On the Relation between Position Information and Sentence Length in Neural Machine Translation

Masato Neishi
The University of Tokyo
neishi@tkl.iis.u-tokyo.ac.jp

Naoki Yoshinaga
Institute of Industrial Science,
the University of Tokyo
ynaga@iis.u-tokyo.ac.jp

Abstract

Long sentences have been one of the major challenges in neural machine translation (NMT). Although some approaches such as the attention mechanism have partially remedied the problem, we found that the current standard NMT model, Transformer, has difficulty in translating long sentences compared to the former standard, Recurrent Neural Network (RNN)-based model. One of the key differences of these NMT models is how the model handles position information which is essential to process sequential data. In this study, we focus on the position information type of NMT models, and hypothesize that relative position is better than absolute position. To examine the hypothesis, we propose RNN-Transformer which replaces positional encoding layer of Transformer by RNN, and then compare RNN-based model and four variants of Transformer. Experiments on ASPEC English-to-Japanese and WMT2014 English-to-German translation tasks demonstrate that relative position helps translating sentences longer than those in the training data. Further experiments on length-controlled training data reveal that absolute position actually causes overfitting to the sentence length.

1 Introduction

Sequence to sequence models for neural machine translation (NMT) are now utilized for various text generation tasks including automatic summarization (Chopra et al., 2016; Nallapati et al., 2016; Rush et al., 2015) and dialogue systems (Vinyals and Le, 2015; Shang et al., 2015); the models are required to take inputs of various length. Early studies on recurrent neural network (RNN)-based model analyze the translation quality with respect to the sentence length, and show that their models improve translations for long sentences, using the long short-term memory (LSTM) (Sutskever et al., 2014) or introducing the attention mechanism (Bahdanau et al., 2015; Luong et al., 2015). However, Koehn and Knowles (2017) report that even RNN-based model with the attention mechanism performs worse than phrase-based statistical machine translation (Koehn et al., 2007) in translating very long sentences, which challenges us to develop an NMT model that is robust to long sentences or more generally, variations in input length.

Have the recent advances in NMT achieved the robustness to the variations in input length? NMT has been advancing by upgrading the model architecture: RNN-based model (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015) followed by convolutional neural network (CNN)-based model (Kalchbrenner et al., 2016; Gehring et al., 2017) and attention-based model (Vaswani et al., 2017) called Transformer (§ 2). Transformer is the de facto standard NMT model today for its better performance compared to the former standard RNN-based model. We thus came up with a question whether Transformer have acquired the robustness to the variations in input length.

On the length of input sentence(s), the key difference between existing NMT models is how they incorporate information on word positions in the input. RNN or CNN-based NMT captures *relative* positions which stem from sequential operation of RNN or convolution operation of CNN. On the other hand, position embeddings or positional encodings (vector representations of positions) are used to handle *absolute* positions in Transformer. Gehring et al. (2017) integrate position embeddings, which are induced together with the other model parameters, into the CNN-based model, and showed that absolute position is still beneficial for their model in addition to the relative position captured by CNN. By contrast, Transformer only em-

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 328–338
Hong Kong, China, November 3-4, 2019.

ploys positional encodings, which give fixed vectors to positions using sine and cosine functions.

In this study, we suspect that these differences in position information types of the models have an impact on the accuracy of translating long sentences, and investigate the impact of position information on translating long sentences to realize an NMT model that is robust to variations in input length. We reveal that RNN-based model (relative position) is better than Transformer with positional encodings (absolute position) in translating longer sentences than those in the training data (§ 5.2). Motivated from this result, we propose a simple modification to Transformer, using RNN as relative positional encoder (§ 4).

Whereas RNN and CNN-based models are inseparable from relative position inside of RNN or CNN, Transformer allows us to change the position information type. We therefore compare the RNN-based model and four variants of Transformer: vanilla Transformer, the modified Transformer using self-attention with relative positional encodings (Shaw et al., 2018), our modified Transformer with RNN instead of positional encoding layer, and a mixture of the last two models (§ 5). On ASPEC English-to-Japanese and WMT2014 English-to-German translation tasks, we show that relative information improves Transformer to be more robust to variations in input length.

Our contribution is as follows:

- We identified a defect in Transformer. Use of absolute position makes it difficult to translate very long sentences.
- We proposed a simple method to incorporate relative position into Transformer; it gives an additive improvement to the existing model by Shaw et al. (2018) which also incorporates relative position.
- We revealed the overfitting property of Transformer to both short and long sentences.

