neural network (RNN) to jointly consider the long-distance dependencies and the sequential information of words. Concerning the second problem, we adopt a latent vector to dynamically condition the parameters for the composition of each node representation. Experimental results reveal that the proposed model significantly outperforms the recently proposed tree-based methods in English–Chinese and English–German translation tasks with even far fewer parameters.

1. Introduction

Neural machine translation (NMT) has recently attained state-of-the-art performance in various translation tasks [1–3]. The most widely adopted framework of NMT is the encoder–decoder model [4,5], which firstly maps the source sentences into the distributed representations, then recurrently generates the target words by exploiting the soft attention mechanism [6]. Most of the encoders in prior studies employed bidirectional recurrent neural networks (RNNs) [7] to encode the source-side sequential context. Sennrich and Haddow [8], Stahlberg et al. [9] and Nooralahzadeh et al. [10] point out that the shortages of such models are lacking in considering the syntax information, whereby the dependencies between long-distance words failed to be fully modeled. In order to alleviate the problem of capturing the long-distance dependencies, several syntax-aware NMT models have recently been proposed [11–14]. From a model architecture perspective, those syntax-based NMT models can be categorized into two approaches, either modeling the syntactic structures by linearization [13,15,16] or by a tree-structured neural network [11,12,14,17].

The **linearized** syntax-aware models aim to incorporate syntactic information as prior knowledge [18] into the conventional sequence-to-sequence NMT [11,13,19,20]. Among them, Wu et al. [19] and Ma et al. [20] and [21] suggested to convert the syntactic tree into a sequence (linearization) and encode it together with the source sentences. While Li et al. [13] considered embedding the syntactic labels of words as additional features, allowing the NMT models to learn the syntactic information of sentences in an explicit way. However, the success of multi-modal learning [22,23] reveals that building the relevance between words directly is best for modeling syntax (i.e. word dependencies).

In this study, we focus on the latter approach, namely the **tree-based** methods. As illustrated in Fig. 1, the sentence is encoded according to its syntactic structure. Under this context, Socher et al. [24], Tai et al. [25] and Blevins et al. [26] have shown that further encoding the source sentence following a syntactic structure upon the sequential RNN layer benefits to capture more the source-side linguistic information. Although the tree-based methods have obtained state-of-the-art results in several translation tasks [17,27], the methods under this category still face several problems:

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105042.

* Corresponding author.
E-mail addresses: nlp2ct.baosong@gmail.com (B. Yang), derekfw@um.edu.mo (D.F. Wong), lidiasc@um.edu.mo (L.S. Chao), minzhang@suda.edu.cn (M. Zhang).

https://doi.org/10.1016/j.knosys.2019.105042
0950-7051/

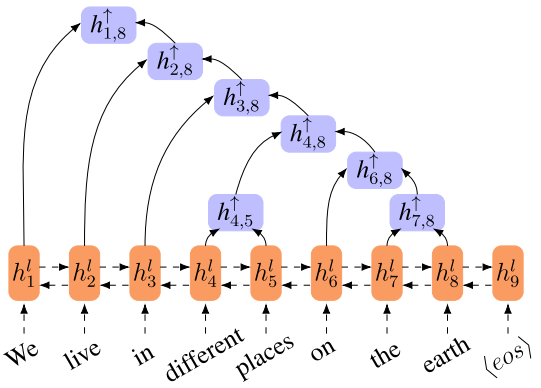


Fig. 1. An example of a conventional tree-based encoder. The leaf (lexical) nodes are generated using bidirectional RNNs, while the phrase representations are recursively summarized in the hidden states of its child nodes using tree-RNNs [25]. As seen, the conventional model recursively encodes a source sentence following its binary phrase structure.

- The lexical information is overused in the composition of the internal nodes of a constituent structure [28]. Yang et al. [12] and Chen et al. [27] noted that employing a phrase structural encoder may lead to over-translation problem. Besides, with the exploitation of the syntactic structure, the number of model parameters increases. We hypothesize that such internal (phrase) nodes may not always benefit to the neural translation model.
- The tree-RNN models adopt the child-sum architecture [25] which relies on a fixed (pre-defined) number of child nodes. Hence, the existing constituent-based models generally deal with binary tree [17,27,29]. This, in some degree, limits it from considering the remote context (sibling nodes) in computing the representation of a phrase. The word dependencies that are crucial to correctly interpret the semantic meaning of a source sentence may not directly be embodied in a consecutive phrase structure. The head and dependent words can be far apart in a sentence [19,30].
- Socher et al. [31] and Liu et al. [32] pointed out that a single set of weight parameters is not efficient enough to fully capture the syntactic and semantic richness of linguistic phrases on parsing and text classification tasks.

In the present study, we wanted to address the mentioned problems by exploiting a novel lexicalized dependency encoder, which encodes the source sentence following the dependency graph from the head to its dependents. With the dependency structure, we are able to remove the additional internal nodes introduced in the constituent structure and regard all the nodes as terminals. Due to the number of dependents is variable, we introduced a variant of the tree-RNN which accounts for the head-dependent relations and sequential information of words simultaneously. Thus, the proposed model is able to simplify the network architecture of the model and reduce the number of model parameters. In addition, to capture the knowledge from the richness of structure as well as to further reduce the model parameters, inspired by the dynamic compositional network [32], a meta-network was designed for the dependency encoder. By conditioning on a latent vector, the parameters of the proposed model can be dynamically generated for the composition of each node representation.

We conducted extensive analyses to show that the phrase representations of high-level nodes in constituent tree contribute less to the translation quality. In evaluations, we chose to re-implement the work of [12] and [19] as the representatives of constituent tree-structured model and dependency linearization model, respectively, as they are the most recent syntax-aware models. Empirical results in the English–Chinese and English–German translation tasks demonstrate that the proposed model outperforms the two prior models with even fewer parameters.

2. Related work

For neural machine translation, modeling syntax is served as an important role [8,33,34]. The efforts could be categorized into two directions, either incorporating the linearized syntactic structure, as additional syntactic annotations, into the translation models [13,15,16,35] or altering the architecture of encoder using the syntactic tree to guide recurrence and attention model to better capture the dependencies and attachments of words [11, 12,17].

Following the principle of sequence-to-sequence model, the first line of research attempts to incorporate the syntactic structure by transforming it into a sequence. Li et al. [13] proposed using an extra RNN to model the sequence of structural label, in addition to the word sequence, to enhance the model's capability in capturing the structural context of the source sentence. Hashimoto et al. [22] and Eriguchi et al. [36] proposed a multi-task framework to jointly parse and translate a source sentence, with an objective to incorporate the linguistic knowledge during training. Wu et al. [19] extended the original bidirectional encoder with two additional RNNs to model the head-enriched structure and child-enriched structure derived from the dependency structure of a source sentence. Instead of using the 1-best parse structure, Ma et al. [20] proposed to linearize the packed forest of parse trees to alleviate the translation quality caused by parsing errors. Apart from using the tree linearization, recent studies had succeeded on severing syntactic labels as external features. For example, Sennrich and Haddow [8] and Chen et al. [16] proposed encoding the dependency relations into the conventional NMT models, one by enriching the word representations while the other enhancing the attention probability. Although the above-mentioned approach transforms the syntactic structure into a sequential representation or treats the dependency relations as labels, Lin et al. [37] pointed out that establishing the dependencies between words in a direct fashion is best for modeling the syntax. Contrary to these studies, we focused on modeling the head-dependent relations of words following the dependency graph of the sentence to enhance the source-side representation.

