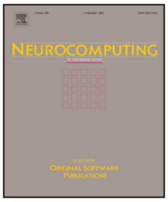
Neurocomputing 568 (2024) 127063

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



RoFormer: Enhanced transformer with Rotary Position Embedding

Jianlin Su, Murtadha Ahmed [*], Yu Lu, Shengfeng Pan, Wen Bo, Yunfeng Liu

Zhuiyi Technology Co., Ltd. Shenzhen, China

ARTICLE INFO

Communicated by F. Orihuela-Espina

Keywords:
Pre-trained language models
Position information encoding
Pre-training
Natural language processing

ABSTRACT

Position encoding has recently been shown to be effective in transformer architecture. It enables valuable supervision for dependency modeling between elements at different positions of the sequence. In this paper, we first investigate various methods to integrate positional information into the learning process of transformer-based language models. Then, we propose a novel method named Rotary Position Embedding (RoPE) to effectively leverage the positional information. Specifically, the proposed RoPE encodes the absolute position with a rotation matrix and meanwhile incorporates the explicit relative position dependency in the self-attention formulation. Notably, RoPE enables valuable properties, including the flexibility of sequence length, decaying inter-token dependency with increasing relative distances, and the capability of equipping linear self-attention with relative position encoding. Finally, we evaluate the enhanced transformer with rotary position embedding, also called RoFormer, on various long text classification benchmark datasets. Our experiments show that it consistently overcomes its alternatives. Furthermore, we provide a theoretical analysis to explain some experimental results. RoFormer is already integrated into Huggingface: https://huggingface.co/docs/transformers/model_doc/roformer.

1. Introduction

The sequential order of words is of great value to natural language understanding. Recurrent neural networks (RRNs) based models encode tokens' order by recursively computing a hidden state along the time dimension. Convolution neural networks (CNNs) based models (CNNs) [1] were typically considered position-agnostic, but recent work [2] has shown that the commonly used padding operation can implicitly learn position information. Recently, the pre-trained language models (PLMs), which were built upon the transformer, have achieved the state-of-the-art performance of various natural language processing (NLP) tasks [3], including context representation learning [4], machine translation [5], and language modeling [6], to name a few. Unlike, RRNs and CNNs-based models, PLMs utilize the self-attention mechanism to semantically capture the contextual representation of a given corpus. As a consequence, PLMs achieve a significant improvement in terms of parallelization over RNNs and improve the modeling ability of longer intra-token relations compared to CNNs.[1]

The self-attention architecture of the current PLMs has shown to be position-agnostic [7]. Following this claim, various approaches have been proposed to encode the position information into the learning process. On one side, generated absolute position encoding through a pre-defined function [5] was added to the contextual representations, while a trainable absolute position encoding [1,4,6,8–10]. On the other side, the previous work [11–19] focuses on relative position encoding, which typically encodes the relative position information into the attention mechanism. In addition to these approaches, the authors of [20] have proposed to model the dependency of position encoding from the perspective of Neural ODE [21], and the authors of [22] have proposed to model the position information in complex space. Despite the effectiveness of these approaches, they often incorporate position information into the context representation, making them unsuitable for linear self-attention architectures.

In this paper, we introduce a novel method, namely Rotary Position Embedding (RoPE), to leverage the positional information into the learning process of PLMs. Specifically, RoPE encodes the absolute position with a rotation matrix and meanwhile incorporates the explicit relative position dependency in the self-attention formulation. Note that the proposed RoPE is prioritized over the existing methods through valuable properties, including the sequence length flexibility, decaying inter-token dependency with increasing relative distances, and the capability of equipping the linear self-attention with relative position encoding. Experimental results on various long text classification benchmark datasets show that the enhanced transformer with rotary position embedding, namely RoFormer, can give better performance compared to baseline alternatives and thus demonstrates the efficacy of the proposed RoPE.

[*] Corresponding author.
E-mail address: murtadha.alrahbi@gmail.com (M. Ahmed).

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

https://doi.org/10.1016/j.neucom.2023.127063
Received 4 August 2022; Received in revised form 19 September 2023; Accepted 19 November 2023
Available online 24 November 2023
0925-2312/

In brief, our contributions are three-folds as follows:

- We investigated the existing approaches to the relative position encoding and found that they are mostly built based on the idea of the decomposition of adding position encoding to the context representations. We introduce a novel method, namely Rotary Position Embedding (RoPE), to leverage the positional information into the learning process of PLMs. The key idea is 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 increase, which is desired for natural language encoding. We kindly argue that previous relative position encoding-based approaches are not compatible with linear self-attention.
- We evaluate the proposed RoFormer on various long-text benchmark datasets. Our experiments show that it consistently achieves better performance compared to its alternatives. Some experiments with pre-trained language models are available on GitHub: https://github.com/ZhuiyiTechnology/roformer.

The remaining of the paper is organized as follows. We establish a formal description of the position encoding problem in self-attention architecture and revisit previous works in Section 2. We then describe the rotary position encoding (RoPE) and study its properties in Section 3. We report experiments in Section 4. Finally, we conclude this paper in Section 6.

2. Background

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 $\mathbb{S}_N$ is 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 into 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 $\boldsymbol{q}_m, \boldsymbol{k}_n$ and $\boldsymbol{v}_n$ incorporate the m^{th} and n^{th} positions through f_q, f_k and f_v, respectively. The query and key values are then used to compute the attention weights, while the output is computed as the weighted sum over the value representation.

$$\begin{aligned}
a_{m,n} &= \frac{\exp(\frac{\boldsymbol{q}_m^{\intercal}\boldsymbol{k}_n}{\sqrt{d}})}{\sum_{j=1}^N \exp(\frac{\boldsymbol{q}_m^{\intercal}\boldsymbol{k}_j}{\sqrt{d}})}\\
\mathbf{o}_m &= \sum_{n=1}^N a_{m,n}\boldsymbol{v}_n
\end{aligned} \tag{2}$$

The existing approaches of transformer-based position encoding mainly focus on choosing a suitable function to form Eq. (1) reviewed in details in Section 5. In this work, we attempt to derive the relative position encoding from Eq. (1) under some constraints. Next, 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

Transformer-based language modeling usually leverages the position information of individual tokens through a self-attention mechanism. As can be observed in Eq. (2), $\boldsymbol{q}_m^{\intercal}\boldsymbol{k}_n$ typically enables knowledge conveyance between tokens at different positions. To incorporate relative position information, we need the inner product of query $\mathbf{q}_m$ and key $\mathbf{k}_n$ to be computed using a function g that exclusively relies on the word embeddings $\mathbf{x}_m$, $\mathbf{x}_n$, and their relative position $m - n$ as input variables. In essence, we aim for the inner product to encode position information solely in the relative form:

$$\langle f_q(\boldsymbol{x}_m, m), f_k(\boldsymbol{x}_n, n)\rangle = g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n). \tag{3}$$

The ultimate goal is to find an equivalent encoding mechanism to solve the functions $f_q(\boldsymbol{x}_m, m)$ and $f_k(\boldsymbol{x}_n, n)$ to conform the aforementioned relation.

3.2. Rotary position embedding

3.2.1. A 2D case

We begin with a simple case with a dimension $d = 2$. Under these settings, we make use of the geometric property of vectors on a 2D plane and its complex form to prove (refer Section 3.4.1 for more details) that a solution to our formulation Eq. (3) is:

$$\begin{aligned}
f_q(\boldsymbol{x}_m, m) &= (\boldsymbol{W}_q\boldsymbol{x}_m)e^{im\theta}\\
f_k(\boldsymbol{x}_n, n) &= (\boldsymbol{W}_k\boldsymbol{x}_n)e^{in\theta}\\
g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n) &= \mathrm{Re}[(\boldsymbol{W}_q\boldsymbol{x}_m)(\boldsymbol{W}_k\boldsymbol{x}_n)^* e^{i(m-n)\theta}]
\end{aligned} \tag{4}$$

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 a multiplication matrix:

$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \begin{pmatrix}\cos m\theta & -\sin m\theta\\ \sin m\theta & \cos m\theta\end{pmatrix}\begin{pmatrix}W_{\{q,k\}}^{(11)} & W_{\{q,k\}}^{(12)}\\ W_{\{q,k\}}^{(21)} & W_{\{q,k\}}^{(22)}\end{pmatrix}\begin{pmatrix}x_m^{(1)}\\ x_m^{(2)}\end{pmatrix} \tag{5}$$

where $(x_m^{(1)}, x_m^{(2)})$ is $\boldsymbol{x}_m$ expressed in the 2D coordinates. Similarly, g can be viewed as a matrix and thus enables the solution of formulation in Section 3.1 under the 2D case. Specifically, incorporating the relative position embedding is straightforward: simply rotate the affine-transformed word embedding vector by the number of angle multiples of its position index and thus interpret the intuition behind *Rotary Position Embedding*.

