

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

Ahmed Murtadha

Zhuiyi Technology Co., Ltd.
Shenzhen
mengjiayi@wezhuiyi.com

Bo Wen

Zhuiyi Technology Co., Ltd.
Shenzhen
brucewen@wezhuiyi.com

Yunfeng Liu

Zhuiyi Technology Co., Ltd.
Shenzhen
glenliu@wezhuiyi.com

November 9, 2023

ABSTRACT

Position encoding recently has shown effective in the 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 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 the 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

특성	왜 유용한가?	RoPE에서 어떻게 얻어지나?
◎ 길이 유연성	훈련 때 본 적 없던 > 길이 문장을 다룰 수 있음	회전 각도는 수식으로 바로 계산 → 위치 벡터를 따로 학습할 필요 X
◎ 감쇠(decay)	멀리 떨어진 단어끼리 영향력이 자연스럽게 줄어들음	$\cos((m-n)\theta)$ 항이 거리↑ → 값↓
◎ 선형 attention 호환	메모리 $O(n)$ 인 Performer Flash-Attention과 함께 사용 가능	회전은 쿼리/키를 사전 변환할 뿐, 이후 계산 방식(커널)을 가리지 않음 <small>www</small>

RoPE

- Absolute Position에 회전변환
- Relative position을 Self-attn에 통합시킴
- Seq length가 유동적이게 됨
- 상대 거리가 멀어지는 토큰간에 의존성을 감소시킴
- Linear Self-attn에 Relative Position Encoding을 적용 가능
- 케 함

Keywords Pre-trained Language Models · Position Information Encoding · Pre-training · Natural Language Processing.

1 Introduction

The sequential order of words is of great value to natural language understanding. Recurrent neural networks (RNNs) based models encode tokens' order by recursively computing a hidden state along the time dimension. Convolution neural networks (CNNs) based models (CNNs) [Gehring et al. [2017]] were typically considered position-agnostic, but recent work [Islam et al. [2020]] 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 [Vaswani et al. [2017]], have achieved the state-of-the-art performance of various natural language processing (NLP) tasks, including context representation learning [Devlin et al. [2019]], machine translation [Vaswani et al. [2017]], and language modeling [Radford et al. [2019]], to name a few. Unlike, RNNs 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 CNNs¹.

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

It is noteworthy that the self-attention architecture of the current PLMs has shown to be position-agnostic [Yun et al. 2020]. 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 [Vaswani et al. 2017] was added to the contextual representations, while a trainable absolute position encoding [Gehring et al. 2017], [Devlin et al. 2019], [Lan et al. 2020], [Clark et al. 2020], [Radford et al. 2019], [Radford and Narasimhan 2018]. On the other side, the previous work [Parikh et al. 2016], [Shaw et al. 2018], [Huang et al. 2018], [Dai et al. 2019], [Yang et al. 2019], [Raffel et al. 2020], [Ke et al. 2020], [He et al. 2020], [Huang et al. 2020] focuses on relative position encoding, which typically encodes the relative position information into the attention mechanism. In addition to these approaches, the authors of [Liu et al. 2020] have proposed to model the dependency of position encoding from the perspective of Neural ODE [Chen et al. 2018a], and the authors of [Wang et al. 2020] have proposed to model the position information in complex space. Despite the effectiveness of these approaches, they commonly add the position information to the context representation and thus render them unsuitable for the linear self-attention architecture.

RoPE는 회전행렬을 통해 Abs Position을 인코딩하는 동시에 Self-Attention에 명시적인 Relative Position Dependency를 통합한다

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 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.

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 increased, 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>

RoPE의 핵심 아이디어는 context representation에 회전 행렬을 곱함으로써 상대 위치 정보를 인코딩하는 것임

RoPE는 Relative Distance가 멀어질 수록 Decay하는 올바른 특성을 보임

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 (5).

2 Background and Related Work

2.1 Preliminary

Let $S_N = \{w_i\}_{i=1}^N$ be a sequence of N input tokens with w_i being the i^{th} element. The corresponding word embedding of S_N is denoted as $E_N = \{x_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ 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.

