Towards Bidirectional Hierarchical Representations for Attention-Based Neural Machine Translation

Baosong Yang[†]   **Derek F. Wong**[†*]   **Tong Xiao**[‡]   **Lidia S. Chao**[†]   **Jingbo Zhu**[‡]

[†]NLP[2]CT Lab, Department of Computer and Information Science,
University of Macau, Macau, China
[‡]NiuTrans Lab, Northeastern University, Shenyang, China
`nlp2ct.baosong@gmail.com, {derekfw,lidiasc}@umac.mo,`
`{xiaotong,zhujingbo}@mail.neu.edu.cn`

Abstract

This paper proposes a hierarchical attentional neural translation model which focuses on enhancing source-side hierarchical representations by covering both local and global semantic information using a bidirectional tree-based encoder. To maximize the predictive likelihood of target words, a weighted variant of an attention mechanism is used to balance the attentive information between lexical and phrase vectors. Using a tree-based rare word encoding, the proposed model is extended to sub-word level to alleviate the out-of-vocabulary (OOV) problem. Empirical results reveal that the proposed model significantly outperforms sequence-to-sequence attention-based and tree-based neural translation models in English-Chinese translation tasks.

1 Introduction

Neural machine translation (NMT) automatically learns the abstract features of and semantic relationship between the source and target sentences, and has recently given state-of-the-art results for various translation tasks (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). The most widely used model is the encoder-decoder framework (Sutskever et al., 2014), in which the source sentence is encoded into a dense representation, followed by a decoding process which generates the target translation. By exploiting the attention mechanism (Bahdanau et al., 2015), the generation of target words is conditional on the source hidden states, rather than on the context vector alone. From a model architecture perspective, prior studies of the attentive
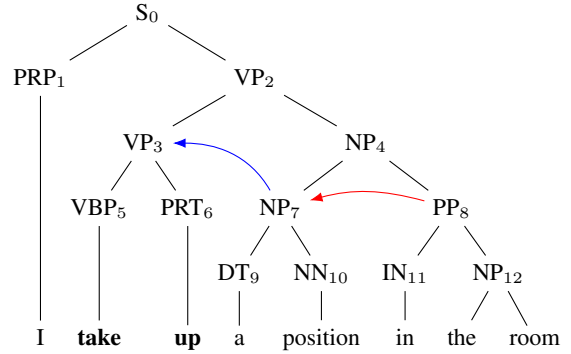
---
[*]Corresponding author

Figure 1: Induction of phrase and sentence representations over the syntactic structure of a sentence.

encoder-decoder translation model are mainly divided into two types.

The **sequence-to-sequence model** treats a sentence as a sequence of tokens. The most fundamental approaches transform the source sentence sequentially into a fixed-length context vector, and the annotation vector of each word summarizes the preceding words (Sutskever et al., 2014; Cho et al., 2014b). Although Bahdanau et al. (2015) used a bidirectional recurrent neural network (RNN) (Schuster and Paliwal, 1997) to consider preceding and following words jointly, these sequential representations are insufficient to fully capture the semantics of a sentence, due to the fact that they do not account for the syntactic interpretations of sentence structure (Eriguchi et al., 2016; Tai et al., 2015). By incorporating additional features into a sequential model, Sennrich and Haddow (2016) and Stahlberg et al. (2016) suggest that a greater amount of linguistic information can improve the translation performance.

The **tree-to-sequence model** encodes a source sentence according to a given syntactic tree

over the sentence. The existing tree-based encoders (Tai et al., 2015; Eriguchi et al., 2016; Zhou et al., 2016) recursively generate phrase (sentence) representations in a bottom-up fashion, whereby the annotation vector of each phrase is derived from its constituent sub-phrases. As a result, the learned representations are limited to local information, while failing to capture the global meaning of a sentence. As illustrated in Figure 1, the phrases "take up"[1] and "a position"[2] have different meanings in different contexts. However, in composing the representations h_{VP_3} and h_{NP_7} for phrases VP_3 and NP_7, the current approaches do not account for the differences in meaning which arise as a result of ignoring the neighboring context as well as the remote context, i.e. $h_{NP_7} \leftarrow h_{PP_8}$ (sibling) and $h_{VP_3} \leftarrow h_{NP_7}$ (child of sibling). More specifically, at the encoding step t, the generated phrase is based on the results at the previous time steps h_{t-1} and h_{t-2}, but has no information about the parent phrases $h_{t'}$ for $t' > t$.

To address the above problems, we propose a novel architecture, a bidirectional hierarchical encoder, which extends the existing attentive tree-structured models (Eriguchi et al., 2016). In contrast to the model of Eriguchi et al. (2016), we first use a bidirectional RNN (Schuster and Paliwal, 1997) at lexical level to concatenate the forward and backward states as the hidden states of source words, to capture the preceding and following contexts (described in Section 3.1). Secondly, we propose a bidirectional tree-based encoder (described in Section 3.2), in which the original bottom-up encoding model is extended using an additional top-down encoding process. In the bidirectional hierarchical model, the vector representations of the sentence, phrases as well as words, are therefore based on the global context rather than local information.

