Dependency-Based Relative Positional Encoding for Transformer NMT

Yutaro Omote and **Akihiro Tamura** and **Takashi Ninomiya**
Ehime University
{omote@ai.cs, tamura@cs, ninomiya@cs}.ehime-u.ac.jp

Abstract

In this paper, we propose a novel model for Transformer neural machine translation that incorporates syntactic distances between two source words into the relative position representations of a self-attention mechanism. In particular, the proposed model encodes pair-wise relative depths on a source dependency tree, which are the differences between the depths of two source words, in the encoder's self-attention. Experiments show that our proposed model achieved a 0.5 point gain in BLEU on the Asian Scientific Paper Excerpt Corpus Japanese-to-English translation task.

1 Introduction

Machine translation (MT) has been actively studied for many decades. In recent years, neural machine translation (NMT) has become dominant. In particular, the Transformer model (Vaswani et al., 2017), which is based solely on attention mechanisms, has advanced the state-of-the-art performance on various translation tasks and has become the focus of many MT researchers nowadays. Unlike recurrent neural network (RNN) based models (Sutskever et al., 2014; Luong et al., 2015) or convolutional neural network (CNN) based models (Gehring et al., 2017), the Transformer model attends to words in the same sentence, i.e., a source sentence or a target sentence, through the self-attention mechanisms in each encoder and decoder. In addition, it encodes the positional information of each word, such as the word order, as positional encoding (PE) so that recurrent and convolutional structures are excluded and training can be parallelized. Since NMT appeared, translation performance has been improved by using the syntactic information, such as phrase structures or dependency structures, of the source-side, target-side, or both (Ding and Palmer, 2005; Chen et al., 2017; Eriguchi et al., 2017; Wu et al., 2018). In semantic role labeling (SRL), though the task is not MT, Strubell et al. (2018) improved a Transformer-based model through the learning of self-attention weighting on the basis of syntactic information, i.e., dependency structures. Hence, it is expected that the performance of Transformer NMT will be improved by incorporating syntax information.

In this paper, we aim to improve Transformer NMT by using dependency structures. Some researchers have improved Transformer NMT by modifying self-attention. Shaw et al. (2018) used relative position information between two words encoded in self-attention in addition to the absolute position information of words.

Inspired by Shaw et al. (2018), we propose a novel Transformer NMT model that incorporates the relationships between two words on source dependency structures into relative position representations in self-attention. In particular, the proposed model adds a vector that encodes relative positional relationships between words on source dependency structures to a word embedding vector. It adds only dependency information to the word embedding vector; hence, there is no need to change the whole Transformer's mechanism or objective function, and it is easy to adapt the mechanism to other extended Transformer models because it is highly extensible. Strubell et al. (2018)'s method is different from our work in that their task is SRL, and to learn the attention between words directly from dependency structures, they largely changed the Transformer's model and objective function.

We evaluate the effectiveness of the proposed model on the WAT'18 Asian Scientific Paper Ex-

Proceedings of Recent Advances in Natural Language Processing, pages 854–861,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_099

cerpt Corpus (ASPEC) Japanese-to-English translation task. The experimental results demonstrate that our approach achieves a 0.5 point gain in BLEU over baseline Transformers (Vaswani et al., 2017; Shaw et al., 2018).

2 Related Work

NMT performance has been improved by using the syntactic information of source language sentences, target language sentences, or both.

Some researchers have focused on phrase structures as syntactic information. Aharoni and Goldberg (2017) incorporated target-side phrase structures into NMT, and Eriguchi et al. (2016) and Ma et al. (2018) incorporated source-side phrase structures. Our work is different from their research in that we focus on dependency structures rather than phrase structures. In addition, while their models are based on RNN-based NMT models, we aim to improve a Transformer NMT model.

