RoFormer: Enhanced Transformer with Rotary Position Embedding

Jianlin Su
Zhuiyi Technology Co., Ltd.
Shenzhen
`bojonesu@wezhuiyi.com`

Yu Lu
Zhuiyi Technology Co., Ltd.
Shenzhen
`julianlu@wezhuiyi.com`

Shengfeng Pan
Zhuiyi Technology Co., Ltd.
Shenzhen
`nickpan@wezhuiyi.com`

Bo Wen
Zhuiyi Technology Co., Ltd.
Shenzhen
`brucewen@wezhuiyi.com`

Yunfeng Liu
Zhuiyi Technology Co., Ltd.
Shenzhen
`glenliu@wezhuiyi.com`

April 21, 2021

ABSTRACT

Position encoding in transformer architecture provides supervision for dependency modeling between elements at different positions in the sequence. We investigate various methods to encode positional information in transformer-based language models and propose a novel implementation named Rotary Position Embedding(RoPE). The proposed RoPE encodes absolute positional information with rotation matrix and naturally incorporates explicit relative position dependency in self-attention formulation. Notably, RoPE comes with valuable properties such as flexibility of being expand to any sequence lengths, decaying inter-token dependency with increasing relative distances, and capability of equipping the linear self-attention with relative position encoding. As a result, the enhanced transformer with rotary position embedding, or RoFormer, achieves superior performance in tasks with long texts. We release the theoretical analysis along with some preliminary experiment results on Chinese data. The undergoing experiment for English benchmark will soon be updated.

1 Introduction

The sequential order of words plays a vital role in natural language. Recurrent-based models (RNNs) encode tokens' order by recursively computing a hidden state along the time dimension. Convolution-based models (CNNs) [5] were typically considered position-agnostic, but recent work [9] has shown that the commonly used padding operation can implicitly learn position information. In recent years, the effectiveness of transformer-based models was shown on various natural language processing (NLP) tasks such as context representation learning [4], machine translation [21], and language modeling [16]. Unlike recurrent-based and convolution-based models, transformer-based models utilize the self-attention architecture to capture the dependency among tokens in the context, which provides better parallelization than RNNs and can model longer intra-token relations than CNNs. [1]

Since transformer-based models contain no recurrence and no convolution, and the self-attention architecture is shown to be position-agnostic [26], different approaches have been proposed to inject position information into the model. One line of works focuses on absolute position encoding, where absolute position encoding which are trainable [5, 4, 12, 2, 16, 15] or generated by pre-defined function [21] were added to context representations. The other line of works [14, 18, 7, 3, 25, 17, 11, 6, 8] focuses on relative position encoding, which typically injects relative position information into the attention calculation. In addition to these approaches, [13] has proposed to model the dependency of position encoding from the perspective with Neural ODE [1], and [22] has proposed to model the

---

[1]A stack of multiple CNN layers can also capture longer intra-token relation, here we only consider single layer setting.

position information in complex space.

In this work, we first establish a formal description of the position encoding problem in self-attention architecture and revisit previous works in Section 2. We then propose the rotary position encoding (RoPE) and study its properties in Section 3. We report preliminary experiments in section 4. Our contributions are as follows:

- We investigate previous works on relative position encoding and find most of them based on the decomposition of adding position encoding to the context representations. We propose to encode relative position by multiplying the context representations with a rotation matrix with a clear theoretical interpretation.

- We study the properties of RoPE and show that it decays with the relative distance increased, which is desired for natural language encoding. We argue that previous relative position encoding approaches are not compatible with linear self-attention and show that RoPE can be used in such mechanism.

- We demonstrate that RoFormer shows superior performance than peer models dealing with long texts. Preliminary experiments with pre-trained Chinese RoFormer [2] is carried out on downstream tasks. Benchmarks on English dataset is undergoing and will be released once finished.

2 Background and Related Work

2.1 Preliminary

Let $\mathbb{S}_N = \{w_i\}_{i=1}^N$ be a sequence of N input tokens with w_i being the ith element. The corresponding word embedding of $\mathbb{S}_N$ can be denoted as $\mathbb{E}_N = \{\boldsymbol{x}_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the d-dimensional word embedding vector of token w_i without position information. The self-attention first incorporates position information to the word embeddings and transforms them into queries, keys, and value representations.

$$
\begin{aligned}
\boldsymbol{q}_m &= f_q(\boldsymbol{x}_m, m) \\
\boldsymbol{k}_n &= f_k(\boldsymbol{x}_n, n) \\
\boldsymbol{v}_n &= f_v(\boldsymbol{x}_n, n)
\end{aligned}
\tag{1}
$$

Where $\boldsymbol{q}_m, \boldsymbol{k}_n, \boldsymbol{v}_n$ incorporates the mth and n^{th} position by functions f_q, f_k and f_v respectively. The attention weights are then calculated using query and key representations, and the output can be computed as the weighted sum of value representations.

$$
a_{m,n} = \frac{\exp(\frac{\boldsymbol{q}_m^\intercal \boldsymbol{k}_n}{\sqrt{d}})}{\sum_{j=1}^N \exp(\frac{\boldsymbol{q}_m^\intercal \boldsymbol{k}_j}{\sqrt{d}})}
$$

$$
\mathbf{o}_m = \sum_{n=1}^N a_{m,n} \boldsymbol{v}_n
\tag{2}
$$

The research on position encoding of transformer mainly focuses on choosing suitable function forms of eq. (1).

