# Syntax-Enhanced Neural Machine Translation with Syntax-Aware Word Representations

Meishan Zhang<sup>1</sup> and Zhenghua  $Li^2$  and Guohong  $Fu^{3*}$  and Min Zhang<sup>2</sup>

1. School of New Media and Communication, Tianjin University, China

2. School of Computer Science and Technology, Soochow University, China

3. Institute of Artificial Intelligence, Soochow University, China

# Abstract

Syntax has been demonstrated highly effective in neural machine translation (NMT). Previous NMT models integrate syntax by representing 1-best tree outputs from a welltrained parsing system, e.g., the representative Tree-RNN and Tree-Linearization methods, which may suffer from error propagation. In this work, we propose a novel method to integrate source-side syntax implicitly for NMT. The basic idea is to use the intermediate hidden representations of a well-trained end-to-end dependency parser, which are referred to as syntax-aware word representations (SAWRs). Then, we simply concatenate such SAWRs with ordinary word embeddings to enhance basic NMT models. The method can be straightforwardly integrated into the widelyused sequence-to-sequence (Seq2Seq) NMT models. We start with a representative RNNbased Seq2Seq baseline system, and test the effectiveness of our proposed method on two benchmark datasets of the Chinese-English and English-Vietnamese translation tasks, respectively. Experimental results show that the proposed approach is able to bring significant BLEU score improvements on the two datasets compared with the baseline, 1.74 points for Chinese-English translation and 0.80 point for English-Vietnamese translation, respectively. In addition, the approach also outperforms the explicit Tree-RNN and Tree-Linearization methods.

# 1 Introduction

In the past few years, neural machine translation (NMT) has drawn increasing interests due to its simplicity and promising performance (Bahdanau et al., 2014; Jean et al., 2015; Luong and Manning, 2015; Luong et al., 2015; Shen et al., 2016; Vaswani et al., 2017). The widely used



• An example of input dependency tree.



• SAWRs, where the encoder outputs are used as inputs for NMT similar to source-side word embeddings.

Figure 1: An example to illustrate our method of encoding source dependency syntax, where the English translation is "Education is the cornerstone of modern civilization" for the source Chinese input.

sequence-to-sequence (Seq2Seq) framework combined with attention mechanism achieves significant improvement over the traditional statistical machine translation (SMT) models on a variety of language pairs, such as Chinese-English (Shi et al., 2016; Mi et al., 2016; Vaswani et al., 2017; Cheng et al., 2018). Under an encoder-decoder architecture, the Seq2Seq framework first encodes the source sentence into a sequence of hidden vectors, and then incrementally predicts the target sentence (Cho et al., 2014a).

Recently, inspired by the success of syntaxbased SMT (Williams et al., 2016), researchers propose a range of interesting approaches for exploiting syntax information in NMT models, as syntactic trees could offer long-distance relations in sentences (Shi et al., 2016; Wu et al., 2017b; Li

<sup>\*</sup>Corresponding author.

et al., 2017; Bastings et al., 2017; Hashimoto and Tsuruoka, 2017).

As a straightforward method, tree-structured recurrent neural network (**Tree-RNN**) can elegantly model the source-side syntax and globally encode the whole trees. Eriguchi et al. (2016), Chen et al. (2017a) and Yang et al. (2017) show that Tree-RNN can effectively integrate syntaxoriented trees into Seq2Seq NMT models.

Regardless of the effectiveness of Tree-RNN, we find that it suffers from a severe low-efficiency problem because of the heterogeneity of different syntax trees, which leads to increasing difficulties for batch computation compared with sequential inputs. Even with deliberate batching method of Neubig et al. (2017), our preliminary experiments show that Tree-RNN with gated recurrent unit (GRU) can lead to nearly four times slower performance when it is integrated into a classical Seq2Seq system.

To solve the problem, **Tree-Linearization** is a good alternative for syntax encoding. The main idea is to linearize syntax trees into sequential symbols, and then exploit the resulting sequences as inputs for NMT. Li et al. (2017) propose a depth-first method to traverse a constituent tree, converting it into a sequence of symbols mixed with sentential words and syntax labels. Similarly, Wu et al. (2017b) combine several strategies of tree traversing for dependency syntax integration.