Other researchers have focused on dependency structures as syntactic information. Chen et al. (2017) proposed a hybrid NMT model of RNNs and CNNs to incorporate syntactic information into an encoder. Their model first learns source dependency representations to compute dependency context vectors by using CNNs. The RNN-based encoder-decoder model learns a translation model, which is provided with the CNNs' syntactic information. Sennrich and Haddow (2016) proposed an RNN-based NMT model that combines embedding vectors of linguistic features such as part-of-speech tags and dependency relation labels on a source sentence with the embedded representations of the source words. Eriguchi et al. (2017) proposed a hybrid model, called NMT+RNNG, that learns parsing and translation by combining recurrent neural network grammar into an RNN-based NMT.

Most existing dependency-based NMT models, including the above-mentioned models, are improvements over RNN-based NMT models, which, in terms of structure, differ greatly from the Transformer model. Because we make the proposed model consider dependency information in self-attention, which is the Transformer's characteristic structure, the usage of dependency information is different from their models.

Recently, Wu et al. (2018) and Ma et al. (2019) incorporated syntactic information into Transformer NMT. Wu et al. (2018) proposed a dependency-based NMT model that uses dependency trees for both source and target languages. Their model encodes source sentences with two extra sequences linearized from source dependency trees and jointly generates both target sentences and their dependency trees. They applied their model not only to bi-directional RNNs but also to the Transformer, but did not improve the Transformer's architecture. In contrast, we improve the Transformer model so that it incorporates source dependency information by encoding pair-wise relative depths on a source dependency tree, which are the differences between the depths of two source words, in the encoder's self-attention.

Ma et al. (2019) proposed several strategies for improving NMT with neural syntax distance (NSD), which has been used for constituent parsing (Shen et al., 2018), and dependency-based NSD, which is an extension of the original NSD for dependency trees. In their work, they proposed a syntactic PE for Transformer NMT in order to incorporate positions on a dependency tree for each word via an absolute PE mechanism. In contrast, our model uses relative dependency-based distances between two words via a relative PE mechanism in the encoder's self-attention.

3 Background

In this section, we first describe the baseline of our proposed model, the Transformer model. Then, we describe a Transformer model that employs relative PE.

3.1 Transformer

Transformer (Vaswani et al., 2017) is an encoder-decoder model that has a distinct architecture based on self-attention. Figure 1 shows the architecture of the model. Unlike RNN-based NMT and CNN-based NMT, Transformer does not have a recurrent or convolutional configuration of networks. Instead, it encodes source sentences as intermediate representations by using self-attention and decodes them by using self-attention and encoder-decoder attention.

The encoder maps an input sequence $(\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})$ to a sequence of vector representations $Z = (\boldsymbol{z_1}, \ldots, \boldsymbol{z_n})$. Given Z, the decoder generates an output sequence $(\boldsymbol{y_1}, \ldots, \boldsymbol{y_{n'}})$. In both the encoder and decoder, the embedding layer converts input tokens (source tokens in the

