Understanding How Positional Encodings Work in Transformer Model

Taro Miyazaki, Hideya Mino, Hiroyuki Kaneko
NHK Science and Technology Research Laboratories
1-10-11, Kinuta, Setagaya-ku, Tokyo, Japan
{miyazaki.t-jw, mino.h-gq, kaneko.h-dk}@nhk.or.jp

Abstract

A transformer model is used in general tasks such as pre-trained language models and specific tasks including machine translation. Such a model mainly relies on positional encodings (PEs) to handle the sequential order of input vectors. There are variations of PEs, such as absolute and relative, and several studies have reported on the superiority of relative PEs. In this paper, we focus on analyzing in which part of a transformer model PEs work and the different characteristics between absolute and relative PEs through a series of experiments. Experimental results indicate that PEs work in both self- and cross-attention blocks in a transformer model, and PEs should be added only to the query and key of an attention mechanism, not to the value. We also found that applying two PEs in combination, a relative PE in the self-attention block and an absolute PE in the cross-attention block, can improve translation quality.

Keywords: Positional Encoding, Transformer, Machine Translation

1. Introduction

A transformer model (Vaswani et al., 2017) is a neural network model that can handle sequential information. Sequential information is traditionally handled using a recurrent or convolutional neural network, which has a structure that handles sequential information in order. A transformer model does not have a recurrent or convolutional structure. It handles sequential information by using positional encodings (PEs) that directly inject positional information of the sequence in the input vectors.

There are variations of PEs such as absolute PEs (Vaswani et al., 2017) and relative PEs (Shaw et al., 2018). There have been several studies on comparing these two PEs (Rosendahl et al., 2019), reporting the superiority of relative PEs, especially in longer sentences.

In this paper, we focused on analyzing in which part of a transformer model PEs work. We also investigated the different characteristics between absolute and relative PEs through a series of experiments.

Our contributions are as follows: (1) PEs mainly work in a self-attention block as well as in the cross-attention block, (2) PEs should be added only to the query and key of an attention mechanism, not to the value, (3) using absolute and relative PEs in combination can improve translation quality, (4) machine translation methods using a transformer model do not have very high generalization ability with respect to sentence length, and their performance degrades if the sentence lengths match between training and testing data, and (5) machine translation quality may rely on the number of tokens in the training data rather than the number of sentences.

2. Transformer Model and Positional Encoding

2.1. Transformer Model

Typical transformer-based encoder-decoder models consist of two structures, an encoder and decoder. The encoder consists of two blocks, self-attention and feed-forward, and the decoder consists of three blocks, self-attention, cross-attention, and feed-forward. The main component of the self-attention and cross-attention blocks is multi-head attention, which is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{D}}\Big)V, \quad (1)$$

where Q, K and V respectively represent the query, key, and value of an input, and D is dimension. Note that we dropped head indices for clarity.

2.2. Positional Encoding

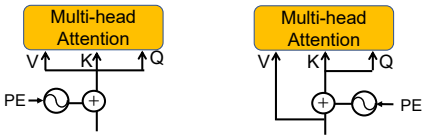
A transformer model injects positional information of tokens in the sequence to handle sequential information. The model uses PEs to incorporate the order of sequences in the self-attention block of the transformer. As variations of PEs (Dufter et al., 2022), we used the widely used absolute and relative PEs to determine their performances.

Absolute Positional Encoding An absolute PE injects absolute positional information into input vectors using fixed or learned positional embeddings.

Fixed positional embeddings are calculated using sinusoidal functions (Vaswani et al., 2017).

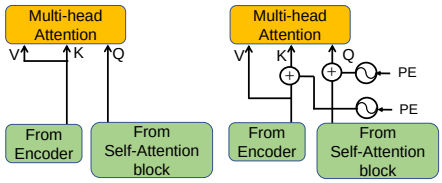
Learned positional embeddings, however, are calculated using the embedding layer of neural net-

LREC-COLING 2024, pages 17011–17018
20-25 May, 2024.