2 Related Work

Early studies on NMT, at that time RNN-based model, analyze the translation quality in terms of sentence length (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015), and a few studies shed light on the details. Shi et al. (2016) examine why RNN-based model generates translations of the right length without special mechanism for the length, and report how LSTM regulates the output length. Koehn and Knowles (2017) reveal that RNN-based model has lower translation quality on very long sentences. Although researchers have proposed various new NMT architecture, they usually evaluate their models only in terms of the overall translation quality and rarely mention how the translation has changed (Gehring et al., 2017; Kalchbrenner et al., 2016; Vaswani et al., 2017). Only a few studies do the analysis on the translation quality in terms of sentence length (Elbayad et al., 2018; Zhang et al., 2019). The robustness of the recent NMT models on very long sentences remains to be assessed.

What we focus on in this study is the word position information which will closely relate to the decodable sentence length. Relative information has been implicitly used in the models using RNN or CNN. Gehring et al. (2017) introduce position embeddings which represent absolute position information to their CNN-based model. Sukhbaatar et al. (2015) introduce another absolute position information, positional encodings, which need no parameter training, and Vaswani et al. (2017) adopt them in their model, Transformer, which has neither RNN nor CNN.

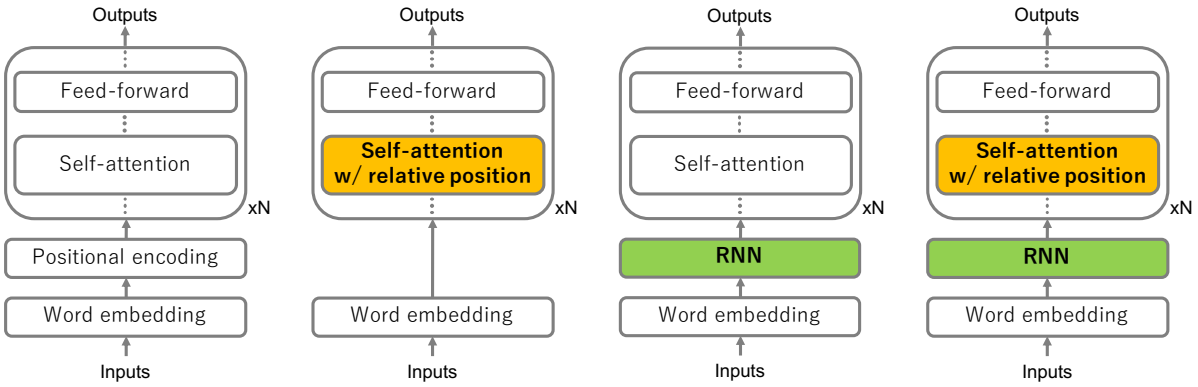
Recently, Shaw et al. (2018) propose to incorporate relative position into Transformer by modifying the self-attention layer while removing positional encodings. Lei et al. (2018) propose a fast RNN named Simple Recurrent Units (SRU) and replace the feed-forward layers of Transformer by SRU considering that recurrent process would better capture sequential information. Although both approaches succeeded in improving BLEU score, the researchers did not report in what respect the models improved the translation.

Chen et al. (2018) propose a RNN-based model, RNMT+, which is based on stacked LSTMs and incorporates some components from Transformer such as layer normalization and multi-head attention. On the other hand, our model is based on Transformer and incorporates RNN into Transformer.

3 Preliminaries

3.1 Transformer

Transformer (Vaswani et al., 2017) is a sequence to sequence model that has an encoder to process and represent input sequence and a decoder to generate output sequence from the encoder outputs. Both the encoder and decoder have a word embedding layer, a positional encoding layer, and



(a) Transformer (Vaswani et al., 2017)

(b) Rel-Transformer (Shaw et al., 2018)

(c) RNN-Transformer Our proposed

(d) RR-Transformer (b) + Our proposed

Figure 1: The architectures of all the Transformer-based models we compare in this study; for simplicity, we show the encoder architectures here since the same modification is applied to their decoders.

stacked encoder/decoder layers. The encoder architecture is shown in Figure 1a.

Word embedding layers encode input words into continuous low-dimension vectors, followed by positional encoding layers that add position information to them. Encoder/decoder layers consist of a few sub-layers, self-attention layer, attention layer (decoder only) and feed-forward layer, with layer normalization (Ba et al., 2016) for each. Both self-attention layer and attention layer employ the same architecture, and we explain the details in § 3.3. Feed-forward layer consists of two linear transformations with a ReLU activation in between. As for the decoder, a linear transformation and a softmax function follow the stacked layers to calculate probabilities of words to output.

Figure 1 illustrates the architectures of all the Transformer-based models we compare in this study including our porposed model which will be introduced in § 4. The model in Shaw et al. (2018) modifies the self-attention layer (§ 3.3).