To effectively leverage hierarchical representations in generating the target words, we adopt a variant weighted tree-based attention mechanism (described in Section 3.4) in which a time-dependent gating scalar is used to control the proportion of conditional information between the word and phrase vectors. To alleviate the out-of-vocabulary (OOV) problem, we further extend the proposed tree-based model to the sub-word level

---

[1]**Take up** has the meanings of *start doing something new*, *use space/time*, *accept an offer*, etc.

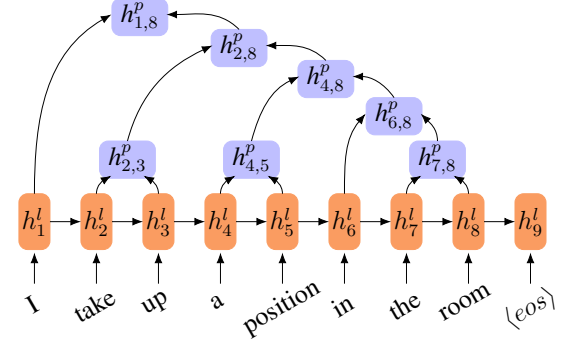[2]**Position** has the meanings of *location*, *job offer*, *rank/status*, etc.

Figure 2: The tree-based model of Eriguchi et al. (2016) comprising a structured and sequential encoder.

by integrating byte-pair encoding (BPE) (Sennrich et al., 2016) into the tree-based model (as described in Section 3.3). Experimental results for the NIST English-to-Chinese translation task reveal that the proposed model significantly outperforms the vanilla tree-based (Eriguchi et al., 2016) and sequential NMT models (Bahdanau et al., 2015) (Section 4.1).

2 Tree-Based Neural Machine Translation

A neural machine translation system (NMT) aims to use a single neural network to build a translation model, which is trained to maximize the conditional distribution of sentence pairs using a parallel training corpus (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014b,a). By incorporating syntactic information, the tree-based NMT exploits an additional syntactic structure of the source sentence to improve the translation. Since most existing NMTs generate one target word at a time, given a source sentence $\mathbf{x} = (x_1, ..., x_N)$ and its corresponding syntactic tree $\mathbf{tr}$, the conditional probability of a target sentence $\mathbf{y} = (y_1, ..., y_M)$ is formally expressed as:

$$p(\mathbf{y} \mid \mathbf{x}, \mathbf{tr}) = \prod_{1}^{M} p(y_j \mid y_1, ..., y_{j-1}, \mathbf{x}, \mathbf{tr}; \theta),$$

where θ represents the model parameters. A tree-based NMT consists of a tree-based encoder and a decoder.

2.1 Tree-Based Encoder

In a tree-based encoder, the source language $\mathbf{x}$ is encoded according to a given syntactic structure

tr of the sentence. As shown in Figure 2, Eriguchi et al. (2016) employed a forward Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) recurrent neural network (RNN) to encode the lexical nodes and a tree-LSTM (Tai et al., 2015) to generate the phrase representations in a bottom-up fashion. In the present study, we utilize the gated recurrent unit (GRU) (Cho et al., 2014b) instead of an LSTM, in view of its comparable performance (Chung et al., 2014) and since it yields even better results for certain tasks (Józefowicz et al., 2015). The lexical annotation vectors $(h_1^l, ..., h_N^l)$ are sequentially generated by using a GRU. The i-th leaf node vector is calculated as:

$$h_i^l = f_{GRU}^l(x_i, h_{i-1}^l), \qquad (1)$$

where x_i is the i-th source word embedding and h_{i-1}^l denotes the previous hidden state. The parent hidden state $h_{i,j}^\uparrow$ summarizes its left child $h_{i,k}^\uparrow$ and right child $h_{k+1,j}^\uparrow$ $(i < k < j)$ by applying the tree-GRU (Zhou et al., 2016) as follows:

$$
\begin{aligned}
z_{i,j}^\uparrow &= \sigma(U_{(z)}^L h_{i,k}^\uparrow + U_{(z)}^R h_{k+1,j}^\uparrow + b_{(z)}^\uparrow) \\
r_{i,k}^\uparrow &= \sigma(U_{(rL)}^L h_{i,k}^\uparrow + U_{(rL)}^R h_{k+1,j}^\uparrow + b_{(rL)}^\uparrow) \\
r_{k+1,j}^\uparrow &= \sigma(U_{(rR)}^L h_{i,k}^\uparrow + U_{(rR)}^R h_{k+1,j}^\uparrow + b_{(rR)}^\uparrow) \\
\widetilde{h}_{i,j}^\uparrow &= \tanh(U_{(h)}^L(r_{i,k}^\uparrow \odot h_{i,k}^\uparrow) \\
&\quad + U_{(h)}^R(r_{k+1,j}^\uparrow \odot h_{k+1,j}^\uparrow) + b_{(h)}^\uparrow) \\
h_{i,j}^\uparrow &= z_{i,j}^\uparrow \widetilde{h}_{i,j}^\uparrow + (1 - z_{i,j}^\uparrow)(h_{i,k}^\uparrow + h_{k+1,j}^\uparrow),
\end{aligned}
$$