The conventional NMT models rely on sequential encoder and decoder [4,6] without any explicit modeling of the syntactic structure of sentences. With this motivation, the second line of research attempts to improve the translation model by modeling the hierarchical structure of language in an explicit way. Eriguchi et al. [17] first proposed a tree-based attentive NMT model, which was further extended by Yang et al. [12] and Chen et al. [27] via a bidirectional encoding mechanism. All the above tree-based models applied constituent tree structure and faced the same problems (discussed in Section 1). Other studies try to improve the NMT by modeling the syntax on the target side [15,36,38–40]. Despite enhancing the decoder, their success also verified the necessity and effectiveness of modeling syntactic information for NMT systems.

Another thread of research to improve the encoding of tree structure is to employ the dynamic compositional network proposed by [32], in which a meta-network is used to generate the parameters and dynamically compose the constituents over the tree structure. However, their models were trained on a small data set and evaluated on the tasks of text classification and text semantic matching, which is in contrast to the present research that we adopted: it was for the NMT that were trained on massive amounts of parallel data with a large number of parameters.

Please cite this article as: B. Yang, D.F. Wong, L.S. Chao et al., Improving tree-based neural machine translation with dynamic lexicalized dependency encoding, Knowledge-Based Systems (2019) 105042, https://doi.org/10.1016/j.knosys.2019.105042.

3. Background: Tree-to-sequence neural machine translation

The idea of tree-to-sequence neural machine translation system is to build a neural network that considers the syntactic structure of a source sentence. Given a source sentence $\mathbf{X} = (x_1, \ldots, x_N)$, and its syntactic tree $\mathbf{T}$, the model is trained to maximize the conditional translation probability $p(\mathbf{Y} \mid \mathbf{X}, \mathbf{T})$ of the target translation $\mathbf{Y} = (y_1, \ldots, y_M)$ over a parallel training corpus [17,27,41].

3.1. Constituency-based encoder

In this paper, we first follow a bidirectional tree-based encoder, which is one of the recent syntax-aware model [12]. The source sentence $\mathbf{X}$ is first encoded into a sequence of hidden states (leaf node representations) $(h_1^l, \ldots, h_N^l)$ using bidirectional RNNs [6], as illustrated in Fig. 1. The hidden state of the ith leaf node h_i^l is the concatenation of the forward and backward vectors, namely $h_i^l = [\overrightarrow{h}_i^l, \overleftarrow{h}_i^l]$, where the forward vector $\overrightarrow{h}_i^l$ and backward vector $\overleftarrow{h}_i^l$ are respectively calculated by two gated recurrent units (GRUs) [42]:

$$\overrightarrow{h}_i^l = f_{\overrightarrow{GRU}}(x_i, \overrightarrow{h}_{i-1}^l)$$
$$\overleftarrow{h}_i^l = f_{\overleftarrow{GRU}}(x_i, \overleftarrow{h}_{i+1}^l),$$

where x_i is the ith source word embedding, $\overrightarrow{h}_{i+1}^l$ and $\overleftarrow{h}_{i-1}^l$ respectively denotes the previous hidden states in left-to-right and right-to-left GRUs. Given the constituent tree $\mathbf{T}$ of the source sentence, as shown in Fig. 1, a tree-GRU [43] is employed to recursively generate the phrase representations in a bottom-up fashion. The parent hidden state $h_{i,j}^{\uparrow}$ is composed of the hidden states of its left child node $h_{i,k}^{\uparrow}$ and right child node $h_{k+1,j}^{\uparrow}$ $(i \le k$ and $k+1 \le j)$,[1] where i, k, j indicate the indexes of the tokens:

$$h_{i,j}^{\uparrow} = f_{tree\_GRU}^{\uparrow}(h_{i,k}^{\uparrow}, h_{k+1,j}^{\uparrow}). \quad (1)$$

Thus, the phrase representations [44,45] are recursively built in an upward direction. To enhance the representations with global semantic information, a GRU network is applied to update the representations of internal tree nodes from the root to the leaves [27,46]. Given the hidden state of a parent node $h_{i,j}^{\downarrow}$, the hidden states of its left and right children $h_{i,k}^{\downarrow}$ and $h_{k+1,j}^{\downarrow}$ are updated as:

$$h_{i,k}^{\downarrow} = f_{GRU}^{lc}(h_{i,k}^{\uparrow}, h_{i,j}^{\downarrow})$$
$$h_{k+1,j}^{\downarrow} = f_{GRU}^{rc}(h_{k+1,j}^{\uparrow}, h_{i,j}^{\downarrow}),$$

where $h_{i,k}^{\uparrow}$ and $h_{k+1,j}^{\uparrow}$ are the left and right child representations generated via the bottom-up tree-GRU network. f_{GRU}^{lc} and f_{GRU}^{rc} represent different parameters to distinguish the left and right structural information.

3.2. Sequential decoder

The target sentence is sequentially predicted by an RNN. The conditional probability of the jth target word y_j is calculated using a non-linear function $f_{softmax}$:

$$p(\mathbf{Y} \mid \mathbf{X}, \mathbf{T}) = \prod_1^M p(y_j \mid y_{<j}, \mathbf{X}, \mathbf{T})$$
$$= \prod_1^M f_{softmax}(s_j, d_j, y_{j-1}),$$

where y_{j-1} is the previously generated target word, s_j denotes the jth target hidden state which is calculated by a standard sequential GRU network:

$$s_j = f_{GRU}^{dec}(y_{j-1}, s_{j-1}, d_{j-1}).$$

The context vector d_j is computed by the attention model [5,6] which is used to softly align each decoded hidden state with the source-side representations:

$$d_j = \sum_{i=1}^{2N-1} \alpha_j(i) h_i^{\downarrow}, \quad (2)$$

where $h_i^{\downarrow}$ indicates the ith representation of the source-side node [17,43].[2] The weight $\alpha_j(t)$ is computed by:

$$\alpha_j(t) = \frac{\exp(e_t)}{\sum_{i=1}^{2N-1} \exp(e_i)}$$
$$e_t = (V_a)^T \tanh(U_a s_j + W_a h_t^{\downarrow} + b_a),$$

where V_a, U_a, W_a, and b_a are the model parameters.

4. Lexicalized dependency encoding

4.1. Motivation

As illustrated in Fig. 1, the composition of a constituent representation vector is derived from its child node representations, the lexical information at leaf nodes underneath a constituent node tend to be repeatedly used and propagated to all of the internal nodes along the path to the root. That leads to the case whichever an constituent node is attended in decoding, the same lexical information is always considered in the prediction of the target words. Yang et al. [12] and Chen et al. [27] independently pointed out that conditioning on the phrase representations without proper control may result in the problem of over-translation, that is, part of the source sentence is translated more than once. An alternative solution to address this problem is to either head-lexicalize the internal nodes of a constituent structure [29] or to model the syntax of a sentence using dependency structure in which every node is naturally lexicalized. Moreover, the dependencies between words that are crucial to correctly interpret the semantic meaning of a source sentence may not directly be embodied in a consecutive phrase structure. The head word and its dependents can be far apart from each other in a sentence [19]. Fig. 2 shows an example of dependency structure of a sentence, where the head-dependent relations between word pairs are directly linked. In encoding perspective, the information flow over a dependency graph is more direct and, at the same time, it gives a more simple network architecture, in contrast to the constituent structure where additional nodes are introduced to represent the sentence constituents. We argue that modeling the syntax of a sentence using dependency structure is able to alleviate the above-mentioned problems. In this study, we proposed a novel syntax-based model to enhance the source representation by accounting for the syntactic relations of words according to a lexicalized dependency graph.

