Explore Better Relative Position Embeddings from Encoding Perspective for Transformer Models

Anlin Qu[1], **Jianwei Niu**[1]*, **Shasha Mo**[2]*
[1] State Key Laboratory of Virtual Reality Technology and Systems,
School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2] School of Cyber Science and Technology, Beihang University, Beijing 100191, China
{anlin781205936, niujianwei, moshasha}@buaa.edu.cn

Abstract

Relative position embedding (RPE) is a successful method to explicitly and efficaciously encode position information into Transformer models. In this paper, we investigate the potential problems in Shaw-RPE and XL-RPE, which are the most representative and prevalent RPEs, and propose two novel RPEs called **L**ow-level **F**ine-grained **H**igh-level **C**oarse-grained (LFHC) RPE and **G**aussian **C**umulative **D**istribution **F**unction (GCDF) RPE. LFHC-RPE is an improvement of Shaw-RPE, which enhances the perception ability at medium and long relative positions. GCDF-RPE utilizes the excellent properties of the Gaussian function to amend the prior encoding mechanism in XL-RPE. Experimental results on nine authoritative datasets demonstrate the effectiveness of our methods empirically. Furthermore, GCDF-RPE achieves the best overall performance among five different RPEs.

1 Introduction

Recently, the fully attention-based Transformer model (Vaswani et al., 2017) has achieved state-of-the-art results across a range of natural language processing (NLP) tasks, including reading comprehension (Yu et al., 2018), machine translation (Raffel et al., 2020), natural language inference (Guo et al., 2019), unsupervised pretraining (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019), etc. Since the self-attention blocks in vanilla Transformer are entirely invariant to sequence order, which is one of the most important features of natural language, how to explicitly encode position information is crucial for the current Transformer based models.

The original method is to use absolute position embedding (APE), such as pre-defined sinusoidal functions (Vaswani et al., 2017) or fully data-driven learnable parameter embeddings (Devlin et al., 2019; Radford et al., 2019), to integrate position information into contextual representation. Although APE can significantly help the Transformer model learn the contextual representation of the tokens at different positions, Ke et al. (2020) pointed out that the coupled method in APE is unreasonable. Besides, APE itself also has many defects, such as the limitation of processing long sequences (Liu et al., 2020) and the gradual loss of position information (Al-Rfou et al., 2019).

To address the drawbacks mentioned above of APE, Shaw et al. (2018); Dai et al. (2019) further proposed the relative position embedding (RPE), which incorporates carefully designed temporal bias term into the self-attention module to encode the relative distance between any two tokens. RPE has been proven to be more effective than APE, and thus it is adopted by many excellent pre-trained language models (Yang et al., 2019; Song et al., 2020; Dai et al., 2020). Despite the success of RPE, the existing methods are not perfect. Although Huang et al. (2020) has made improvement to RPE, this improvement is only focused on the perspective of interaction rather than the perspective of encoding[1]. Moreover, to the best of our knowledge, there is currently no unified and comprehensive evaluation of various RPEs. Since almost every RPE is proposed for specific tasks, it is unknown whether these RPEs really have high universality and generalization ability.

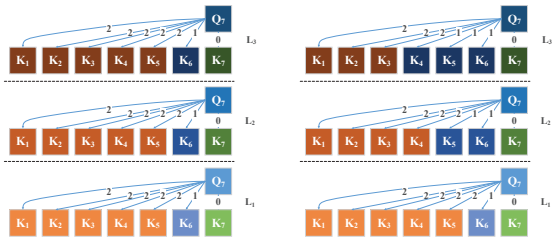
In this paper, we focus on the most widely adopted Shaw-RPE (Shaw et al., 2018) and XL-RPE (Dai et al., 2019), and improve each of them from encoding perspective. Concretely, for Shaw-RPE, to overcome its weak ability to perceive the relative position at medium and long distance, we design an ingenious **L**ow-level **F**ine-grained **H**igh-

* Corresponding author.

[1] Interaction perspective refers to how to calculate the attention weights between query and key. Encoding perspective refers to how to generate an embedding vector for each relative position.