2.2 Absolute position embedding

A typical choice of eq. (1) is
$$
f_{t:t\in\{q,k,v\}}(\boldsymbol{x}_i, i) := \boldsymbol{W}_{t:t\in\{q,k,v\}}(\boldsymbol{x}_i + \boldsymbol{p}_i)
\tag{3}
$$

Where $\boldsymbol{p}_i \in \mathbb{R}^d$ is a d-dimensional vector depending of the position of token $\boldsymbol{x}_i$. [4, 12, 2, 16, 15] used a set of trainable vectors $\boldsymbol{p}_i \in \{\boldsymbol{p}_t\}_{t=1}^L$, where L is the maximum sequence length. On the other hand, [21] has proposed to generate $\boldsymbol{p}_i$ using the sinusoidal function.

$$
\begin{cases}
\boldsymbol{p}_{i,2t} &= \sin(k/10000^{2t/d}) \\
\boldsymbol{p}_{i,2t+1} &= \cos(k/10000^{2t/d})
\end{cases}
\tag{4}
$$

Where $\boldsymbol{p}_{i,2t}$ is the $2t^{th}$ element of the d-dimensional vector $\boldsymbol{p}_i$. In Section x, we will show that our proposed RoPE is related to this approach from the perspective of using the sinusoidal function, but ours incorporates relative position information by multiplying sinusoidal function to the context representation instead of adding.

---

[2] The code and pre-trained Chinese model are available at https://github.com/ZhuiyiTechnology/roformer

2.3 Relative position embedding

[18] used a different setting of eq. (1) as following:

$$f_q(\boldsymbol{x}_m) := \boldsymbol{W}_q \boldsymbol{x}_m$$
$$f_k(\boldsymbol{x}_n, n) := \boldsymbol{W}_k(\boldsymbol{x}_n + \tilde{\boldsymbol{p}}_r^k) \tag{5}$$
$$f_v(\boldsymbol{x}_n, n) := \boldsymbol{W}_v(\boldsymbol{x}_n + \tilde{\boldsymbol{p}}_r^v)$$

Where $\tilde{\boldsymbol{p}}_r^k, \tilde{\boldsymbol{p}}_r^v \in \mathbb{R}^d$ are trainable relative position embeddings. Note that $r = \text{clip}(m - n, r_{\min}, r_{\max})$ represents the relative distance between position m and n. They clipped the relative distance with the hypothesis that precise relative position information is not useful beyond a certain distance.

Keeping the form of eq. (3), [3] has proposed to decompose the $\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n$ term in eq. (2) as

$$\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n = \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{p}_n + \boldsymbol{p}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{p}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{p}_n \tag{6}$$

They replaced absolute position embedding $\boldsymbol{p}_n$ with its sinusoid-encoded relative counterpart $\tilde{\boldsymbol{p}}_{m-n}$ and replaced absolute position $\boldsymbol{p}_m$ in the third and fourth term with two trainable vectors $\mathbf{u}, \mathbf{v}$ independent of the query positions. Further, $\boldsymbol{W}_k$ is distinguished for the content-based and location-based key vectors $\boldsymbol{x}_n$ and $\boldsymbol{p}_n$, denoted as $\boldsymbol{W}_k$ and $\widetilde{\boldsymbol{W}}_k$, resulting in:

$$\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n = \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \widetilde{\boldsymbol{W}}_k \tilde{\boldsymbol{p}}_{m-n} + \mathbf{u}^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + \mathbf{v}^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \widetilde{\boldsymbol{W}}_k \tilde{\boldsymbol{p}}_{m-n} \tag{7}$$

It is worth mentioning that they remove the position information in the value term by setting $f_v(\boldsymbol{x}_j) := \boldsymbol{W}_v \boldsymbol{x}_j$. Later works [17, 6, 11, 8] followed this step by only considering inject relative position information into the attention weights. [17] revised eq. (6) as

$$\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n = \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + b_{i,j} \tag{8}$$

Where $b_{i,j}$ is a trainable bias. [11] investigated the middle two terms of **??** and found little correlations between absolute positions and words. Follow [17], they have proposed to model a pair of words or positions by using different projection matrices.

$$\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n = \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{p}_m^\mathsf{T} \mathbf{U}_q^\mathsf{T} \mathbf{U}_k \boldsymbol{p}_n + b_{i,j} \tag{9}$$