In this work, we present an implicit syntax encoding method for NMT, enhancing NMT models by syntax-aware word representations (SAWRs). Figure 1 illustrates the basic idea, where trees are modeled indirectly by sequential vectors extracted from an encoder-decoder dependency parser. On the one hand, the method avoids the structural heterogeneity and thus can be integrated efficiently, and on the other hand, it does not require discrete 1-best tree outputs, alleviating the error propagation problem induced from syntax parsers. Concretely, the vector outputs are extracted from the encoding outputs of the encoder-decoder dependency parser. As shown in Figure 1, the encoding outputs, denoted as  $o = o_1 \cdots o_6$ , are then integrated into Seq2Seq NMT models by directly concatenated with the source input word embeddings after a linear projection.

We start with a Seq2Seq baseline with attention mechanism (Bahdanau et al., 2014) for study, following previous studies of the same research line, and then integrate source dependency syntax by SAWRs. We conduct experiments on Chinese-English and English-Vietnamese translation tasks, respectively. The results show that our method is very effective in source syntax integration. With source dependency syntax, the performances of Chinese-English and English-Vietnamese translation can be significantly boosted by 1.74 BLEU points and 0.80 BLEU points, respec-We also compare the method with the tively. representative Tree-RNN and Tree-Linearization approaches of syntax integration, finding that our method is able to achieve larger improvements than the two approaches for both tasks. All the codes are released publicly available at https://github.com/zhangmeishan/SYN4NMT under Apache License 2.0.

# 2 Baseline

We take the simple yet effective Seq2Seq model with attention mechanism proposed by Luong et al. (2015) as our baseline. Under the standard encoder-decoder architecture, an encoder first maps the source-language input sentence into a sequence of hidden vectors, and a decoder then incrementally predicts the target output sentence. In particular, we should notice that several recent models (Vaswani et al., 2017; Zheng et al., 2017; Cheng et al., 2018) which have been shown to be more powerful can also serve as our baseline, since these models focus on very different aspects of NMT, which could be potentially complementary with our focus of syntax integration. We will demonstrate it by experimental analysis as well.

# 2.1 Encoder

In the encoder part, a single-layer bi-directional recurrent neural network (Bi-RNN) is employed to encode the sentence in order to capture features from the current word and the unbounded left and right contextual words. Given a source-language input sentence  $x = x_1 \cdots x_n$  and its embedding sequence  $e^{x_1} \cdots e^{x_n}$ , the Bi-RNN produces an encoding sequence of dense vectors  $h = h_1 \cdots h_i \cdots h_n$ :

$$\begin{aligned} \boldsymbol{h}_{i} &= \overrightarrow{\boldsymbol{h}}_{i} \oplus \overleftarrow{\boldsymbol{h}}_{i}, \\ \overrightarrow{\boldsymbol{h}}_{i} &= \operatorname{rnn}^{\mathrm{L}}(\boldsymbol{e}^{x_{i}}, \overrightarrow{\boldsymbol{h}}_{i-1}) \\ \overleftarrow{\boldsymbol{h}}_{i} &= \operatorname{rnn}^{\mathrm{R}}(\boldsymbol{e}^{x_{i}}, \overleftarrow{\boldsymbol{h}}_{i+1}) \end{aligned}$$
(1)

where  $rnn^{L/R}$  can be either GRU (Cho et al., 2014b) or LSTM. We use GRU all through this

paper for efficiency following Chen et al. (2017a).

# 2.2 Decoder

The decoder part incrementally predicts the target word sequence  $y = y_1 \cdots y_m$ , whose translation probability is defined as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{j=1}^{m} p(y_j|y_1\cdots y_{j-1}, \boldsymbol{h}).$$
(2)

The training objective is to maximize the probability of the reference translation. During evaluation, we aim to search for a target sentence with the highest probability for a given source sentence.

The probability of the j-th target word is computed by a two-layer feed-forward neural network:

$$p(y_j|y_1\cdots y_{j-1},\boldsymbol{h}) = g(\boldsymbol{s}_{j-1},\boldsymbol{c}_j), \quad (3)$$

where  $s_{j-1} = \operatorname{rnn}^{\operatorname{tgt}}(e^{y_{j-1}} \oplus c_{j-1}, s_{j-2})$  is the output of a left-to-right RNN over the predicted words, and the  $c_j/c_{j-1}$  is the weighted sum over the encoding sequence h of the source sentence via the attention mechanism, which is computed as follows:

$$c_{j} = \sum_{k=1}^{n} \alpha_{j,k} h_{k}$$

$$\alpha_{j,k} = \frac{\exp(\beta_{j,k})}{\sum_{l=1}^{n} \exp(\beta_{j,l})}$$

$$\beta_{j,l} = s_{j-1}^{T} W^{a} h_{l}$$
(4)

where  $W^a$  is the model parameter in attention.