3.2.2. General form

To extend our results from the 2D case to any $\mathbf{x}_i \in \mathbb{R}^d$, where d is even, we partition the d-dimensional space into $d/2$ subspaces and combine them, leveraging the linearity of the inner product. This transformation modifies $f_{\{q,k\}}$ into:

$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \boldsymbol{R}_{\Theta,m}^d \boldsymbol{W}_{\{q,k\}}\boldsymbol{x}_m \tag{6}$$

where

$$\boldsymbol{R}_{\Theta,m}^d = \begin{pmatrix}
\cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0\\
\sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0\\
0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0\\
0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\
0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2}\\
0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2}
\end{pmatrix} \tag{7}$$

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

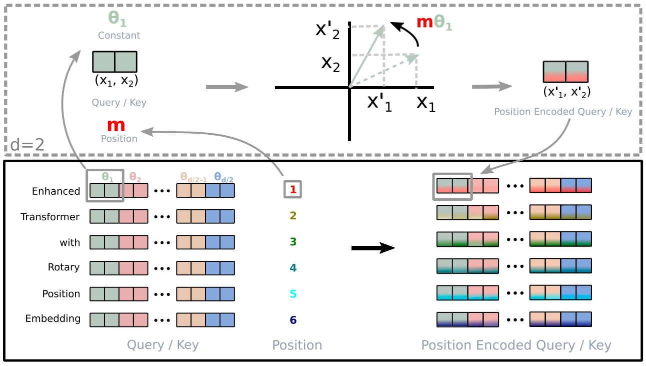


Fig. 1. Implementation of Rotary Position Embedding (RoPE).

is shown in Fig. 1. Applying our RoPE to self-attention in Eq. (2), we obtain:

$$\boldsymbol{q}_m^\intercal \boldsymbol{k}_n = (\boldsymbol{R}^d_{\Theta,m} \boldsymbol{W}_q \boldsymbol{x}_m)^\intercal (\boldsymbol{R}^d_{\Theta,n} \boldsymbol{W}_k \boldsymbol{x}_n) = \boldsymbol{x}^\intercal \boldsymbol{W}_q R^d_{\Theta,n-m} \boldsymbol{W}_k \boldsymbol{x}_n \tag{8}$$

where $\boldsymbol{R}^d_{\Theta,n-m} = (\boldsymbol{R}^d_{\Theta,m})^\intercal \boldsymbol{R}^d_{\Theta,n}$. Note that $\boldsymbol{R}^d_{\Theta}$ is an orthogonal matrix, which ensures stability during the process of encoding position information. In addition, due to the sparsity of R^d_{Θ}, applying matrix multiplication directly as in Eq. (8) is not computationally efficient; we provide another realization in theoretical explanation.

In contrast to the additive nature of the position embedding method adopted in the previous works, i.e., Eqs. (30) to (37), our approach is multiplicative. Moreover, RoPE naturally incorporates relative position information through rotation matrix product instead of altering terms in the expanded formulation of additive position encoding when applied with self-attention.

3.3. Properties of RoPE

Long-term decay. Following [5], we set $\theta_i = 10000^{-2i/d}$. One can prove that this setting provides a long-term decay property (refer to Section 3.4.3 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 a long relative distance should have less connection.

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

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V})_m = \frac{\sum_{n=1}^{N} \text{sim}(\boldsymbol{q}_m, \boldsymbol{k}_n)\boldsymbol{v}_n}{\sum_{n=1}^{N} \text{sim}(\boldsymbol{q}_m, \boldsymbol{k}_n)}. \tag{9}$$

The original self-attention chooses $\text{sim}(\boldsymbol{q}_m, \boldsymbol{k}_n) = \exp(\boldsymbol{q}_m^\intercal \boldsymbol{k}_n / \sqrt{d})$. Note that the original self-attention should compute the inner product of the query and key for every pair of tokens, which has a quadratic complexity $\mathbb{O}(N^2)$. Following Katharopoulos et al. [23], the linear attentions reformulate Eq. (9) as

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_m = \frac{\sum_{n=1}^{N} \phi(\boldsymbol{q}_m)^\intercal \varphi(\boldsymbol{k}_n)\boldsymbol{v}_n}{\sum_{n=1}^{N} \phi(\boldsymbol{q}_m)^\intercal \varphi(\boldsymbol{k}_n)}, \tag{10}$$

where $\phi(\cdot), \varphi(\cdot)$ are usually non-negative functions. The authors of [23] have 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. A softmax function is used in [24] to normalize queries and keys separately before the inner product, which is equivalent to $\phi(\boldsymbol{q}_i) = \text{softmax}(\boldsymbol{q}_i)$ and $\phi(\boldsymbol{k}_j) = \exp(\boldsymbol{k}_j)$. Note that the complexity of standard linear attention is typically $O(n \times d^2)$. For more details about linear attention, we encourage readers to refer to original papers. In this section, we focus on discussing incorporating RoPE with Eq. (10). Since RoPE injects position information by rotation, which keeps the norm of hidden representations unchanged, we can combine RoPE with linear attention by multiplying the rotation matrix with the outputs of the non-negative functions.

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V})_m = \frac{\sum_{n=1}^{N} \left(\boldsymbol{R}^d_{\Theta,m}\phi(\boldsymbol{q}_m)\right)^\intercal \left(\boldsymbol{R}^d_{\Theta,n}\varphi(\boldsymbol{k}_n)\right)\boldsymbol{v}_n}{\sum_{n=1}^{N} \phi(\boldsymbol{q}_m)^\intercal \varphi(\boldsymbol{k}_n)}. \tag{11}$$

It is noteworthy 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 Eq. (11) are not strictly probabilistic normalized, we kindly argue that the computation can still model the importance of values.

3.4. Theoretical explanation

3.4.1. 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:

$$\begin{aligned} \boldsymbol{q}_m &= f_q(\boldsymbol{x}_q, m), \\ \boldsymbol{k}_n &= f_k(\boldsymbol{x}_k, n), \end{aligned} \tag{12}$$

where the subscripts of $\boldsymbol{q}_m$ and $\boldsymbol{k}_n$ indicate the encoded positions information. Assume that there exists a function g that defines the inner product between vectors produced by $f_{\{q,k\}}$:

$$\boldsymbol{q}_m^\intercal \boldsymbol{k}_n = \langle f_q(\boldsymbol{x}_m, m), f_k(\boldsymbol{x}_n, n)\rangle = g(\boldsymbol{x}_m, \boldsymbol{x}_n, n-m), \tag{13}$$

we further require below initial condition to be satisfied:

$$\begin{aligned} \boldsymbol{q} &= f_q(\boldsymbol{x}_q, 0), \\ \boldsymbol{k} &= f_k(\boldsymbol{x}_k, 0), \end{aligned} \tag{14}$$

which can be read as the vectors with empty position information encoded. Given these settings, we attempt 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. (12) and (13) into:

$$\begin{aligned} f_q(\boldsymbol{x}_q, m) &= R_q(\boldsymbol{x}_q, m)e^{i\Theta_q(\boldsymbol{x}_q, m)}, \\ f_k(\boldsymbol{x}_k, n) &= R_k(\boldsymbol{x}_k, n)e^{i\Theta_k(\boldsymbol{x}_k, n)}, \\ g(\boldsymbol{x}_q, \boldsymbol{x}_k, n-m) &= R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n-m)e^{i\Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n-m)}, \end{aligned} \tag{15}$$

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. (13), we get the relation:

$$\begin{aligned} R_q(\boldsymbol{x}_q, m)R_k(\boldsymbol{x}_k, n) &= R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n-m), \\ \Theta_k(\boldsymbol{x}_k, n) - \Theta_q(\boldsymbol{x}_q, m) &= \Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n-m), \end{aligned} \tag{16}$$

with the corresponding initial condition as:

$$\begin{aligned} \boldsymbol{q} &= \|\boldsymbol{q}\|e^{i\theta_q} = R_q(\boldsymbol{x}_q, 0)e^{i\Theta_q(\boldsymbol{x}_q, 0)}, \\ \boldsymbol{k} &= \|\boldsymbol{k}\|e^{i\theta_k} = R_k(\boldsymbol{x}_k, 0)e^{i\Theta_k(\boldsymbol{x}_k, 0)}, \end{aligned} \tag{17}$$