[6] argued that a relative positions of word pair can only be fully modeled by using both the middle two terms of **??**, so they have proposed to replace the absolute position embeddings $\boldsymbol{p}_m$ and $\boldsymbol{p}_n$ in these two terms with relative position embeddings $\tilde{\boldsymbol{p}}_{m-n}$.

$$\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n = \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{x}_m^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \tilde{\boldsymbol{p}}_{m-n} + \tilde{\boldsymbol{p}}_{m-n}^\mathsf{T} \boldsymbol{W}_q^\mathsf{T} \boldsymbol{W}_k \boldsymbol{x}_n \tag{10}$$

[15] has compared four variants of relative position embeddings and shown that the variant similar to eq. (10) is the most efficient among the other three.

All these works modified eq. (6) based on the decomposition of eq. (3) under the self-attention setting in eq. (2), which is originally from [21]. They share the same nature that the position information is injected by deliberately adding to the context representations. Different from these work, our approach aims to derive the relative position encoding directly from eq. (1) under some constraints. In Section xxx, we show that the derived approach is more interpretable by incorporating relative position information with the rotation of context representations.

3 Proposed Approach

In this section, we discuss the proposed rotary position embedding (RoPE). We first formulate the relative position encoding problem in section (**3.1**), we then derive the RoPE in section (**3.2**) and investigate its properties in section (**3.3**).

3.1 Formulation

Language modeling in Transformer integrates position information of individual tokens through self-attention. We start from eq. (1) and notice that the $\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n$ term in eq. (2) actually facilitates information exchange between tokens at different positions. In order to incorporate relative position information, we require the inner product of query $\boldsymbol{q}_m$ and

key $\boldsymbol{k}_n$ be formulated by a function g, which takes only the word embeddings $\boldsymbol{x}_m$, $\boldsymbol{x}_n$, and their relative position $m - n$ as input variables. In other words, we hope the inner product encodes position information only in the relative form:

$$\langle f_q(\boldsymbol{x}_m, m), f_k(\boldsymbol{x}_n, n) \rangle = g(\boldsymbol{x}_m, \boldsymbol{x}_n, m - n) \tag{11}$$

Next, finding such a encoding mechanism is equivalent to solve the function $f_q(\boldsymbol{x}_m, m)$ and $f_k(\boldsymbol{x}_n, n)$ that conforms above relation.

3.2 Rotary Position Embedding

3.2.1 A 2D case

We start from simple case with dimension $d = 2$. Under this setting, we make use of the geometric property of vectors on 2D plane and its complex form to prove (refer to Appendix A for more details) that a solution to our formulation eq. (11) is:

$$
\begin{aligned}
f_q(\boldsymbol{x}_m, m) &= (\boldsymbol{W}_q \boldsymbol{x}_m) e^{im\theta} \\
f_k(\boldsymbol{x}_n, n) &= (\boldsymbol{W}_k \boldsymbol{x}_n) e^{in\theta} \\
g(\boldsymbol{x}_m, \boldsymbol{x}_n, m - n) &= \mathrm{Re}[(\boldsymbol{W}_q \boldsymbol{x}_m)(\boldsymbol{W}_k \boldsymbol{x}_n)^* e^{i(m-n)\theta}]
\end{aligned} \tag{12}
$$

where $\mathrm{Re}[\cdot]$ is the real part of a complex number and $(\boldsymbol{W}_k \boldsymbol{x}_n)^*$ represents the conjugate complex number of $(\boldsymbol{W}_k \boldsymbol{x}_n)$. $\theta \in \mathbb{R}$ is a preset non-zero constant. We can further write $f_{\{q,k\}}$ in matrix multiplication:

$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} W_{\{q,k\}}^{(11)} & W_{\{q,k\}}^{(12)} \\ W_{\{q,k\}}^{(21)} & W_{\{q,k\}}^{(22)} \end{pmatrix} \begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix} \tag{13}$$

where $(x_m^{(1)}, x_m^{(2)})$ is $\boldsymbol{x}_m$ expressed in the 2D coordinates. Similarly, function g can be turned into matrix form. Thus, we come up with a solution to formulation in section 3.1 under the 2D case. Specifically, incorporating relative position embedding is straightforward: simply rotate the affine-transformed word embedding vector by amount of angle in multiples of its position index. Due to this characteristic, we name it *Rotary Position Embedding*.

3.2.2 General form

In order to generalize our result in 2D to any $\boldsymbol{x}_i \in \mathbb{R}^d$ where d is even, we divide the d-dimension space in to $d/2$ sub-spaces and combine them in merit of the linearity of inner product, turning $f_{\{q,k\}}$ into:

$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \boldsymbol{R}_{\Theta,m}^d \boldsymbol{W}_{\{q,k\}} \boldsymbol{x}_m \tag{14}$$

where

$$\boldsymbol{R}_{\Theta,m}^d = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix} \tag{15}$$

is the rotary matrix with pre-defined parameters $\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, ..., d/2]\}$. An graphic illustration of RoPE is shown in fig. (1).