[1] The equal sign is used when $h_{i,k}^{\uparrow}$ or $h_{k+1,j}^{\uparrow}$ denotes the representation of leaf node.

[2] For simplistic, we simply the subscript of the source-side representation, since attention function summarizes source nodes regardless of their positions.

Please cite this article as: B. Yang, D.F. Wong, L.S. Chao et al., Improving tree-based neural machine translation with dynamic lexicalized dependency encoding, Knowledge-Based Systems (2019) 105042, https://doi.org/10.1016/j.knosys.2019.105042.

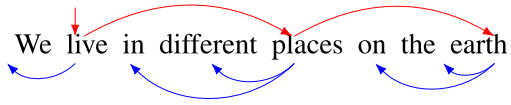


Fig. 2. The head-dependent relations between words are represented in a dependency tree.

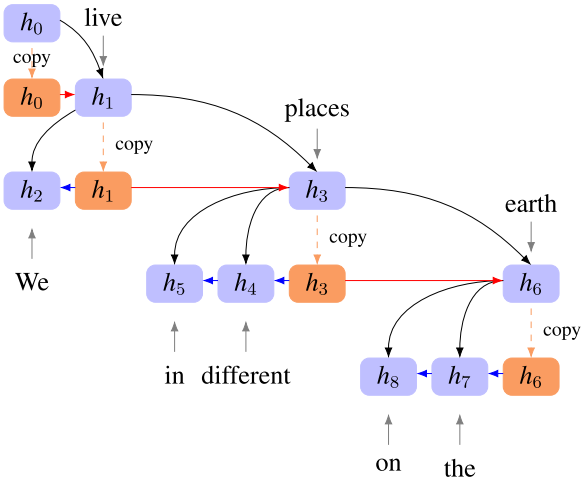


Fig. 3. Illustration of the lexicalized dependency encoder, where the gray lines indicate the input of the word embeddings, head-dependent relations are shown in black lines, leftward (red) and rightward (blue) straight lines indicate the direction of head-to-right-dependents and head-to-left-dependents sequential sibling encoding.[3] (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Modeling dependents

Recent studies have shown that the top-down encoding method can better propagate and capture the global syntactic and semantic information [12,27]. Under the dependency representation, the dependents are treated as the modifiers (or attributes) to the heads. To carry the syntactic information and reduce the distance between the corresponding words, we propose to recursively encode a source sentence from head word to its dependent words. As shown in Fig. 3, taking h_0 as the hidden state of root node (discussed in Section 4.4), the head word "live" is firstly encoded into h_1, followed by its left and right dependents, "We" and "places", respectively into h_2 and h_3. The black lines with arrow indicate the dependencies. The intuition behind our approach is that we first model the head words (most important information) followed by its dependents (the supplementary information). The encoding process is demonstrated in Table 1 .

4.3. Modeling sequential context

In a vanilla tree-based model, the representations of tree nodes are recursively built following a binarized constituent tree which increases the depth of the structure and insufficiently considers the sequential context. A dependency tree is an ordered graph which embraces the head-dependent relations as well as the relative position of a dependent to its head, either on the left- or right-hand side. In the proposed method, in order to model the sequential information, the dependent words sharing the same head are encoded sequentially in a center to a round fashion, i.e., head-to-left dependents and head-to-right dependents. Take the head node h_3 as an example, h_4 and h_5 are the left dependents while h_6 is the right dependent. The arrows indicate the direction in which the dependents are recurrently computed, as illustrated in Fig. 3.

Thus, the long-distance dependencies and the sequential information can be incorporated into the representation of each node. To distinguish the syntactic information of head-to-left-dependents and head-to-right-dependents, inspired by [46], two variant GRU networks f^{ld}_{GRU} and f^{rd}_{GRU} with different parameters are employed to model the left- and right-hand side dependents. Contrary to f^{lc}_{GRU} and f^{rc}_{GRU} (Section 7.1) which merely handle binary structure, the proposed method is able to encode multiple dependents in a layer. In detail, the representations of the left dependent h_{l_t} and the right dependent h_{r_t} are calculated as:

$$h_{l_t} = f^{ld}_{GRU}(h_p, h_{l_{t-1}}, x_{l_t})$$
$$h_{r_t} = f^{rd}_{GRU}(h_p, h_{r_{t-1}}, x_{r_t}),$$

where t indexes the time step, h_p denotes the hidden state of the head (parent) word, x_{l_t} and x_{r_t} indicate the word embeddings or the encoded representations calculated by bidirectional RNN layer. In the rest of this paper, word representations from RNN layers were used as default if no confusion is possible. $h_{l_{t-1}}$ and $h_{r_{t-1}}$ are the generated hidden states of the sibling nodes at the last time step. In detail, f^{ld}_{GRU} uses an update gate u_{l_t}, a reset gate r_{l_t} and a candidate activation $\widetilde{h}_{l_t}$ to calculate h_{l_t}, as follows:

$$\begin{aligned}
u_{l_t} &= \sigma(W^l_{(u)}h_p + U^l_{(u)}h_{l_{t-1}} + V^l_{(u)}x_{l_t} + b^l_{(u)}) \\
r_{l_t} &= \sigma(W^l_{(r)}h_p + U^l_{(r)}h_{l_{t-1}} + V^l_{(r)}x_{l_t} + b^l_{(r)}) \\
\widetilde{h}_{l_t} &= \tanh((W^l_{(h)}h_p + U^l_{(h)}h_{l_{t-1}}) \odot r_{l_t} + V^l_{(h)}x_{l_t} + b^l_{(h)}) \\
h_{l_t} &= (1 - u_{l_t})\widetilde{h}_{l_t} + u_{l_t}(h_p + h_{l_{t-1}}),
\end{aligned} \tag{3}$$

where $W^l_{(*)}$, $V^l_{(*)}$ and $U^l_{(*)}$ represent the weight matrices, $b^l_{(*)}$ denotes the bias vectors ($* \in \{u, r, h\}$), σ is the logistic sigmoid function, and the operator $\odot$ indicates element-wise multiplication between vectors. f^{rd}_{GRU} is defined in a similar way with different parameters. The update gate is to control the proportion of the information flow from the word embedding and the representations of the parent node, as well as the neighborhood nodes. The dependent relation between the parent node and the sequential information from the sibling nodes are retained by the reset gate, besides dropping the irrelevant information.

4.4. Initial states

To enrich the source representations with global information [47], we simply treat initial hidden state h_0 as the representation of root node and initialize it by feeding the mean of its

Table 1
An example illustrating the encoding process at each time step. "Current", "Head" and "Sibling" denote the currently encoded word, the parent word and the sibling word whose representations are input to the variant GRU at each time step. The blue (with underline) and red (without underline) colors indicate employing f^l_{GRU} and f^r_{GRU} respectively. "INIT" denotes the root node whose initial hidden state is h_0.