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. (16) and take into account initial conditions in Eq. (17):

$$R_q(\boldsymbol{x}_q, m)R_k(\boldsymbol{x}_k, m) = R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 0) = R_k(\boldsymbol{x}_q, 0)R_k(\boldsymbol{x}_k, 0) = \|\boldsymbol{q}\|\|\boldsymbol{k}\|, \tag{18a}$$

$$\begin{aligned} \Theta_k(\boldsymbol{x}_k, m) - \Theta_q(\boldsymbol{x}_q, m) = \Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 0) &= \|\Theta_k(\boldsymbol{x}_k, 0) - \Theta_q(\boldsymbol{x}_q, 0)\| \\ &= \|\theta_k - \theta_q\|. \end{aligned} \tag{18b}$$

On one hand, a straightforward solution of R_f could be formed from Eq. (18a) :

$$\begin{aligned} R_q(\boldsymbol{x}_q, m) &= R_q(\boldsymbol{x}_q, 0) = \|\boldsymbol{q}\| \\ R_k(\boldsymbol{x}_k, n) &= R_k(\boldsymbol{x}_k, 0) = \|\boldsymbol{k}\| \\ R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, n-m) &= R_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 0) = \|\boldsymbol{q}\|\|\boldsymbol{k}\| \end{aligned} \tag{19}$$

which interprets the radial functions R_q, R_k and R_g are independent from the position information. On the other hand, as can be noticed in Eq. (18b), $\Theta_q(\boldsymbol{x}_q, m) - \theta_q = \Theta_k(\boldsymbol{x}_k, m) - \theta_k$ 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:

$$\Theta_f(\boldsymbol{x}_{\{q,k\}}, m) = \phi(m) + \theta_{\{q,k\}}, \tag{20}$$

Further, by plugging $n = m + 1$ to Eq. (16) and consider the above equation, we can get:

$$\phi(m+1) - \phi(m) = \Theta_g(\boldsymbol{x}_q, \boldsymbol{x}_k, 1) + \theta_q - \theta_k, \tag{21}$$

Since RHS is a constant irrelevant to m, $\phi(m)$ with continuous integer inputs produce an arithmetic progression:

$$\phi(m) = m\theta + \gamma, \tag{22}$$

where $\theta, \gamma \in \mathbb{R}$ are constants and θ is non-zero. To summarize our solutions from Eqs. (19) to (22):

$$\begin{aligned} f_q(\boldsymbol{x}_q, m) &= \|\boldsymbol{q}\| e^{i\theta_q + m\theta + \gamma} = \boldsymbol{q} e^{i(m\theta+\gamma)}, \\ f_k(\boldsymbol{x}_k, n) &= \|\boldsymbol{k}\| e^{i\theta_k + n\theta + \gamma} = \boldsymbol{k} e^{i(n\theta+\gamma)}. \end{aligned} \tag{23}$$

Note that we do not apply any constrains to f_q and f_k of Eq. (14), thus $f_q(\boldsymbol{x}_m, 0)$ and $f_k(\boldsymbol{x}_n, 0)$ are left to choose freely. To make our results comparable to Eq. (30), we define:

$$\begin{aligned} \boldsymbol{q} &= f_q(\boldsymbol{x}_m, 0) = \boldsymbol{W}_q \boldsymbol{x}_n, \\ \boldsymbol{k} &= f_k(\boldsymbol{x}_n, 0) = \boldsymbol{W}_k \boldsymbol{x}_n. \end{aligned} \tag{24}$$

Then, we simply set $\gamma = 0$ in Eq. (23) of the final solution:

$$\begin{aligned} f_q(\boldsymbol{x}_m, m) &= (\boldsymbol{W}_q \boldsymbol{x}_m) e^{im\theta}, \\ f_k(\boldsymbol{x}_n, n) &= (\boldsymbol{W}_k \boldsymbol{x}_n) e^{in\theta}. \end{aligned} \tag{25}$$

3.4.2. Computational efficient realization of rotary matrix multiplication

Taking the advantage of the sparsity of $\boldsymbol{R}^d_{\Theta,m}$ in Eq. (7), a more computational efficient realization of a multiplication of R^d_Θ and $\boldsymbol{x} \in \mathbb{R}^d$ is:

$$\boldsymbol{R}^d_{\Theta,m}\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ x_3 \\ \vdots \\ -x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix} \tag{26}$$

3.4.3. 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 Eq. (8) can be written as a complex number multiplication.

$$(\boldsymbol{R}^d_{\Theta,m} \boldsymbol{W}_q \boldsymbol{x}_m)^\intercal (\boldsymbol{R}^d_{\Theta,n} \boldsymbol{W}_k \boldsymbol{x}_n) = \mathrm{Re}\left[\sum_{i=0}^{d/2-1} \boldsymbol{q}_{[2i:2i+1]} \boldsymbol{k}^*_{[2i:2i+1]} e^{i(m-n)\theta_i}\right] \tag{27}$$

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

$$\begin{aligned} \sum_{i=0}^{d/2-1} \boldsymbol{q}_{[2i:2i+1]} \boldsymbol{k}^*_{[2i:2i+1]} e^{i(m-n)\theta_i} &= \sum_{i=0}^{d/2-1} h_i (S_{i+1} - S_i) \\ &= -\sum_{i=0}^{d/2-1} S_{i+1}(h_{i+1} - h_i). \end{aligned} \tag{28}$$

Thus,

$$\begin{aligned} \left|\sum_{i=0}^{d/2-1} \boldsymbol{q}_{[2i:2i+1]} \boldsymbol{k}^*_{[2i:2i+1]} e^{i(m-n)\theta_i}\right| &= \left|\sum_{i=0}^{d/2-1} S_{i+1}(h_{i+1} - h_i)\right| \\ &\le \sum_{i=0}^{d/2-1} |S_{i+1}||(h_{i+1} - h_i)| \\ &\le \left(\max_i |h_{i+1} - h_i|\right) \sum_{i=0}^{d/2-1} |S_{i+1}| \end{aligned} \tag{29}$$

Note that 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.

4. Experiments and evaluation

We evaluate the proposed RoFormer on various NLP tasks as follows. We validate the performance of the proposed solution on machine translation task Section 4.1. Then, we compare our RoPE implementation with BERT [4] during the pre-training stage in Section 4.2. Based on the pre-trained model, in Section 4.3, we further carry out evaluations across different downstream tasks from GLUE benchmarks [25]. In addition, we conduct experiments using the proposed RoPE with the linear attention of PerFormer [26] in Section 4.4. By the end, additional tests on Chinese data are included in Section 4.5. All the experiments were run on two cloud severs with 4 x V100 GPUs.

4.1. Machine translation

We first demonstrate the performance of RoFormer on sequence-to-sequence language translation tasks.

4.1.1. Experimental settings

We choose the standard WMT 2014 English–German dataset [27], which consists of approximately 4.5 million sentence pairs. We compare to the transformer-based baseline alternative [5].

4.1.2. Implementation details

We carry out some modifications on the self-attention layer of the baseline model [5] to enable RoPE to its learning process. We replicate the setup for English-to-German translation with a vocabulary of 37k based on a joint source and target byte pair encoding (BPE) [28]. In NMT, checkpoint averaging is a simple method to improve model performance at a low computational cost. The procedure is straightforward: select specific model checkpoints, average their parameters, and thus improve the model. Compared to ensembling, checkpoint averaging is computationally more efficient and eliminates the need to store and query multiple models during testing [29,30]. During the evaluation, a single model is obtained by averaging the last 5 checkpoints, and the result uses beam search with a beam size of 4 and length penalty 0.6 [31], which were widely adopted in the comparative approaches [5,32]. We implement the experiment in PyTorch in the fairseq toolkit (MIT License) [33]. Our model is optimized with the Adam optimizer using $\beta_1 = 0.9$, $\beta_2 = 0.98$, the learning rate is increased linearly from $1e{-}7$ to $5e{-}4$ and then decayed proportionally to the inverse square root of the step number. Label smoothing with 0.1 is also adopted. We report the BLEU [34] score on the test set as the final metric.

4.1.3. Results

We train the baseline models and our RoFormer under the same settings and report the results in Table 1. As can be seen, our model gives better BLEU scores compared to learned absolute position embedding Transformer [5] and the learned relative position embedding RPE [32].