# 3 Our Method

Syntax information has been demonstrated to be valuable for NMT. Previously, there were two representative approaches to encode syntax into an NMT model. The first approach directly represents an input syntax tree by **Tree-RNN**, and then uses the Tree-RNN outputs as additional encoder inputs for NMT. The second approach models source syntax trees indirectly by first converting a hierarchical tree into a sequence of symbols, and then use the symbols as inputs for NMT. The second method is referred to as **Tree-Linearization** here.

Tree-RNN is able to represent the syntax structures fully and comprehensively. However, because of the heterogeneity of different syntax trees, this approach suffers serious inefficiency



Figure 2: The framework of the SAWR approach, where the left part shows the encoder-decoder of a supervised dependency parsing model and the right part shows the NMT encoder-decoder.

problem as the increased difficulty of batch computation for GPU neural computation. The second approach exploits an alternative sequence to substitute the original trees, which solves the inefficiency problem. But it may bring loss of syntax information because the hierarchical tree structure is no longer maintained in the new representation, which could be potentially useful for NMT.

Both the two syntax integration approaches are based on discrete 1-best outputs of a supervised dependency parser, which may suffer from the error propagation problem. Incorrect syntax trees as inputs for NMT may produce erroneous outputs, leading to inappropriate translation results. In order to alleviate the problem, we present a novel method not using the discrete parsing outputs.

We focus on supervised dependency parsing models which can be formalized as an encoderdecoder architecture, and exploit the encoder outputs as the inputs for our Seq2Seq NMT model. The encoder outputs are sequences of dense vectors aligning with the source sentential words, as shown in Figure 1, and thus they could be easily combined with the encoder part of our NMT model. We refer to this method as **SAWR** for brief. Our approach takes the implicit hidden outputs from a supervised parser as inputs for NMT, which greatly reduces the direct influence brought from discrete 1-best parser outputs.

Figure 2 shows the framework of SAWR. Concretely, we first project the encoder outputs of a dependency parsing model into a sequence of vectors by a feed-forward linear layer, as shown by the projection module in Figure 2:

$$\boldsymbol{s}_i = \boldsymbol{W} \boldsymbol{o}_i + \boldsymbol{b} \tag{5}$$

where  $o = o_1 \cdots o_n$  is the encoder output of a parsing model, W and b are model parameters. Then we concatenate the resulting vectors with the source embeddings as inputs for the baseline Bi-RNN Encoder. Thus the encoder process can be formalized as follows:

$$\boldsymbol{h} = \operatorname{Bi-RNN}(\boldsymbol{e}^{x_1} \oplus \boldsymbol{s}_1, \cdots, \boldsymbol{e}^{x_n} \oplus \boldsymbol{s}_n).$$
 (6)

Noticeably, the SAWR method can be regarded as an adaption of joint learning as well. We can train both dependency parsing and machine translation model parameters concurrently. In this work, we focus on the machine translation task and do not involve the training objective of dependency parsing. However, we can still finetune model parameters of the encoder part of dependency parsing by back-propagating the training losses of NMT into this part as well.

Actually, SAWRs are also similar to the ELMO embeddings (Peters et al., 2018). ELMO learns context word representations by using language model as objective, while SAWRs learn syntaxaware word representations by using dependency parsing as objective. On the other hand, compared with the Tree-RNN and Tree-Linearization methods which encode syntax trees by neural networks directly, SAWRs are less sensitive to the output syntax trees. Thus the SAWR method can alleviate the error propagation problem.

# 4 Experiments

### 4.1 Settings

**Data.** We conduct experiments on the Chinese-English and English-Vietnamese translation tasks, respectively. For Chinese-English, we use the parallel training data from the publicly available LDC corpora,<sup>1</sup> with 28.3M Chinese words and 34.5M English words, respectively, consisting of 1.25M sentence pairs, and test model performances on the NIST datasets, using NIST MT02 as the development data, and MT03-06 as test datasets. For English-Vietnamese, we use the standard IWSLT 2015 dataset,<sup>2</sup> which consists of about 133K sentence pairs, and evaluate our models by exploiting

<sup>1</sup>LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

the TED tst2012 and tst2013 as the development and test datasets, respectively.