S_N : N 개 Input Tokens를 가진 Sequence
 w_i : N 개의 Token의 i 번째 token
 E_N : N 개 Token의 Embedding
 x_i : 각 Token의 dimension d 차원의 벡터값
(Position Information을 갖기 전 임베딩 벡터)

일반적인 Pos + Emb

$$\begin{aligned} q_m &= f_q(x_m, m) \\ k_n &= f_k(x_n, n) \\ v_n &= f_v(x_n, n), \end{aligned}$$

m, n 은 m 번째 Position, n 번째 Position을 의미,
 f_q, f_k, f_v 는 Position Information을 통합시킨 값을 의미 (1)

where q_m, k_n and 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

Position Information이 통합된 후, Attention Weight를 계산하게됨

representation.

$$a_{m,n} = \frac{\exp(\frac{q_m^\top k_n}{\sqrt{d}})}{\sum_{j=1}^N \exp(\frac{q_m^\top k_j}{\sqrt{d}})} \quad (2)$$

$$\mathbf{o}_m = \sum_{n=1}^N a_{m,n} \mathbf{v}_n \quad \text{Attn Weight} = Q^\top * K / V \text{ 공식}$$

The existing approaches of transformer-based position encoding mainly focus on choosing a suitable function to form Equation (1).

2.2 Absolute position embedding 기존의 Abs Pos 임베딩 방식

A typical choice of Equation (1) is

$$f_{t:t \in \{q,k,v\}}(\mathbf{x}_i, i) := \mathbf{W}_{t:t \in \{q,k,v\}}(\mathbf{x}_i + \mathbf{p}_i), \quad (3)$$

where $\mathbf{p}_i \in \mathbb{R}^d$ is a d-dimensional vector depending of the position of token \mathbf{x}_i . Previous work [Devlin et al. [2019], Lan et al. [2020], Clark et al. [2020], Radford et al. [2019], Radford and Narasimhan [2018] introduced the use of a set of trainable vectors $\mathbf{p}_i \in \{\mathbf{p}_t\}_{t=1}^L$, where L is the maximum sequence length. The authors of [Vaswani et al. [2017] have proposed to generate \mathbf{p}_i using the sinusoidal function. Transformers 논문에서 \mathbf{p}_i 는 Sinusoidal Function임

$$\begin{cases} \mathbf{p}_{i,2t} &= \sin(k/10000^{2t/d}) \\ \mathbf{p}_{i,2t+1} &= \cos(k/10000^{2t/d}) \end{cases} \quad \mathbf{p}_{\{i, 2t\}}: d\text{-차원 벡터 } \mathbf{p}_i \text{의 매 2번째 Element} \quad (4)$$

in which $\mathbf{p}_{i,2t}$ is the $2t^{th}$ element of the d-dimensional vector \mathbf{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.

2.3 Relative position embedding

The authors of [Shaw et al. [2018] applied different settings of Equation (1) as following:

$$\begin{aligned} \text{Shaw et al. [2018]의 Relative PE} \quad f_q(\mathbf{x}_m) &:= \mathbf{W}_q \mathbf{x}_m \\ f_k(\mathbf{x}_n, n) &:= \mathbf{W}_k (\mathbf{x}_n + \tilde{\mathbf{p}}_r^k) \\ f_v(\mathbf{x}_n, n) &:= \mathbf{W}_v (\mathbf{x}_n + \tilde{\mathbf{p}}_r^v) \end{aligned} \quad (5)$$

$\mathbf{p}^{\{k,v\}}_{\{r\}}$ 은 학습 가능한 상대위치 임베딩
 $r = \text{clip}(m - n, r_{\min}, r_{\max})$ 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 Equation (3), the authors [Dai et al. [2019] have proposed to decompose $q_m^\top k_n$ of Equation (2) as

각 위치를 단순히 더하기
 $\rightarrow m$ 과 n 의 거리 벡터 \mathbf{p}_r 를 만들어서 k, v 에 더하기
 $|r|$ 이 너무 크면 clip
 r 은 Position m 과 n 의 상대 위치를 의미
 r 을 clipping한 이유는, 특정 거리 이상으로 상대위치가 벌어지면, 의미가 없어지기 때문

where $\tilde{\mathbf{p}}_r^k, \tilde{\mathbf{p}}_r^v \in \mathbb{R}^d$ 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 Equation (3), the authors [Dai et al. [2019] have proposed to decompose $q_m^\top k_n$ of Equation (2) as

$$q_m^\top k_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{p}_n + \mathbf{p}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{p}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{p}_n, \quad (6)$$