where $z_{i,j}^\uparrow$ is the update gate; $r_{i,k}^\uparrow$, $r_{k+1,j}^\uparrow$ are the reset gates for the left and right children; $\widetilde{h}_{i,j}^\uparrow$ denotes the candidate activation; $U_{(\cdot)}^L$ and $U_{(\cdot)}^R$ represent weight matrices; $b_{(\cdot)}^\uparrow$ denote bias vectors; σ is the logistic sigmoid function; and the operator $\odot$ denotes element-wise multiplication between vectors. The phrase representations are recursively built in an upward direction.

2.2 Decoding with a Tree-Based Attention Mechanism

In generating the target words, we employ a sequential decoder with an input-feeding method (Luong et al., 2015) and attention mechanism (Bahdanau et al., 2015). The conditional probability of the j-th target word y_j is calculated using

a non-linear function $f_{softmax}$:

$$p(y_j \mid y_1, ..., y_{j-1}, \mathbf{x}, \mathbf{tr}; \theta) = f_{softmax}(c_j),$$

where c_j is the composite hidden state, which consists of a target hidden state s_j and a context vector d_j:

$$c_j = f_{\tanh}([s_j, d_j]).$$

Given the previous target word y_{j-1}, the concatenation of the previous hidden state s_{j-1} and the previous context vector c_{j-1} (input-feeding) (Luong et al., 2015), s_j, is calculated using a standard sequential GRU network:

$$s_j = f_{gru}^{dec}(y_{j-1}, [s_{j-1}, c_{j-1}]).$$

The context vector d_j is computed using an attention model which is used to softly summarize the attended part of the source-side representations. Eriguchi et al. (2016) adopted a tree-based attention mechanism to consider both the word and phrase vectors:

$$d_j = \sum_{i=1}^N \alpha_j(i) h_i^l + \sum_{k=1}^{N-1} \alpha_j(k) h_k^p, \qquad (2)$$

where h_i^l is the i-th hidden state of the source word at leaf level, and h_k^p is the k-th hidden state of the source phrase. The weight $\alpha_j(t)$ of node t is computed by:

$$
\begin{aligned}
\alpha_j(t) &= \frac{\exp(e_t)}{\sum_{i=1}^N \exp(e_i^l) + \sum_{k=1}^{N-1} \exp(e_k^p)} \\
e_t &= (V_a)^T \tanh(U_a s_j + W_a h_t + b_a),
\end{aligned}
$$

where h_t is the hidden state of the node. V_a, U_a, W_a and b_a are the model parameters.

3 The Bidirectional Hierarchical Model

Although the tree-based encoder of Eriguchi et al. (2016) has shown certain advantages in translation tasks involving distant language pairs, e.g. English-Japanese, the representation of a phrase relies solely on its child nodes, and the word representation at leaf level only takes into account the sequential information. We argue that the incorporation of more hierarchical information into the representations may contribute to an improvement in the translation. In particular, the use of global information can help in distinguishing the differences between word meanings. Based on this hypothesis, we propose an alternative architecture, the bidirectional hierarchical model, to enhance the source-side representations.
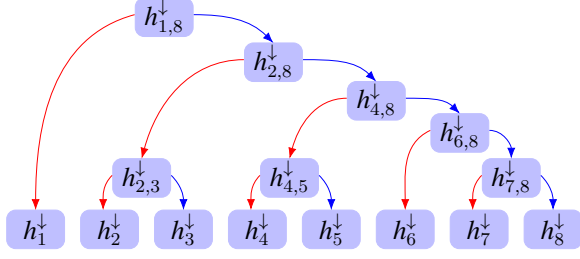
Figure 3: A top-down encoding process updates the hidden states recursively from root to leaf nodes. The red and blue lines denote the use of different learning parameters.

3.1 Bidirectional Leaf-Node Encoding

As discussed in Section 1, the unidirectional recurrent neural network reads an input sequence in order, from the first symbol to the last. In order to generate leaf node annotation vectors which jointly take into account both preceding and following annotations, we exploit a bidirectional RNN encoder (Bahdanau et al., 2015). The hidden state of the i-th leaf node h_i^l is the concatenation of the forward and backward vectors:

$$h_i^l = [\overrightarrow{h}_i^l, \overleftarrow{h}_i^l],$$

where $\overrightarrow{h}_i^l$ is obtained by a rightward GRU, as shown in Equation 1, and a leftward GRU calculates $\overleftarrow{h}_i^l$, as follows:

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

where $\overleftarrow{h}_{i-1}^l$ is the previous hidden state.