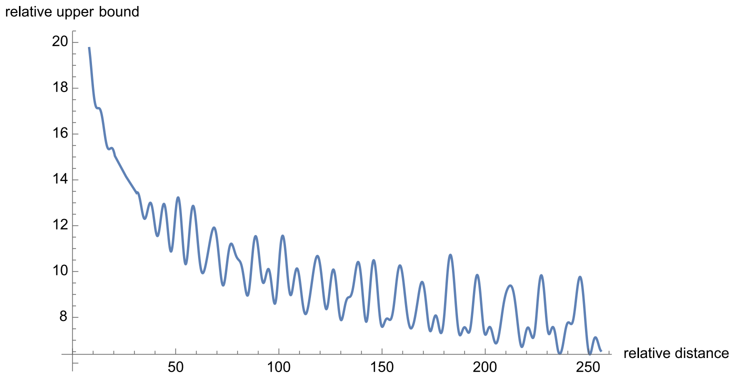


Fig. 2. Long-term decay of RoPE.

Table 1
Comparative results of learned absolute position embedding, Transformer-base [5], the learned relative position embedding RPE [32] and our proposed RoFormer on the WMT 2014 English-to-German translation task [27]. Note that the results of Transformer-base [5] and RPE are retrieved from the original paper of RPE [32]. Best scores are highlighted in bold.

Model	BLEU
Transformer-base [5]	26.5
RPE [32]	26.8
RoFormer	**27.5**

4.2. Pre-training language modeling

The second experiment is to validate the performance of our proposal in terms of learning contextual representations. To achieve this, we replace the original learned absolute position encoding of BERT with our RoPE during the pre-training step.

4.2.1. Experimental settings

We use the BookCorpus [35] and the Wikipedia Corpus [36] from Huggingface Datasets library (Apache License 2.0) for pre-training. The corpus is further split into train and validation sets at 8:2 ratio. We use the masked language-modeling (MLM) loss values of the training process as an evaluation metric. The well-known BERT [4] is adopted as our baseline model. Note that we use bert-base-uncased in our experiments.

4.2.2. Implementation details

For RoFormer, we replace the sinusoidal position encoding in the self-attention block of the baseline model with our proposed RoPE and realizes self-attention according to Eq. (8). We train both BERT and RoFormer with batch size 64 and maximum sequence length of 512 for 100k steps. AdamW [37] is used as the optimizer with learning rate 1e−5.

4.2.3. Results

The MLM loss during pre-training is shown on the left plot of Fig. 3. Compare to the vanilla BERT, RoFormer experiences faster convergence.

4.3. Fine-tuning on GLUE tasks

Consistent with the previous experiments, we fine-tune the weights of our pre-trained RoFormer across various GLUE tasks in order to evaluate its generalization ability on the downstream NLP tasks.

4.3.1. Experimental settings

We look at several datasets from GLUE, i.e. MRPC [38], SST-2 [39], QNLI [40], STS-B [41], QQP [42] and MNLI [43]. We use F1-score for MRPC and QQP dataset, spearman correlation for STS-B, and accuracy for the remaining as the evaluation metrics. We use Huggingface Transformers library (Apache License 2.0) [44] to fine-tune each of the aforementioned downstream tasks for 3 epochs with a maximum sequence length of 512, batch size of 32 and learning rates 2, 3, 4, 5e−5. Following [4], we report the best-averaged results on the validation set.

4.3.2. Results

The evaluation results of the fine-tuning tasks are reported in Table 2. For a significant test, we average the performance of three runs with different randomization and report the standard deviation. As can be seen, RoFormer can significantly outperform BERT in three out of six datasets, and the improvements are considerable.

4.4. Performer with RoPE

Performer [26] introduces an alternative attention mechanism, linear attention, which is designed to avoid quadratic computation cost that scales with input sequence length. As discussed in Section 3.3, the proposed RoPE can be easily implemented in the PerFormer model to realize the relative position encoding while keeping its linearly scaled complexity in self-attention. We demonstrate its performance with the pre-training task of language modeling.

4.4.1. Implementation details

We carry out tests on the Enwik8 dataset [45], which is from English Wikipedia that includes markup, special characters and text in other languages in addition to English text. We incorporate RoPE into the 12 layer char-based PerFormer with 768 dimensions and 12 heads.[2] To better illustrate the efficacy of RoPE, we report the loss curves of the pre-training process with and without RoPE under the same settings, i.e., learning rate 1e−4, batch size 128 and a fixed maximum sequence length of 1024, etc.

4.4.2. Results

As shown on the right plot of Fig. 3, substituting RoPE into Performer leads to rapid convergence and lower loss under the same amount of training steps. These improvements, in addition to the linear complexity, make Performer more attractive.

[2] For this experiment, we adopt code (MIT License) from https://github.com/lucidrains/performer-pytorch.

Table 2
Comparing RoFormer and BERT by fine tuning on downstream GLUE tasks. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. All comparative results are retrieved from [4]. For RoFormer, we report the averaged scores of three runs with different randomization and the standard deviation.

Model	MRPC	SST-2	QNLI	STS-B	QQP	MNLI (m/mm)
Pre-OpenAI	86.0	93.2	82.3	81.0	66.1	80.6/80.1
BiLSTM+ELMo+Attn	84.9	90.4	79.8	73.3	64.8	76.4/76.1
OpenAI GPT	82.3	91.3	87.4	80.0	70.3	82.1/81.4
BERT [4]	88.9	93.5	90.5	85.8	71.2	84.6/83.4
RoFormer	**90.0** (±0.38)	90.2 (±0.28)	87.6 (±0.08)	**87.7** (±0.37)	**86.5** (±0.05)	80.5 (±0.24)/80.5 (±0.24)

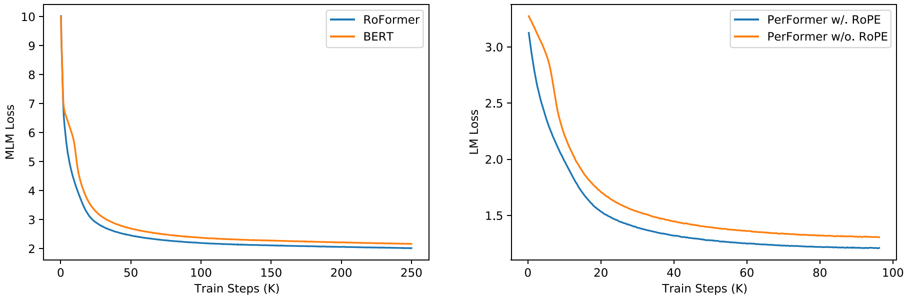


Fig. 3. Evaluation of RoPE in language modeling pre-training. **Left**: training loss for BERT and RoFormer. **Right**: training loss for PerFormer with and without RoPE.

Table 3
Cross-comparison between our RoFormer and other pre-trained models on Chinese data. 'abs' and 'rel' annotates absolute position embedding and relative position embedding, respectively.

Model	BERT [4]	WoBERT [46]	NEZHA [47]	RoFormer
Tokenization level	char	word	char	word
Position embedding	abs.	abs.	rel.	RoPE

Table 4
Pre-training strategy of RoFormer on Chinese dataset. The training procedure is divided into various consecutive stages. In each stage, we train the model with a specific combination of maximum sequence length and batch size.

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 5
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.

Model	Validation	Test
BERT-512	**64.13**%	67.77%
WoBERT-512	64.07%	68.10%
RoFormer-512	**64.13**%	**68.29**%

4.5. *Evaluation on Chinese data*

In addition to experiments on English data, we show additional results on Chinese data. To validate the performance of RoFormer on long texts, we conduct experiments on long documents whose length exceeds 512 characters.

4.5.1. *Implementation*

In these experiments, we carried out some modifications on WoBERT [46] by replacing the absolute position embedding with our proposed RoPE. As a cross-comparison with other pre-trained Transformer-based models in Chinese, i.e. BERT [4], WoBERT [46], and NEZHA [47], we tabulate their tokenization level and position embedding information in Table 3.

4.5.2. *Pre-training*

We pre-train RoFormer on approximately 34 GB of data collected from Chinese Wikipedia, news and forums. The pre-training is carried out in multiple stages with changing batch size and maximum input sequence length in order to adapt the model to various scenarios. As shown in Table 4, the accuracy of RoFormer elevates with an increasing upper bound of sequence length, which demonstrates the ability of RoFormer in dealing with long texts. We claim that this is the attribute to the excellent generalizability of the proposed RoPE.