3.2 Word Position Information

Transformer has positional encoding layers which follow the word embedding layers and capture absolute position. The process of positional encoding layer is to add positional encodings (position vectors) to input word embeddings. The positional encodings are generated using sinusoids of varying frequencies, which is designed to allow the model to attend to relative positions from the periodicity of positional encodings (sinusoids). This is in contrast to the position embeddings (Gehring et al., 2017), a learned position vectors, which are not meant to attend to relative positions. Vaswani et al. (2017) report that both approaches produced nearly identical results in their experiments, and also mentioned that the model with positional encodings may handle longer inputs in testing than those in training, which implies that absolute position approach might have problems at this point.[1]

3.3 Self-attention with Relative Position

Some studies modify Transformer to consider relative position instead of absolute position. Shaw et al. (2018) propose an extension of self-attention mechanism which handles relative position inside in order to incorporate relative position into Transformer. We hereafter refer to their model as Rel-Transformer. In what follows, we explain the self-attention mechanism and their extension.

Self-attention is a special case of general attention mechanism, which uses three elements called query, key and value. The basic idea is to compute weighted sum of values where the weights are computed using the query and keys. Each weight represents how much attention is paid to the corresponding value. In the case of self-attention, the input set of vectors behaves as all of the three elements (query, key and value) using three different transformations. When taking a sentence as input, it is processed as a set in the self-attention.

Self-attention operation is to compute output sequence $z = (z_1, ..., z_n)$ out of input sequence $x = (x_1, ..., x_n)$, where both sequences have the same langth n and $x_i \in \mathbb{R}^{d_x}, z_i \in \mathbb{R}^{d_z}$. The output

[1] Our preliminary experiment confirmed that positional encodings perform better for longer sentences than those in the training data, while position embeddings perform slightly better for the other length.

element z_i is computed as follows.

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V) \quad (1)$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}} \quad (2)$$

$$e_{ij} = \frac{x_i W^Q (x_j W^K)^T}{\sqrt{d_z}}, \quad (3)$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_x \times d_z}$ are the matrices that transform input elements into querys, keys, and values, respectively.

The extension proposed by Shaw et al. (2018) adds only two terms to the original self-attention: the relative position vectors $w^K_{j-i}, w^V_{j-i} \in \mathbb{R}^{d_z}$.

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V + w^V_{j-i}) \quad (4)$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}} \quad (5)$$

$$e_{ij} = \frac{x_i W^Q (x_j W^K + w^K_{j-i})^T}{\sqrt{d_z}}, \quad (6)$$

Note that when using the relative position vectors, the input is processed as a directed graph instead of a set. Maximum distance k is employed to clip the relative distance within a certain distance so that the value of relative distance is limited as $-k < j - i < k$.

4 RNN as a Relative Positional Encoding

The approach by Shaw et al. (2018) is not the only way to incorporate relative position into Transformer. Lei et al. (2018) replace feed-forward layers by their proposed SRU which also incorporates relative position. Both approaches modify the encoder and decoder layers that are repeatedly stacked, which means their models handle position information multiple times. However, the original Transformer does only once at the positional encoding layer which locates shallow layer of the deep layered network.

To conduct a clear comparison of the position information types, we propose another simple method that replaces the positional encoding layer of Transformer by RNN. As the RNN has the nature to handle a sequence using relative position information, it can be used not only as a main processing unit of RNN-based model, but also as a relative positional encoder. While Lei et al. (2018) also employ RNN, they use position embeddings. Our approach is a pure replacement of position information type for Transformer.

In the original Transformer, the positional encoding layer adds the i-th position vector $pe(i) \in \mathbb{R}^{d_{wv}}$ to the i-th input word vector $wv_i \in \mathbb{R}^{d_{wv}}$ and outputs the position informed word vector $wv'_i \in \mathbb{R}^{d_{wv}}$:

$$wv'_i = wv_i + pe(i) \quad (7)$$

In our approach, we adopt RNN, specifically GRU (Cho et al., 2014) in this study, as a relative positional encoder. GRU computes its output or its i-th time hidden state $h_i \in \mathbb{R}^{d_{wv}}$ given the input word vector wv_i and the previous hidden state $h_{i-1} \in \mathbb{R}^{d_{wv}}$, and we take h_i as the position informed word vector wv'_i:

$$h_i = \mathrm{GRU}(wv_i, h_{i-1}) \quad (8)$$

$$wv'_i = h_i \quad (9)$$

Although LSTM (Hochreiter and Schmidhuber, 1997) is more often used as an RNN module in RNN-based models, we employed GRU which has less parameters. This is because, in our approach, RNN is just a positional encoder which we do not expect to work more, even though it can. We refer to our proposed model as RNN-Transformer.

We also consider the mixture of Shaw et al. (2018) and our method to investigate whether the two methods of considering relative position have additive improvements. Although both methods are intended to incorporate relative position into Transformer, they modify different parts of Transformer. By combining both, we can see either of modification suffices to incorporate relative position. We refer to this model as RR-Transformer.