For the source side sentences, we construct vocabularies of the most frequent 50K words, while for the target side sentences, we apply byte-pair encodings (BPE) (Sennrich et al., 2016) with 32K merges to obtain subword units, and construct the target vocabularies by the most frequent 32K subwords. During training, we use only the sentence pairs whose source and target lengths both are no longer than 50 and 150 for Chinese-English and English-Vietnamese translations, respectively.

**Evaluation.** We use the case insensitive 4gram BLEU score as the main evaluation metrics (Papineni et al., 2002), and adopt the script multi-bleu.perl in the Mose toolkit.<sup>3</sup> Significance tests are conducted based on the best-BLEU results for each approach by using bootstrap resampling (Koehn, 2004).

Alternatively, in order to compare the effectiveness of our model with other syntax integration methods, we implement a **Tree-RNN** approach and a **Tree-Linearization** approach, respectively:

- Tree-RNN: We build a one-layer bidirectional Tree-RNN with GRU over input word embeddings, producing syntaxenhanced word representations, which are then fed into the encoder of NMT as basic inputs. The method is similar to the model proposed by Chen et al. (2017a).
- Tree-Linearization: We first convert dependency trees into constituent trees (Sun and Wan, 2013), and then feed it into the NMT model proposed by Li et al. (2017).

**Hyperparameters.** We set the dimension sizes of all hidden neural layers to 1024, except the input layers for RNNs (i.e. input word embeddings and the projection layer of SAWR), which are set to 512. We initialize all model parameters by random uniform distribution between [-0.1, 0.1]. We apply dropout on the output layer of word translation with a ratio of 0.5.

We adopt the Adam algorithm (Kingma and Ba, 2014) for parameter optimization, with the initial learning rate of  $5 \times 10^{-4}$ , the gradient clipping threshold of 5, and the mini-batch size of 80. During translation, we employ beam search for decoding with the beam size of 5.

<sup>&</sup>lt;sup>2</sup>https://nlp.stanford.edu/projects/nmt/

<sup>&</sup>lt;sup>3</sup>http://www.statmt.org/moses

System	MT03	MT04	MT05	MT06	Average/ $\Delta$		
Baseline	36.44	39.35	36.26	36.32	37.09		
SAWR	38.42	40.60	38.27	38.04	38.83/+1.74		
Tree-RNN	38.12	40.35	37.86	37.32	38.41/+1.32		
Tree-Linearization	37.95	40.24	37.64	37.44	38.32/+1.23		
Previous Work							
Chen et al. (2017a)	35.64	36.63	34.35	30.57	34.30/ <b>+2.59</b>		
Li et al. (2017)	34.9	38.6	35.5	35.6	<b>36.15</b> /+1.45		
Chen et al. (2017b)	35.91	38.73	34.18	33.76	35.65/+1.52		

Table 1: Final results of Chinese-English translation. All syntax-integrated approaches are significantly better than the baseline system (p < 0.05).

**Source-Side Parsing.** We employ the state-ofthe-art BiAffine dependency parser recently proposed by Dozat and Manning (2016) to obtain the source-side dependency syntax information. The BiAffine parser can also be understood as an encoder-decoder model, where the encoder part is a three-layer bi-directional LSTM over the input words, and the decoder uses BiAffine operations to score all candidate dependency arcs and finds the highest-scoring trees via dynamic programming.

For Chinese-English translation, we train the dependency parser on Chinese Treebank 7.0 with Stanford dependencies,<sup>4</sup> using 50K random sentences as the training data and the remaining as the test data. The parser achieves 81.02% parsing accuracy (labeled attached score, LAS) on the test dataset. For English-Vietnamese translation, we train the dependency parser on English WSJ corpus, following the same data split as Dozat and Manning (2016), and obtaining a LAS of 93.84% on the test dataset.<sup>5</sup>

### 4.2 Speed Comparison

All our experiments are run on a single GPU NVIDIA TITAN Xp. We report the averaged one-epoch training time on the Chinese-English translation dataset (consuming all 125M sentence pairs) as follows:

Baseline	105 min
SAWR	142 min
Tree-RNN	498 min
Tree-Linearization	137 min

<sup>4</sup>https://nlp.stanford.edu/software/stanforddependencies.shtml

<sup>5</sup>For simplicity, we use only words as inputs for both Chinese and English dependency parsing, avoiding the influences brought by other inputs, such as automatic POS tags.