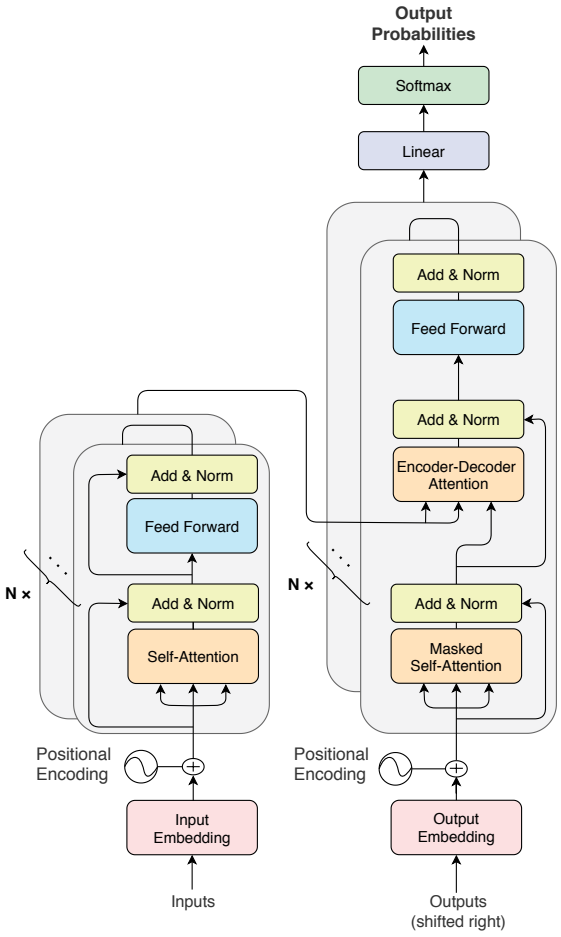

Figure 1: Architecture of Transformer

encoder and target tokens in the decoder) to vectors of dimension d_{model}. Because the information on proximity between tokens is not considered in self-attention itself, the information on a token's position is embedded by using positional encoding (PE). Specifically, PE provides a matrix that represents the absolute position information of tokens in a sentence, and Transformer adds PE to the embedding matrix of the input tokens. Each element of PE is computed by the following equations, which are sine and cosine functions of different frequencies.

$$PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}}),$$
$$PE(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}),$$

where pos is the position of each input token, i is the dimension of each element, and d_{model} is the embedding dimension of an input token. The input of the encoder's or decoder's first layer is the embedding matrix added with the positional encoding.

The encoder's layer has two sub-layers. The first sub-layer is a multi-head self-attention mechanism, and the second layer is a simple, position-wise fully connected feed-forward network (FFN). The decoder's layer has three sub-layers. The first sub-layer is a masked multi-head self-attention mechanism, the second sub-layer is a multi-head encoder-decoder attention mechanism, and the third sub-layer is the FFN.

Residual connection (He et al., 2016) is applied to the sub-layers, followed by layer normalization (Ba et al., 2016), i.e., the output of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the output of the original sub-layer.

The self-attention and the encoder-decoder attention employ a multi-head attention mechanism. The multi-head attention first computes h dot-product attentions after linearly mapping three input vectors, $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}$[*1] $\in \mathbb{R}^{1 \times d_{model}}$, from d_{model} dimension to d_k dimension with parameter matrices, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d_k} (i = 1, \ldots, h)$, where d_{model} is the dimension of input vectors, and $d_k = d_{model}/h$. In what follows, each dot-product attention is referred to as a head (H_i $(i = 1, \ldots, h)$).

$$H_i = Attention(\boldsymbol{q'}, \boldsymbol{k'}, \boldsymbol{v'}), \quad (1)$$

$$Attention(\boldsymbol{q'}, \boldsymbol{k'}, \boldsymbol{v'}) = softmax(\frac{\boldsymbol{q'}\boldsymbol{k'}^T}{\sqrt{d_k}})\boldsymbol{v'}, \quad (2)$$

$$\boldsymbol{q'} = \boldsymbol{q}W_i^Q, \boldsymbol{k'} = \boldsymbol{k}W_i^K, \boldsymbol{v'} = \boldsymbol{v}W_i^V. \quad (3)$$

Then, multi-head attention linearly maps concatenated heads with a parameter matrix, $W^o \in \mathbb{R}^{d_{model} \times d_{model}}$.

$$MultiHead(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = Concat(H_1, \ldots, H_h)W^O. \quad (4)$$

The encoder's self-attention computes Equation 4 by substituting the intermediate states of the encoder, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, for $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}$. Specifically, each head computes the following weighted sum.

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_{ij} \boldsymbol{x}_j W^V, \quad (5)$$

where $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ are the outputs of the self-attention. Each coefficient, α_{ij}, is computed by using a softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}, \quad (6)$$

[*1] In this paper, we treat a vector as a row vector according to the original paper (Vaswani et al., 2017) unless otherwise noted.

where e_{ij} is computed:

$$e_{ij} = \frac{(\boldsymbol{x}_i W^Q)(\boldsymbol{x}_j W^K)^T}{\sqrt{d_z}}, \quad (7)$$

where d_z is the dimension of $\boldsymbol{z}_i$.