3.2 Bidirectional Tree-Node Encoding

Since the hidden states of leaf nodes are derived in a sequential, context-sensitive way, by generating phrase annotations in a bottom-up fashion, the sequential context can be propagated to tree nodes. However, the learned annotation vectors still fail to capture global information from the upper nodes. To enhance the representations with global semantic information, we propose to use a standard GRU recurrent network to update representations in a **top-down fashion**, as shown in Figure 3. The annotation vectors, which are learned by the previous encoding steps, are fed to the updating process.

First, we treat the bottom-up hidden state of root $h_{root}^{\uparrow}$, which covers the global meaning as well as

the syntactic information of the source sentence, as the initial state of the top-down GRU network:

$$h_{root}^{\downarrow} = h_{root}^{\uparrow}.$$

Given an updated hidden state of the parent node $h_{i,j}^{\downarrow}$, the hidden states of left and right children $h_{i,k}^{\downarrow}$ and $h_{k+1,j}^{\downarrow}$ are calculated as:

$$
\begin{aligned}
h_{i,k}^{\downarrow} &= f_{GRU}^{ld}(h_{i,k}^{\uparrow}, h_{i,j}^{\downarrow}) \\
h_{k+1,j}^{\downarrow} &= f_{GRU}^{rd}(h_{k+1,j}^{\uparrow}, h_{i,j}^{\downarrow}),
\end{aligned}
$$

where $h_{i,k}^{\uparrow}$ and $h_{k+1,j}^{\uparrow}$ are the left and right child annotation vectors generated via the bottom-up tree-GRU network. Contrary to the similar top-down encoding for sentiment classification (Kokkinos and Potamianos, 2017), which uses same weighting parameters to handle both left and right child nodes, f_{GRU}^{ld} and f_{GRU}^{rd} with different parameters are applied in the proposed model to distinguish the left and right structural information. According to the definition of a GRU (Cho et al., 2014b), f_{GRU}^{ld} uses an update gate $z_{i,k}^{\downarrow}$, a reset gate $r_{i,k}^{\downarrow}$ and a candidate activation $\widetilde{h}_{i,k}^{\downarrow}$ to generate $h_{i,k}^{\downarrow}$, as follows:

$$
\begin{aligned}
z_{i,k}^{\downarrow} &= \sigma(W_{(z)}^{ld}h_{i,k}^{\uparrow} + U_{(z)}^{ld}h_{i,j}^{\downarrow} + b_{(z)}^{ld}) \\
r_{i,k}^{\downarrow} &= \sigma(W_{(r)}^{ld}h_{i,k}^{\uparrow} + U_{(r)}^{ld}h_{i,j}^{\downarrow} + b_{(r)}^{ld}) \\
\widetilde{h}_{i,k}^{\downarrow} &= \tanh(W_{(h)}^{ld}h_{i,k}^{\uparrow} + U_{(h)}^{ld}(r_{i,k}^{\downarrow} \odot h_{i,j}^{\downarrow}) + b_{(h)}^{ld}) \\
h_{i,k}^{\downarrow} &= (1 - z_{i,k}^{\downarrow})h_{i,j}^{\downarrow} + z_{i,k}^{\downarrow}\widetilde{h}_{i,k}^{\downarrow}, \qquad (3)
\end{aligned}
$$

where $W_{(\cdot)}^{ld}$ and $U_{(\cdot)}^{ld}$ represent weight matrices, and $b_{(\cdot)}^{ld}$ denote bias vectors. f_{GRU}^{rd} is defined in a similar way.

From a linguistic point of view, in the top-down GRU network, the reset gate is able to retain the useful global information and drop irrelevant information from the parent state $h_{i,j}^{\downarrow}$, while the proportions of the global context from the top-down state $h_{i,j}^{\downarrow}$, and the local context from the bottom-up state $h_{i,k}^{\uparrow}$ are controlled by the update gate. As it covers both the partial meaning of the phrase and the whole meaning of the sentence, $h_{i,k}^{\downarrow}$ is regarded as the final representation of $node_{i,k}$:

$$h_{i,k}^{p} = h_{i,k}^{\downarrow}.$$

With the propagation of information from root to leaf nodes, the i-th leaf node representation is updated as:

$$h_i^l = h_i^{\downarrow}.$$

As each source-side hidden state of the leaf nodes and tree nodes carries the hierarchical information of the sentence, we interpret such an encoded state as a **hierarchical representation**.