Step	Current	Head	Sibling	Coverage
t_1	h_1 ←live	INIT	INIT	○●○○○○○○
t_2	h_2 ←We	live	live	●●○○○○○○
t_3	h_3 ←places	live	live	●●○○●○○○
t_4	h_4 ←different	places	places	●●○●●○○○
t_5	h_5 ←in	places	different	●●●●●○○○
t_6	h_6 ←earth	places	places	●●●●●○○●
t_7	h_7 ←the	earth	earth	●●●●●○●●
t_8	h_8 ←on	earth	the	●●●●●●●●

[3] Both the purple and orange nodes refer to the same hidden state of a head node. The use of the additional orange node is for better illustration that, in computing the left and right siblings, the head node is used as the initial state to the respective left and right RNNs. The *copy* means to use the same embedding for sequential encoding.

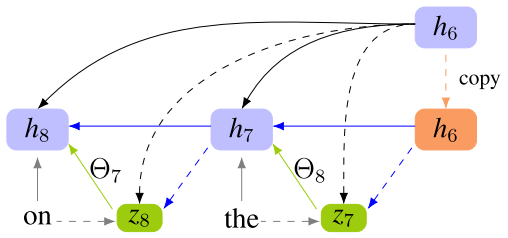


Fig. 4. Illustration of the framework of the meta-network (dashed lines) and its basic network (solid lines), where Θ denotes the context-dependent parameters conditioned on the latent vector z.

lower layer (i.e. representations generated by bidirectional RNN), namely:

$$h_0 = \tanh(W_{init}^{enc} \frac{\sum_{i=1}^{N} x_i}{N}),$$

where W_{init}^{enc} is the model parameters. Therefore, the global information can be propagated from root to leaves following the dependency structure.

Since the source-side hidden states are calculated in a lexicalized way, it is easy to combine the proposed model with a standard NMT decoder. Following [48], we defined the mean of source-side hidden states as the initial hidden state of the decoder:

$$s_{init} = \tanh(W_{init}^{dec} \frac{\sum_{i=1}^{N} h_i}{N}).$$

Then, the source hidden states are passed to the standard attention model to predict the conditional probabilities of target words.

5. Dynamic context-specific parameters

5.1. Motivation

Another drawback of the conventional methods is that it recursively uses the same shared compositional function throughout the whole compositional process and lacks expressive power due to the inability to capture the syntactic and semantic richness of linguistic phrases. Liu et al. [32] mentioned that the same parameters sharing across all semantic compositional rules may fail to capture the richness of semantic structure on text classification and semantic matching tasks. To handle the under-fitting problem caused by the diversities in semantic compositions, partially inspired by [32], we further adopt a low dimensional latent vector z, which was generated by a meta-network, to dynamically condition the context-specific parameters.

5.2. Latent vector

As shown in Fig. 4, a meta-network is built in the same framework as the proposed dependency model, but with a smaller hidden state size. According to Eq. (3), the meta-network of f_{GRU}^{ld} is formally expressed as:

$$\hat{u}_{l_t} = \sigma(\hat{W}_{(u)}^{l} h_p + \hat{U}_{(u)}^{l} h_{l_{t-1}} + \hat{V}_{(u)}^{l} x_{l_t} + \hat{b}_{(u)}^{l})$$
$$\hat{r}_{l_t} = \sigma(\hat{W}(r)^{l} h_p + \hat{U}_{(r)}^{l} h_{l_{t-1}} + \hat{V}_{(r)}^{l} x_{l_t} + \hat{b}_{(r)}^{l})$$
$$\hat{h}'_{l_t} = \tanh((\hat{W}_{(h)}^{l} h_p + \hat{U}_{(h)}^{l} h_{l_{t-1}}) \odot \hat{r}_{l_t} + \hat{V}_{(h)}^{l} x_{l_t} + \hat{b}_{(h)}^{l})$$
$$\hat{h}_{l_t} = (1 - \hat{u}_{l_t}) \hat{h}'_{l_t} + \hat{u}_{l_t} (h_p + h_{l_{t-1}})$$
$$z_{l_t} = W_z^{l} \hat{h}_{l_t},$$

Table 2
Statistics of the experimental data.

Task	Train	Dev	Test
En-Zh	1,249,372	1357	1664
En-De	4,398,299	3000	3003

where $\hat{W}_{(*)}^{l} \in \mathbb{R}^{d \times k}$, $\hat{U}_{(*)}^{l} \in \mathbb{R}^{d \times k}$, $\hat{V}_{(*)}^{l} \in \mathbb{R}^{d \times k}$ and $\hat{b}_{(*)}^{l} \in \mathbb{R}^{k}$ are the trainable parameters of the meta-network ($* \in \{u, r, h\}$), $W_z^{l} \in \mathbb{R}^{k \times k}$ is a scale matrix, d denotes the size of hidden state of f_{GRU}^{ld}, k is the dimensionality of the latent vector z_{l_t}, while $\hat{u}_{l_t}$, $\hat{r}_{l_t}$ and $\hat{h}'_{l_t}$ represent update gate, reset gate and candidate activation, respectively. Thus, the latent vector z_{l_t} is generated according to the context.

5.3. Dynamic parameters

Analogous to the Singular Value Decomposition, the context-specific low-rank factorized representation z_{l_t} is able to extract the semantic structure features to condition the parameters $W_{(*)}^{l}$, $V_{(*)}^{l}$, $U_{(*)}^{l}$ and $b_{(*)}^{l}$ used in Eq. (3), e.g., the static parameter matrix $W_{(*)}^{l}$ can be dynamically calculated by:

$$W_{(*)}^{l}(z_l) = P_{W_{(*)}}^{l} D(z_{l_t}) Q_{W_{(*)}}^{l},$$

where $D(z_{l_t}) \in \mathbb{R}^{k \times k}$ is the diagonal matrix of vector z_{l_t}, $P_{W_{(*)}}^{l} \in \mathbb{R}^{d \times k}$ and $Q_{W_{(*)}}^{l} \in \mathbb{R}^{k \times d}$ are the weight matrices. Similarly, the static parameters $W_{(*)}^{l}$, $V_{(*)}^{l}$ and $U_{(*)}^{l}$ are now substituted with the dynamic parametric functions $W_{(*)}^{l}(z_{l_t})$, $V_{(*)}^{l}(z_{l_t})$ and $U_{(*)}^{l}(z_{l_t})$, respectively. Given the learned parameters $B_{W_{(*)}}^{l} \in \mathbb{R}^{d \times k}$, the bias vector $b_{(*)}^{l}$ is substituted by $b_{(*)}^{l}(z_{l_t})$:

$$b_{(*)}^{l}(z_{l_t}) = B_{W_{(*)}}^{l} z_{l_t}.$$

The meta-network of f_{GRU}^{rd} is defined in a similar way using different parameters. In addition, with a small dimensionality of z, the model even needs fewer parameters.

6. Experiments

6.1. Setup

In this section, we evaluated the effectiveness of the proposed models. We conducted a group of experiments on English–Chinese translation task following the experiments reported in [12].[3] The models are trained on the parallel data from LDC corpora,[4] developed using the NIST 08 data set, and evaluated on the NIST 06 test set. The Chinese sentences are segmented using the Chinese word segmentation toolkit of *NiuTrans* [49]. We also trained the proposed models on a more prominent data set available at WMT14 for the English–German translation task, in which the models are developed using the newstest13 data set and examined on the newstest14 data set. We use the *Stanford Parser*[5] [50] based on Shift-Reduce algorithm [51] to

[3] The approaches of [27] and [12] are of the same principle. We chose to re-implement the latter method considering that it is the most recent model which has been further extended to distinguish the leftward and rightward representations using different parameterized GRUs, which are not considered in the former model.