5 Experiments

We conduct two experiments to evaluate our modification to Transformer and to investigate the impact of using relative position in NMT models. The first experiment is a basic translation experiment which uses all the training data. We carry out analysis on the translations generated by the NMT models in terms of sentence length, especially focusing on long sentences. In the second experiment, we control the training data by the sentence length so that the NMT models are trained only on sentences with lengths in a certain range. We also analyze the result in terms of sentence length, focusing on the short sentences.

	Train (orig.)	Dev	Test
ASPEC (En-Ja)	1,166,725 (3,008,500)	1790	1812
WMT2014 (En-De)	3,661,035 (4,468,840)	3000	2737

Table 1: Number of sentence pairs in the preprocessed corpus.

5.1 Setup

Dataset and Preprocess: We perform a series of experiments on English-to-Japanese and English-to-German translation tasks. For English-to-Japanese translation task, we exploit ASPEC (Nakazawa et al., 2016), a parallel corpus compiled from abstract sections of scientific papers. For English-to-German translation task, we exploit a dataset in WMT2014, which is one of the most common dataset for translation task.

For ASPEC English-to-Japanese data, we used scripts of Moses toolkit[2](ver. 2.2.1) (Koehn et al., 2007) for English tokenization and truecasing, and KyTea[3] (ver. 0.4.2) (Neubig et al., 2011) for Japanese segmentations. Following those word-level preprocess, we further applied SentencePiece (Kudo and Richardson, 2018) to segment texts down to subword level with shared vocabulary size of 16,000. Finally we selected the first 1,500,000 sentence pairs for the poor quality of the latter part, and filtered out sentence pairs with more than 49 subwords in either of the languages.

For WMT2014 English-to-German translation task, we used preprocessed data provided from the Stanford NLP Group,[4] and used newstest2013 and newstest2014 as development and test data, respectively. We also applied SentencePiece to this data to segment into subwords with shared vocabulary size of 40,000. We filtered out the sentence pairs in the same way as the ASPEC. Table 1 shows the number of sentence pairs of preprocessed data.

Figure 2 shows the distributions of the sentences plotted against the length of input sentence. Althought ASPEC data has slightly larger peak at sentence length of 20-29 subwords, both datasets have no big difference in length distributions. The training and test data have almost identical curves.

Model: We compare the following five NMT models:

[2] http://www.statmt.org/moses/

[3] http://www.phontron.com/kytea/

[4] https://nlp.stanford.edu/projects/nmt/

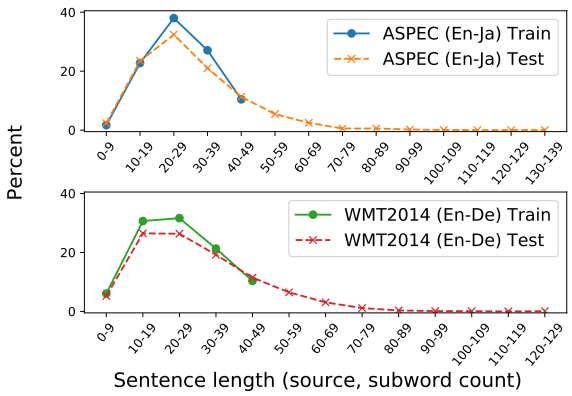


Figure 2: Sentence length ratio of preprocessed corpus.

RNN-NMT is a RNN-based NMT model with dot-attention and input-feeding (Luong et al., 2015). This model consists of four layered bi-directional LSTM for encoder and three layered uni-directional LSTM for decoder.

Transformer is a vanilla Transformer model (the base model in Vaswani et al. (2017)). This model consists of six-layered Transformer encoder and decoder.

Rel-Transformer is a modified version of Transformer by Shaw et al. (2018). Since the modifications do not increase the number of model parameter much, this model consists of the same number of encoder/decoder layers as **Transformer**, with the modified self-attention layer. We set the hyperparameter k, relative distance limit, to 16 following the base model in Shaw et al. (2018).

RNN-Transformer is another modified version of Transformer proposed in § 4. Because the replacement of the original positional encoding layer with RNN increases the number of model parameter, we employ uni-directional GRU as relative positional encoder for both encoder and decoder and reduced the number of decoder layer instead. Our RNN-Transformer model consists of six layered encoder and five layered decoder with one GRU layer for each.[5]

RR-Transformer is the mixture model of Rel-Transformer and RNN-transformer. With the same logic as Rel-Transformer, this model

[5] This configuration was chosen because it performed better than a model with five-layered encoder and six-layered decoder, and was comparable to five-layred encoder and decoder with bi-directional (instead of uni-directional) GRU for the relative position encoder in preliminary experiments.