The decoder's self-attention computes Equation 4 by substituting the intermediate states of the decoder for $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}$. During inference, however, it is not possible for the decoder to get the information on the words that will be generated later when predicting a word, i.e., only the intermediate states of the sub-sequence that has been generated can be used for self-attention. Hence, masked self-attention is introduced to the decoder's self-attention so as not to calculate the self-attention between a predicted word and succeeding words. Masked self-attention is calculated by changing Equation 7:

$$e_{ij} = \begin{cases} \frac{(\boldsymbol{x}_i W^Q)(\boldsymbol{x}_j W^K)^T}{\sqrt{d_z}} & (i \geq j), \\ -\infty & (otherwise). \end{cases} \quad (8)$$

The coefficient representing the strength of the relationship between a certain word and the word located behind it ($i < j$) becomes zero, and it can be controlled so as not to consider the relationship. Hence, Equation 6 is changed:

$$\alpha_{ij} = \begin{cases} \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} & (i \geq j), \\ 0 & (otherwise). \end{cases} \quad (9)$$

In the encoder-decoder attention, the intermediate states of the decoder are used for $\boldsymbol{q}$, and the outputs of the encoder are used for $\boldsymbol{k}, \boldsymbol{v}$.

The FFN for input $\boldsymbol{x}$ compute as follows:

$$FFN(\boldsymbol{x}) = max(0, \boldsymbol{x}W_1 + \boldsymbol{b}_1)W_2 + \boldsymbol{b}_2, \quad (10)$$

where $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$ are parameter matrices, and $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ are biases.

3.2 Transformer with Relative Positional Encoding

Shaw et al. (2018) proposed an extended transformer model that captures the pairwise relationships between input elements in terms of relative positions in both the encoder and decoder. In their method, the relationships between the intermediate representations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, i.e., relative position information between the i-th and j-th words in an input sentence, are represented by vectors $\boldsymbol{a}_{ij}^V, \boldsymbol{a}_{ij}^K \in \mathbb{R}^{d_k}$. The relative position representations are added to the output of the sub-layer to be the input to the next layer. Specifically, the following equation is used instead of Equation 5.

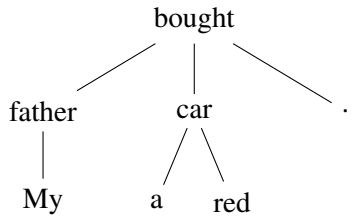

Figure 2: Example of Dependency Tree

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\boldsymbol{x}_j W^V + \boldsymbol{a}_{ij}^V). \quad (11)$$

The following equation is also used for the substitution of Equation 7 in order to consider relative position relationships between words in calculating e_{ij}:

$$e_{ij} = \frac{\boldsymbol{x}_i W^Q (\boldsymbol{x}_j W^K + \boldsymbol{a}_{ij}^K)^T}{\sqrt{d_z}}. \quad (12)$$

Shaw et al. (2018) assume that relative position information is not useful when the distance is long. They define the maximum relative position as a constant k. In addition, the relative position relationships between two words are captured by $2k + 1$ unique labels as follows, considering that succeeding words are in a positive direction and preceding words are in a negative direction.

$$\boldsymbol{a}_{ij}^K = \boldsymbol{w}_{clip(j-i,k)}^K, \quad (13)$$

$$\boldsymbol{a}_{ij}^V = \boldsymbol{w}_{clip(j-i,k)}^V, \quad (14)$$

$$clip(x, k) = max(-k, min(k, x)), \quad (15)$$

where $\boldsymbol{w}^K = (\boldsymbol{w}_{-k}^K, \ldots, \boldsymbol{w}_k^K)$ and $\boldsymbol{w}^V = (\boldsymbol{w}_{-k}^V, \ldots, \boldsymbol{w}_k^V)$ $(\boldsymbol{w}_k^K, \boldsymbol{w}_k^V \in \mathbb{R}^{d_k})$ are relative position representations to be learned.

4 Dependency-Based Relative Positional Encoding for Transformer

In this section, we explain our proposed method, which encodes relative positions on source dependency trees in Transformer. We first introduce the inter-word distance on source dependency trees and then explain dependency-based relative positional encoding, which provides relative position representations on the trees. The dependency-based encoding is incorporated into

	My	father	bought	a	red	car	.
My	0	-1	-2	0	0	-1	-1
father	1	0	-1	1	1	0	0
bought	2	1	0	2	2	1	1
a	0	-1	-2	0	0	-1	-1
red	0	-1	-2	0	0	-1	-1
car	1	0	-1	1	1	0	0
.	1	0	-1	1	1	0	0

Table 1: Examples of Dependency-based Inter-Word Distances

the self-attention mechanism, following the idea of relative positional encoding (Shaw et al., 2018).