[4] Our training data consists of: *LDC2000T46*, *LDC2000T50*, *LDC2003E14*, *LDC2004T08*, *LDC2004T08* and *LDC2005T10*.

[5] To have a fair comparison among the compared models, we use the Stanford PCFG Parser to parse all the English sentences to obtain their constituency trees, followed by converting it into its dependency counterpart using the accompanied conversion tool "EnglishGrammaticalStructure" of the Stanford NLP package. https://nlp.stanford.edu/software.

parse the English sentences and obtain the binary constituent trees and the dependency trees, respectively, for the conventional tree-based models and our proposed models. The parser have yielded promising results on various parsing tasks [50]. The parser achieves accuracy of 92.0% on English Penn Treebank (PTB). For the reason of computational efficiency, the sentences longer than 50 words are excluded from our training sets. We use a shortlist of the most frequent 30K (English–Chinese) and 50K (English–German) words as both the source and target vocabulary, covering approximately 99.8%/97.9% and 97.6%/94.5% on the source and target side of the two parallel corpora respectively. All the out-of-vocabulary words are mapped to a special token "*UNK*" [52]. The sentence counts for the filtered data sets are reported in Table 2.

For assuring the comparability, in this study, all the compared NMT models were implemented or re-implemented based on a widely-used attentional NMT source code *dl4mt*.[6] The hyper parameters of NMT are set as follows: The models use the 500-dimensional word embeddings and hidden units. For a fair comparison, we build an additional sequential model, with a dimensionality of 700, which requires a comparable number of parameters as that of the proposed models. In order to prevent over-fitting, the training data is shuffled following each epoch. In each iteration, we set the mini-batch size to 50 sentences. In decoding process, we adopt beam search algorithm to predict the output word at each step, and the beam search size was 5. The training of each model was early-stopped to minimize the perplexity on the development set. In addition, we use Adadelta [53] to optimize the model parameters, in which the learning rate is adapted according to the changes in gradients. The accuracy of the translation relative to the reference is assessed using the BLEU metric [54]. The contrasted systems are:

- *Sequential encoder*: An widely used sequence-to-sequence baseline [6] without any explicit modeling of the syntactic structure of sentences.
- *Constituency encoder*: The conventional tree-to-sequence baseline, which employs a bidirectional tree-based encoder [12].
- *Sequential dependency encoder*: A representative of sequence-to-sequence framework with source-side dependency information [19]. The model extended the original sequential encoder with two additional RNNs to model the head-enriched and child-enriched sequences derived from the source dependency structure.
- *Lexicalized dependency encoder*: Our method immediately exploits the source dependency structure into the encoder.

6.2. English-Chinese translation

6.2.1. Lexicalized dependency

The experimental results are summarized in Table 3. Both the dependency-based encoders (Models #6 and #8) outperform the traditional sequential encoders [6] (Models #1 and #2) and the conventional constituency model [12] (Model #3). This reveals the effectiveness of the dependency approaches on modeling the source-side representations from the structural context. Concerning the dependency-based models, the proposed lexicalized dependency model (Model #8) outperforms its linearized dependency counterpart [19] (Model #6). We attribute the improvement to the fact that the latter builds dependencies in the sequential order, which is still limited to capture the long-distance dependencies, while the proposed model is able to directly model the dependencies through the hierarchical structure.

Table 3
The translation results on English–Chinese translation tasks. "500d" and "700d" indicate the hidden size of the sequential encoder which is set to 500 and 700, respectively. The number of parameters (M = million) of each encoder is reported in the fourth column ("Para."). "Train" and "Decode" denote the training and decoding speed (sentences per second) which was examined on one GeForce GTX 1080 GPU. The last column is the translation accuracy on English–Chinese test sets evaluated by BLEU scores (%). Bold text indicates that the dynamic lexicalized dependency encoder is significantly better than the vanilla constituent-based encoder (Model #3) ($p < 0.01$).

#	Model	Para.	Train	Decode	BLEU
1	Sequential encoder-500d	3.50M	109.68	22.78	25.68
2	Sequential encoder-750d	7.88M	96.56	19.73	25.93
3	Constituency encoder [12]	9.50M	21.20	13.87	25.89
4	− bidirectional-RNNs	5.75M	33.41	15.60	19.16
5	+ dynamic parameters ($z = 100d$)	8.11M	19.77	13.07	26.40
6	Sequential dependency encoder [19]	7.00M	69.92	18.25	26.10
7	− bidirectional-RNNs	3.50M	83.05	19.01	22.82
8	Lexicalized dependency encoder	8.75M	58.79	17.18	26.32
9	− bidirectional-RNNs	5.00M	61.08	18.06	23.05
10	+ dynamic parameters ($z = 100d$)	6.97M	52.54	16.82	**26.73**

We further evaluated the models on the ability to capture the sequential contextual information. As observed, by removing the bidirectional sequential encoding layer, it has a tremendous impact on the translation quality of the constituency model (Model #4). There is a big drop of 6.73 BLEU scores in English–Chinese translation tasks, revealing that the success of constituent model relies on the context learned by the sequential RNNs. On the contrary, without using the explicit word sequence, the dependency-based approaches (Model #7 and #10) outperform that of the conventional tree method (Model #4), yielding significantly higher results. This confirms our hypothesis that, to some extent, the proposed dependency model is able to capture the sequential context by modeling the relative position of the sibling nodes.

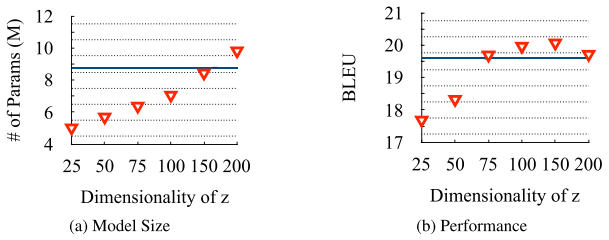
6.2.2. Dynamic parameters

The latent vector z controls the performance of the basic network and its dimensionality determines the number of model's parameters. We investigated the dynamic compositional mechanism across models with different dimensionality of [25, 50, 75, 100, 150 and 200] against the work of [32], which evaluated z only on small dimension sizes up to 25. The experimental results in Fig. 5 demonstrate that even when the size of vector z is reduced to 25, the model can still achieve considerable performances with 2 times fewer parameters in the encoder. By investigating the combination of translation quality and the model size, we set the dimension of z to 100.

As seen, by exploiting the dynamic parameters, both the constituency and the proposed dependency models (Model #5 and #10) achieve progressive improvements with far less parameters. We believe the dynamic parameters that are conditioned on a latent vector for compositional function helps in capturing various syntactic patterns and therefore can more accurately model the source-side information. It is worth noting that, the sequential dependency encoder (Model #6) does not benefit from the advantage brought by the meta-network due to its incompatibility issue of the network's architecture. In addition, the statistics of parameters required by each encoder[7] indicate that our proposed model requires fewer model parameters and yields faster processing speed than that of the conventional tree-based models.

[6] https://github.com/nyu-dl/dl4mt-tutorial.

[7] The statistics do not include the parameters for modeling the source-side word embeddings.