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

$$q_m^\top k_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{x}_m^\top \mathbf{W}_q^\top \tilde{\mathbf{W}}_k \tilde{\mathbf{p}}_{m-n} + \mathbf{u}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{v}^\top \mathbf{W}_q^\top \tilde{\mathbf{W}}_k \tilde{\mathbf{p}}_{m-n} \quad (7)$$

It is noteworthy that the position information in the value term is removed by setting $f_v(\mathbf{x}_j) := \mathbf{W}_v \mathbf{x}_j$. Later work [Raffel et al. [2020], He et al. [2020], Ke et al. [2020], Huang et al. [2020] followed these settings by only encoding the relative position information into the attention weights. However, the authors of [Raffel et al. [2020] reformed Equation (6) as:

$$q_m^\top k_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + b_{i,j} \quad b_{\{i,j\}} \text{는 학습 가능한 bias} \quad (8)$$

where $b_{i,j}$ is a trainable bias. The authors of [Ke et al. [2020] investigated the middle two terms of Equation (6) and found little correlations between absolute positions and words. The authors of [Raffel et al. [2020] proposed to model a pair of words or positions using different projection matrices.

$$q_m^\top k_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{p}_m^\top \mathbf{U}_q^\top \mathbf{U}_k \mathbf{p}_n + b_{i,j} \quad (9)$$

식 6에서 2, 3번째 항을 살펴보면,
 절대위치정보와 단어 임베딩의 상관관계가 적었음
 따라서 각각의 다른 Projection Matrix인 $\mathbf{U}^\top_{q,k}$ 로 변환하여 각각 계산함

The authors of [He et al. [2020]] argued that the relative positions of two tokens could only be fully modeled using the middle two terms of Equation (6). As a consequence, the absolute position embeddings p_m and p_n were simply replaced with the relative position embeddings \tilde{p}_{m-n} :

$$q_m^T k_n = x_m^T W_q^T W_k x_n + x_m^T W_q^T W_k \tilde{p}_{m-n} + \tilde{p}_{m-n}^T W_q^T W_k x_n \quad (10)$$

A comparison of the four variants of the relative position embeddings [Radford and Narasimhan [2018]] has shown that the variant similar to Equation (10) is the most efficient among the other three. Generally speaking, all these approaches attempt to modify Equation (6) based on the decomposition of Equation (3) under the self-attention settings in Equation (2), which was originally proposed in [Vaswani et al. [2017]]. They commonly introduced to directly add the position information to the context representations. Unlikely, our approach aims to derive the relative position encoding from Equation (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.

GPT-1에 따르면,
다른 3개의 방법보다
Deberta의 식(10)이
가장 효율적이라고 함

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 Equation (2), $q_m^T k_n$ typically enables knowledge conveyance between tokens at different positions. In order to incorporate relative position information, we require the inner product of query q_m and key k_n to be formulated by a function g , which takes only the word embeddings x_m, x_n , and their relative position $m - n$ as input variables. In other words, we hope that the inner product encodes position information only in the relative form:

함수 g로 $q_m^T k_n$ 을 내적하는 것을 표기함

Input은 단어 임베딩 x_m, x_n 과

상대적 위치 값인 $m-n$ 2개로 함

내적에서 상대적 위치 형태로만 인코딩 되기를 기대함

$$\langle f_q(x_m, m), f_k(x_n, n) \rangle = g(x_m, x_n, m - n). \quad (11)$$

The ultimate goal is to find an equivalent encoding mechanism to solve the functions $f_q(x_m, m)$ and $f_k(x_n, n)$ to conform the aforementioned relation.

3.2 Rotary position embedding

3.2.1 A 2D case

이해하기 쉽게 d = 2를 샘플로 설명

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 Equation (11) is:

$$\begin{aligned} f_q(x_m, m) &= (W_q x_m) e^{im\theta} \\ f_k(x_n, n) &= (W_k x_n) e^{in\theta} \\ g(x_m, x_n, m - n) &= \text{Re}[(W_q x_m)(W_k x_n)^* e^{i(m-n)\theta}] \end{aligned} \quad (12)$$

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

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

where $(x_m^{(1)}, x_m^{(2)})$ is 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 amount of angle multiples of its position index and thus interprets the intuition behind Rotary Position Embedding.