(a) Applying PE for *Q*, *K*, *V* (ORIGINAL) (b) Applying PE only for *Q* and *K* (LESS-FEEDING)

Figure 1: Two types of self attention block.



(a) No PE applied (NONE). (b) Applying PE (CROSS-PE).

Figure 2: Two types of cross attention block.

Language-pair	Train	Devel	Test
fr → de	17,975,604	1,026	2,006
de → fr		1,000	1,984
jp → en	33,875,119	1,005	2,008
en → jp		1,000	2,037

Table 1: Number of sentences in datasets.

works. This is typically done by embedding absolute position IDs $P = \{0, 1, \cdots, n\}$, where n is the number of tokens in the sentence into the embedding network (Devlin et al., 2019). The position embedding vectors are then added to the Q, K and V of an input of a self-attention.
Typical transformer-based machine translation methods utilize fixed position embeddings, so we utilize fixed position embeddings in this study.

Relative Positional Encoding A relative PE injects the distance between elements of input sequences instead of its position as:

$$\text{Attention}_{\text{rel}} = \text{Softmax}\Big(\frac{QK^T + S}{\sqrt{D}}\Big)V, \quad (2)$$

where S is relative positional embedding that embeds the distance between input elements (Huang et al., 2019).

3. Experiments

We conducted a series of experiments to understand the behaviors and characteristics of PEs. We first clarified in which blocks of the transformer model PEs work. We then conducted experiments on machine translation involving two language-pairs datasets, i.e., French(fr)-German(de) and Japanese(ja)-English(en) to determine whether our findings can be generally adopted.

3.1. Experimental settings

We used The Conference on Machine Translation (WMT22) (2022) General MT (News) task dataset as the training and test data, and the test data of The Conference on Machine Translation (WMT21) (2021) News task dataset as the development data[1]. We chose a similar language pair (fr-de) and non-similar language pair (ja-en) to determine performance in as wide a range of situations as possible. Note that we regard that language pairs

[1]This is because WMT22 dataset does not include development data.

Self-attention	Cross-attention		BLEU	COMET
ORIGINAL	NONE	Median	29.05	-8.07
(Original transformer)		Average	29.13 ± 0.16	-7.52 ± 1.12
ORIGINAL	CROSS-PE	Median	29.02	-8.87
		Average	29.03 ± 0.15	8.91 ± 0.46
LESS-FEEDING	NONE	Median	36.59	34.14
		Average	36.65 ± 0.29	34.49 ± 0.48
LESS-FEEDING	CROSS-PE	Median	**38.11**	**38.78**
		Average	**38.32** ± 0.27	**39.00** ± 0.88
NONE	NONE	Median	10.59	-73.42
		Average	10.57 ± 0.18	-73.40 ± 0.74
NONE	CROSS-PE	Median	10.43	-73.57
		Average	10.41 ± 0.10	-74.09 ± 1.17

Table 2: Results of experiments for analyzing which blocks of transformer require PE using fr → de task. We show median, average, and standard deviation of three experiments. **Bold** represents best results. Note that we utilized 1-layer transformer models for this experiment.

with significantly different word orders, such as SVO and SOV, as "non-similar languages." The volumes of each dataset are listed in Table 1. The translation models were implemented in PyTorch (Paszke et al., 2019) and learned with the RAdam optimizer (Liu et al., 2020) with a learning rate of 5.0×10^{-4} with a cosine scheduler with warmup steps of 5,000 and mini batch size of 256. We used beam search with a beam width of 10 in the decoding process, and COMET[2] (Rei et al., 2020) and SacreBLEU[3] (Post, 2018) as evaluation metrics. We utilized sentencepiece (Kudo and Richardson, 2018) as the tokenizer with a vocabulary size of 16,000. We trained the models with 10 epochs for the fr-de pair and 5 epochs for the ja-en pair (almost equal to 650K steps for both language pairs), and evaluated every 50K steps using development data. We chose the best models on the basis of the COMET score on the development data and used those models to evaluate on the test data. We show the average, standard deviation, and median of three experiments with the same settings but using dif-

[2]We used wmt20-comet-da model.