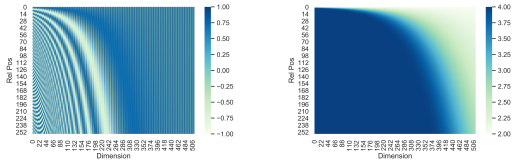
Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2989–2997
November 7–11, 2021.

level **C**oarse-grained (**LFHC**) embedding strategy without changing the number of parameters. For XL-RPE, we recognize the potential problems of its prior sinusoidal encoding functions under the relative position setting and propose a more reasonable encoding mechanism based on the **G**aussian **C**umulative **D**istribution **F**unction (**GCDF**). We conduct a unified evaluation of five RPEs on nine authoritative datasets, including language modeling, question generation, and text classification. The experimental results show that both LFHC-RPE and GCDF-RPE outperform their respective baseline, and GCDF-RPE achieves the best overall performance among the five methods[2].



(a) Shaw-RPE (b) LFHC-RPE

Figure 1: The illustration of two different pure data-driven RPEs ($k = 2$) in self-attention mechanism.



(a) XL-RPE (b) GCDF-RPE

Figure 2: The prior encoding matrix of XL-RPE and GCDF-RPE. The horizontal axis represents dimension and the vertical axis represents relative position.

2 Background

2.1 Vanilla Self-attention

The self-attention layer is the core component of Transformer, which provides a bridge for semantic interaction between tokens. In this layer, Transformer performs scaled dot-product self-attention over the input sequence by H individual attention heads and then concatenates the summary output of each head. For simplicity, we ignore the head index in the following formula. The summary output of each head is calculated as follows:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = I\mathbf{W}_q, I\mathbf{W}_k, I\mathbf{W}_v \tag{1}$$

$$\mathbf{P}_{i,j} = \frac{\mathbf{Q}_i \mathbf{K}_j^T}{\sqrt{d_{head}}} \tag{2}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{P})\mathbf{V} \tag{3}$$

where I is the input sequence representations. $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d_{model} \times d_{head}}$ are three independent linear transformation matrices, and d_{head} is the dimension of each head that satisfies $d_{head} = d_{model}/H$.

2.2 Relative Position Embeddings

Shaw-RPE (Shaw et al., 2018) is the earliest proposed RPE method. As shown in Figure 1(a), it employs fully data-driven embedding to represent different relative positions and incorporates them into the attention mechanism. In Shaw-RPE, Eq. (2) is revised as follows:

$$\text{clip}(x, k) = \max(-k, \min(k, x)) \tag{4}$$

$$\mathbf{P}_{i,j} = \frac{\mathbf{Q}_i (\mathbf{K}_j + \mathbf{w}_{\text{clip}(i-j,k)})^T}{\sqrt{d_{head}}} \tag{5}$$

where k is the maximum absolute value of relative distance and $\mathbf{w}_i \in \mathbb{R}^{d_{head}}$.

XL-RPE (Dai et al., 2019) offers a different derivation. It utilizes the sinusoidal encoding functions (Vaswani et al., 2017) to generate a prior vector embedding for each relative position (as shown in Figure 2(a)). In XL-RPE, Eq. (2) is revised as follows:

$$\mathbf{R}_d^{(2k)} = \sin(d/10000^{2k/d_{model}}) \tag{6}$$

$$\mathbf{R}_d^{(2k+1)} = \cos(d/10000^{2k/d_{model}}) \tag{7}$$

$$\mathbf{P}_{i,j} = \frac{1}{\sqrt{d_{head}}}(\mathbf{Q}_i \mathbf{K}_j^T + \mathbf{Q}_i \mathbf{W}_r \mathbf{R}_{i-j} + \mathbf{u}^T \mathbf{K}_j^T + \mathbf{v}^T \mathbf{W}_r \mathbf{R}_{i-j}) \tag{8}$$

where $\mathbf{W}_r \in \mathbb{R}^{d_{model} \times d_{head}}$ and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_{head}}$ are trainable parameters.

3 Methodology

3.1 Low-level Fine-grained High-level Coarse-grained Embedding

In Shaw-RPE, the authors discovered that precise relative position information is not useful beyond a certain distance, and this phenomenon has also been confirmed in subsequent work. Therefore,

[2]The code and training scripts will be released at https://github.com/menghuanlater/LFHC-GCDF-RPE.

Shaw-RPE sets the maximum relative distance to a relatively small value. However, this phenomenon is more likely to be caused by: (1) more independent embedding parameters will increase model optimization difficulty. (2) the greater the relative embedding distance, the more serious the optimization imbalance problem of this embedding strategy[3]. Moreover, it is necessary to distinguish the relative position at medium and long distances most of the time, especially for learning long-term dependency.