3.3 Handling Out-of-Vocabulary: Tree-Based Rare Word Encoding

In NMT, the translation of rare words and unknown words is an open problem, since the computational cost increases with the size of the vocabulary. Sennrich et al. (2016) proposed a simple and effective approach to handling out-of-vocabulary by representing rare words as a sequence of sub-word units, which are segmented using byte-pair encoding (BPE) (Gage, 1994).
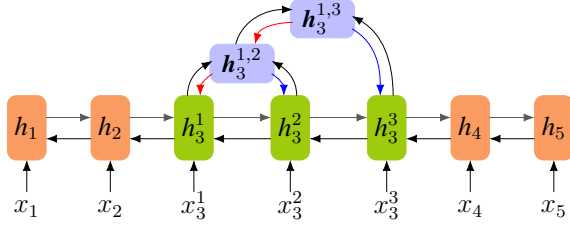

Figure 4: Encoding sub-word units with an additional binary lexical tree, where x_3^1, x_3^2, x_3^3 are the sub-units of word x_3.

We propose a variant tree-based rare word encoding approach which extends the tree-based model to the sub-word level. Sub-word units are encoded following an additional binary lexical tree. For a sentence $\mathbf{x} = (x_1, ..., x_i, ..., x_N)$, BPE segments the word x_i into a sequence of sub-word units $(x_i^1, ..., x_i^n)$. The binary lexical tree is simply built by composing two nodes in a rightwards fashion, $(((x_i^1, x_i^2), x_i^3)...), x_i^n)$, as shown in Figure 4. From the i-th leaf node, the original syntactic tree is extended downwards using the binary lexical tree, and the set of leaf nodes are replenished as $\mathbf{x} = (x_1, ..., x_i^1, x_i^2, ..., x_i^n, ..., x_N)$. Sub-word units can therefore be regarded as leaf nodes, and can be encoded using the proposed encoder, as illustrated in Figure 5. The experimental results in Section 4.1 demonstrate the effectiveness of this simple approach.

3.4 Decoder with Weighted Variant of Attention Mechanism

Since each representation carries both local and global information, in this case, attending fairly to the lexical and phrase representations in each
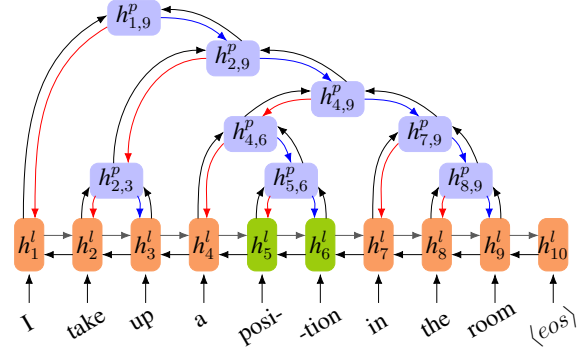

Figure 5: Illustration of the bidirectional hierarchical encoder: representations are enhanced by a bidirectional leaf-node encoding and a bidirectional tree-node encoding. The green nodes indicate the sub-word representations.

decoding step may cause the problem of over-translation (repeatedly attending and translating the same constituent of a sentence). An alternative approach is to balance the attentive information between the lexical and phrase vectors in the context vector. To effectively leverage these hierarchical representations, we propose a weighted variant of the tree-based attention mechanism (the original is defined in Equation 2). Formally, the calculation of the context vector d_j at step j is modified as:

$$d_j = (1 - \beta_j) \sum_{i=1}^{n} \alpha_j(i)h_i^l + \beta_j \sum_{k=1}^{n-1} \alpha_j(k)h_k^p \quad (4)$$

where $\beta_j \in [0, 1]$ is used to weight the expected importance of the representations. Inspired by work on a multi-modal NMT (Calixto et al., 2017) which exploits a gating scalar (Xu et al., 2015) to weight the image context vector, we use such a scalar in our model in order to dynamically adapt the weighting scalar. The gating scalar β_j at step j is calculated by :

$$\beta_j = \sigma(W_\beta c_{j-1} + b_\beta),$$

where W_β and b_β represent the model parameters. In contrast with α, which denotes the correspondence between each source annotation and the current target hidden state, β is dominated by the target composite hidden state alone. In other words, β is a time-dependent scalar in relation to the current target word, and therefore enables the attention model to explicitly quantify how far the

leaf and no-leaf states contribute to the word prediction at each time step. In the proposed model, the phrase and lexical context vectors are learned by a single attention model, meaning that they are dependent, and the gating scalar weights the phrase and lexical context vectors in complementary fashion, as shown in Equation 4. This distinguishes the model from that introduced by Calixto et al. (2017), in which the context vectors of the source sentence and image (bi-modal) are measured using two independent attention models and the gating scalar is merely used to weight the image context vector.

4 Experiments

4.1 Data

Training	Dev	Test		
LDC En-Zh	mt08	mt04	mt05	mt06
1,435,575	1,357	1,788	1,082	1,664

Table 1: Data used in the experiments.