[3]The signatures are nrefs:1|case:mixed|eff:no|tok:ja-mecab-0.996-IPA|smooth:exp|version:2.3.1 for en → jp and refs:1|case:mixed|eff:no|tok:13a| smooth:exp|version:2.3.1 for other tasks.

PE	Less-Feeding	Cross-PE		fr→de BLEU	fr→de COMET	de→fr BLEU	de→fr COMET	ja→en BLEU	ja→en COMET	en→ja BLEU	en→ja COMET
Absolute	(Original transformer)		Median	37.15	36.51	26.36	16.24	18.82	15.39	20.09	37.00
			Average	37.15 ± 0.04	37.34 ± 1.58	26.32 ± 0.13	16.39 ± 0.69	18.87 ± 0.49	15.10 ± 0.51	20.06 ± 0.18	36.46 ± 1.13
	✓		Median	38.82	39.37	**26.56**	19.11	18.35	14.54	20.31	**38.43**
			Average	38.82 ± 0.12	39.81 ± 0.96	**26.57** ± 0.21	18.93 ± 0.97	18.16 ± 0.69	14.73 ± 0.45	20.29 ± 0.04	38.18 ± 0.69
	✓	✓	Median	38.86	41.13	26.54	18.92	18.13	15.86	**20.46**	38.27
			Average	38.85 ± 0.26	41.06 ± 0.45	26.55 ± 0.42	**19.38** ± 1.25	18.32 ± 0.34	15.19 ± 1.19	**20.50** ± 0.17	**38.42** ± 0.34
Relative	✓		Median	38.79	40.29	26.41	**19.67**	17.80	14.91	20.38	37.66
			Average	**38.92** ± 0.33	40.37 ± 0.65	26.56 ± 0.27	19.28 ± 0.86	17.97 ± 1.30	15.04 ± 1.67	20.29 ± 0.27	37.50 ± 0.62
	✓	✓	Median	**38.89**	**41.22**	26.52	19.10	**19.18**	**17.66**	20.35	37.76
			Average	38.86 ± 0.05	**41.47** ± 1.29	26.55 ± 0.13	19.05 ± 0.38	**19.05** ± 0.25	**17.22** ± 1.08	20.33 ± 0.28	37.98 ± 0.62

Table 3: Experimental results of machine translation. **Bold** represents best results and underline represents those models outperforming original transformer.

ferent random seeds for each experiment.

3.2. In Which Blocks Do PEs work?

We conducted experiments to clarify which blocks of the transformer model require information of PEs, we prepared two types of self-attention blocks. One, denoted as Original, is applying a PE to Q, K, and V (Figure 1-(a)), which has the same structure as that in the original transformer model. The other, denoted as Less-Feeding, is applying PE only to Q and K (Figure 1-(b)) to reduce PE-derived information to feed to the next block (Press et al., 2022). For comparison, we also prepared a model that PE is not applied for self-attention, denoted as None. We also prepared two types of cross-attention block. One is with no PE applied (Figure 2-(a)), denoted as None, which has the same structure as that of the original transformer model, and the other, denoted as Cross-PE, is applying PE for cross attention (Figure 2-(b)) (Li et al., 2023). We utilized 1-layer encoder-decoder models for this experiment with absolute PEs to eliminate effects other than the difference in how PEs applied as much as possible. We utilized the fr-de translation dataset.

The results are given in Table 2. The best result was using Less-Feeding and Cross-PE. The results show that PEs mainly work in the self-attention block but also work in the cross-attention block.