	ASPEC (En-Ja)	WMT2014 (En-De)
RNN-NMT	70,521,476	107,409,476
Transformer	68,736,644	105,624,644
Rel-Transformer	68,787,332	105,675,332
RNN-Transformer	67,684,484	104,572,484
RR-Transformer	67,730,948	104,618,948

Table 2: Number of the model parameters.

consists of the same number of encoder and decoder layers as RNN-Transformer model, with the modified self-attention layer.

We implemented all the models using PyTorch[6] (ver. 0.4.1). Taking the base model of Transformer (Vaswani et al., 2017) which consists of six-layered encoder and decoder as a reference model, we built the other models to have almost the same number of model parameters for a fair comparison. For all models, we set word embedding dimension and model dimension (or hidden size for RNNs) to 512. For the Transformer-based models, we set feed-forward layer dimension to 2048, and the number of attention head to 8.

Table 2 shows the total number of model parameters for all the models in our implementation. The difference of the numbers by the datasets comes from the difference in vocabulary size.

Training: We used Adam optimizer (Kingma and Ba, 2015) with initial learning rate of 0.0001, and set dropout rate of 0.2 and gradient clipping value of 3.0. We adopted warm-up strategy (Vaswani et al., 2017) for fast convergence with warm-up step of 4k, and trained all the model for 300k steps. The mini-batch size was set to 128.

Evaluation: We performed greedy search for translation with the models, and evaluated the translation quality in terms of BLEU score (Papineni et al., 2002) using `multi-bleu.perl` in the Moses toolkit. We checked model's BLEU score on the development data at every 10k steps during the training, and took the best performing model for evaluation on the test data.

5.2 Long Sentence Translation

Table 3 shows the BLEU scores of the NMT models on the test data of ASPEC English-to-Japanese and WMT2014 English-to-German when using all the preprocessed training data for training. Table 4 lists the results of statistical significance test using bootstrapping of 10,000 samples. The evaluation is done on word-level, which means that we converted the outputs of NMT models from subword-level into word-level before scoring. On both datasets, Transformer outperforms RNN-NMT, and all of the three modified versions of Transformer outperform the Transformer. RNN-Transformer was comparable to Rel-Transformer, and RR-Transformer, the mixture of RNN-Transformer and Rel-Transformer, gives the best score.

[6] https://pytorch.org/

	ASPEC (En-Ja)	WMT2014 (En-De) newstest2014
RNN-NMT	36.67	19.95
Transformer	38.44	21.00
Rel-Transformer	39.58	22.51
RNN-Transformer	39.17	22.35
RR-Transformer	**40.34**	**23.01**

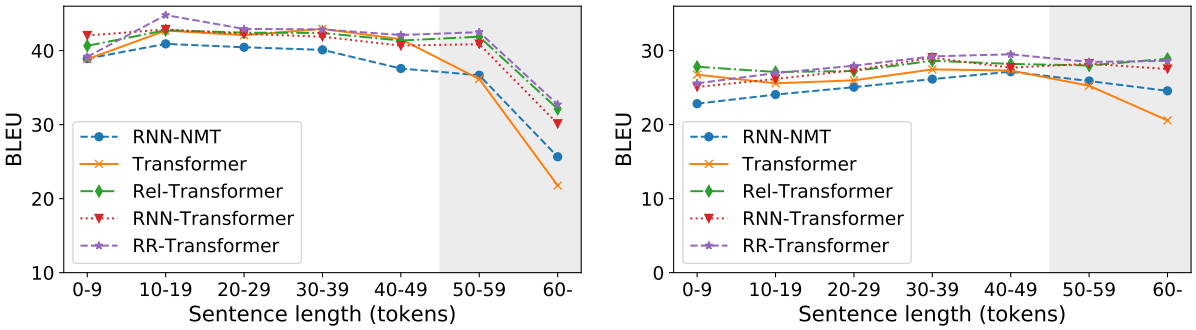
Table 3: BLEU scores on test data.

	RNN-NMT	Trans	Rel	RNN	RR
RNN-NMT		$<<$	$<<$	$<<$	$<<$
Trans	$>>$		$<<$	$<<$	$<<$
Rel	$>>$	$>>$		$\sim$	$<$
RNN	$>>$	$>>$	$\sim$		$<<$
RR	$>>$	$>>$	$>>$	$>>$	

Table 4: Results of statistical significance test on ASPEC English-to-Japanese (lower-left) and WMT2014 English-to-German (upper-right): "$>>$" or "$<<$" means $p < 0.01$, "$>$" or "$<$" means $p < 0.05$ and "$\sim$" means $p \geq 0.05$.