Applying our RoPE to self-attention in eq. (2), we have:

$$\boldsymbol{q}_m^\intercal \boldsymbol{k}_n = (\boldsymbol{R}_{\Theta,m}^d \boldsymbol{W}_q \boldsymbol{x}_m)^\intercal (\boldsymbol{R}_{\Theta,n}^d \boldsymbol{W}_k \boldsymbol{x}_n) = \boldsymbol{x}^\intercal \boldsymbol{W}_q \boldsymbol{R}_{\Theta,n-m}^d \boldsymbol{W}_k \boldsymbol{x}_n \tag{16}$$

where $\boldsymbol{R}_{\Theta,n-m}^d = (\boldsymbol{R}_{\Theta,m}^d)^\intercal \boldsymbol{R}_{\Theta,n}^d$. Notice that $\boldsymbol{R}_\Theta^d$ is an orthogonal matrix, which ensures the stability during the process of encoding position information. In addition, due to the sparsity of R_Θ^d, applying matrix multiplication directly as in eq. (16) is not computational efficient, we provide another realization in Appendix B.

4

In contrast to the additive nature of position embedding method use by other works, i.e. eqs. (3) to (10), our approach is multiplicative. Moreover, our RoPE naturally incorporates relative position through rotation matrix product instead of altering terms of additive position encoding in self-attention.
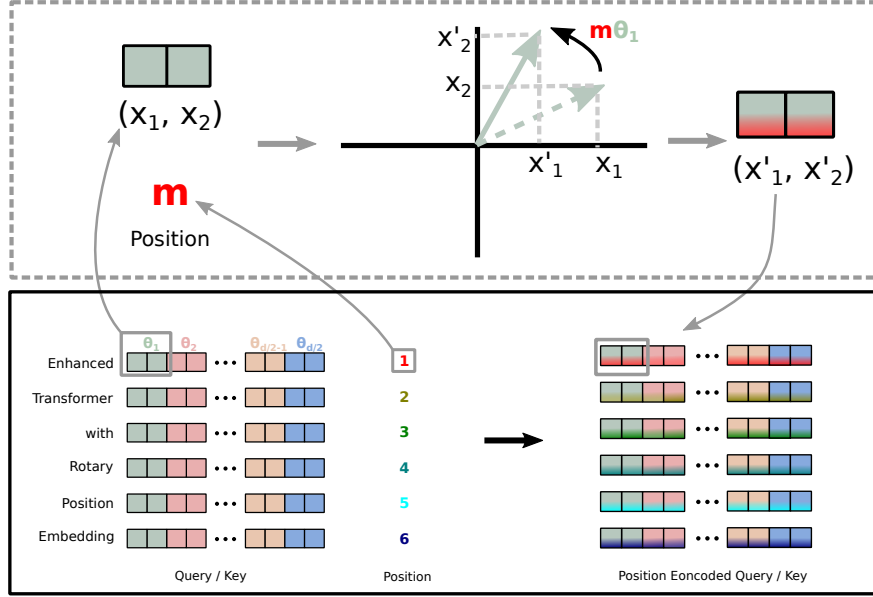

Figure 1: Implementation of Rotary Position Embedding(RoPE).

3.3 Properties of RoPE

Long-term decay: Following [21], we choose $\theta_i = 10000^{-2i/d}$. One can prove that this setting provides a long-term decay property (refer to Appendix C for more details), which means the inner-product will decay when the relative position increase. This property coincides with the intuition that a pair of tokens with long relative distance should have less connection.

RoPE with linear attention: The self-attention can be rewritten in a more general form.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_m = \frac{\sum_{n=1}^{N} \text{sim}(\boldsymbol{q}_m, \boldsymbol{k}_n)\boldsymbol{v}_n}{\sum_{n=1}^{N} \text{sim}(\boldsymbol{q}_m, \boldsymbol{k}_n)} \tag{17}$$

The original self-attention chooses $\text{sim}(\boldsymbol{q}_m, \boldsymbol{k}_n) = \exp(\boldsymbol{q}_m^\intercal \boldsymbol{k}_n / \sqrt{d})$. Notice that the original self-attention need to compute the inner product of query and key for every pair of tokens, which has quadratic complexity $\mathbb{O}(N^2)$. Follow [10] , linear attentions reformulate equation 17 as