To improve the model's ability to perceive medium and long relative distances without changing the number of parameters, inspired by the analysis conclusions of many works (Jawahar et al., 2019; Ethayarajh, 2019) on Transformer that the lower layers learn local syntactic features and the higher layers capture global semantic features, we propose the *LFHC embedding strategy*. Concretely, as shown in Figure 1(b), each embedding represents a relative position range instead of a single position. At the low layers, the range is small and the embedding granularity is fine, which keep the maximum relative distance consistent with Shaw-RPE. As the level of layers increases, the range becomes larger and the embedding granularity becomes coarser, which expand the maximum relative distance gradually. In LFHC-RPE, Eq. (4) in l-th layer is revised as follows:

$$\text{clip}^l(x,k) = \begin{cases} k, & x > kl \\ \lfloor x/l \rfloor, & 0 \le x \le kl \\ \lceil x/l \rceil, & -kl \le x < 0 \\ -k, & x < -kl \end{cases} \quad (9)$$

3.2 Gaussian Cumulative Distribution Function Encoding

Intuitively and empirically, for RPEs using a prior encoding mechanism, the following two properties are important[4]:

Property 1. For an offset k and two relative position i and j where $0 <= i < j$, the proximity between the prior encoding vectors satisfies the following condition:

$$\phi(x,y) = \|\mathbf{R}_x - \mathbf{R}_y\| \quad (10)$$

$$\phi(i+k,i) > \phi(j+k,j) \quad (11)$$

Property 2. For two relative position i and j where $0 <= i < j$, the changing trend of the Euclidean distance between the prior encoding vectors satisfies the following condition:

$$\phi(i,j+2) > \phi(i,j+1) > \phi(i,j) \quad (12)$$

$$\phi(i,j+1) - \phi(i,j) > \phi(i,j+2) - \phi(i,j+1) \quad (13)$$

However, the prior sinusoidal encoding mechanism in XL-RPE does not satisfy either of these properties, especially for property 1[5]. To design a prior encoding mechanism that can satisfy the above properties, we propose the *GCDF encoding mechanism*. Specifically, each dimension of all relative positions is encoded by the GCDF with different variances. As shown in Figure 2(b), the higher the dimension, the greater the variance. In GCDF-RPE, Eq. (6) and Eq. (7) are revised as follows:

$$\sigma_k = (d_{model})^{k/d_{model}} \quad (14)$$

$$\mathbf{R}_d^k = \frac{\lambda}{\sigma_k\sqrt{2\pi}} \int_{-\infty}^{d} \exp(-\frac{x^2}{2\sigma_k^2})\, dx \quad (15)$$

where λ is the scale factor, and its default value is 4[6].

4 Experiments

In this section, we evaluate the performance of five different RPEs (T5 (Raffel et al., 2020)[7], Shaw, LFHC, XL, GCDF) on text classification, question generation and language modeling.

4.1 Experimental Setup

For text classification, IMDB (Maas et al., 2011), SNLI (Bowman et al., 2015), and four datasets (SST-2, QQP, QNLI, MNLI) belonging to GLUE (Wang et al., 2019) are used. An 8-layers 8-heads 512-dimension Transformer-encoder is used. For question generation, SQuAD (Rajpurkar et al., 2018) and CMRC (Cui et al., 2019) are employed. Pre-trained BERT-base model (Devlin et al., 2019) is chosen as the encoder, and a 3-layers 12-heads 768-dimension Transformer-decoder is employed as the decoder. For language modeling, WikiText-103 (Merity et al., 2017) is adopted. Following previous work (Dai et al., 2019), a 16-layers

[3] See Appendix A.1 for details

[4] For simplicity, we only describe these two properties in a single direction.

[5] See Appendix A.2 for proofs

[6] In our experiments, the test result is relatively stable when λ is set to be 4.

[7] T5-RPE is the simplest form of RPE, which only contains bias terms.