4.5.3. *Downstream tasks & dataset*

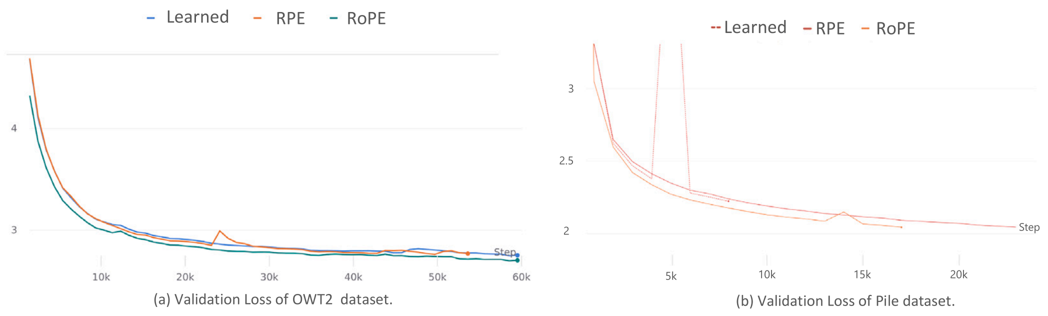
We choose Chinese AI and Law 2019 Similar Case Matching (CAIL2019-SCM) [48] dataset to illustrate the ability of RoFormer in dealing with long texts, i.e., semantic text matching. CAIL2019-SCM contains 8964 triplets of cases published by the Supreme People's Court of China. The input triplet, denoted as (A, B and 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. Note that existing methods mostly cannot perform significantly on CAIL2019-SCM dataset due to the length of documents (i.e., mostly more than 512 characters). We split train, validation and test sets based on the well-known ratio 6:2:2.

4.5.4. *Results*

We apply the pre-trained RoFormer model to 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 5. With short text cut-offs, i.e., 512, RoFormer can achieve competitive performance compared to WoBERT and slightly better than the BERT.

4.6. *RoPE against various position embeddings*

Thanks to researchers in EleutherAI [49], extensive experiments were conducted to validate the performance of learned absolute and learned relative positional embeddings methods as follows. (1) The learned absolute positional embeddings used in GPT-3 [50], denoted as Learned. (2) The learned relative positional embeddings used in



(a) Validation Loss of OWT2 dataset.

(b) Validation Loss of Pile dataset.

Fig. 4. Validation sets' losses where (a) depicts the loss of the small parameter model trained on OpenWebText2 dataset, while (b) illustrates the loss of the large scale parameter model trained on Pile dataset.

Table 6
The validation loss and PPL scores on both validation sets of OpenWebText2 (OWT2) at 55k steps (30B tokens, i.e., small parameter model), and Pile at 8k steps (8B tokens, i.e., large scale parameter model). The best scores are highlighted in **bold**.

Method	Loss		PPL	
	OWT2	Pile	OWT2	Pile
Learned absolute [50]	2.809	2.240	16.59	9.393
RPE (T5) [51]	2.801	2.223	16.46	9.234
Rotary (ours)	**2.759**	**2.173**	**15.78**	**8.784**

T5 [51], denoted as RPE. (3) Our rotary embeddings are denoted as RoPE. More details are available in the training logs.[3]

4.6.1. Experimental settings

The empirical evaluation was carried out in both small and large parameter model settings as follows. For the small parameter model, the same hyperparameters of the 125M parameter model [50] were adopted using GPT-Neox codebase. All comparative approaches were trained on OpenWebText2 (OWT2), which is a large and diverse dataset of online text. For the large-scale parameter model, 1.4B parameter models were trained with the mesh-transformer-jax codebase. Note that the same hyperparameters of GPT3's 1.3B [50] model were used on the Pile dataset [52].

4.6.2. Results

The loss values of the validation sets for all comparative approaches are reported in Fig. 4, whereas (a) illustrates the evaluation of the small parameter model, while (b) depicts the evaluation of the large parameter model. As can be seen, (1) RPE can significantly perform better than Learn and thus confirm the idea of relative embeddings; (2) RoPE can achieve better results than its alternative RPE in terms of loss decrease and convergence. For instance, RoPE, in the small parameter model settings Fig. 4(a), can achieve the same performance within only 35k steps (<55%). Moreover, the loss and PPL scores of validation sets are illustrated in Table 6 for both datasets. The scores clearly illustrate that the proposed method can outperform its alternatives, and the improvements can be deemed significant. Overall, all these results and the observations confirm that a well-designed rotary position encoding method is of great value to the performance of the current Transformer's architecture.

4.7. RoPE against GPT3

RoPE has been added to the architecture of GPT3 to examine the efficacy of relative embedding mechanism. The same experimental settings of the large-scale parameter model in Section 4.6.1 were used. The experimental logs are publicly available.[4]

4.7.1. Results

The loss values of the training and validation sets are reported in Fig. 5 in (a) and (b), respectively. As can be observed, GPT-3 with RoPE can give better scores than the original GPT-3, and the same performance can be achieved within only <75% steps on the validation set.

Moreover, the pre-trained models of both GPT-3 and GPT-3 with RoPE in various training steps have been evaluated on different fine-tuning tasks, including text understanding, [53], commonsense natural language inference [54], reading comprehension [55], etc. The scores are reported in Fig. 6 in which the performance is measured by the accuracy metric. RoPE consistently shows its computational capability across all datasets, and the improvement is significant.

4.8. Performance under different optimizers

Recently, a novel optimizer with a theoretically supported and adaptive schedule for learning rate and weight decay, called Amos [56,57] has shown significant performance compared to the state-of-the-art AdamW settings for PLMs. A comparative evaluation of various PLMs under efferent optimizers, including AdmW and Amos, has been conducted for base model size 12-layer 768-hidden by [56].

4.8.1. Experimental settings

The comparison includes BERT [4], Relative Position Embeddings, denoted as RPE, [32], and our proposed method denoted as RoPE. All the comparative models were trained on Wikipedia and the Books Corpus [58] with batch size 1024 and pre-train 200k or 300k steps; the same settings used by [59].

4.8.2. Results

The comparative evaluation on downstream tasks is reported in Fig. 7 in terms of train and validation losses as well as the MRR scores on the validation set. Our proposed solution can achieve better performance than BERT and is very competitive with RPE.

4.9. Limitations

Although we provide theoretical groundings as well as promising experimental justifications, our method is limited by the following facts:

- Despite the fact that we mathematically format the relative position relations as rotations under 2D sub-spaces there lacks thorough explanations on why it converges faster than baseline models that incorporate other position encoding strategies.

[3] EleutherAI's training logs are publicly available.

[4] RoPE vs GPT3.

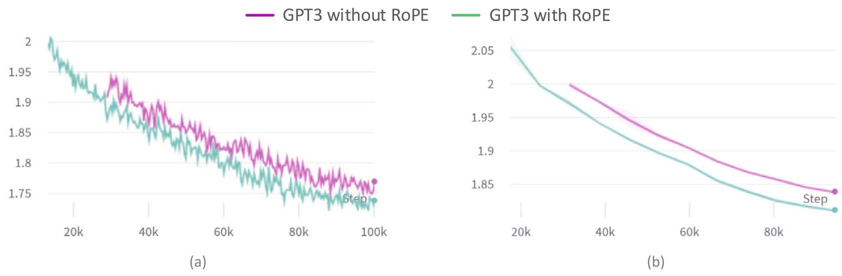
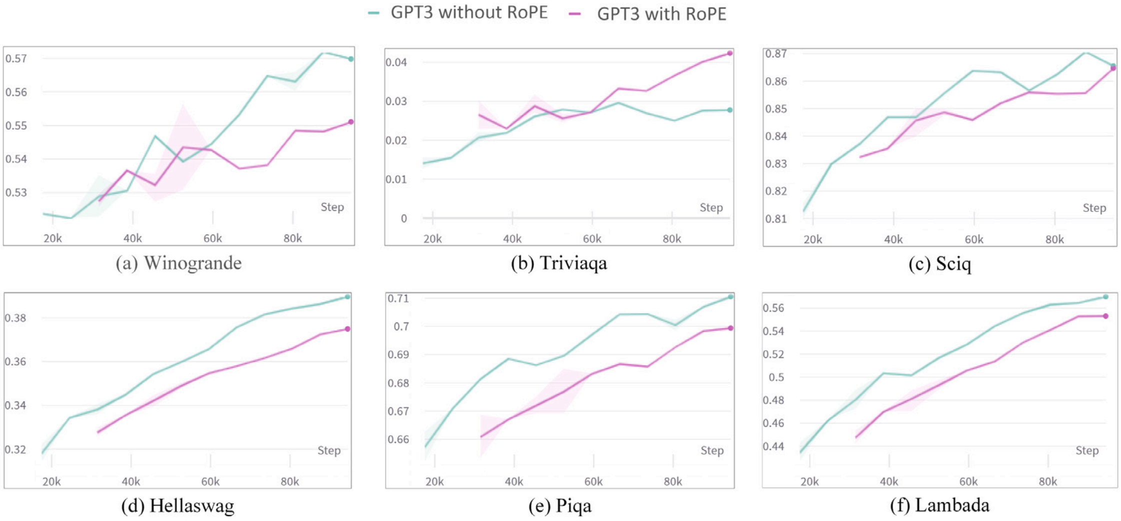


Fig. 5. The loss values of the first runs for GPT3 in large scale parameter settings trained on Pile dataset, whereas (a) illustrates the loss of training set, while (b) stands for the validation set.