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})_m = \frac{\sum_{n=1}^{N} \phi(\boldsymbol{q}_m)^\intercal \varphi(\boldsymbol{k}_n)\boldsymbol{v}_n}{\sum_{n=1}^{N} \phi(\boldsymbol{q}_m)^\intercal \varphi(\boldsymbol{k}_n)} \tag{18}$$

where $\phi(\cdot), \varphi(\cdot)$ are usually non-negative functions. [10] has proposed $\phi(x) = \varphi(x) = \text{elu}(x) + 1$ and first computed the multiplication between keys and values using the associative property of matrix multiplication. [19] has proposed to use softmax function to normalize queries and keys separately before the inner product, which is equivalent to $\phi(\boldsymbol{q}_i) = \text{softmax}(\boldsymbol{q}_i)$ and $\phi(\boldsymbol{k}_j) = \exp(\boldsymbol{k}_j)$. For more details about linear attentions, we encourage readers to refer to original papers. In this section, we focus on discussing incorporating RoPE with equation 18. Since RoPE injects

position information by rotation, which keeps the norm of hidden representations unchanged, we can combine RoPE with linear attentions by multiplying the rotation matrix with the outputs of the non-negative functions.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_m = \frac{\sum_{n=1}^{N} \left(\boldsymbol{R}_{\Theta,m}^d \phi(\boldsymbol{q}_m) \right)^{\mathsf{T}} \left(\boldsymbol{R}_{\Theta,n}^d \varphi(\boldsymbol{k}_n) \right) \boldsymbol{v}_n}{\sum_{n=1}^{N} \phi(\boldsymbol{q}_m)^{\mathsf{T}} \varphi(\boldsymbol{k}_n)} \tag{19}$$

It is worth mentioning that we keep the denominator unchanged to avoid the risk of dividing zero, and the summation in the numerator could contain negative terms. Although the weights for each value $\boldsymbol{v}_i$ in equation 19 are not strictly probabilistic normalized, we argue that such computation can still model the importance of values.

4 Experiment

In this section, we show some preliminary experiment results of RoFormer in Chinese language modeling, complete benchmark on English tasks is in progress and will be released once done. We discuss the implementation of RoFormer in section (**4.1**) and show the pre-training results with Chinese language in section (**4.2**). To illustrate the performance of RoFormer on long texts, we choose a task with most documents exceeding 512 characters (section (**4.3**)) for downstream evaluation and discuss the results in section (**4.4**).

4.1 Implementation

Our RoFormer model is based on WoBERT[20], from which we replace the absolute position embedding with our proposed RoPE. For cross-comparison with other pre-trained Transformer-based models in Chinese, i.e. BERT[4], WoBERT[20], and NEZHA[23], we tabulate their tokenization level and position embedding in table (**1**).

	BERT	WoBERT	NEZHA	RoFormer
tokenization level	char	word	char	word
position embedding	abs.	abs.	rel.	RoPE

Table 1: Cross-comparison between our RoFormer and other pre-trained models in Chinese. 'abs' and 'rel' annotates absolute position embedding and relative position embedding, respectively.

4.2 Pre-training

We pre-train RoFormer on approximately 34GB data which consists of contents from Chinese Wikipedia, news, forums, etc. The pre-training was done in multiple stages with changing batch size and maximum input sequence length in order to adapt the model with various scenarios, as shown in table (**2**).

stage	max seq. length	batch size	training steps	loss	accuracy
1	512	256	200k	1.73	65.0%
2	1536	256	12.5k	1.61	66.8%
3	256	256	120k	1.75	64.6%
4	128	512	80k	1.83	63.4%
5	1536	256	10k	1.58	67.4%
6	512	512	30k	1.66	66.2%

Table 2: Pre-training strategy of RoFormer on Chinese dataset. The training procedure is divided into various stages. In each stage, we train the model with a specific combination of maximum sequence length and batch size.

According to table (**2**), the accuracy of RoFormer elevates with increasing upper bound of sequence length, which demonstrates the ability of RoFormer in dealing with long texts. We claim that this is attribute to the excellent generalizability of the proposed RoPE.

4.3 Downstream Tasks & Dataset

We choose Chinese AI and Law 2019 Similar Case Matching dataset(CAIL2019-SCM)[24] to illustrate the ability of RoFormer dealing with long texts in semantic text matching task.

CAIL2019-SCM contains 8964 triplets of cases published by the Supreme People's Court of China. The input triplet, denoted as (A, B, C), are fact descriptions of three cases. The task is to predict whether the pair (A, B) is closer than (A, C) under a predefined similarity measure. Due to the background of CAIL2019-SCM dataset, most of its documents contain more than 512 characters, which is challenging for existing methods to capture document level information.

The amount of data we used in our experiment in shown in table (**3**).