The SAWR system spends averaged 142 minutes,<sup>6</sup> 37 minutes slower than the baseline model. The Tree-Linearization spends averaged 137 minutes per epoch, which is the fastest syntax integration method. Our SAWR approach spends 5 more minutes than Tree-Linearization, appropriate 3.5% of the total spend time per epoch, which could be negligible. The Tree-RNN model spends 498 minutes per epoch, nearly four times slower than the baseline model.<sup>7</sup> According to the results, we can conclude that the Tree-RNN model is highly inefficient for encoding dependency syntax, whereas the SAWR and Tree-Linearization are almost as efficient as the baseline Seq2Seq system.

#### 4.3 Main Results

# 4.3.1 Chinese-English Translation

Table 1 shows the main results of all approaches on Chinese-English datasets. Considering the effect of random initialization, we train three individual models for each approach, and use the averaged BLEU scores for fair comparisons.

According to the results, we can see that all syntax-integrated approaches can bring significant improvements over the baseline system, which denotes that syntax is highly effective for Chinese-English machine translation. In addition, the proposed SAWR approach obtains the largest BLEU improvements, averaged  $\Delta = 1.74$  BLEU points better than the baseline system. The Tree-RNN and Tree-Linearization approaches bring improve-

<sup>&</sup>lt;sup>6</sup>We exclude the time consumed by the encoder part of the dependency parsing model for fair comparisons, as other methods require to perform parsing in an offline way.

<sup>&</sup>lt;sup>7</sup>The Tree-RNN model is implemented with deliberate batching motivated by Neubig et al. (2017), without which the model is intolerably slow, reaching about 1,900 minutes per epoch.

System	tst 2013 / $\Delta$
Baseline	28.29
SAWR	29.09/+0.80
Tree-RNN	28.51/+0.22
Tree-Linearization	28.93/+0.64

Table 2: Final Results on the IWSLT 2015 English-Vietnamese translation task. Only SAWR is significantly better than the baseline system (p < 0.05).

ments of averaged  $\Delta = 1.32$  and  $\Delta = 1.23$  BLEU points, respectively. The results show that our implicit syntax-aware encoding method is better than Tree-RNN and Tree-Linearization.

We compare our NMT models with other stateof-the-art methods as well. The results are just for reference since experimental details could be very different. In particular, we list the relative improvements over the corresponding baseline models by integrating syntax structures, which are calculated according to their papers. All these studies exploit lower baselines compared with our models. The Tree-RNN and Tree-Linearization are essentially similar to Chen et al. (2017a) and Li et al. (2017), respectively. As shown, our approaches can still obtain large improvements based on a stronger baseline.

# 4.3.2 English-Vietnamese Translation

Table 2 shows the final results on the IWSLT 2015 English-Vietnamese translation task. The overall tendency is similar to that of Chinese-English translation. The syntax information can boost the translation performances by using any of the three approaches. The SAWR approach gives the best translation performance, significantly outperform the baseline system by  $\Delta = 0.80$  BLEU points. While although the other two approaches bring better performances, the improvements are not significant. The results demonstrate the advantage of the proposed implicit SAWR approach. By not using the 1-best parser outputs, our approach can reduce the error propagation problem, thus bring larger improvements with syntax.

In particular, we find that the increases of BLEU scores are smaller than that of Chinese-English translation by integrating syntactic features. The averaged BLEU increases are 0.55 for English-Vietnamese and 1.43 for Chinese-English. The possible reason may be due to that the source English sentences are more grammatically rigorous

					Average
no Tune	38.42	40.60	38.27	38.04	38.83
no Tune Tune	37.33	39.45	36.93	37.03	37.69

Table 3: The influence of fine-tuning parser parameters in the SAWR system.

than Chinese sentences. For example, the English functional words such as "of" and "'s" which indicate the possessive relationship, should be always kept in sentences by standard, while their Chinese correspondence "的" may be omitted in sentences.

# 4.4 Analysis

In this section, we conduct analysis on Chinese-English translation from different aspects to better understand the SAWR approach of integrating source-side dependency syntax for NMT.

### 4.4.1 Fine-Tuning Syntax-Oriented Inputs

The SAWR approach directly uses the encoder outputs of a dependency parser as extra inputs for NMT. In the above experiments, we keep the parser model parameters fixed, letting them uninfluenced from NMT optimization. Actually, this part can be further fine tuned along with the NMT learning, by treating them as one kind model parameters. Thus there arises a question that whether fine-tuning the parser model parameters can bring better performance.