3.3. Comparing Absolute and Relative PEs

We also conducted experiments on machine translation tasks with the two language pairs to confirm whether the findings can be adopted generally. We prepared three models, the original transformer model, one applying Less-Feeding, and one applying Less-Feeding and Cross-PE. We used a 6-layer encoder-decoder model and applied PEs for the first layer of the models. We also prepared absolute and relative PEs for the self-attention blocks to compare translation quality. We used a relative PE with skew (Huang et al., 2019). This relative PE does not add position embedding vectors to V, which is the same as Less-Feeding, so we did not use Original with the relative PE. Note that we used an absolute PE for the cross-attention block when applying Cross-PE regardless of the type of PEs applied in the self-attention blocks. The experimental results are shown in Table 3.

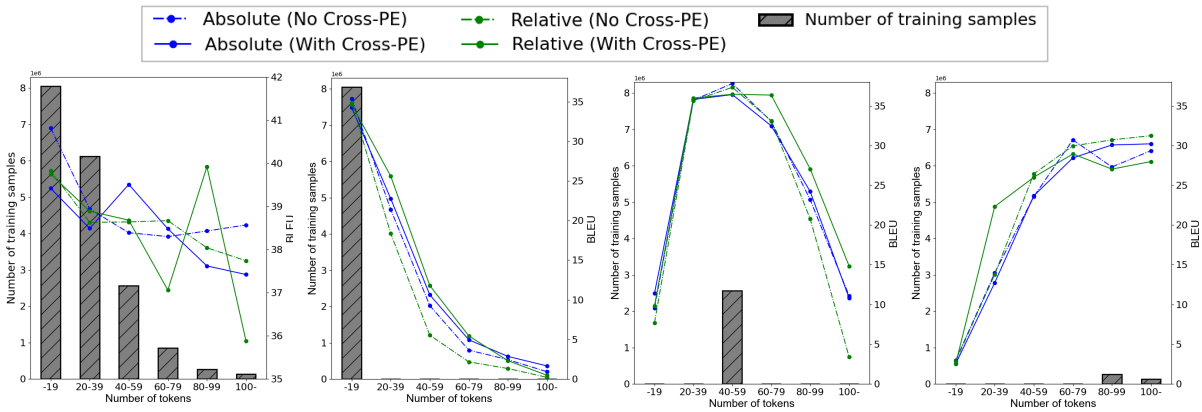
Less-Feeding with an absolute PE outperformed Original in almost all tasks except ja→en, which shows the effectiveness of Less-Feeding. Applying Cross-PE also improved translation quality especially in term of COMET scores. Therefore, we confirmed that our findings presented in Section 3.2 can be generally adopted.

Cross-PE with a relative PE outperformed Original in all tasks and using a relative PE without Cross-PE in most cases, so we confirmed the effectiveness of using Cross-PE with a relative PE.

3.4. Discussions

How PEs Work in Transformer Model As shown in Table 2, PEs work in the cross-attention block as well as work in a self-attention block, so using Cross-PE in the cross-attention block improved translation quality for both tasks. As Garg et al. (2019) showed, word alignment is calculated in the transformer model, and it is thought that the word alignment is calculated mostly in the cross-attention block. We believe that Cross-PE is useful for improving the word alignment calculation.

When comparing the two types of self-attention block, the translation qualities were better using Less-Feeding than using Original. If adding PEs to V as Original, the output of self-attention block contains too much PEs-derived information, and the information is a combination of PEs from various input words. This is difficult to handle and translation quality worsens. Surprisingly, using Less-Feeding with absolute PEs can achieve almost the same results as using Less-Feeding with relative PEs. We presume that one of the reasons of the reportedly superiority of relative PEs is that relative PEs basically do not add position embedding vectors to the V of the an attention mechanism.



(a) All (b) tokens < 20 (c) 40 < tokens < 60 (d) 80 < tokens

Figure 3: Comparison of translation qualities by sentence length in training data using fr → de task. Note that scale of vertical axis for BLEU score differs in (a) and others.