Split	Count
Train	5102
Valid	1500
Test	1536
Total	8138

Table 3: The number of triplets used in different splits of CAIL2019-SCM

4.4 Results

We apply the pre-trained RoFormer model discussed in section (**4.2**) to downstream task CAIL2019-SCM with different input lengths. The model is compared with the pre-trained BERT and WoBERT model on the same pre-training data, as shown in table (**4**). With short text cut-offs, i.e. 512, RoFormer achieves comparable result to WoBERT and is slightly better than BERT implementation. However, when increase the maximum input text length to 1024 RoFormer outperforms WoBERT by an absolute improvement of 1.5%.

	validation	test
BERT-512	64.13%	67.77%
WoBERT-512	64.07%	68.10%
RoFormer-512	64.13%	68.29%
RoFormer-1024	**66.07**%	**69.79**%

Table 4: Experiment results on CAIL2019-SCM task. Numbers in the first column denote the maximum cut-off sequence length. The results are presented in terms of percent accuracy.

5   Conclusions

In this work, we proposed a new position embedding method that incorporates explicit relative position dependency in self-attention to enhance the performance of transformer architectures. Our theoretical analysis showed that relative position can be naturally formulated using vector production in self-attention, after absolution position information being encoded through rotation matrix. In addition, we mathematically illustrated the advantageous properties of the proposed method when applied in transformer. Finally, our preliminary experiment on Chinese data demonstrated that the enhanced transformer performs superior in tasks with long text. Detailed experiments on English benchmarks are in progress and will be released soon.

References

[1] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6572–6583, 2018.

[2] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[3] Zihang Dai, Z. Yang, Yiming Yang, J. Carbonell, Quoc V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, 2019.

[4] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR, 2017.

[6] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654, 2020.

[7] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, I. Simon, C. Hawthorne, Andrew M. Dai, M. Hoffman, M. Dinculescu, and D. Eck. Music transformer. *arXiv: Learning*, 2018.

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

[9] Md. Amirul Islam, Sen Jia, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? *ArXiv*, abs/2001.08248, 2020.

[10] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[11] Guolin Ke, Di He, and T. Liu. Rethinking positional encoding in language pre-training. *ArXiv*, abs/2006.15595, 2020.

[12] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[13] Xuanqing Liu, Hsiang-Fu Yu, Inderjit S. Dhillon, and Cho-Jui Hsieh. Learning to encode position for transformer with continuous dynamical model. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6327–6335. PMLR, 2020.

[14] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, 2016.

[15] A. Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[16] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[18] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL-HLT*, 2018.

[19] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3531–3539, 2021.

[20] Jianlin Su. Wobert: Word-based chinese bert model - zhuiyiai. Technical report, 2020.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[22] Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. Encoding word order in complex embeddings. In *International Conference on Learning Representations*, 2020.

[23] Victor Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. Nezha: Neural contextualized representation for chinese language understanding. 08 2019.

[24] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen hu, Heng Wang, and Jianfeng Xu. Cail2019-scm: A dataset of similar case matching in legal domain. 11 2019.

[25] Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[26] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.

A   Derivation of RoPE under 2D

Under the case of $d = 2$, we consider two word embedding vectors $\boldsymbol{x}_q$, $\boldsymbol{x}_k$ corresponds to query and key and their position m and n, respectively. According to eq. (1), their position-encoded counterparts are:

$$\begin{aligned}
\boldsymbol{q}_m &= f_q(\boldsymbol{x}_q, m) \\
\boldsymbol{k}_n &= f_k(\boldsymbol{x}_k, n)
\end{aligned} \tag{20}$$

Here the subscripts of $\boldsymbol{q}_m$, $\boldsymbol{k}_n$ indicates the encoded position information. Assume there exists function g that defines inner product between vectors produced by $f_{\{q,k\}}$:

$$\boldsymbol{q}_m^\mathsf{T} \boldsymbol{k}_n = \langle f_q(\boldsymbol{x}_m, m), f_k(\boldsymbol{x}_n, n) \rangle = g(\boldsymbol{x}_m, \boldsymbol{x}_n, n - m) \tag{21}$$

We further ask below initial condition to be satisfied:

$$\begin{aligned}
\boldsymbol{q} &= f_q(\boldsymbol{x}_q, 0) \\
\boldsymbol{k} &= f_k(\boldsymbol{x}_k, 0)
\end{aligned} \tag{22}$$

which denotes the vectors with empty position information encoded. With above settings, we manage to find a solution of f_q, f_k.

First, we take advantage of the geometric meaning of vector in 2D and its complex counter part, decompose functions in eqs. (20) and (21) into:

$$\begin{aligned}
f_q(\boldsymbol{x}_q, m) &= R_q(\boldsymbol{x}_q, m) e^{i\Theta_q(\boldsymbol{x}_q, m)} \\
f_k(\boldsymbol{x}_k, n) &= R_k(\boldsymbol{x}_k, n) e^{i\Theta_k(\boldsymbol{x}_k, n)} \\
g(\boldsymbol{x}_q, \boldsymbol{x}_k, n - m) &= R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n - m) e^{i\Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n - m)}
\end{aligned} \tag{23}$$