As an interesting attempt, we can simultaneously fine tune the parameters of both the parser and the Seq2Seq NMT model during training. Figure 3 shows the results. We can see that fine-tuning decreases the average BLEU score by 38.83 - 37.69 = 1.14 significantly. This may be because that fine-tuning disorders the representation ability of the parser and makes its function more overlapping with other network components. This further demonstrates that pretrained syntax-aware word representations are helpful for NMT.

#### 4.4.2 Alignment Study

Alignment quality is an important metric to illustrate and evaluate machine translation outputs. Here we study how syntax features influence the alignment results for NMT. We approximate the alignment scores by the attention probabilities as shown in Equation 4.<sup>8</sup> For better understanding

<sup>&</sup>lt;sup>8</sup>We aim to offer an intuitive interpretation by a carefullyselected example. In fact, the alignment computation method here may be problematic (Koehn and Knowles, 2017).

System	MT03	MT04	MT05	MT06	Average/ $\Delta$
Baseline×3	40.90	43.25	40.64	40.16	41.24
SAWR×3	41.94	44.59	41.91	41.97	42.60/+1.36
Tree-RNN×3	42.03	44.15	41.50	41.41	42.27/+1.03
Tree-Linearization×3	41.74	44.23	41.32	41.44	42.18/+0.94
Hybrid	42.72	45.14	42.38	42.15	43.10/+1.86

Table 4: Ensemble performances, where the Hybrid model denotes SAWR + Tree-RNN + Tree-Linearization.



Figure 3: Alignments for the baseline and syntaxintegrated systems, where the same example in Figure 1 is analyzed and the target English word is "of".

the effectiveness of syntax, we choose the targetside English word "of" for comparison, which is a grammatical functional word.

Figure 3 shows the alignment probability distributions returned by different approaches. Intuitively, this word should be aligned with the Chinese word "的(de)". But according to the results, we can see that only the SAWR model distributes a high attention score to it, which is consistent with our intuition. The other three models are all aligned to the source word "现代 (modern)" with high confidence over 85%. The possible reason for "of" being aligned to "现代 (modern)" could be due to that "of modern" is a high-frequency collocation in the training corpora.

# 4.4.3 Ensemble Study

Here we perform model ensembles to examine the divergences of the three syntax-integration approaches (Zhou et al., 2017b; Denkowski and Neubig, 2017). Intuitively, the heteroapproach ensemble which combines three NMT models of different methods should obtain better performances than homo-approach ensembles which combine three NMT models of the same method, since NMT models of different syntaxintegrations approaches have larger divergences.

Table 4 shows the results. First, we can see that ensemble is one effective technique to improve the translation performances. More impor-



Figure 4: The effect of source input length.

tantly, the results show that the heterogeneous ensemble achieves averaged BLEU improvements by 43.10 - 41.24 = 1.86 points, better than the gains achieved by all three homo-approach ensembles, denoting that the three approaches could be mutually complementary in representing dependency syntax, and the resulting models of the three approaches are highly diverse.

# 4.4.4 Analysis by Source Sentence Length

Intuitively, by introducing the source syntax into the NMT model, relations between long-distance words are explicitly modeled by dependency trees, thus we can expect that models enhanced by source syntax are able to bring better translations for longer sentences. Figure 4 shows the performances of the baseline and all syntax-enriched models in terms of source sentence lengths, where we bin all the MT03-MT06 sentences by their lengths into six intervals. The results show that the BLEU scores are improved significantly when source sentential lengths are over 10, which confirms our intuition.

### 4.4.5 Effect of Parsing Performance

Finally, we examine how the performance of the dependency parser influences the final translation quality. While the full dependency parser is

System	MT03	MT04	MT05	MT06	Average/ $\Delta$
Transformer	40.45	42.76	40.09	39.67	40.74
SAWR	41.63	43.60	41.68	40.21	41.78/+1.04
Tree-RNN	41.24	43.38	41.04	40.02	41.42/+0.68
Tree-Linearization	41.12	43.02	41.04	39.86	41.26/+0.52

Table 5: Final results based on the transformer. Only the SAWR results are significantly better (p < 0.05).