Data	# Sentences	# Tokens	BLEU (Absolute PE)	BLEU (Relative PE)
tokens < 10 (Figure 3-(b))	8,052,523	93,327,696	34.27	34.66
40 < tokens < 60 (Figure 3-(c))	2,560,810	122,431,120	36.42	36.46
80 < tokens (Figure 3-(d))	394,505	44,813,104	29.44	25.97

Table 4: Number of sentences and tokens in training data used in Figure 3-(b),(c),(d). We show BLEU scores using absolute and relative PEs with CROSS-PE for sentences with same number range of tokens as training data.

Differences in Generalizing Ability regarding Sentence Lengths The inability of a transformer model to generalize to data of unseen lengths was reported by Varis and Bojar (2021). To clarify the differences among the three transformer models we used, we categorized the BLEU scores on the basis of source sentence length. The results are shown in Figure 3-(a). There are different characteristics among the models. We believe this difference is due to the generalizing ability for sentence lengths among models. To determine the differences in generalizing ability, we conducted experiments in which the training data were selected on the basis of sentence length. Specifically, we extracted three sets of training data where the number of source language tokens was under 20, between 40 and 59, and over 80, and compared the performance.

The results are shown in Figure 3-(b),(c),(d). Using absolute PEs when CROSS-PE was not applied performed better for unseen sentence lengths than when using relative PEs. Using relative PEs when CROSS-PE was applied, the BLEU scores for unseen sentence lengths improved and outperformed using absolute PEs when applying CROSS-PE. We believe that this is because applying two different PEs, relative PEs in a self-attention block and absolute PEs in the cross-attention block increases the amount of information handled in the translation model, improving performance.

Interestingly, the translation quality for sentences shorter than training data was poor for all settings, which indicates that machine translation methods using the transformer model fail to generalize data with different sentence lengths. Translation performances degrade if the sentence lengths do not match between training and test data.

Effect of Number of Sentences and Tokens in Training Data Even though there was only around 400,000 sentence pairs of the selected training data of $80 <$ tokens, the BLEU score for sentences with the same length as the training data was high around 30 (Figure 3-(d)). There were small differences in BLEU scores with the training data of tokens < 20, which had around 8,000,000 sentence pairs. Table 4 shows the number of sentences and tokens in the source language of the training data. It is suggested that it is the number of tokens as well as the number of sentences that affects to the BLEU score.

4. Related Work

PEs are commonly utilized to incorporate the order of sequences. Vaswani et al. (2017) proposed a pre-defined PE with the original transformer model, which uses sinusoidal functions of different frequencies to add to the input vectors. Also, there are absolute PEs with trainable parameters have been widely used (Devlin et al., 2019; Radford et al.; Liu et al., 2019).

Shaw et al. (2018) proposed a relative PE, and demonstrated its effectiveness regarding machine translation quality. There are variants of relative PEs such as TUPE (Transformer with Untied Positional Encoding) (Ke et al., 2021), RoPE (Rotary Po-

sitional Embedding) (su2, 2024), and other (Press et al., 2022; Chowdhery et al., 2023; Sun et al., 2023). Most are used to handle longer sentences as well as improve performance.

Li et al. (2023) proposed a model that does not feed PEs to the value of self-attention in a transformer model, and Press et al. (2022) proposed a model that utilize PEs in cross-attention. On the basis of these models, we conducted further detailed analysis to analyze the behavior of PEs. We not only improved overall performance by changing the manner in which a PE is given but also found differences in behavior in generalizing differences in sentence length between training and evaluation data. It has been reported that the performance of a transformer model deteriorates due to differences in sentence length (Varis and Bojar, 2021). As one solution to this problem, we argued that using different types of the PEs in one model may lead to improved generalizability for sentence lengths.

5. Conclusion

In this paper, we analyzed in which part of the transformer model positional encodings (PEs) work. We also investigated the different characteristics between absolute and relative PEs through a series of experiments.