(a) Model Size (b) Performance

Fig. 5. The left and right diagrams show the model sizes and the BLEU scores with respect to different dimension sizes of the latent vector z. The x-axis represents the dimensionality of z. The y-axis of the left diagram (a) reports the number of parameters required by the encoder, while the y-axis of the right diagram (b) shows the BLEU scores that were evaluated on the English–Chinese validation set. The basic model (blue lines) is the proposed lexicalized dependency encoder. The red lines profile the influences of the basic model using different sizes of latent vector z on the translation quality and the model size.

Table 4
The translation results on English–German translation tasks. Bold text indicates that the dynamic lexicalized dependency encoder is significantly better than the constituency encoder ($p < 0.01$).

Model	BLEU
Sequential encoder-500d	18.22
Constituency encoder	18.02
+ dynamic parameters ($z = 100d$)	18.39
Lexicalized dependency encoder	18.41
+ dynamic parameters ($z = 100d$)	**18.74**

6.2.3. Main results

Both of the re-implemented baseline models, namely the sequential encoder (Model #1) and constituency-based encoder (Model #3), outperform the reported results [12] on the same data, and we believe the evaluations are convincing. By integrating all the above improvements into the NMT system, the proposed dynamic lexicalized dependency encoder (Model #10) significantly outperforms (+0.84 BLEU score) that of the constituency-based encoder (Model #3), while with a smaller parameter size. The proposed model demonstrates the ability to effectively model the source-side representations from both the sequential and structural contexts.

6.3. English-German translation

To the best of our knowledge, this is the first time to evaluate the tree-based NMT models on a large training data (4.5M) and for English–German translation. As seen in Table 4, the proposed model consistently outperforms constituency and sequential encoder on English–German translation task, showing the effectiveness and the universality of our method. However, against the sequential model, the proposed approach show gains (+0.52 BLEU score) in English–German translation tasks that are minuscule compared to English–Chinese experiments (+1.05 BLEU scores). The reason might be that German belongs to a morphologically-rich language. In the English to German translation, we used a limited vocabulary of frequent 50k words for both the English and German. It only covers 94.5% of vocabularies on the German training data, resulting that the problem of out-of-vocabulary is introduced during the model training.

Table 5
Translation qualities on the validation set. The second column denotes the layer which is considered by attention mechanism (e.g., Model #2 is constrained to attend to the hidden states of the nodes at the lowest two levels, i.e., the leaf nodes and their immediate parent nodes).

Model	Attended layer(s)	BLEU	△
Sequential		18.97	–
Lexicalized dependency		19.55	–
Constituency encoder			
1	1	19.33	–
2	≤2	19.45	+0.12
3	≤3	19.22	−0.11
4	≤5	19.24	−0.09
5	≤7	18.90	−0.43
6	≤9	19.06	−0.27
7	All	19.16	−0.17
8	>1	17.52	−1.81

7. Analysis

7.1. Lexicalized vs. phrase-based

In this section, we take a further step to investigate why the proposed lexical dependency model outperforms its constituent-based counterpart. According to Eq. (1), the phrase representations are merely derived from the hidden states of their child nodes considering the syntactic and semantic relations of words. With the increasing depth of the sub-tree, we argue that the hidden states of upper level phrase representations are difficult to learn the syntactic information well. To take an insightful assessment on the impact of the phrase representations at various levels in terms of the quality of the translation, we conducted several experiments on English–Chinese translation tasks. The models are allowed to expose on attending limit phrase representations on different levels of the constituent tree. We trained these models by controlling the variable i of Eq. (2).

7.1.1. The higher nodes, the worse translation quality

As the experimental results shown in Table 5, the models (Model #5, #6, #7), which consider the representations of the nodes close to the root (upper level) in the constituent tree, give slightly worse translation qualities than those models (Model #1, #2, #3, #4) attended on lower-level phrase vectors. Furthermore, the model that neglects the leaf representations (Model #8) is likely to underperform the others that are also conditioned on the leaf nodes. This reveals that considering the source-side high-level phrase representations in decoding process explicitly harms the translation quality.

7.1.2. Lexicalized formalism

Concerning the representation of lower nodes (e.g. the leaf nodes and their immediate parent nodes), there is only a marginal improvement between the proposed lexicalized model and Model #1, #2. We attribute the narrowing gaps to the fact that the lexical representations can be updated with the syntactic information, which is propagated from upper nodes through the top-down encoding. By lexicalizing the constituent tree, conventional tree-based model is able to achieve the comparable results to the dependency encoder, showing the effectiveness of lexicalized formalism on neural machine translation context. We believe the remained gaps between the dependency-based model and constituent-based model were caused by which the former model directly capture long-distance dependencies of words while the latter cannot.

Please cite this article as: B. Yang, D.F. Wong, L.S. Chao et al., Improving tree-based neural machine translation with dynamic lexicalized dependency encoding, Knowledge-Based Systems (2019) 105042, https://doi.org/10.1016/j.knosys.2019.105042.

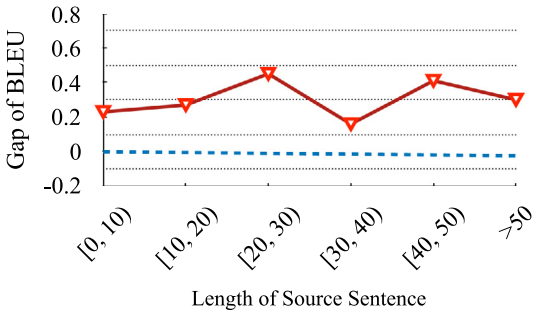


Fig. 6. Performance improvement according to various input sentence lengths. *Y*-axis denotes the gap of BLEU score between our model and constituency encoder (blue dash line).

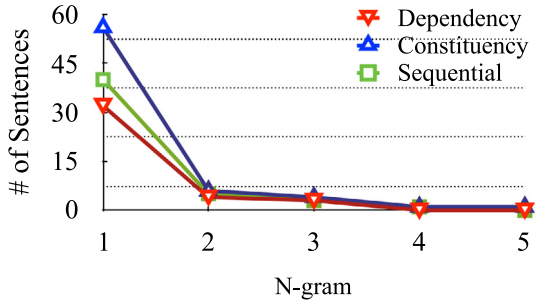


Fig. 7. The number of sentences with repeated translations respect to different phrase lengths. The evaluations were conducted on English-to-Chinese development set.

7.1.3. Analysis on long sentences

We further evaluate the effect of the proposed dependency model on long sentences. The sentences were divided into 10 disjoint groups according to their lengths and their BLEU scores were evaluated, as presented in Fig. 6. The proposed approaches outperform the constituency model in almost all the length segments. The results demonstrated that the proposed dependency encoder performs better on modeling long-distance dependencies, thus achieves better translation quality on long sentences.

7.1.4. Alleviating the problem of over-translation

As pointed out by Yang et al. [12] and Chen et al. [16], without properly controlling the proportion of information flow from the structure representations, their constituent models tend to over-translate (repeatedly translate) the sentence, particularly the translation of constituents. This phenomenon is further confirmed by our observations of the translation results. Furthermore, we take a further step to analyze the over-translation errors generated by different models on English-to-Chinese development set. The sentences which consist of two or more consecutive same phrases or segments, i.e. *n*-gram, were selected and manually checked. The statistics were divided into 5 disjoint groups according to the length of a repeated phrase, as shown in Fig. 7. Obviously, the constituent model generates more over-translations than that of a sequential model, especially on the shorter phrases ($N < 4$). The proposed dependency model is able to alleviate this problem and generate slightly less over-translations than its sequential counterpart.