Figure 5: The effect of dependency parsing performances on our proposed approaches.

trained on 50K sentences, we retrain three weaker dependency parsers on 30K, 10K and 5K sentences, respectively. Figure 5 shows the NMT BLEU scores and the parsing accuracies. It is clear that the parsing accuracy directly influences the translation quality, indicating the effectiveness and importance of exploiting syntactic information.

## 4.4.6 Transformer as Baseline

Here we conduct experiments based on the transformer NMT model (Vaswani et al., 2017), which is a stronger baseline, to further verify the effectiveness of our proposed method. This also demonstrates that the proposed SAWR method does not limit to a certain NMT baseline. Concretely, we extend the bottom word representations by incorporating syntactic encodings  $s=s_1\cdots s_n$  (shown in Equation 5) into them, and then feed them into the transformer encoder by a linear projection layer to align with the input dimension. We implement Tree-RNN and Tree-Linearization for Transformer in a similar way, only adapting the source input word representing. We adopt a widely-used setting with 8 heads, 6 layers and the hidden dimension size of 512.

Table 5 shows the results. As shown, the transformer results are indeed much better than RNNbased baseline. The BLEU scores show an average increase of 40.74 - 37.09 = 3.65. In addition, we can see that syntax information can still give positive influences based on the transformer. The SAWR approach can also outperform the baseline system significantly. Particularly, we find that our SAWR approach is much more effective than the Tree-RNN and Tree-Linearization approaches. The results further demonstrate the effectiveness of SAWRs in syntax integration for NMT.

### 5 Related Work

By explicitly expressing the structural connections between words and phrases, syntax trees been demonstrated helpful in SMT (Liu et al., 2006; Cowan et al., 2006; Marton and Resnik, 2008; Xie et al., 2011; Li et al., 2013; Williams et al., 2016). Although the representative Seq2Seq NMT models are able to capture latent long-distance relations by using neural network structures such GRU and LSTM (Sutskever et al., 2014; Wu et al., 2016), recent studies show that explicitly integrating syntax trees into NMT models can bring further gains (Sennrich and Haddow, 2016; Shi et al., 2016; Zhou et al., 2017a; Wu et al., 2017a; Aharoni and Goldberg, 2017). Under the NMT setting, the exploration of syntax trees could be more flexible, because of the strong capabilities of neural network in representing arbitrary structures.

Recursive neural networks based on LSTM or GRU have been one natural method to model syntax trees (Zhu et al., 2015; Tai et al., 2015; Li et al., 2015; Zhang et al., 2016; Teng and Zhang, 2016; Miwa and Bansal, 2016; Kokkinos and Potamianos, 2017), which are capable of representing the entire trees globally. Eriguchi et al. (2016) present the first work to apply a bottom-up Tree-LSTM for NMT. The major drawback is that its bottomup composing strategy is insufficient for bottom nodes. Thus bi-directional extensions have been suggested (Chen et al., 2017a; Yang et al., 2017). Since Tree-RNN suffers serious inefficiency problem, Li et al. (2017) suggest a Tree-Linearization alternative, which converts constituent trees into a sequence of symbols mixed with words and syntactic tags. The method is as effective as Tree-RNN approaches yet more effective. Noticeably, all these studies focus on constituent trees.

There have been several studies for NMT using dependency syntax. Hashimoto and Tsuruoka (2017) propose to combine the head information with sequential words together as source encoder inputs, where their input trees are latent dependency graphs. Recently, there are several studies by using convolutional neural structures to represent source dependency trees, where tree nodes are modeled individually (Chen et al., 2017b; Bastings et al., 2017). Wu et al. (2017b) build a syntax enhanced encoder by multiple Bi-RNNs over several different word sequences based on different traversing orders over dependency trees, i.e., the original sequential order and several tree-based orders. All these methods require certain extra efforts to encode the source dependency syntax over a baseline Seq2Seq NMT.

# 6 Conclusion

We proposed a novel syntax integration method, SAWR, to incorporate source dependency-based syntax for NMT. It encodes dependency syntax implicitly, not requiring discrete syntax trees as inputs. Experiments showed that the method can bring significantly better performances for both Chinese-English and English-Vietnamese translation tasks. In addition, we compared the method with two approaches based on Tree-RNN and Tree-Linearization, which has been previously exploited for syntax integration, finding that our method is more effective and meanwhile very efficient. We conducted several experimental analyses to study our proposed methods deeper.