	IMDB	SST-2	SNLI	QNLI	QQP	MNLI
	acc	acc	acc	acc	avg(acc/f1)	avg(m/mm)
T5	88.37±0.08	82.91±0.32	86.09±0.16	80.09±0.53	72.28±0.51	73.54±0.14
Shaw	88.39±0.08	83.03±0.24	86.13±0.11	80.07±0.64	71.68±0.52	73.79±0.19
LFHC	**88.78±0.06**	**83.26±0.29**	**86.48±0.14**	**80.62±0.39**	**72.00±0.43**	**74.18±0.18**
XL	88.45±0.09	84.25±0.35	86.27±0.15	80.16±0.45	71.13±0.48	75.24±0.21
GCDF	**88.98±0.04**	**84.63±0.22**	**86.52±0.08**	**81.04±0.36**	**71.64±0.46**	**75.83±0.18**

Table 1: The performance of five RPEs on text classification tasks. All metrics are consistent with GLUE Benchmark. For QQP, the average score of accuracy and f1 is selected as the final evaluation metric. For MNLI, the average score of matched-accuracy and mismatched-accuracy is selected as the final evaluation metric.

	SQuAD	CMRC
	Rouge-L	Rouge-L
T5	47.14±0.11	60.06±0.06
Shaw	47.26±0.06	60.14±0.18
LFHC	**47.42±0.08**	**60.33±0.12**
XL	47.21±0.05	60.33±0.10
GCDF	**47.55±0.08**	**60.63±0.09**

Table 2: The dev set results of five RPEs on question generation tasks.

	M=150	M=640	Best
T5	25.26	25.02	25.02±0.20
Shaw	25.34	25.67	25.34±0.18
LFHC	**24.26**	**23.72**	**23.72±0.16**
XL	24.43	23.89	23.89±0.25
GCDF	**23.87**	**23.34**	**23.34±0.22**

Table 3: The test perplexity (ppl) of five RPEs on WikiText-103. M means the memory length during evaluation.

10-heads 410-dimension Transformer-encoder is adopted. All our experiments are conducted on 4 RTX2080Ti or single V100 GPU. To eliminate randomness, we run each experiment ten times and report the average performance. For more detailed experimental settings, please see Appendix A.3.

4.2 Main Results

The performance of five RPEs on text classification is shown in Table 1. The dev set performance on question generation is shown in Table 2. Table 3 reports the test perplexity on language modeling[8]. As can be seen from the above experimental results, both LFHC-RPE and GCDF-RPE outperform their respective baseline on all datasets. On the long-term dependency language modeling task, LFHC-RPE has a significant improvement over Shaw-RPE, which fully proves the effectiveness of the LFHC embedding strategy. Even on datasets with relatively short sentence length, such as SST-2, SNLI and QQP, LFHC-RPE does not lose accuracy, but obtains a certain degree of improvement. GCDF-RPE has a stable improvement on all datasets compared with XL-RPE, and achieves the best overall performance among the five RPEs, demonstrating the reasonability of the Gaussian prior encoding mechanism. Besides, from the overall point of view, it is obvious that RPEs based on prior encoding mechanism are better than pure data-driven RPEs, especially on SST-2 and MNLI.

4.3 Discussion

From a qualitative point of view, each type of RPE has its advantages and disadvantages. For pure data-driven RPEs (e.g., Shaw-RPE, LFHC-RPE), all their positional embedding parameters are learned autonomously by the neural network according to the characteristics of the data, so in theory, their solution space has a very high degree of freedom and can be flexibly adapted to different tasks or datasets. However, in traditional machine learning and deep learning, a high degree of freedom usually means that the model easily falls into overfitting and obtains a local suboptimal solution (the experimental results on SST-2 and MNLI can corroborate this phenomenon). For RPEs based on prior encoding mechanism (e.g., XL-RPE, GCDF-RPE), their positional parameter optimization is constrained by the prior encoding mechanism, which is equivalent to regularize the freedom of the parameter space implicitly, thus reduce the complexity of the model space and enhance the generalization of the obtained model. The overall experimental results show that RPEs based on prior encoding mechanisms achieve better performance. However, if the prior hypothesis deviates too much from reality, adverse effects will appear (e.g., the poor performance on QQP dataset).

From a quantitative point of view, it is evident from the experimental results that there does not

[8]For Shaw-RPE and LFHC-RPE, the results with different k are shown in Appendix A.4

exist any RPE that can perform best on all datasets. Even for GCDF-RPE, which has the best overall performance, there still exists a considerable gap between its performance and the optimal results on QQP dataset. Therefore, it is still very challenging and necessary to design an RPE capable of all tasks for Transformer models. We hope that our LFHC-RPE and GCDF-RPE will give some impetus to this direction.