(a) Winogrande (b) Triviaqa (c) Sciq (d) Hellaswag (e) Piqa (f) Lambada

Fig. 6. The performance of both GPT-3 and GPT-3 with RoPE fine-tuned in various training steps on various text classification datasets.

- Although we have proved that our model has favorable property of long-term decay for intern-token products, Section 3.3, which is similar to the existing position encoding mechanisms, our model shows superior performance on long texts than peer models; we have not come up with a faithful explanation.

5. Related work

The existing approaches of transformer-based position encoding mainly focus on choosing a suitable function to form Eq. (1). They can be best categorized into absolute position embedding and Relative position embedding. In this section, we review the literature for these two lines of research.

5.1. Absolute position embedding

A typical choice of Eq. (1) is

$$f_{t:t\in\{q,k,v\}}(\boldsymbol{x}_i, i) := \boldsymbol{W}_{t:t\in\{q,k,v\}}(\boldsymbol{x}_i + \boldsymbol{p}_i), \tag{30}$$

where $\boldsymbol{p}_i \in \mathbb{R}^d$ is a d-dimensional vector depending of the position of token $\boldsymbol{x}_i$. Previous work [4,6,8–10] introduced the use of a set of trainable vectors $\boldsymbol{p}_i \in \{\boldsymbol{p}_t\}_{t=1}^{L}$, where L is the maximum sequence length. The authors of [5] have proposed to generate $\boldsymbol{p}_i$ using the sinusoidal function.

$$\begin{cases} \boldsymbol{p}_{i,2t} &= \sin(k/10000^{2t/d}) \\ \boldsymbol{p}_{i,2t+1} &= \cos(k/10000^{2t/d}) \end{cases} \tag{31}$$

in which $\boldsymbol{p}_{i,2t}$ is the $2t^{th}$ element of the d-dimensional vector $\boldsymbol{p}_i$. In the next section, we show that our proposed RoPE is related to this intuition from the sinusoidal function perspective. However, instead of directly adding the position to the context representation, RoPE proposes to incorporate the relative position information by multiplying with the sinusoidal functions.

5.2. Relative position embedding

The authors of [12] applied different settings of Eq. (1) as following:

$$\begin{aligned} f_q(\boldsymbol{x}_m) &:= \boldsymbol{W}_q \boldsymbol{x}_m \\ f_k(\boldsymbol{x}_n, n) &:= \boldsymbol{W}_k(\boldsymbol{x}_n + \tilde{\boldsymbol{p}}_r^k) \\ f_v(\boldsymbol{x}_n, n) &:= \boldsymbol{W}_v(\boldsymbol{x}_n + \tilde{\boldsymbol{p}}_r^v) \end{aligned} \tag{32}$$

where $\tilde{\boldsymbol{p}}_r^k, \tilde{\boldsymbol{p}}_r^v \in \mathbb{R}^d$ are trainable relative position embeddings. Note that $r = \text{clip}(m-n, r_{\text{min}}, r_{\text{max}})$ 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. (30), the authors [14] have proposed to decompose $\boldsymbol{q}_m^{\intercal}\boldsymbol{k}_n$ of Eq. (2) as

$$\boldsymbol{q}_m^{\intercal}\boldsymbol{k}_n = \boldsymbol{x}_m^{\intercal}\boldsymbol{W}_q^{\intercal}\boldsymbol{W}_k\boldsymbol{x}_n + \boldsymbol{x}_m^{\intercal}\boldsymbol{W}_q^{\intercal}\boldsymbol{W}_k\boldsymbol{p}_n + \boldsymbol{p}_m^{\intercal}\boldsymbol{W}_q^{\intercal}\boldsymbol{W}_k\boldsymbol{x}_n + \boldsymbol{p}_m^{\intercal}\boldsymbol{W}_q^{\intercal}\boldsymbol{W}_k\boldsymbol{p}_n, \tag{33}$$

the key idea is to replace the absolute position embedding $\boldsymbol{p}_n$ with its sinusoid-encoded relative counterpart $\tilde{\boldsymbol{p}}_{m-n}$, while the absolute position $\boldsymbol{p}_m$ in the third and fourth term with two trainable vectors $\mathbf{u}$ and $\mathbf{v}$ 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:

$$\boldsymbol{q}_m^{\intercal}\boldsymbol{k}_n = \boldsymbol{x}_m^{\intercal}\boldsymbol{W}_q^{\intercal}\boldsymbol{W}_k\boldsymbol{x}_n + \boldsymbol{x}_m^{\intercal}\boldsymbol{W}_q^{\intercal}\widetilde{\boldsymbol{W}}_k\tilde{\boldsymbol{p}}_{m-n} + \mathbf{u}^{\intercal}\boldsymbol{W}_q^{\intercal}\boldsymbol{W}_k\boldsymbol{x}_n + \mathbf{v}^{\intercal}\boldsymbol{W}_q^{\intercal}\widetilde{\boldsymbol{W}}_k\tilde{\boldsymbol{p}}_{m-n} \tag{34}$$

It is noteworthy that the position information in the value term is removed by setting $f_v(\boldsymbol{x}_j) := \boldsymbol{W}_v\boldsymbol{x}_j$. Later work [16–19] followed

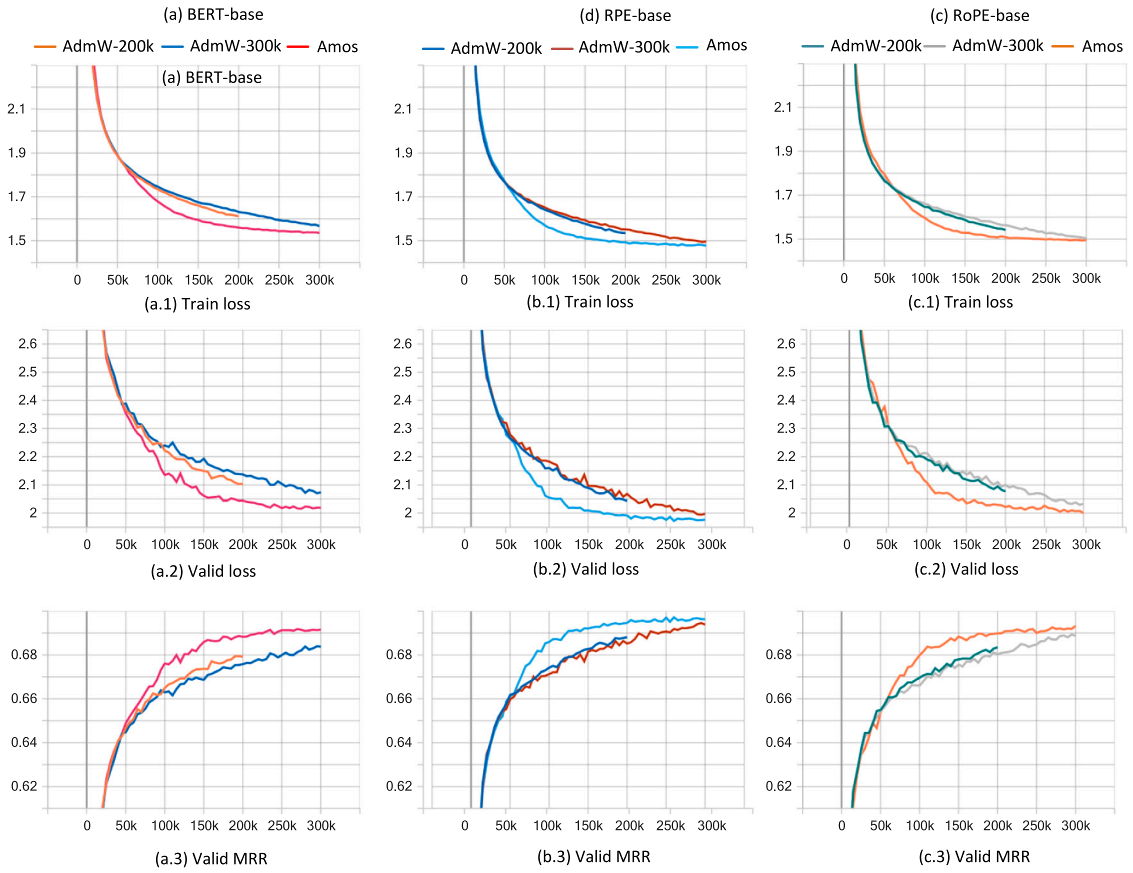


Fig. 7. Comparative evaluation of BERT [4], RPE [32] and our RoPE with various optimizers, including AdamW and Amos.