## Acknowledgments

We thank all anonymous reviewers for their valuable comments. We thank Huadong Chen, Haoran Wei and Zaixiang Zheng for their help in implementing baseline neural machine translation models. This work is supported by National Natural Science Foundation of China (NSFC) grants 61525205, U1836222, and 61672211.

# References

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th ACL*, pages 132–140.

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of EMNLP*, pages 1957–1967.
- Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017a. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of ACL*, pages 1936–1945.
- Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. 2017b. Neural machine translation with source dependency representation. In *Proceedings* of *EMNLP*, pages 2846–2852.
- Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *ACL*.
- 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-8*, pages 103– 111.
- Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.
- Brooke Cowan, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of EMNLP*, pages 232–241.
- Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *The First Workshop on Neural Machine Translation (NMT)*.
- Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.
- Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of ACL*, pages 823–833.
- Kazuma Hashimoto and Yoshimasa Tsuruoka. 2017. Neural machine translation with source-side latent graph parsing. In *Proceedings of EMNLP*, pages 125–135.
- Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt'15. In *Proceedings of SMT*, pages 134–140.

- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395.
- Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.
- Filippos Kokkinos and Alexandros Potamianos. 2017. Structural attention neural networks for improved sentiment analysis. In *Proceedings of EACL*, pages 586–591.
- Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When are tree structures necessary for deep learning of representations? In *Proceedings of the EMNLP*, pages 2304–2314.
- Junhui Li, Philip Resnik, and Hal Daumé III. 2013. Modeling syntactic and semantic structures in hierarchical phrase-based translation. In *Proceedings of NAACL*, pages 540–549.
- Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. In *Proceed*ings of ACL, pages 688–697.
- Yang Liu, Qun Liu, and Shouxun Lin. 2006. Treeto-string alignment template for statistical machine translation. In *Proceedings of ACL*, pages 609–616.
- Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of IWSLT* 2015, pages 76–79.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the* 2015 EMNLP, pages 1412–1421.
- Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL*, pages 1003–1011.
- Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings* of *EMNLP*, pages 955–960.
- Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*, pages 1105–1116.
- Graham Neubig, Yoav Goldberg, and Chris Dyer. 2017. On-the-fly operation batching in dynamic computation graphs. In *Conference on Neural Information Processing Systems (NIPS)*.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*, pages 2227– 2237.
- Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th ACL*, pages 1715–1725, Berlin, Germany.
- Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of ACL*, pages 1683–1692.
- Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of EMNLP*, pages 1526–1534.
- Weiwei Sun and Xiaojun Wan. 2013. Data-driven, pcfg-based and pseudo-pcfg-based models for chinese dependency parsing. *TACL (TACL)*, 1(1):301–314.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, 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 ACL, pages 1556–1566.
- Zhiyang Teng and Yue Zhang. 2016. Bidirectional tree-structured lstm with head lexicalization. *arXiv* preprint arXiv:1611.06788.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.
- Philip Williams, Rico Sennrich, Matt Post, and Philipp Koehn. 2016. Syntax-based statistical machine translation. *Synthesis Lectures on Human Language Technologies*, 9(4):1–208.
- Shuangzhi Wu, Dongdong Zhang, Nan Yang, Mu Li, and Ming Zhou. 2017a. Sequence-to-dependency neural machine translation. In *Proceedings of ACL*, pages 698–707.
- Shuangzhi Wu, Ming Zhou, and Dongdong Zhang. 2017b. Improved neural machine translation with source syntax. In *Proceedings of IJCAI 2017*, pages 4179–4185.

- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proceedings of EMNLP*, pages 216–226.
- Baosong Yang, Derek F. Wong, Tong Xiao, Lidia S. Chao, and Jingbo Zhu. 2017. Towards bidirectional hierarchical representations for attention-based neural machine translation. In *Proceedings of the 2017 Conference on EMNLP*, pages 1432–1441.
- Xingxing Zhang, Liang Lu, and Mirella Lapata. 2016. Top-down tree long short-term memory networks. In *Proceedings of NAACL*, pages 310–320.
- Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2017. Modeling past and future for neural machine translation. arXiv preprint arXiv:1711.09502.
- Hao Zhou, Zhaopeng Tu, Shujian Huang, Xiaohua Liu, Hang Li, and Jiajun Chen. 2017a. Chunk-based bi-scale decoder for neural machine translation. In *Proceedings of the 55th ACL*, pages 580–586.
- Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017b. Neural system combination for machine translation. In *Proceedings of ACL*, pages 378–384.
- Xiao-Dan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *ICML*, pages 1604–1612.