5 Conclusion and Future Work

In this paper, we explore better RPEs from encoding perspective for Transformer models. For pure data-driven RPEs, we propose LFHC-RPE to strengthen the sensitivity at medium and long relative positions. For RPEs based on prior encoding mechanisms, we present GCDF-RPE with stronger generalization. Extensive experimental results on nine datasets show the effectiveness of our methods. We leave adjusting our methods to different kinds of pre-trained language models as our future work.

Acknowledgements

We thank anonymous reviewers for their responsible attitude and helpful comments. This work was supported by National Natural Science Foundation of China (61772060, 62106013).

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3159–3166.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. A span-extraction dataset for Chinese machine reading comprehension. 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 5883–5889, Hong Kong, China. Association for Computational Linguistics.

Zihang Dai, Guokun Lai, Yiming Yang, and Quoc Le. 2020. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. In *Advances in neural information processing systems*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

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.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Maosheng Guo, Yu Zhang, and Ting Liu. 2019. Gaussian transformer: a lightweight approach for natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6489–6496.

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.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure

of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking the positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit S. Dhillon, and Cho-Jui Hsieh. 2020. Learning to encode position for transformer with continuous dynamical model. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119, pages 6327–6335. PMLR.

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*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. 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.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

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

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*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

A Appendices

A.1 Optimization Imbalance Problem

For Shaw-RPE, if truncation is not considered, which means k is set to the maximum relative distance in the training set, then for an input token sequence with length L, when performing self-attention, as shown in Figure 3, the frequency of each relative position will gradually decrease when the absolute value of the distance increases. Since each relative position embedding parameters are independent in Shaw-RPE, this frequency decline phenomenon may lead to *inner optimization imbalance problem*.

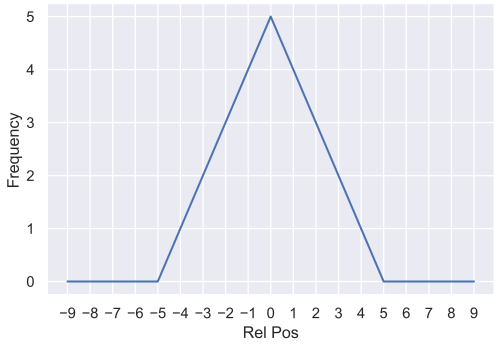


Figure 3: The frequency distribution of different relative positions when $L = 5$.

On the other hand, due to the unbalanced distribution of the input sequence length L itself (as shown in Figure 4, the length distributions on six different datasets all show characteristics similar to the long-tailed distribution), the number of samples used to optimize the medium and long relative positions is relatively small, which makes the relevant parameters easy to fall into overfitting state. We refer to this phenomenon as *internal optimization imbalance problem*.

The above two optimization imbalance problems have a greater impact on pure data-driven Shaw-RPE and LFHC-RPE when truncation is not considered. However, RPEs based on prior encoding mechanisms hardly suffer from these problems because the learnable parameters of these RPEs are shared for all relative positions. Besides, although T5-RPE is also purely data-driven, it is less affected because its parameters are only bias scalars. Perhaps in the future, we can learn from Baevski and Auli (2019) to combine Shaw-RPE and T5-RPE.

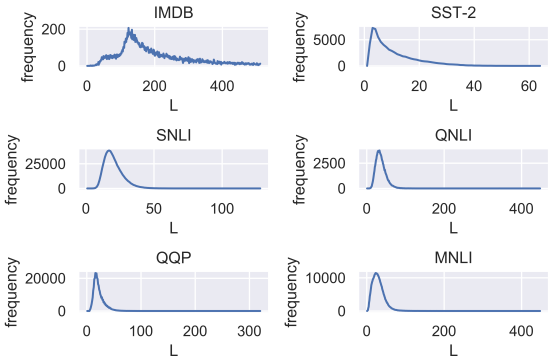


Figure 4: The length distributions of the input token sequence on six different datasets.

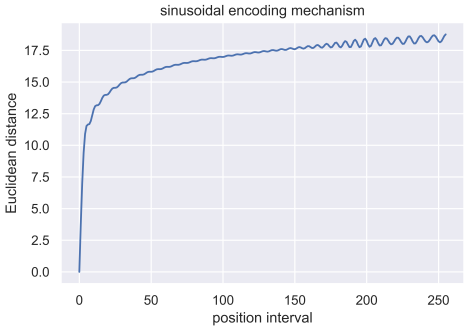


Figure 5: The Euclidean distance change for different position intervals in XL-RPE.