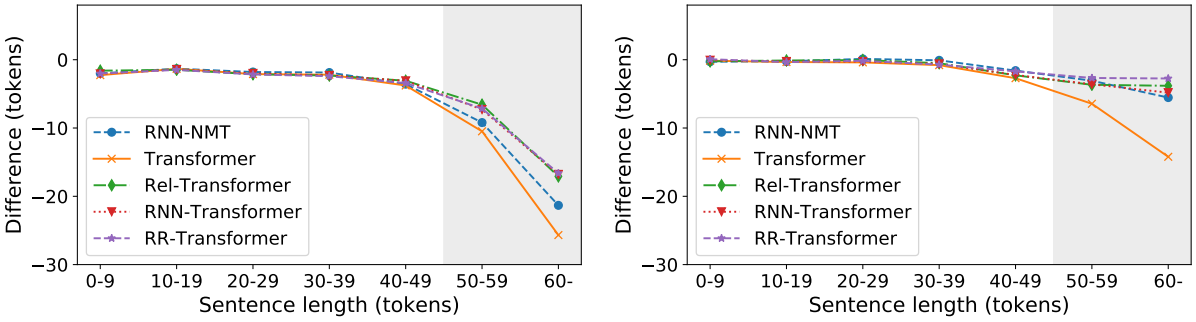
In order to see the capability of translating long sentences of the models, we split the test data into different bins according to the length of input sentences, and then calculated BLEU scores on each bin. The following evaluation uses the raw subword-level outputs of the models since the sentence length is based on subwords.

Figure 3a and 3b show the BLEU scores on the split test data of ASPEC English-to-Japanese and WMT2014 English-to-German, respectively. The BLEU score of Transformer, the only model that uses absolute position, more sharply drops than the BLEU scores of the other models at the input length of 50-59, which is outside of the length range of the training data. As for the input length of 60-, Transformer performs the worst among all the models. These results indicate that relative position works better than absolute position in translating sentences longer than those of the training data. Meanwhile, for the lengths with enough



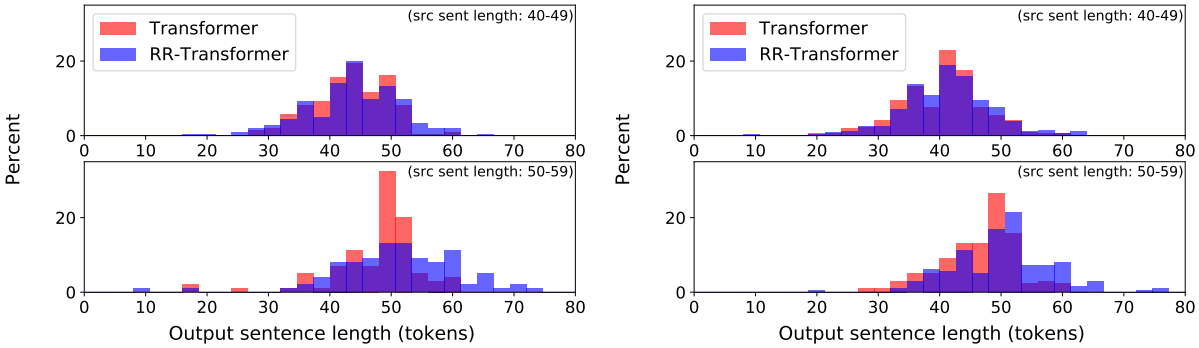
(a) ASPEC English-to-Japanese (b) WMT2014 English-to-German

Figure 3: BLEU scores on test data split by the sentence length (no training data in the gray-colored area).



(a) ASPEC English-to-Japanese (b) WMT2014 English-to-German

Figure 4: Averaged difference of sentence length between NMT model's output and the reference translation (no training data in the gray-colored area).



(a) ASPEC English-to-Japanese (b) WMT2014 English-to-German

Figure 5: Distributions of output sentence length of Transformer and RR-Transformer.

amount of training data, both position information types seem to work almost equally. On WMT2014 English-to-German, all the models except Transformer successfully keep as good performance in 50-59 and 60- bins as the other bins.

To figure out the effect of position information on the ability of the models to generate output of proper length, we look into the difference of sentence length between the model's output and the reference translation. Figure 4a and 4b show the averaged differences plotted against the input sentence length on both language pairs. We can observe that all the models tend to output shorter sentence than the reference. However, Transformer shows the largest drop at the input length of 50-59 again among all the models, which is even more than RNN-NMT. The difference between Transformer and RNN-Transformer indicates the advantage of relative position against absolute position, while the difference between the three modified Transformer-based models and RNN-NMT indicates the structural advantage of Transformer to RNN-based model in generating translations with appropriate lengths.

	min len.	max len.	# of sentences	# of tokens
Short	2	26	555,922	10,392,775
Middle	26	34	350,176	10,392,797
Long	34	49	260,626	10,392,729

(a) ASPEC English-to-Japanese

	min len.	max len.	# of sentences	# of tokens
Short	1	24	1,878,354	29,841,533
Middle	24	34	1,041,794	29,841,531
Long	34	49	740,887	29,841,519

(b) WMT2014 English-to-German

Table 5: Statistics of the split training data.