The inter-word distance on dependency trees is defined as the relative depth between two words in dependency trees. The relative depth $dist_{ij}$ between node n_i and node n_j corresponding to word w_i and word w_j is defined as follows:

$$dist_{ij} = depth(n_j) - depth(n_i), \quad (16)$$

where $depth(n)$ is the depth of node n in a dependency tree. For example, in Figure 2, the depth of "bought" (w_3) relative to "My" (w_1) is calculated by $dist_{1,3} = 0 - 2 = -2$. Table 1 shows a list of the inter-word distances on the dependency tree shown in Figure 2.

The relative position between node n_i and node n_j in a source dependency tree is represented by vectors, $\boldsymbol{b}^V_{ij}, \boldsymbol{b}^K_{ij} \in \mathbb{R}^{d_k}$, and the following equations are used instead of Equations 11 and 12.

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\boldsymbol{x}_j W^V + \boldsymbol{b}^V_{ij}), \quad (17)$$

$$e_{ij} = \frac{\boldsymbol{x}_i W^Q (\boldsymbol{x}_j W^K + \boldsymbol{b}^K_{ij})^T}{\sqrt{d_z}}. \quad (18)$$

We assume that the influence of a distance decreases if the distance is longer than some certain threshold. We limit the maximum distance to a constant l. The relative position representations $\boldsymbol{b}^V_{ij}$ and $\boldsymbol{b}^K_{ij}$ between node n_i and node n_j in a dependency tree are defined with inter-word distance labels:

$$\boldsymbol{b}^K_{ij} = \boldsymbol{w}^K_{clip(dist_{ij},l)}, \quad (19)$$

$$\boldsymbol{b}^V_{ij} = \boldsymbol{w}^V_{clip(dist_{ij},l)}. \quad (20)$$

Using these expressions, the encoder's self-attention networks learn the relative position representations on a source dependency structure. We call this model $Transformer_{dep}$.

We also describe a hybrid model that learns both relative position representations on dependency structures and relative position representations for linear relations in sentences, i.e., the relative positional encoding explained in Section 3.2. This hybrid method is called $Transformer_{dep+rel}$. The $Transformer_{dep+rel}$ model uses the sum of $\boldsymbol{a}^V_{ij}$ and $\boldsymbol{b}^V_{ij}$ and the sum of $\boldsymbol{a}^K_{ij}$ and $\boldsymbol{b}^K_{ij}$ as relative position information between two words. $\boldsymbol{z}_i$ and e_{ij} are defined in $Transformer_{dep+rel}$ as follows:

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\boldsymbol{x}_j W^V + \boldsymbol{a}^V_{ij} + \boldsymbol{b}^V_{ij}), \quad (21)$$

$$e_{ij} = \frac{\boldsymbol{x}_i W^Q (\boldsymbol{x}_j W^K + \boldsymbol{a}^K_{ij} + \boldsymbol{b}^K_{ij})^T}{\sqrt{d_z}}. \quad (22)$$

5 Experiments

5.1 Experimental Setup

We experimented on the WAT'18 Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016) by using the Japanese-to-English language pair. We tokenized English sentences by using Moses (Koehn et al., 2007) and Japanese sentences by using KyTea (Neubig et al., 2011). We also parsed the dependency of the Japanese sentences by using EDA[*2].

For model learning, we used 1,341,417 sentence pairs of 50 words or less for both the English and Japanese sentences from the first 1.5 million sentence pairs of the training data (train-1.txt, train-2.txt). The Japanese dictionary was comprised of words that appeared 7 times or more in the training data, and the English dictionary was comprised of words that appeared 10 times or more in the training data. The other words were replaced with $\langle UNK \rangle$ tags representing unknown words. We used 1,790 sentences (dev.txt) as validation data and 1,812 sentences (test.txt) as test data.