A.2 Prior Encoding Mechanism

As shown in Section 3.2, a good prior encoding mechanism should satisfy property 1 and property 2. Property 1 represents the *translation attenuation*: for the same interval between two relative positions, the divergence between two relative positions at a close distance is greater than that at a long distance. Property 2 means that as the interval increases, the divergence between two different relative positions will become larger, but the increasing trend should gradually stabilize. Both properties are summarized from intuition and various previous research work on representation learning. The core of these two properties is that the attention mechanism is more sensitive to relative position changes at close distances and less sensitive to relative position changes at long distances. For example, the discrepancy between $\mathbf{R}_1$ and $\mathbf{R}_5$ should be higher than the discrepancy $\mathbf{R}_{101}$ and $\mathbf{R}_{105}$.

In XL-RPE, sinusoidal functions with different periods are used as the prior encoding matrix. For an offset k and a relative position i where $i >= 0$, the divergence (squared Euclidean distance) be-

	IMDB	SST-2	SNLI	QNLI	QQP	MNLI	SQuAD	CMRC	WT103
batch size	64	64	64	64	64	64	32	32	60
FFN size	2048	2048	2048	2048	2048	2048	3072	3072	2100
lr rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2.5e-4
dropout	0.15	0.15	0.15	0.15	0.15	0.15	0.10	0.10	0.00
clip norm	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
warmup steps	200	1000	5000	4000	4000	4000	4000	600	0
maximum steps	4000	20000	80000	40000	80000	120000	30000	4500	200000
eval interval	200	500	5000	1000	4000	4000	2000	300	4000

Table 4: The other hyperparameters for different datasets.

	IMDB		SST-2		SNLI		QNLI		QQP		MNLI	
	acc	k	acc	k	acc	k	acc	k	avg(acc/f1)	k	avg(m/mm)	k
Shaw-RPE	88.28	2	82.80	2	85.81	2	79.25	2	71.24	2	73.59	2
	88.39	4	83.03	4	85.90	4	80.07	4	71.68	4	73.66	4
	88.19	8	82.68	8	86.01	8	79.70	8	71.48	8	73.78	8
	88.21	16	-	-	86.13	16	79.60	16	71.25	16	73.79	16
LFHC-RPE	88.52	2	83.14	2	86.48	2	80.51	2	71.38	2	73.87	2
	88.78	4	83.26	4	86.10	4	80.62	4	72.00	4	73.96	4
	88.59	8	83.16	8	86.18	8	80.25	8	71.45	8	74.18	8
	88.63	16	-	-	86.11	16	80.24	16	71.40	16	73.96	16

Table 5: The performance of Shaw-RPE and LFHC-RPE on text classification tasks with different k. All metrics are consistent with GLUE Benchmark. For QQP, the average score of accuracy and f1 is selected as the final evaluation metric. For MNLI, the average score of matched-accuracy and mismatched-accuracy is selected as the final evaluation metric.

tween $\mathbf{R}_i$ and $\mathbf{R}_{i+k}$ is formulated as follows:

$$w_t = (1/10000)^{2t/d_{model}} \tag{16}$$

$$\mathbf{R}_d = \begin{bmatrix} \sin(w_0 d) \\ \cos(w_0 d) \\ \vdots \\ \sin(w_{\frac{d_{model}}{2}-1} d) \\ \cos(w_{\frac{d_{model}}{2}-1} d) \end{bmatrix} \tag{17}$$

$$\begin{aligned} \|\mathbf{R}_{i+k} - \mathbf{R}_i\|^2 = & \sum_{j=0}^{\frac{d_{model}}{2}-1} [\\ & \sin^2(w_j i) + \sin^2(w_j(i+k)) + \\ & \cos^2(w_j i) + \cos^2(w_j(i+k)) - \\ & 2\sin(w_j i)\sin(w_j(i+k)) - \\ & 2\cos(w_j i)\cos(w_j(j+k))] \\ = & d_{model} - 2 \sum_{j=0}^{\frac{d_{model}}{2}-1} \cos(k) \end{aligned} \tag{18}$$

From Eq. 18, it is extremely obvious that the sinusoidal prior encoding mechanism is *translation invariant*, which completely violates property 1. And according to this equation, we plot the divergence change curve between $\mathbf{R}_0$ and other relative position encoding vertors in Figure 5. Although the sinusoidal encoding mechanism conforms to property 2 on the whole, it can be clearly seen that there are a lot of burrs on the curve, and there is a serious jitter at the medium and long intervals.