The above result that the models tend to output shorter sentences suggests that the models may have a limit in the range of output length. To confirm this possibility, we look into the distributions of the model's output length. Figure 5a and 5b show distributions of output length of Transformer and RR-Transformer for the input length of 40-49 (length within the training data) and 50-59 (length outside of the training data). For the input length of 40-49, the distributions of both models are flat and have no big difference. For the input length of 50-59, on the other hand, we can see a sharp peak in the distribution of Transformer in which most of the values distribute around 50 tokens or less. These results indicate that Transformer tends to overfit to a range of length of input sentences.

5.3 Length-Controlled Training Data

The above experiments focus on trainslation of long sentences, or, strictly speaking, sentences longer than those in the training data. With the use of absolute position, it is no surprise that the model fails to handle longer sentences since those sentences demand the model to handle the position vectors which are never seen during training.

In this section, we focus on short sentences to investigate whether Transformer overfits to the length of input sentences in the training data. Note that position vectors of small numbers are included in long sentences. If the problem is only unseen position vectors, then the model shall be able to handle short sentences because short sentences do not include any unseen position numbers.

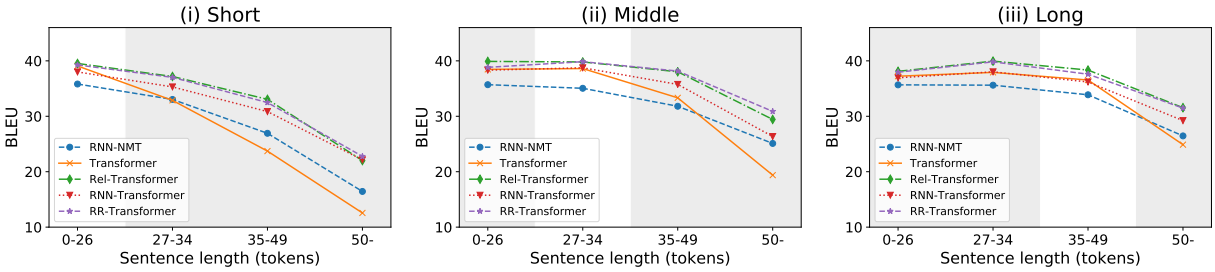
To figure out how the NMT models behave on sentences shorter than those in the training data, we conduct another experiment in which the length of the training data is controlled. We split the training data of both ASPEC English-to-Japanese and WMT2014 English-to-German into three portions according to the length of input sentences so that each of them has almost the same number of tokens. We then trained the five NMT models on each of the three training data. We hereafter refer to these three length-controlled training data as Short, Middle and Long. The statistics of these data is summarized in Table 5a and 5b.

To see how the translation quality changes between inside and outside of the length within the training data, we split the test data with respect to the lengths of split training data. Figure 6a and 6b show the BLEU scores on all the three training data of both language pairs. Transformer shows the worst performance among the four Transformer-based models on the sentences longer than those in the training data for any controlled length. However, on the shorter sentences than those in the training data, RNN-Transformer scores almost the same as Transformer on the Middle and Long training data of ASPEC English-to-Japanese and also shows a larger drop than RNN-NMT at length of -24 on the Long training data of WMT2014 English-to-German. This implies that our proposed method to replace absolute positional encoding layer by RNN does not work well in translating shorter sentences.

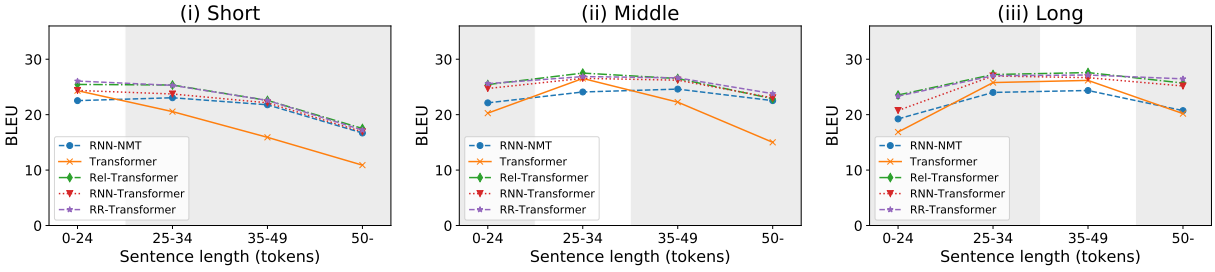
We can also see that Rel-Transformer and RR-Transformer are quite competitive across all the situations. This suggests that one Transformer decoder layer and two GRUs contribute almost equally to the translation quality.