We evaluate the proposed model on an English-to-Chinese translation task. For reasons of computational efficiency, we extracted 1.4M sentence pairs, in which the maximum length of the sentence was 40, from the LDC parallel corpus[3] as our training data. The models were developed using NIST mt08 data and were examined using NIST mt04, mt05, and mt06 data. The number of sentences in each dataset is shown in Table 1. On the English side, we used the *constituent parser* (Zeng et al., 2014, 2015) to produce a binary syntactic tree for each sentence, in constrast to the use of the *HPSG parser* by Eriguchi et al. (2016). On the Chinese side, the sentences are segmented using the Chinese word segmentation toolkit of *NiuTrans* (Xiao et al., 2012).

To avoid data sparsity, words referring to time, date and number, which are low in frequency, are generalized as '$time', '$date' and '$number'. In addition, as described in Section 3.3, the vocabularies are further compressed by segmenting the rare words into sub-word units using BPE.

4.2 Experimental Settings

As shown in Table 2, which gives the statistics of the token types, we limit the source and target vo-

Training Set	Original	Generalization	BPE
$|V|$ in En	159k	120k	40k
$|V|$ in Zh	198k	125k	40k

Table 2: The vocabulary size of the training set before and after applying the generalization and BPE segmentation.

cabulary size to 40,000, in order to cover all the English and Chinese tokens. The dimensions of word embedding and hidden layer are respectively set as 620 and 1,000. Due to the concatenation in the bidirectional leaf-node encoding, the dimensions of the forward and backward vectors, which are half of those of the other hidden states, are set to 500. In order to prevent over-fitting, the training data is shuffled following each epoch. Moreover, the model parameters are optimized using AdaDelta (Zeiler, 2012), due to its capability for dynamically adapting the learning rate. We set the mini-batch size to 16 and the beam search size to 5. The accuracy of the translation relative to a reference is assessed using the BLEU metric (Papineni et al., 2002). In order to give an equitable comparison, all the NMT models used for comparison are implemented or re-implemented using GRU in our code, based on *dl4mt*[4].

4.3 Enhanced Hierarchical Representations

Firstly, the effectiveness of the enhanced hierarchical representations is evaluated through a set of experiments, the results of which are summarized in Table 3.

Compared with the original **tree-based encoder** (Eriguchi et al., 2016), the model with bidirectional leaf-node encoding (described in Section 3.1) shows better performance. This also reveals that the future context at leaf level can contribute to word prediction. Secondly, although the representations of leaf nodes are learned in a sequential, context-sensitive way, the translation quality is further improved by considering the global semantic information in the top-down encoding (Section 3.2).

By incorporating the above enhancements into the model, the proposed **hierarchical encoder** yields significant improvements over both the sequential and the tree-based models. The problem of OOV is alleviated by further extending the tree-

Model	BPE	# of params	MT04	MT05	MT06	Dev.
sequential encoder	no	86.8M	31.26	23.98	24.02	17.20
+ sequential rare word encoding	yes	86.8M	32.54	25.09	25.07	18.19
+ tree-based rare word encoding	yes	104.1M	32.56	25.30	24.96	18.33
tree-based encoder	no	95.0M	31.90	24.68	24.40	17.63
+ bidirectional leaf-node encoding	no	92.0M	32.13	24.94	25.02	18.12
+ top-down encoding	no	101.1M	32.85	25.37	25.30	18.26
+ tree-based rare word encoding	yes	95.0M	33.02	25.62	25.24	18.59
hierarchical encoder ($\beta = 0.5$)	no	104.1M	32.91↑	25.55↑	25.52↑	18.46↑
hierarchical encoder ($\beta = 0.5$)	yes	104.1M	33.81⇑	26.47⇑	26.31⇑	19.41⇑
+ gating scalar	yes	105.1M	**34.33**⇑	**26.72**⇑	**26.58**⇑	**20.10**⇑

Table 3: Translation results for the various models. The first column shows the models; the second column indicates whether the corresponding experiment uses BPE data. The number of parameters (M = millions) in each model is given in the third column. The remaining columns are the translation accuracies for the test sets and development set, evaluated using BLEU scores (%). "↑ / ⇑": indicates that the hierarchical encoder is significantly better than the vanilla tree-based encoder ($p < 0.05/p < 0.01$).

based model to sub-word level (Section 3.3). In addition, we evaluate our tree-based rare word encoding method against the conventional rare word encoding (Sennrich et al., 2016) using the **sequential encoder** (Bahdanau et al., 2015). The empirical results confirm that our proposed tree-based BPE method achieves performance comparable to that of the standard BPE in the sequential model, but is applicable to the tree-based NMT model.

Overall, the proposed **hierarchical encoder** has demonstrated the ability to effectively model source-side representations from both the sequential and structural context. The NMT systems based on the proposed model significantly outperform those of conventional models using the **sequential encoder** and the **tree-based encoder**.