these settings by only encoding the relative position information into the attention weights. However, the authors of [16] reformed Eq. (33) as:

$$q_m^\intercal k_n = x_m^\intercal W_q^\intercal W_k x_n + b_{i,j} \tag{35}$$

where $b_{i,j}$ is a trainable bias. The authors of [17] investigated the middle two terms of Eq. (33) and found little correlations between absolute positions and words. The authors of [16] proposed to model a pair of words or positions using different projection matrices.

$$q_m^\intercal k_n = x_m^\intercal W_q^\intercal W_k x_n + p_m^\intercal \mathbf{U}_q^\intercal \mathbf{U}_k p_n + b_{i,j} \tag{36}$$

The authors of [18] argued that the relative positions of two tokens could only be fully modeled using the middle two terms of Eq. (33). As a consequence, the absolute position embeddings p_m and p_n were simply replaced with the relative position embeddings $\tilde{p}_{m-n}$:

$$q_m^\intercal k_n = x_m^\intercal W_q^\intercal W_k x_n + x_m^\intercal W_q^\intercal W_k \tilde{p}_{m-n} + \tilde{p}_{m-n}^\intercal W_q^\intercal W_k x_n \tag{37}$$

A comparison of the four variants of the relative position embeddings [10] has shown that the variant similar to Eq. (37) is the most efficient among the other three. Generally speaking, all these approaches attempt to modify Eq. (33) based on the decomposition of Eq. (30) under the self-attention settings in Eq. (2), which was originally proposed in [5]. They commonly introduced to directly add the position information to the context representations. Unlikely, our approach aims to derive the relative position encoding from Eq. (1) under some constraints.

6. Conclusion and future work

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 indicates that relative position can be naturally formulated using vector production in self-attention, with absolution position information being encoded through a rotation matrix. In addition, we mathematically illustrated the advantageous properties of the proposed method when applied to the Transformer. Finally, experiments on both English and Chinese benchmark datasets demonstrate that our method encourages faster convergence in pre-training. The experimental results also show that our proposed RoFormer can achieve better performance on long texts task. It is noteworthy RoPE in the current architecture cannot effectively extrapolate past the sequence length compared to ALiBi [60] that, instead of embeddings, directly attenuating the attention scores based on how far away the keys/queries are. For future work, we consider a hybrid collaborative approach; however, further investigation is needed.

CRediT authorship contribution statement

Jianlin Su: Methodology. **Murtadha Ahmed:** Supervision, Formal analysis, Validation. **Yu Lu:** Conceptualization, Software, Validation. **Shengfeng Pan:** Writing – original draft. **Wen Bo:** Writing – review & editing. **Yunfeng Liu:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 1243–1252, URL http://proceedings.mlr.press/v70/gehring17a.html.

[2] M.A. Islam, S. Jia, N.D.B. Bruce, How much position information do convolutional neural networks encode? in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020, URL https://openreview.net/forum?id=rJeB36NKvB.

[3] A. Murtadha, S. Pan, W. Bo, J. Su, X. Cao, W. Zhang, Y. Liu, Rank-Aware Negative Training for Semi-Supervised Text Classification, Transactions of the Association for Computational Linguistics (ISSN: 2307-387X) 11 (2023) 771–786, http://dx.doi.org/10.1162/tacl_a_00574, https://doi.org/10.1162/tacl_a_00574.

[4] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/n19-1423.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008, URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019.

[7] C. Yun, S. Bhojanapalli, A.S. Rawat, S.J. Reddi, S. Kumar, Are transformers universal approximators of sequence-to-sequence functions? in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020, URL https://openreview.net/forum?id=ByxRM0Ntvr.

[8] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: a lite BERT for self-supervised learning of language representations, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020, URL https://openreview.net/forum?id=H1eA7AEtvS.

[9] K. Clark, M. Luong, Q.V. Le, C.D. Manning, ELECTRA: pre-training text encoders as discriminators rather than generators, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020, URL https://openreview.net/forum?id=r1xMH1BtvB.

[10] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving Language Understanding by Generative Pre-Training, OpenAI, 2018.

[11] A.P. Parikh, O. Täckström, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, in: J. Su, X. Carreras, K. Duh (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, The Association for Computational Linguistics, 2016, pp. 2249–2255, http://dx.doi.org/10.18653/v1/d16-1244.

[12] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in: M.A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), Association for Computational Linguistics, 2018, pp. 464–468, http://dx.doi.org/10.18653/v1/n18-2074.

[13] C.A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A.M. Dai, M.D. Hoffman, M. Dinculescu, D. Eck, Music transformer: Generating music with long-term structure, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, OpenReview.net, 2019, URL https://openreview.net/forum?id=rJe4ShAcF7.

[14] Z. Dai, Z. Yang, Y. Yang, J.G. Carbonell, Q.V. Le, R. Salakhutdinov, Transformer-XL: Attentive language models beyond a fixed-length context, in: A. Korhonen, D.R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 2978–2988, http://dx.doi.org/10.18653/v1/p19-1285.

[15] Z. Yang, Z. Dai, Y. Yang, J.G. Carbonell, R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 5754–5764, URL https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67, URL http://jmlr.org/papers/v21/20-074.html.

[17] G. Ke, D. He, T. Liu, Rethinking positional encoding in language pre-training, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021, URL https://openreview.net/forum?id=09-528y2Fgf.

[18] P. He, X. Liu, J. Gao, W. Chen, Deberta: decoding-enhanced bert with disentangled attention, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021, URL https://openreview.net/forum?id=XPZIaotutsD.

[19] Z. Huang, D. Liang, P. Xu, B. Xiang, Improve transformer models with better relative position embeddings, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 3327–3335, http://dx.doi.org/10.18653/v1/2020.findings-emnlp.298, URL https://www.aclweb.org/anthology/2020.findings-emnlp.298.

[20] X. Liu, H. Yu, I.S. Dhillon, C. 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, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 6327–6335, URL http://proceedings.mlr.press/v119/liu20n.html.

[21] T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural ordinary differential equations, in: S. Bengio, H.M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, MontrÉAl, Canada, 2018, pp. 6572–6583, URL https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

[22] B. Wang, D. Zhao, C. Lioma, Q. Li, P. Zhang, J.G. Simonsen, Encoding word order in complex embeddings, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020, URL https://openreview.net/forum?id=Hke-WTVtwr.

[23] A. Katharopoulos, A. Vyas, N. Pappas, F. Fleuret, Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 5156–5165, URL http://proceedings.mlr.press/v119/katharopoulos20a.html.

[24] Z. Shen, M. Zhang, H. Zhao, S. Yi, H. Li, Efficient attention: Attention with linear complexities, in: IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021, IEEE, 2021, pp. 3530–3538, http://dx.doi.org/10.1109/WACV48630.2021.00357.

[25] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S.R. Bowman, GLUE: a multi-task benchmark and analysis platform for natural language understanding, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, OpenReview.net, 2019, URL https://openreview.net/forum?id=rJ4km2R5t7.

[26] K.M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J.Q. Davis, A. Mohiuddin, L. Kaiser, D.B. Belanger, L.J. Colwell, A. Weller, Rethinking attention with performers, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021, URL https://openreview.net/forum?id=Ua6zuk0WRH.

[27] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, A. Tamchyna, Findings of the 2014 workshop on statistical machine translation, in: Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA, The Association for Computer Linguistics, 2014, pp. 12–58, http://dx.doi.org/10.3115/v1/w14-3302.

[28] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers, The Association for Computer Linguistics, 2016, http://dx.doi.org/10.18653/v1/p16-1162.