Experimental results indicate that PEs work in both self- and cross-attention blocks in a transformer model, and PEs should be added only to the query and key of an attention mechanism, not to the value. We also found that applying two PEs in combination, applying a relative PE in the self-attention block and an absolute PE in the cross-attention block, can improve translation quality.

Machine translation methods using a transformer model do not have very high generalization ability with respect to sentence length, and the number of tokens rather than the number of sentences affects translation quality.

We compared the performance using basic PEs, but it is necessary to investigate whether the same conclusion can be made with more recent PEs. We also need to investigate the performance difference due to differences in syntax, such as Det-ee–Det-er in French and Det-er–Det-ee in German. These are left as future work.

Acknowledgement

The authors are grateful to the anonymous reviewers who provided important comments to improve this paper.

6. Bibliographical References

2024. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. Journal of Machine Learning Research, 24(240):1–113.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position information in transformers: An overview. Computational Linguistics, 48(3):733–763.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music transformer. In International Conference on Learning Representations.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. Improve transformer models with better relative position embeddings. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3327–3335, Online. Association for Computational Linguistics.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking positional encoding in language pre-training. In International Conference on Learning Representations.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yachao Li, Junhui Li, Jing Jiang, Shimin Tao, Hao Yang, and Min Zhang. 2023. P-transformer: Towards better document-to-document neural machine translation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31:3859–3870.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In International Conference on Learning Representations.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2685–2702, Online. Association for Computational Linguistics.

Jan Rosendahl, Viet Anh Khoa Tran, Weiyue Wang, and Hermann Ney. 2019. Analysis of positional encodings for neural machine translation. In Proceedings of the 16th International Conference on Spoken Language Translation, Hong Kong. Association for Computational Linguistics.

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), pages 464–468, New Orleans,

Louisiana. Association for Computational Linguistics.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. 2023. A length-extrapolatable transformer. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14590–14604, Toronto, Canada. Association for Computational Linguistics.

Dusan Varis and Ondřej Bojar. 2021. Sequence length is a domain: Length-based overfitting in transformer models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8246–8257, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 30.

7. Language Resource References

The Conference on Machine Translation (WMT21). 2021. WMT21 News MT dataset. The Conference on Machine Translation.

The Conference on Machine Translation (WMT22). 2022. WMT22 General MT (News) dataset. The Conference on Machine Translation.

A. Comparison between Where PEs Are Applied in Attention Mechanism.

To analyze where a PE is used in an attention mechanism, we conducted another experiment by changing the conditions of applying a PE to Q, K, and V. We utilized training data that included 10% of the data randomly extracted from the training data of the fr-de task. We utilized a 1-layer transformer model. This is because in such models with multiple layers, information of PEs is fed from the previous layer, making it difficult to limit PE information. The PEs were applied under the same conditions for all three attention mechanisms in the encoder and decoder. We utilized absolute PE.

The experimental results are given in Table 5. It is clear that it is better to give PE information to Q and K. Although there was a slight difference, better performance was obtained by applying to Q and K without applying to V.

Even when a PE was applied only to V, the performance was better than when no PE was used. PE information given to V cannot be used in the calculation of attention weight, but is added to the output of an attention mechanism. In other words, this setting is almost equivalent to applying PEs only to the cross-attention block of the transformer decoder. This also suggests that applying PEs to the cross-attention block leads to improved performance.

Q	K	V		BLEU	COMET
			Median	7.51	-103.40
			Average	7.57 ± 0.18	-102.84 ± 0.95
✓			Median	12.61	-85.41
			Average	12.61 ± 0.03	-85.37 ± 0.08
	✓		Median	12.84	-78.46
			Average	12.57 ± 0.54	-78.27 ± 0.78
		✓	Median	12.03	-85.99
			Average	12.29 ± 0.22	-85.97 ± 0.36
✓	✓		Median	**24.05**	**-36.88**
			Average	**24.12** ± 0.13	**-37.05** ± 1.04
	✓	✓	Median	13.31	-78.37
			Average	13.20 ± 0.22	-78.73 ± 1.87
✓		✓	Median	12.79	-86.81
			Average	12.76 ± 0.07	-86.65 ± 0.33
✓	✓	✓	Median	23.86	-38.77
			Average	23.81 ± 0.23	-38.47 ± 0.67

Table 5: Comparison between where PEs are applied in attention mechanism.