We compared our models, $Transformer_{dep}$ and $Transformer_{dep+rel}$, with two baseline Transformer NMT models, $Transformer_{abs}$ (Vaswani et al., 2017), which learns absolute position representations, and $Transformer_{rel}$ (Shaw et al., 2018), which learns relative position representations in a sentence.

Hyper-parameters of all Transformer models were determined, following the settings of Vaswani et al. (2017). We set the number of stacks

[*2] http://www.ar.media.kyoto-u.ac.jp/tool/EDA/

Model	BLEU
$Transformer_{abs}$	25.91
$Transformer_{rel}$	26.72
$Transformer_{dep}$	26.10
$Transformer_{dep+rel}$	27.22

Table 2: Experimental Results

$\boldsymbol{b}^V_{ij}$	$\boldsymbol{b}^K_{ij}$	BLEU
✓	✓	16.71
×	✓	16.60
✓	×	15.62
×	×	8.69

Table 3: Experimental Results for Ablating Relative Position Representations $\boldsymbol{b}^V_{ij}, \boldsymbol{b}^K_{ij}$

of the encoder and decoder layers to 6, the number of heads to 8, and the embedding dimension to 512. We used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$. We used the same warm-up and decay strategy for the learning rate as Vaswani et al. (2017), with $4,000$ warm-up steps.

The maximum distances of the relative position for linear relations in the sentences and dependency-based relations were set as $k = 2$, $l = 2$ for $Transformer_{rel}$, $Transformer_{dep}$, and $Transformer_{dep+rel}$[*3]. The batch size was 256, the number of epochs was 30, and the model with the best accuracy for the validation data was applied to the test data. In our experiments, target sentences were generated by a greedy algorithm.

5.2 Results

The evaluation results are shown in Table 2. We used BLEU to evaluate the translation performance. As shown in the table, $Transformer_{dep}$ improved by 0.19 BLEU points against $Transformer_{abs}$. This means that the dependency-based positional encoding was effective for the Transformer model. Although the effectiveness of our dependency-based positional encoding ($Transformer_{dep}$) was not as great as the relative positional encoding ($Transformer_{rel}$), the hybrid model ($Transformer_{dep+rel}$) achieved the best result among these models. $Transformer_{dep+rel}$ improved by 1.31 BLEU points against $Transformer_{abs}$ and by 0.50 BLEU points against $Transformer_{rel}$. From these results, on the Japanese-to-English translation task, the performance of Transformer NMT can be improved by incorporating source dependency structures into relative position representations.

5.3 Discussion

Shaw et al. (2018) verified the effectiveness of both $\boldsymbol{a}^V_{ij}$ and $\boldsymbol{a}^K_{ij}$, representing relative positional relationships. They showed that the translation accuracy was comparative when $\boldsymbol{a}^V_{ij}$ was removed from their model, but the translation accuracy decreased when $\boldsymbol{a}^K_{ij}$ was removed. This means that $\boldsymbol{a}^K_{ij}$ was an effective representation, but $\boldsymbol{a}^V_{ij}$ was less effective. In this section, to confirm the effectiveness of the dependency-based representations, we conducted ablation experiments on $\boldsymbol{b}^V_{ij}$ and $\boldsymbol{b}^K_{ij}$. We evaluated the Japanese-English translation performance for $Transformer_{dep}$. The absolute positional encoding was removed from all models, following the settings of Shaw et al. (2018)'s verification. Specifically, we evaluated (i) the $Transformer_{dep}$ model that used only $\boldsymbol{b}^V_{ij}$, where Equation 18 was changed to Equation 7, (ii) the $Transformer_{dep}$ model that used only $\boldsymbol{b}^K_{ij}$, where Equation 17 was changed to Equation 5, and (iii) the $Transformer_{dep}$ model that used neither $\boldsymbol{b}^V_{ij}$ and $\boldsymbol{b}^K_{ij}$, i.e., $Transformer_{abs}$ without the absolute positional encoding.