4.4 Weighted Attention Model

As discussed in Section 3.4, in order to effectively leverage hierarchical representations in generating the target word, we adopt a variant weighted tree-based attention mechanism which incorporates a scalar to control the proportion of conditional information between the word and phrase vectors. By manually or automatically varying the weight β, the utilization of the weighted attention model is assessed for four cases:

- $\beta = 0.0$: We manually set the weight of phrase vectors to 0.0; in other words, the decoder is forced to ignore the phrase vectors. The final translation is therefore generated by merely summarizing the leaf vectors.

- $\beta = 0.5$: The representations of non-leaf nodes and leaf nodes participate equally in the translation process. The decoder of this case therefore employs the same attention mechanism as that of the original model (Section 2.2).

- $\beta = 1.0$: In the reverse of the first case, the weight of the leaf nodes is manually set to 0.0. Thus, only the phrase vectors are used to predict the target words.

- Gating scalar (GS): A gating scalar is used for dynamically learning to control the proportion in which the lexical and phrase contexts contribute to the generation of the target words (Section 3.4).

Model	BLEU	Perplexity	Avg. Length
$\beta = 1.0$	17.16	98.65	21.13
$\beta = 0.5$	19.41	94.73	23.08
$\beta = 0.0$	19.83	94.68	23.33
GS	**20.10**	**94.18**	23.24

Table 4: Translation results for the development set. The last column indicates the average length of translation sentences, and the average length of reference sentences is 23.19.

The experimental results are shown in Table 4. The model which attends only to lexical annotation vectors ($\beta = 0.0$) gives slightly better performance than that which uses equal weights for
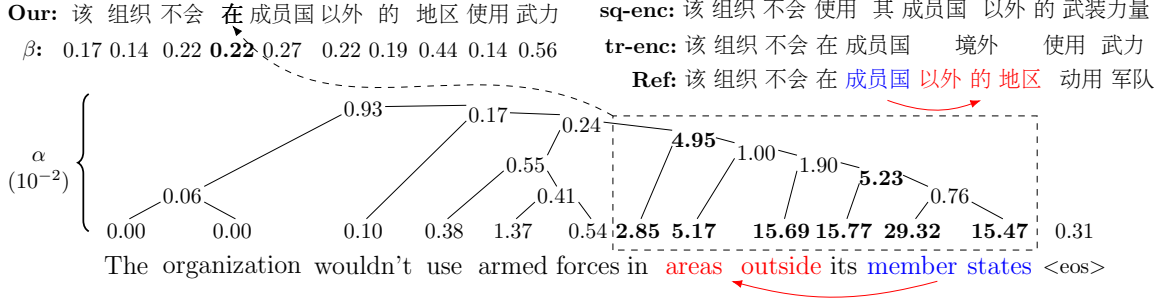
Our: 该 组织 不会 在 成员国 以外 的 地区 使用 武力
β: 0.17 0.14 0.22 **0.22** 0.27 0.22 0.19 0.44 0.14 0.56

sq-enc: 该 组织 不会 使用 其 成员国 以外 的 武装力量
tr-enc: 该 组织 不会 在 成员国 境外 使用 武力
Ref: 该 组织 不会 在 成员国 以外 的 地区 动用 军队

α (10⁻²)

0.93 — 0.17 — 0.24 — 4.95 — 1.00 — 1.90 — 5.23 — 0.76
0.06 0.55 0.41
0.00 0.00 0.10 0.38 1.37 0.54 2.85 5.17 15.69 15.77 29.32 15.47 0.31

The organization wouldn't use armed forces in areas outside its member states <eos>

Figure 6: Translations of an English sentence output using the NMT models with bidirectional hierarchical model (**our**), sequential encoder (**seq-enc**) and original tree-based encoder (**tr-enc**). **Ref** indicates the reference Chinese sentence. The attention scores (α), which are noted over the source-side syntactic tree, are output by the bidirectional hierarchical model at the step where the fourth target word "在" is translated. The sequence of scores β denote the value of the gating scalar at each translation step.

lexical and phrase vectors ($\beta = 0.5$). The use of global information contributes to distinguishing the differences between word meanings, although the similar semantic information in the lexical and phrase representations aggravates the over-translation problem observed in the translation results. However, we found that the model which attends only to phrase representations tends to generate shorter translation of an average of 21.13 words in length, as shown in the last column of the first row of Table 4. Furthermore, the model that neglects the leaf representations ($\beta = 1.0$) is likely to underperform the others that are also conditioned on the leaf nodes. Even though the phrase representations are derived from the lexical level via a bottom-up encoding, we believe it is unable to fully capture the lexical information of the source sentence. Through the use of the gating scalar, the hierarchical model achieves progressive improvements, as shown in Tables 3 and 4, the problem of over-translation is also alleviated. The representations of non-leaf nodes can be regarded as supplements in the translation process.