Where R_f, R_g and Θ_f, Θ_g are the radical and angular components for $f_{\{q,k\}}$ and g, respectively. Plug them into eq. (21), we get relation:

$$\begin{aligned}
R_q(\boldsymbol{x}_q, m) R_k(\boldsymbol{x}_k, n) &= R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n - m) \\
\Theta_k(\boldsymbol{x}_k, n) - \Theta_q(\boldsymbol{x}_q, m) &= \Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n - m)
\end{aligned} \tag{24}$$

with the corresponding initial condition as:

$$\begin{aligned}
\boldsymbol{q} &= \|\boldsymbol{q}\| e^{i\theta_q} = R_q(\boldsymbol{x}_q, 0) e^{i\Theta_q(\boldsymbol{x}_q, 0)} \\
\boldsymbol{k} &= \|\boldsymbol{k}\| e^{i\theta_k} = R_k(\boldsymbol{x}_k, 0) e^{i\Theta_k(\boldsymbol{x}_k, 0)}
\end{aligned} \tag{25}$$

Where $\|\boldsymbol{q}\|$, $\|\boldsymbol{k}\|$ and θ_q, θ_k are the radial and angular part of $\boldsymbol{q}$ and $\boldsymbol{k}$ on the 2D plane.

Next, we set $m = n$ in eq. (24) and take into account initial conditions in eq. (25):

$$R_q(\boldsymbol{x}_q, m) R_k(\boldsymbol{x}_k, m) = R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 0) = R_k(\boldsymbol{x}_q, 0) R_k(\boldsymbol{x}_k, 0) = \|\boldsymbol{q}\| \|\boldsymbol{k}\| \tag{26a}$$

$$\Theta_k(\boldsymbol{x}_k, m) - \Theta_q(\boldsymbol{x}_q, m) = \Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 0) = \|\Theta_k(\boldsymbol{x}_k, 0) - \Theta_q(\boldsymbol{x}_q, 0)\| = \|\theta_k - \theta_q\| \tag{26b}$$

On one hand, from eq. (26a), we have a straightforward solution of R_f:

$$\begin{aligned}
R_q(\boldsymbol{x}_q, m) &= R_q(\boldsymbol{x}_q, 0) = \|\boldsymbol{q}\| \\
R_k(\boldsymbol{x}_k, n) &= R_k(\boldsymbol{x}_k, 0) = \|\boldsymbol{k}\| \\
R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n - m) &= R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 0) = \|\boldsymbol{q}\| \|\boldsymbol{k}\|
\end{aligned} \tag{27}$$

Which means the radial functions R_q, R_k and R_g are functions independent to position information.

On the other hand, according to eq. (26b), notice $\Theta_q(\boldsymbol{x}_q, m) - \theta_q = \Theta_k(\boldsymbol{x}_k, m) - \theta_k$ indicates that the angular functions does not dependent on query and key, we set them to $\Theta_f := \Theta_q = \Theta_k$ and term $\Theta_f(\boldsymbol{x}_{\{q,k\}}, m) - \theta_{\{q,k\}}$ is a function of position m and is independent of word embedding $\boldsymbol{x}_{\{q,k\}}$, we denote it as $\phi(m)$, yielding:

$$\Theta_f(\boldsymbol{x}_{\{q,k\}}, m) = \phi(m) + \theta_{\{q,k\}} \tag{28}$$

Further, by plugging in $n = m + 1$ in eq. (24) and consider above equation, we have:

$$\phi(m + 1) - \phi(m) = \Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 1) + \theta_q - \theta_k \tag{29}$$

10

Since the RHS of the equation is a constant irrelevant to m, function $\phi(m)$ with continuous integer inputs produce an arithmetic progression. Thus, it is straightforward to get:

$$\phi(m) = m\theta + \gamma \tag{30}$$

Where $\theta, \gamma \in \mathbb{R}$ are constants and θ is non-zero.

To summarize our solutions from Equations (27) to (30):

$$\begin{aligned} f_q(\boldsymbol{x}_q, m) &= \|\boldsymbol{q}\|e^{i\theta_q + m\theta + \gamma} = \boldsymbol{q}e^{i(m\theta + \gamma)} \\ f_k(\boldsymbol{x}_k, n) &= \|\boldsymbol{k}\|e^{i\theta_k + n\theta + \gamma} = \boldsymbol{k}e^{i(n\theta + \gamma)} \end{aligned} \tag{31}$$

Finally, notice that we haven't set any constrains to functions in eq. (22), thus $f_q(\boldsymbol{x}_m, 0)$ and $f_k(\boldsymbol{x}_n, 0)$ are left to choose freely. To make our result be comparable to 3, here we simply set:

$$\begin{aligned} \boldsymbol{q} &= f_q(\boldsymbol{x}_m, 0) = \boldsymbol{W}_q \boldsymbol{x}_n \\ \boldsymbol{k} &= f_k(\boldsymbol{x}_n, 0) = \boldsymbol{W}_k \boldsymbol{x}_n \end{aligned} \tag{32}$$