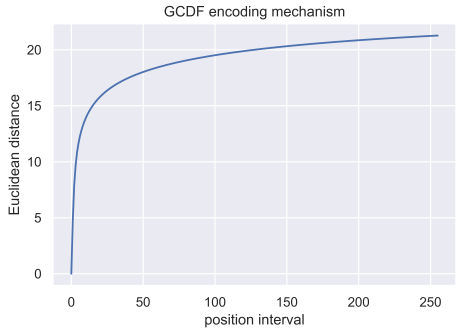


Figure 6: The Euclidean distance change for different position intervals in GCDF-RPE.

In our GCDF-RPE, Eq. 18 is revised as follows:

$$\begin{aligned} \|\mathbf{R}_{i+k} - \mathbf{R}_i\|^2 = & \sum_{j=1}^{d_{model}} (\\ & \frac{\lambda}{\sigma_j\sqrt{2\pi}} \int_i^{i+k} \exp(-\frac{x^2}{2\sigma_j^2})\, dx)^2 \end{aligned} \tag{19}$$

By converting the integral to the area, it can be easily concluded that GCDF-RPE satisfies property

1. Similar to Figure 5, we plot the same curve for GCDF-RPE. As shown in Figure 6, GCDF-RPE also satisfies property 2.

A.3 Detailed Experimental Setup

For text classification tasks, we utilize Stanford CoreNLP toolkit (Manning et al., 2014) for word segmentation, and employ pre-trained GloVe (Pennington et al., 2014) word vectors[9] to initialize the word embedding matrix. Concretely, for words with a frequency greater than three and occurring in the GloVe vocabulary, the initial parameters are pre-trained word vectors, while for other words, we treat them as unregistered words and mark them uniformly as *[UNK]*. For datasets where the input context is a single sentence, we use the max pooling representation of the output in the last layer as the classification feature. For datasets where the input context is composed by two independent sentences, we adopt the same input construction method in BERT (Devlin et al., 2019), and the representation of the *[CLS]* token in the last layer is chosen as the classification feature.

For question generation tasks, we employ the regular sequence-to-sequence structure (Sutskever et al., 2014). It should be noted that we test the performance of different RPEs only in the decoder part, which means their encoder parts are the same. For SQuAD dataset, we utilize bert-base[10] as the encoder. For CMRC dataset, we choose roberta-base-wwm-ext[11] as the encoder. Besides, beam search, copy mechanism (Gu et al., 2016), length penalty, tri-gram blocking, and token embedding sharing (Inan et al., 2017) are also been adopted. We set the beam width to 5 and the length penalty to 0.6.

For auto-regressive language modeling task, we keep the same experimental setup as in Transformer-XL[12] (Dai et al., 2019). In training, the memory length is set to 150. In validation, we follow Transformer-XL's strategy to validate the perplexity when the memory length is 150 and 640, and the best perplexity is chosen as the final result.

We choose AdamW optimizer (Loshchilov and Hutter, 2018) for all three tasks. The other hyper-parameters for different tasks are shown in Table 4.

[9]https://nlp.stanford.edu/data/wordvecs/glove.840B.300d.zip
[10]https://huggingface.co/bert-base-cased
[11]https://github.com/ymcui/Chinese-BERT-wwm
[12]https://github.com/kimiyoung/transformer-xl

	SQuAD		CMRC	
	Rouge-L	k	Rouge-L	k
Shaw-RPE	47.20	2	60.12	2
	47.22	4	60.14	4
	47.26	8	60.09	8
LFHC-RPE	47.42	2	60.33	2
	47.32	4	60.19	4
	47.28	8	60.21	8

Table 6: The dev set results on question generation tasks with different k.

	M=150	M=640	Best	k
Shaw-RPE	25.83	26.14	25.83	4
	25.34	25.67	25.34	8
	25.49	25.79	25.49	12
	25.45	25.76	25.45	16
LFHC-RPE	24.79	24.57	24.57	4
	24.58	24.02	24.02	8
	24.48	23.91	23.91	12
	24.26	23.72	23.72	16

Table 7: The test perplexity on WikiText-103 with different k.

A.4 Results with Different K

In this section, we report the full results of Shaw-RPE and LFHC-RPE with different k. Table 5 shows the results on text classification tasks. Table 6 shows the results on question generation tasks. Table 7 shows the results on language modeling.