The settings for the ablation experiment were as follows. In model training, we used the first 100,000 sentence pairs of 50 words or less for both English and Japanese sentences, which were extracted from the training data (train-1.txt). Both the Japanese dictionary and the English dictionary were comprised of words that appeared 2 times or more in the training data, and the other words were treated as unknown words with UNK tags. The batch size was 100, and the number of epochs was 50. Other settings were the same as the main experiments in Section 5.1.

The results are shown in Table 3. Table 3 shows that $Transformer_{dep}$ using only $\boldsymbol{b}^V_{ij}$ was 1.09 points lower than the baseline $Transformer_{dep}$, which used both $\boldsymbol{b}^V_{ij}$ and

[*3] We chose $k = 2$ because Shaw et al. (2018) showed that BLEU scores for $k \geq 2$ are nearly unchanged. l was tuned on development data.

$\boldsymbol{b}^K_{ij}$, while $Transformer_{dep}$ using only $\boldsymbol{b}^K_{ij}$ was 0.11 points slightly lower than the baseline $Transformer_{dep}$. Table 3 also shows that the $Transformer_{dep}$ that used neither $\boldsymbol{b}^V_{ij}$ and $\boldsymbol{b}^K_{ij}$ was 8.02 points lower than the baseline $Transformer_{dep}$, which was significantly worse.

These results were consistent with the experimental results in Shaw et al. (2018). The dependency-based relative position representations, $\boldsymbol{b}^K_{ij}$ and $\boldsymbol{b}^V_{ij}$, were shown to be effective, but $\boldsymbol{b}^K_{ij}$ was more effective than $\boldsymbol{b}^V_{ij}$.

6 Conclusion

In this paper, we proposed a novel Transformer NMT model that incorporates syntactic distances between two source words into the relative positional encoding of an encoder's self-attention mechanism. We demonstrated that our proposed model improved the translation accuracy, in terms of BLUE score, on the ASPEC Japanese-to-English translation task.

For future work, we would like to improve our model by introducing relative positional encoding to target dependency structures, i.e., dependency-based relative positional encoding for decoders. For example, we would like to integrate our encoding into the dependency-based decoder in (Wu et al., 2018).

Acknowledgement

The research results have been achieved by "Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation", the Commissioned Research of National Institute of Information and Communications Technology (NICT) , JAPAN. This work was partially supported by JSPS KAKENHI Grant Number JP18K18110.

References

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 132–140. https://doi.org/10.18653/v1/P17-2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. 2017. Neural machine translation with source dependency representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 2846–2852. https://doi.org/10.18653/v1/D17-1304.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, pages 541–548. http://aclweb.org/anthology/P05-1067.

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 (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 823–833. https://doi.org/10.18653/v1/P16-1078.

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

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*. PMLR, International Convention Centre, Sydney, Australia, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 770–778. https://doi.org/10.1109/CVPR.2016.90.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and*

Poster Sessions. Association for Computational Linguistics, Prague, Czech Republic, pages 177–180. https://www.aclweb.org/anthology/P07-2045.

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*. Association for Computational Linguistics, pages 1412–1421.

Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2019. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 2032–2037. https://www.aclweb.org/anthology/N19-1205.

Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Tiejun Zhao, and Eiichiro Sumita. 2018. Forest-based neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 1253–1263. https://www.aclweb.org/anthology/P18-1116.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*. pages 2204–2208.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 529–533. http://aclweb.org/anthology/P11-2093.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*. Association for Computational Linguistics, pages 83–91. https://doi.org/10.18653/v1/W16-2209.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, pages 464–468. https://doi.org/10.18653/v1/N18-2074.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018. Straight to the tree: Constituency parsing with neural syntactic distance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 1171–1180. https://doi.org/10.18653/v1/P18-1108.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 5027–5038. http://aclweb.org/anthology/D18-1548.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Shuangzhi Wu, Dongdong Zhang, Zhirui Zhang, Nan Yang, Mu Li, and Ming Zhou. 2018. Dependency-to-dependency neural machine translation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 26(11):2132–2141. https://doi.org/10.1109/TASLP.2018.2855968.