With above and simply set $\gamma = 0$ in eq. (31), the ultimate solution is:

$$\begin{aligned} f_q(\boldsymbol{x}_m, m) &= (\boldsymbol{W}_q \boldsymbol{x}_m)e^{im\theta} \\ f_k(\boldsymbol{x}_n, n) &= (\boldsymbol{W}_k \boldsymbol{x}_n)e^{in\theta} \end{aligned} \tag{33}$$

B Computational efficient realization of rotary matrix multiplication

Taking the advantage of the sparsity of $\boldsymbol{R}_{\Theta,m}^d$ in eq. (15), a more computational efficient realization of multiplication operation between matrix R_Θ^d and vector $\boldsymbol{x} \in \mathbb{R}^d$ is:

$$\boldsymbol{R}_{\Theta,m}^d \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ x_3 \\ \vdots \\ -x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix} \tag{34}$$

C Long-term decay of RoPE

We can group entries of vectors $\boldsymbol{q} = \boldsymbol{W}_q \boldsymbol{x}_m$ and $\boldsymbol{k} = \boldsymbol{W}_k \boldsymbol{x}_n$ in pairs, and the inner product of RoPE in 16 can be written as complex number multiplication.

$$(\boldsymbol{R}_{\Theta,m}^d \boldsymbol{W}_q \boldsymbol{x}_m)^\intercal (\boldsymbol{R}_{\Theta,n}^d \boldsymbol{W}_k \boldsymbol{x}_n) = \operatorname{Re}\left[\sum_{i=0}^{d/2-1} \boldsymbol{q}_{[2i:2i+1]} \boldsymbol{k}_{[2i:2i+1]}^* e^{i(m-n)\theta_i}\right] \tag{35}$$

where $\boldsymbol{q}_{[2i:2i+1]}$ represents the $2i^{th}$ to $(2i+1)^{th}$ entries of $\boldsymbol{q}$. Denote $h_i = \boldsymbol{q}_{[2i:2i+1]}\boldsymbol{k}_{[2i:2i+1]}^*$ and $S_j = \sum_{i=0}^{j-1} e^{i(m-n)\theta_i}$, and let $h_{d/2} = 0$ and $S_0 = 0$, we can rewrite the summation using Abel transformation

$$\sum_{i=0}^{d/2-1} \boldsymbol{q}_{[2i:2i+1]}\boldsymbol{k}_{[2i:2i+1]}^* e^{i(m-n)\theta_i} = \sum_{i=0}^{d/2-1} h_i(S_{i+1} - S_i) = -\sum_{i=0}^{d/2-1} S_{i+1}(h_{i+1} - h_i) \tag{36}$$
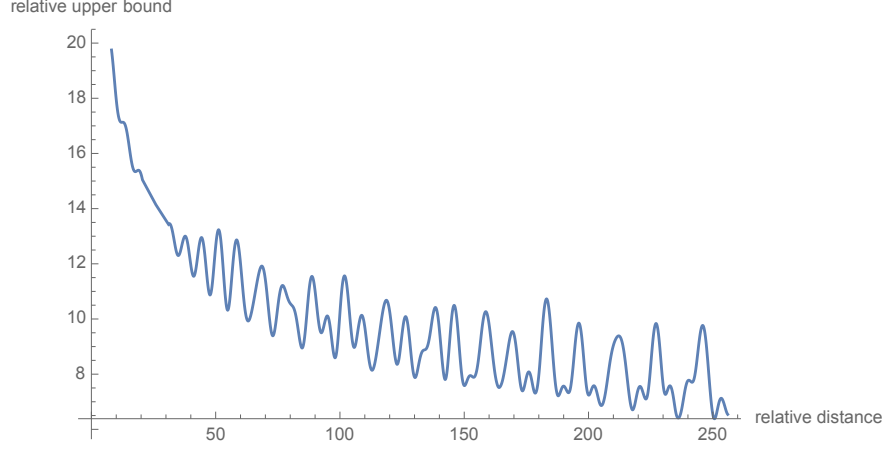
11

Figure 2: Long-term decay of RoPE.

So we have

$$\left| \sum_{i=0}^{d/2-1} \boldsymbol{q}_{[2i:2i+1]} \boldsymbol{k}^*_{[2i:2i+1]} e^{i(m-n)\theta_i} \right| = \left| \sum_{i=0}^{d/2-1} S_{i+1}(h_{i+1} - h_i) \right|$$

$$\leq \sum_{i=0}^{d/2-1} |S_{i+1}||(h_{i+1} - h_i)| \tag{37}$$

$$\leq \left(\max_i |h_{i+1} - h_i| \right) \sum_{i=0}^{d/2-1} |S_{i+1}|$$

the value of $\frac{1}{d/2} \sum_{i=1}^{d/2} |S_i|$ decay with the relative distance $m - n$ increases by setting $\theta_i = 10000^{-2i/d}$, as shown in fig. **(2)**.