B. In which layer do PEs work?

Some methods use PEs for only the first layer (Vaswani et al., 2017; Devlin et al., 2019) and others for every layer (Huang et al., 2020; Raffel et al., 2020). We also conducted experiments to determine in which layer PEs work using the fr-de task. We prepared models that apply PEs in the first n layer and compared their performances. We utilized training data that included 10% of the data randomly extracted from the training data of the fr-de task. The results are given in Table 6. In almost all models except for the original transformer model, their performance was best when the PE added to the first two to five layers, and was almost the same among them. In contrast, the best performance was obtained by applying PEs for all layers for the original transformer model.

The word order information is calculated not only in the first layer of the model but also in the subsequent layers. Therefore it is better to add PEs to not only the first layer but also subsequent layers.

		Absolute PE						Relative PE			
Less-Feeding		(Original transformer)		✓		✓		✓		✓	
Cross-PE						✓				✓	
No. of Layer		BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
1	Median	31.16	12.01	31.36	13.41	31.32	12.38	31.05	8.84	30.33	8.83
	Average	31.14 ± 0.15	12.46 ± 1.22	31.34 ± 0.31	13.40 ± 0.85	31.33 ± 0.17	12.56 ± 0.51	30.94 ± 0.29	9.15 ±1.12	30.38 ± 0.19	8.95 ± 1.06
2	Median	31.38	12.29	31.36	13.67	31.69	13.34	31.64	13.20	31.13	11.58
	Average	31.44 ± 0.13	12.21 ± 0.44	31.31 ± 0.18	13.59 ± 0.59	**31.75** ± 0.20	13.88 ± 0.99	31.63 ± 0.04	13.20 ± 0.89	31.08 ± 0.18	11.61 ± 0.92
3	Median	31.50	13.84	**31.84**	15.51	31.41	13.63	**32.02**	**15.01**	31.58	13.59
	Average	31.46 ± 0.15	13.70 ± 1.02	**31.79** ± 0.18	**15.38** ± 0.57	31.50 ± 0.29	**13.99** ± 1.23	**32.02** ± 0.13	**14.92** ± 0.46	31.53 ± 0.20	13.26 ± 0.69
4	Median	31.45	13.02	31.58	**15.69**	**31.74**	**13.74**	31.89	13.98	31.72	**15.36**
	Average	31.49 ± 0.08	13.18 ± 0.99	31.77 ± 0.38	15.10 ± 1.16	**31.75** ±0.07	13.94 ± 0.35	31.86 ± 0.15	14.33 ± 0.66	31.73 ± 0.27	**15.12** ± 0.56
5	Median	31.49	11.60	31.57	13.16	31.50	13.41	31.80	14.76	**31.88**	12.69
	Average	31.45 ± 0.23	12.04 ± 1.28	31.52 ± 0.12	13.53 ± 1.06	31.55 ± 0.16	13.69 ± 0.72	31.85 ± 0.09	14.01 ± 0.11	**31.79** ±0.46	12.84 ± 1.81
6	Median	**31.61**	**14.53**	31.58	14.30	31.35	12.50	31.36	12.81	31.74	12.30
	Average	**31.62** ± 0.04	**14.21** ± 0.01	31.55 ± 0.17	14.25 ± 1.02	31.41 ± 0.12	12.67 ± 0.62	31.32 ± 0.18	12.39 ± 1.31	31.71 ± 0.16	12.29 ± 0.03

Table 6: Results of experiments for analyzing which layers of transformer require positional encoding. **Bold** means best results in same setting. We used fr $\rightarrow$ de task.