5 Qualitative Analysis

Figure 6 shows an English sentence and its binary tree representation, together with the corresponding Chinese translations produced by the different NMT models. All the models successfully give the correct Chinese translation "该 组织 不会" for the first three words of the English sentence "the organization wouldn't". Differences appear in the translation of the fourth word, and these lead to markedly different meanings. The translation

"使用 其 成员国 以外 的 武装力量" output by the sequential model, means "use the armed forces other than its member states" where "other than its member states" is incorrectly interpreted as a complement to "armed forces". This is caused by the intrinsic limitations of the sequential model, whereby it is unable to properly interpret the syntactic relationship of words. By explicitly incorporating the syntactic information, both the proposed hierarchical model and the tree-based model can accurately attend to the dashed section of Figure 6, and the translations can be correctly generated to reflect the meaning of the source sentence. The distinction between the translations produced by the original tree-based model and our hierarchical model is the interpretation of the words "areas outside". The tree-based model interprets it into "境外 (outside)", while our model correctly translates it into "以外 的 地区 (areas outside)". We believe that, with the help of global and local contextual information, our model is able to capture the short as well as long range dependencies.

We conducted an in-depth analysis of the BPE segmented units of rare words. It was observed that the sub-word units could be categorized into three groups. The first group of units involve the phonetic Romanization (Pinyin) of Chinese. In translation, these are simply transliterated into the corresponding Chinese characters. As shown in the second row of Table 5, "Liu/jing/min" is a person's name. The segmented units are the phonetic representations. Both models can successfully transliterate this into the Chinese equivalent, "刘/敬/民". The second group of sub-

Source	Reference	Hierarchical	Sequential
liu/jing/min	刘/敬/民	刘/敬/民	刘/敬/民
	Liú/jìng/mín	Liú/jìng/mín	Liú/jìng/mín
adventur/er	探险家	探险家	探险者
	Tàn xiǎn jiā	Tàn xiǎn jiā	Tàn xiǎn zhě
hi/k/ed	上调	上升	发生
	Shàng tiáo	Shàng shēng	Fā shēng

Table 5: Translation examples of sub-words, where '/' indicates a separation between sub-word units. The first two columns show the segmented words and their Chinese references. The last two columns report the translations given by the hierarchical and sequential models respectively.

word units are likely to represent the word morphemes. The words are segmented into sub-word units, which are to some extent close to the linguistic word stems and suffixes. For example, the word "adventurer" is segmented into "adventur/er", which is correctly translated into the Chinese translations "探险/家" and "探险/者" respectively by the hierarchical and sequential models, while the third group of sub-word units offer no linguistic interpretation. It is easy to see, using the BPE algorithm, that the identification of sub-word units is merely based on their frequency in the training data, with the result that not all units are well-formed linguistic morphemes. However, an interesting finding arises regarding the translation of these segmented units. In the sequential model, the word is incorrectly translated; however, it can be correctly translated by the hierarchical model. Taking "hi/k/ed" as an example, the sequential model gives an incorrect translation "发生(happened)", while the hierarchical model translates it into "上升(rise)" which is a synonym of "hiked". This result indicates that in our hierarchical model, the parent node of hierarchical representation for sub-word units "hi/k/ed" is better able to capture the meaning of the word as a whole; this cannot be captured independently by the sequential model.

6 Conclusion

In this paper, we propose an improved NMT system with a novel bidirectional hierarchical encoder, which enhances the source-side representations of a sentence, that is, both phrases and words, with local and global context information. By introducing a tree-based rare word encoding, the hi-

erarchical model is extended to sub-word level in order to alleviate the problem of OOVs. To effectively leverage the enhanced hierarchical representations, we also propose a weighted variant of the attention model which dynamically adjusts the proportion of conditional information between the lexical and phrase annotation vectors. Experimental results for NIST English-Chinese translation tasks demonstrate that the proposed model significantly outperforms the vanilla tree-based and sequential NMT models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-Attentive Decoder for Multi-modal Neural Machine Translation. *CoRR*, abs/1702.01287.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Advances in neural information processing systems (NIPS 2014) Deep Learning and Representation Learning Workshop*.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-Sequence Attentional Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 823–833.

Philip Gage. 1994. A New Algorithm for Data Compression. *The C Users Journal*, 12(2):23–38.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2342–2350.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Filippos Kokkinos and Alexandros Potamianos. 2017. Structural Attention Neural Networks for Improved Sentiment Analysis. *CoRR*, abs/1701.01811.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–90.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016. Syntactically Guided Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 299–305.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in neural information processing systems (NIPS 2014)*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 1556–1566.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. 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 Demonstrations*, pages 19–24.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 77–81.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Isabel Trancoso. 2015. Graph-Based Lexicon Regularization for PCFG With Latent Annotations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):441–450.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, Isabel Trancoso, Liangye He, and Qiuping Huang. 2014. Lexicon Expansion for Latent Variable Grammars. *Pattern Recognition Letters*, (42):47–55.

Yao Zhou, Cong Liu, and Yan Pan. 2016. Modelling Sentence Pairs with Tree-structured Attentive Encoder. In *Proceedings of 26th International Conference on Computational Linguistics*, pages 2912–2922.