7.2. Qualitative analysis

In this section, we try to answer the questions of: (1) Does the model improve the translation quality? And (2) whether the proposed model addresses the problem of over-translation in tree-based models. We dig into the translations and provide two translation examples produced by different models, as shown in Table 6.

7.2.1. Better performance on word sense ambiguation

In the first example of Table 6, the sequential model mistranslates "signed up" into "署名 (signature)" which is incorrect in the context, while both the tree models give the correct translation of "签约 (sign a contract)". We attribute this to the ability of tree models which in some degree is able to capture the relation of "sign" and "up" and disambiguate the word sense, revealing that modeling the syntactic structure in NMT is effective to the translation. Regarding the second translation mistake, both the sequential and constituent models cannot well interpret the relation between "play", "music" and "piano", leading to the wrong translation "play ... with a music player (播放)" instead of "perform piano (演奏)". However, in the proposed model, the dependencies of those words are connected in a more direct way regardless of their position in a sentence, resulted in a more faithful translation "演奏".

7.2.2. An example of over-translation

As illustrated by the second example in Table 6. The phrase "the same aspiration working in the same industry" was translated twice by the constituent model. We infer the problem is mainly caused by the recursive structure of sentence. Where it allows the attention model to condition on different level constituents which may carry the lexical information in common. In the proposed dependency model, the recursive structure is implicit in a dependency structure, and hence the problem of over-translation is naturally alleviated.

8. Conclusions

The major contributions in this study are that: (1) we proposed a novel lexicalized dependency encoder which encodes the source sentences following the dependency graph, and jointly learns the head-dependent relations and the relative position of the dependents, i.e., sequential context; (2) to capture the syntactic and semantic richness of linguistic phrases, a latent vector is adopted into the proposed model to dynamically generate the composition parameters; and (3) our extensive experiments and in-depth analysis show that the lexicalized formalism outperforms its phrasal counterpart. By integrating the above enhancements, the proposed dynamic lexicalized dependency model significantly outperforms the conventional tree-based models in English–Chinese and English–German translation tasks with even a smaller parameter size.

It is interesting to validate our model in other NLP tasks, such as reading comprehension, language inference, and sentence classification. Another promising direction is to design a strategy to integrate the proposed RNN-based model into the self-attention networks [55,56].

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61672555 and 61525205), the Joint Project of the Science and Technology Development Fund, Macau SAR and National Natural Science Foundation of China (Grant Nos. 045/2017/AFJ), the Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), and the Multi-year Research Grant from the University of Macau, Macau SAR (Grant No. MYRG2017-00087-FST). We thank the anonymous reviewers for their insightful comments.

Table 6
Translation examples generated by different models.

Model	Translation
Source	Xiao Huang , whose job after college was not satisfying , eventually **signed up** with a hotel to **play** background music on the piano .
Reference	大学生 小黄 毕业 后 , 找 的 工作 不 尽如人意 , 最后 在 一 家 宾馆 **挂 了 名** , 用 钢琴 **弹奏** 背景 音乐 。
Sequential Encoder	小黄 , 在 大学 毕业 后 的 工作 并不 满意 , 最终 与 一家 酒店 **署名** , 在 钢琴 上 **播放** 背景 音乐 。
Constituency Encoder	小黄 , 他的 大学 毕业 后 的 工作 不是 满意的 , 最后 和 一家 旅店 **签约** , 用 钢琴 **播放** 背景 音乐 。
Dependency Encoder	小黄 在 大学 毕业 后 的 工作 并不 令人满意 , 最终 还是 和 一家 酒店 **签约** , 用 钢琴 **演奏** 背景 音乐 。
Source	Writing some texts inspirational both to myself and to others on a freshly clean-cut interface and exchanging details in my work and life with friends **of the same aspiration working in the same industry**.
Reference	在 一 个 干干净净 的 界面 下 , 书写 一些 令 自己 感悟 和 别人 启发 的 文字 , 和 **同行业 的 同样 追求 的** 朋友 交流 工作 和 生活 中 的 点点滴滴 。
Sequential Encoder	在 一个 新鲜 干净 的 界面 上 为 自己 和 其他 人 写 一些 鼓舞 的 文章 , 交流 我 工作 的 细节 并且 与 **同行业 中 同样 愿望 的** 朋友 们 。
Constituency Encoder	为 自己 和 其他 人 在 一个 新鲜 干净 的 界面 上 写 一些 励志 的 文章 , 并 在 **同行业 中 与 同样 愿望 的 同行业 中 同样 愿望 的** 朋友 交换 我 工作 和 生活 中 的 细节 。
Dependency Encoder	为 自己 和 其他 人 在 一个 清新 简洁 的 界面 上 写 一些 励志 的 文章 , 并 在 **同行业 中 与 同样 愿望 的** 朋友 们 交流 我 工作 和 生活 中 的 细节 。

Appendix

Contrast with the existing sequential encoder [6,19] and constituent encoder [12,17] which fail to directly build connections between head and its dependents, a superiority of the proposed model lies in its shortcut path between dependents. The major drawback faced in RNN-based networks, e.g. RNN, LSTM and GRU, is the long-term problem [42,57]. The gradient decent of long-term recurrent unit might explode or vanish [58], thus, they always fail to learn features from long-distance tokens.

For a sequential encoder, given the derivatives of loss L at time step t, the back-propagated gradient of parameter W can be calculated as:

$$\frac{\partial L_t}{\partial W} = \sum_{\gamma \leqslant t} \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial h_\gamma} \frac{h_\gamma}{\partial W}$$

For a term with $\gamma \ll t$, $|\frac{\partial L_t}{\partial h_\gamma} \frac{h_\gamma}{\partial W}| \rightarrow 0$ due to the cumulative multiplication. Detailed proof can be learned in Bengio *et al.* [58]. The gradient of the term at γ step becomes very small when the two elements have a long distance, that is, the cost of tth element fails to effect the γth token.

On the contrary, the proposed lexicalized dependency encoder builds direct connection for head-dependent pairs, which have the most important relevance compare to other word pairs in a given sentence. The strategy limits the distance between head and dependent to 1. Accordingly, the effect of a small change in W would be felt mostly by head-dependent pairs, since they are on the near past (γ close to t). The ability of such kind of shortcut technique to alleviate the problem of gradient vanishing in deep networks can be also found in residual networks [59] and self-attention networks [37].

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[2] M.X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, M. Hughes, The best of both worlds: combining recent advances in neural machine translation, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 76–86.

[3] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, S. Liu, T. Liu, R. Luo, A. Menezes, T. Qin, F. Seide, X. Tan, F. Tian, L. Wu, S. Wu, Y. Xia, D. Zhang, Z. Zhang, M. Zhou, Achieving human parity on automatic chinese to english news translation, in: CoRR, 2018.

[4] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[5] T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.

[6] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of the 3rd International Conference on Learning Representations, 2015.

[7] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.

[8] R. Sennrich, B. Haddow, Linguistic Input Features Improve Neural Machine Translation, in: Proceedings of the First Conference on Machine Translation, 2016, pp. 83–90.

[9] F. Stahlberg, E. Hasler, A. Waite, B. Byrne, Syntactically guided neural machine translation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016 pp. 299–305.

[10] F. Nooralahzadeh, L. Øvrelid, Syntactic dependency representations in neural relation classification, in: Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP, 2018, pp. 47–53.