Figure 7a and 7b show the averaged difference of length between NMT model's output and the reference translation on Long training data of both datasets.[7] These figures indicate that Transformer and RNN-Transformer tend to generate inappropriately long sentences in translating much shorter sentences than those in the training data. As mentioned above, when translating short sentences, there is no unseen positions in Transformer, while there is no concrete position representation in RNN-Transformer; the above results suggest that these two models overfit to the (longer) length of input sentences. In contrast, the result of Rel-

[7] Note that Figure 7a and 7b use different x-axis scale from Figure 6a and 6b in order to show the difference clearly.

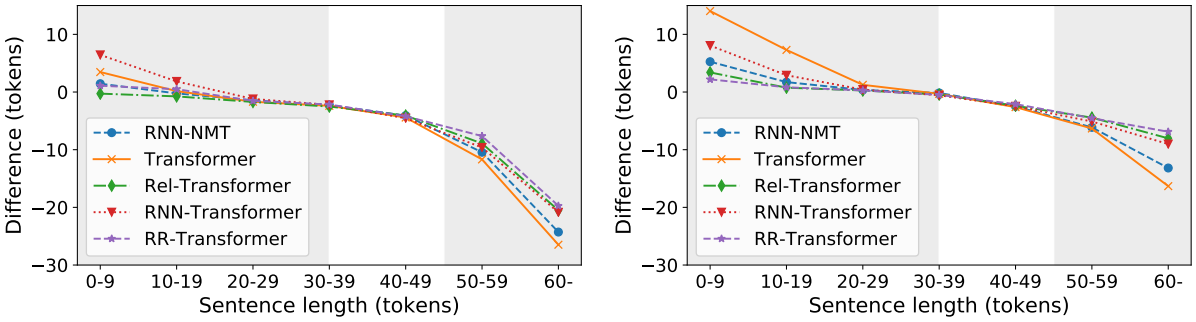


(a) ASPEC English-to-Japanese



(b) WMT2014 English-to-German

Figure 6: BLEU scores of models trained on three length-controlled training data on test data split in the same way as the training data (almost no training data in the gray-colored area).



(a) ASPEC English-to-Japanese

(b) WMT2014 English-to-German

Figure 7: Averaged difference of sentence length between NMT model's output translation and reference translation (almost no training data in the gray-colored area).

Transformer and RR-Transformer indicates that self-attention with relative position prevents this overfitting.

6 Conclusions

In this paper, we examined the relation between position information and the length of input sentences by comparing absolute position and relative position using RNN-based model and variations of Transformer models. Experiments on all the preprocessed training data revealed the crucial weakness of the original Transformer, which uses absolute position, in translating sentences longer than those of the training data. We also confirmed that incorporating relative position into Transformer helps to handle those long sentences and improves the translation quality. Another experiment on the length-controlled training data revealed that absolute position of Transformer causes overfitting to the input sentence length. To conclude, all the experiments suggest to use relative position and not to use absolute position.

Considering that the available data is not balanced in terms of the sentence length in practice, preventing the overfitting is useful for building a practical NMT system.

Acknowledgments

We deeply thank Satoshi Tohda for proofreading the draft of our paper. This work was partially supported by JST CREST Grant Number JPMJCR19A4, Japan. This research was also partially supported by NII CRIS Contract Research 2019.

References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the third International Conference on Learning Representations (ICLR)*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 76–86.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 93–98.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2018. Pervasive attention: 2D convolutional neural networks for sequence-to-sequence prediction. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, pages 97–107.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1243–1252.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *CoRR*, abs/1610.10099.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the third International Conference on Learning Representations (ICLR)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej 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 (ACL): Demo and Poster Sessions*, pages 177–180.

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.

Taku. Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 66–71.

Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4470–4481.

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 (EMNLP)*, pages 1412–1421.

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 tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2204–2208.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Guçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 280–290.

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 (ACL-HLT), Short Papers*, pages 529–533.

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 of the Association for Computational Linguistics (ACL)*, pages 311–318.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 379–389.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1577–1586.

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 (NAACL-HLT), Short Papers*, pages 464–468.

Xing Shi, Kevin Knight, and Deniz Yuret. 2016. Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2278–2282.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NIPS) 28*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS) 27*, pages 3104–3112.

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 *Advances in Neural Information Processing Systems (NIPS) 30*, pages 5998–6008.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Proceedings of Deep Learning Workshop held at the 31st International Conference on Machine Learning (ICML)*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4334–4343.