[29] M. Junczys-Dowmunt, T. Dwojak, H. Hoang, Is neural machine translation ready for deployment? A case study on 30 translation directions, in: Proceedings of the 13th International Conference on Spoken Language Translation, IWSLT 2016, Seattle, WA, USA, December 8-9, 2016, International Workshop on Spoken Language Translation, 2016, URL https://aclanthology.org/2016.iwslt-1.5.

[30] M. Popel, O. Bojar, Training tips for the transformer model, Prague Bull. Math. Linguist. 110 (2018) 43–70, URL http://ufal.mff.cuni.cz/pbml/110/art-popel-bojar.pdf.

[31] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, CoRR abs/1609.08144. arXiv:1609.08144. URL http://arxiv.org/abs/1609.08144.

[32] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in: M.A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), Association for Computational Linguistics, 2018, pp. 464–468, http://dx.doi.org/10.18653/v1/n18-2074.

[33] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, M. Auli, Fairseq: A fast, extensible toolkit for sequence modeling, in: W. Ammar, A. Louis, N. Mostafazadeh (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations, Association for Computational Linguistics, 2019, pp. 48–53, http://dx.doi.org/10.18653/v1/n19-4009.

[34] K. Papineni, S. Roukos, T. Ward, W. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA, ACL, 2002, pp. 311–318, http://dx.doi.org/10.3115/1073083.1073135, URL https://aclanthology.org/P02-1040/.

[35] Y. Zhu, R. Kiros, R.S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, IEEE Computer Society, 2015, pp. 19–27, http://dx.doi.org/10.1109/ICCV.2015.11.

[36] W. Foundation, Wikimedia downloads, 2021, https://dumps.wikimedia.org.

[37] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, OpenReview.net, 2019, URL https://openreview.net/forum?id=Bkg6RiCqY7.

[38] W.B. Dolan, C. Brockett, Automatically constructing a corpus of sentential paraphrases, in: Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005, Asian Federation of Natural Language Processing, 2005, URL https://aclanthology.org/I05-5002/.

[39] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, a Meeting of SIGDAT, a Special Interest Group of the ACL, ACL, 2013, pp. 1631–1642, URL https://aclanthology.org/D13-1170/.

[40] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100, 000+ questions for machine comprehension of text, in: J. Su, X. Carreras, K. Duh (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, The Association for Computational Linguistics, 2016, pp. 2383–2392, http://dx.doi.org/10.18653/v1/d16-1264.

[41] H.T. Al-Natsheh, L. Martinet, F. Muhlenbach, D.A. Zighed, Udl at SemEval-2017 task 1: Semantic textual similarity estimation of english sentence pairs using regression model over pairwise features, in: S. Bethard, M. Carpuat, M. Apidianaki, S.M. Mohammad, D.M. Cer, D. Jurgens (Eds.), Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017, Association for Computational Linguistics, 2017, pp. 115–119, http://dx.doi.org/10.18653/v1/S17-2013.

[42] Z. Chen, H. Zhang, X. Zhang, L. Zhao, Quora question pairs, 2018, URL https://www.kaggle.com/c/quora-question-pairs.

[43] A. Williams, N. Nangia, S.R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: M.A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 1112–1122, http://dx.doi.org/10.18653/v1/n18-1101.

[44] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T.L. Scao, S. Gugger, M. Drame, Q. Lhoest, A.M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45, URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[45] M. Mahoney, Large text compression benchmark, 2011.

[46] J. Su, WoBERT: Word-based Chinese BERT model - ZhuiyiAI, Tech. rep., 2020, URL https://github.com/ZhuiyiTechnology/WoBERT.

[47] J. Wei, X. Ren, X. Li, W. Huang, Y. Liao, Y. Wang, J. Lin, X. Jiang, X. Chen, Q. Liu, NEZHA: neural contextualized representation for Chinese language understanding, 2019, CoRR abs/1909.00204. arXiv:1909.00204. URL http://arxiv.org/abs/1909.00204.

[48] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, T. Zhang, X. Han, Z. Hu, H. Wang, J. Xu, CAIL2019-SCM: a dataset of similar case matching in legal domain, 2019, CoRR abs/1911.08962. arXiv:1911.08962. URL http://arxiv.org/abs/1911.08962.

[49] S. Biderman, S. Black, C. Foster, L. Gao, E. Hallahan, H. He, B. Wang, P. Wang, Rotary embeddings: A relative revolution, 2021, [Online; accessed].

[50] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual, 2020, URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[51] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67, URL http://jmlr.org/papers/v21/20-074.html.

[52] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy, The pile: An 800gb dataset of diverse text for language modeling, 2021, CoRR abs/2101.00027. arXiv:2101.00027. URL https://arxiv.org/abs/2101.00027.

[53] D. Paperno, G. Kruszewski, A. Lazaridou, Q.N. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, R. Fernández, The LAMBADA dataset: Word prediction requiring a broad discourse context, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers, The Association for Computer Linguistics, 2016, http://dx.doi.org/10.18653/v1/p16-1144.

[54] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, Y. Choi, HellaSwag: Can a machine really finish your sentence? in: A. Korhonen, D.R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 4791–4800, http://dx.doi.org/10.18653/v1/p19-1472.

[55] M. Joshi, E. Choi, D.S. Weld, L. Zettlemoyer, Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, in: R. Barzilay, M. Kan (Eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, Association for Computational Linguistics, 2017, pp. 1601–1611, http://dx.doi.org/10.18653/v1/P17-1147.

[56] R. Tian, A.P. Parikh, Amos: An adam-style optimizer with adaptive weight decay towards model-oriented scale, 2022, http://dx.doi.org/10.48550/arXiv.2210.11693, CoRR abs/2210.11693. arXiv:2210.11693.

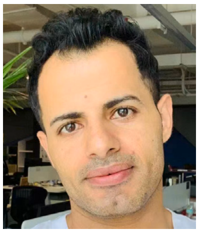
[57] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, 2020, CoRR abs/2005.12872. arXiv:2005.12872. URL https://arxiv.org/abs/2005.12872.

[58] Y. Zhu, R. Kiros, R.S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, IEEE Computer Society, 2015, pp. 19–27, http://dx.doi.org/10.1109/ICCV.2015.11.

[59] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, CoRR abs/1907.11692. arXiv:1907.11692. URL http://arxiv.org/abs/1907.11692.

[60] O. Press, N.A. Smith, M. Lewis, Train short, test long: Attention with linear biases enables input length extrapolation, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022, URL https://openreview.net/forum?id=R8sQPpGCv0.

Jianlin Su received his master degree from Sun Yat-Sen University, Guangzhou, China. He is currently an NLP Researcher at Zhuiyi Technology Co., Ltd, Shenzhen. His current research interests focus on various NLP problems, including Pre-trained Language Models, Named-Entity Recognition, and Few-Shot Learning. One of his contribution to the research community is RoFormer: Enhanced Transformer with Rotary Position Embedding.



Ahmed Murtadha (Member, IEEE) received his Pd.D from School of Computer Science in Northwestern Polytechnical University, China, in 2021. He is currently an NLP Researcher at Zhuiyi Technology Co., Ltd. Shenzhen, China. His current research interests focus machine learning solutions for NLP, including enabling machine learning with less human labeling effort, learning with noisy labels and robust defense training against adversarial attacks.

Yu Lu received his Ph.D. degree in Physics from the University of Texas at Austin, 2019 and B.S. degree from the University of Science and Technology of China, 2014. His current research interests in AI include cross-modal deep learning, AI content generation and digital human.

Shengfeng Pan received his master degree from Hong Kong University of Science and Technology, Hong Kong. He is currently an NLP Researcher at Zhuiyi Technology Co., Ltd. His current research interests focus on the effective contextual semantics learning, Chatbots and Dialogue Systems.

Wen Bo received his B.sc and M.sc degrees from Harbin institute of technology, China. He is currently an NLP algorithm expert engineer at Zhuiyi Technology. His current research interests focus on Machine Learning and Deep Neural Networks-based solutions for NLP applications, including Multilingual Pre-trained Language Models, Chatbots and Dialogue Systems, Semantic Learning of Low-resource Languages, e.g., Arabic and Japanese.

Yunfeng Liu received his Pd.D from School of Computer Science in Huazhong University of Science and Technology, China. He was the former head of technology research channel and technical director of Tencent Technology Vocational Development Association. He is currently the CTO of Zhuiyi Technology Co., Ltd, Shenzhen. His research interests focus on ML theory and application in various fields, including Computer vision, Voice Recognition and Natural Language Process.