[11] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Simaan, Graph convolutional encoders for syntax-aware neural machine translation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1957–1967.

[12] B. Yang, D.F. Wong, T. Xiao, L.S. Chao, J. Zhu, Towards bidirectional hierarchical representations for attention-based neural machine translation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1432–1441.

[13] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, G. Zhou, Modeling source syntax for neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 688–697.

[14] A. Eriguchi, K. Hashimoto, Y. Tsuruoka, Incorporating source-side phrase structures into neural machine translation, Comput. Linguist. 45 (2) (2019) 267–292.

[15] S. Wu, D. Zhang, N. Yang, M. Li, M. Zhou, Sequence-to-dependency neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 698–707.

[16] K. Chen, R. Wang, M. Utiyama, L. Liu, A. Tamura, E. Sumita, T. Zhao, Neural machine translation with source dependency representation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2846–2852.

[17] A. Eriguchi, K. Hashimoto, Y. Tsuruoka, Tree-to-sequence attentional neural machine translation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 823–833.

Please cite this article as: B. Yang, D.F. Wong, L.S. Chao et al., Improving tree-based neural machine translation with dynamic lexicalized dependency encoding, Knowledge-Based Systems (2019) 105042, https://doi.org/10.1016/j.knosys.2019.105042.

[18] P. Frasconi, M. Gori, G. Soda, Recurrent neural networks and prior knowledge for sequence processing: a constrained nondeterministic approach, Knowl.-Based Syst. 8 (6) (1995) 313–332.
[19] S. Wu, M. Zhou, D. Zhang, Improved Neural Machine Translation with Source Syntax, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 4179–4185.
[20] C. Ma, A. Tamura, M. Utiyama, T. Zhao, E. Sumita, Forest-based neural machine translation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2018, pp. 1253–1263.
[21] C. Ma, A. Tamura, M. Utiyama, E. Sumita, T. Zhao, Improving neural machine translation with neural syntactic distance, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 2032–2037.
[22] K. Hashimoto, Y. Tsuruoka, Neural machine translation with source-side latent graph parsing, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 125–135.
[23] E. Strubell, P. Verga, D. Andor, D. Weiss, A. McCallum, Linguistically-informed self-attention for semantic role labeling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 5027–5038.
[24] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[25] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015, pp. 1556–1566.
[26] T. Blevins, O. Levy, L. Zettlemoyer, Deep RNNs encode soft hierarchical syntax, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 14–19.
[27] H. Chen, S. Huang, D. Chiang, J. Chen, Improved neural machine translation with a syntax-aware encoder and decoder, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vol. 1, 2017, pp. 1936–1945.
[28] Y. Shen, S. Tan, A. Sordoni, A. Courville, Ordered neurons: Integrating tree structures into recurrent neural networks, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019.
[29] Z. Teng, Y. Zhang, Head-lexicalized bidirectional tree LSTMs, Trans. Assoc. Comput. Linguist. 5 (1) (2017) 163–177.
[30] S. Wu, D. Zhang, Z. Zhang, N. Yang, M. Li, M. Zhou, Dependency-to-dependency neural machine translation, IEEE/ACM Trans. Audio Speech Lang. Process. 26 (11) (2018) 2132–2141.
[31] R. Socher, J. Bauer, C.D. Manning, A.Y. Ng, Parsing with compositional vector grammars, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 455–465.
[32] P. Liu, X. Qiu, X. Huang, Dynamic compositional neural networks over tree structure, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017 pp. 4054–4060.
[33] X. Shi, I. Padhi, K. Knight, Does string-based neural MT learn source syntax?, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1526–1534.
[34] Y. Wang, C. Wu, R.T. Tsai, Cross-language article linking with different knowledge bases using bilingual topic model and translation features, Knowl.-Based Syst. 111 (2016) 228–236.
[35] Q. Li, D.F. Wong, L.S. Chao, M. Zhu, T. Xiao, J. Zhu, M. Zhang, Linguistic knowledge-aware neural machine translation, IEEE/ACM Trans. Audio Speech Lang. Process. 26 (12) (2018) 2341–2354.
[36] A. Eriguchi, Y. Tsuruoka, K. Cho, Learning to parse and translate improves neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 72–78.
[37] Z. Lin, M. Feng, C.N.d. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, in: Proceedings of the Fifth International Conference on Learning Representations, 2017.
[38] R. Aharoni, Y. Goldberg, Towards string-to-tree neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 132–140.
[39] X. Wang, H. Pham, P. Yin, G. Neubig, A tree-based decoder for neural machine translation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4772–4777.
[40] W. Du, A.W. Black, Top-down structurally-constrained neural response generation with lexicalized probabilistic context-free grammar, in: Proceedings of the Seventh International Conference on Learning Representations, 2019, pp. 3762–3771.
[41] D.F. Wong, Y. Lu, L.S. Chao, Bilingual recursive neural network based data selection for statistical machine translation, Knowl.-Based Syst. 108 (2016) 15–24.
[42] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.
[43] Y. Zhou, C. Liu, Y. Pan, Modelling sentence pairs with tree-structured attentive encoder, in: Proceedings of 26th International Conference on Computational Linguistics, 2016, pp. 2912–2922.
[44] B. Yang, Z. Tu, D.F. Wong, F. Meng, L.S. Chao, T. Zhang, Modeling localness for self-attention networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4449–4458.
[45] B. Yang, L. Wang, D.F. Wong, L.S. Chao, Z. Tu, Convolutional self-attention networks, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4040–4045.
[46] N. Kalchbrenner, P. Blunsom, Recurrent continuous translation models, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1700–1709.
[47] B. Yang, J. Li, D.F. Wong, L.S. Chao, X. Wang, Z. Tu, Context-aware self-attention networks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 387–394.
[48] R. Sennrich, B. Haddow, A. Birch, Edinburgh neural machine translation systems for WMT 16, in: Proceedings of the First Conference on Machine Translation, 2016, pp. 371–376.
[49] T. Xiao, J. Zhu, H. Zhang, Q. Li, NiuTrans: An Open Source Toolkit for Phrase- based and Syntax-based Machine Translation, in: Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics, System Demonstra- tions, 2012, pp. 19–24.
[50] D. Chen, C. Manning, A fast and accurate dependency parser using neural networks, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 740–750.
[51] S.M. Shieber, Sentence disambiguation by a shift-reduce parsing technique, in: Proceedings of the 21st Annual Meeting on Association for Computational Linguistics, 1983, pp. 113–118.
[52] S. Jean, K. Cho, R. Memisevic, Y. Bengio, On using very large target vocabulary for neural machine translation, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015, pp. 1–10.
[53] M.D. Zeiler, ADADELTA: An Adaptive Learning Rate Method, in: CoRR, 2012.
[54] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002, pp. 311–318.
[55] J. Hao, X. Wang, B. Yang, L. Wang, J. Zhang, Z. Tu, Modeling recurrence for transformer, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 1198–1207.
[56] B. Yang, L. Wang, D.F. Wong, L.S. Chao, Z. Tu, Assessing the ability of self-attention networks to learn word order, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019, pp. 3635–3644.
[57] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[58] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2) (1994) 157–166.
[59] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778.

Please cite this article as: B. Yang, D.F. Wong, L.S. Chao et al., Improving tree-based neural machine translation with dynamic lexicalized dependency encoding, Knowledge-Based Systems (2019) 105042, https://doi.org/10.1016/j.knosys.2019.105042.