단어 임베딩을 Position Index 만큼 곱하여 회전시킴

3.2.2 General form

RoPE는 복소수 회전을 통해 2차원 공간에서 쿼리와 키 벡터에 위치 정보를 곱셈하여 삽입함
따라서 전체 d차원 공간을 d/2 개의 2차원 공간으로 쪼개게 됨

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

m = 위치 인덱스값
 \mathbf{x}_m = m 위치의 단어 임베딩
 i = pair의 인덱스
 $\theta_{i,m}$ = 주파수

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

where

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

is the rotary matrix with pre-defined parameters $\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \dots, d/2]\}$. A graphic illustration of RoPE is shown in Figure (1). Applying our RoPE to self-attention in Equation (2), we obtain:

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

where $\mathbf{R}_{\Theta, n-m}^d = (\mathbf{R}_{\Theta, m}^d)^\top \mathbf{R}_{\Theta, n}^d$. Note that \mathbf{R}_{Θ}^d is an orthogonal matrix, which ensures stability during the process of encoding position information. In addition, due to the sparsity of \mathbf{R}_{Θ}^d , applying matrix multiplication directly as in Equation (16) is not computationally efficient; we provide another realization in theoretical explanation.

In contrast to the additive nature of position embedding method adopted in the previous works, i.e., Equations (3) to (10), 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.

이전의 방식과 다르게
RoPE는 더하는게 아니라
Attention에 그대로 곱함

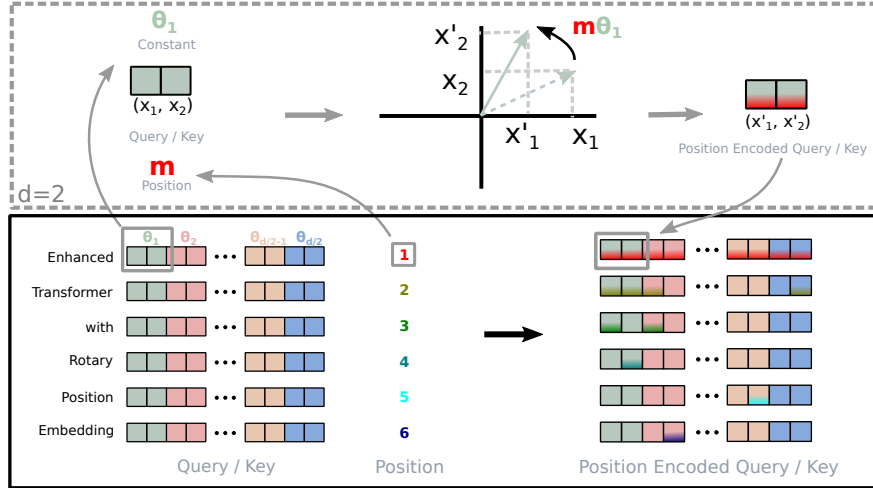


Figure 1: Implementation of Rotary Position Embedding (RoPE).

3.3 Properties of RoPE

주파수는 Transformer(2017)을 따라서 설정

Long-term decay: Following Vaswani et al. [2017], 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.

상대위치의 거리가 늘어날 수록 쿼리-키 내적값이 줄어듦
=> 거리가 먼 토큰은 연관성이 적다는 직관과 일치함

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

다른 Linear Attention에도 적용하기 쉬움

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

The original self-attention chooses $\text{sim}(\mathbf{q}_m, \mathbf{k}_n) = \exp(\mathbf{q}_m^\top \mathbf{k}_n / \sqrt{d})$. Note that the original self-attention should compute the inner product of query and key for every pair of tokens, which has a quadratic complexity $\mathcal{O}(N^2)$. Follow Katharopoulos et al. [2020], the linear attentions reformulate Equation (17) as

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

where $\phi(\cdot), \varphi(\cdot)$ are usually non-negative functions. The authors of Katharopoulos et al. [2020] 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 Shen et al. [2021] to normalize queries and keys separately before the inner product, which is equivalent to $\phi(\mathbf{q}_i) = \text{softmax}(\mathbf{q}_i)$ and $\varphi(\mathbf{k}_j) = \exp(\mathbf{k}_j)$. For more details about linear attention, 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 attention by multiplying the rotation matrix with the outputs of the non-negative functions.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_m = \frac{\sum_{n=1}^N (\mathbf{R}_{\Theta, m}^d \phi(\mathbf{q}_m))^\top (\mathbf{R}_{\Theta, n}^d \varphi(\mathbf{k}_n)) \mathbf{v}_n}{\sum_{n=1}^N \phi(\mathbf{q}_m)^\top \varphi(\mathbf{k}_n)}. \quad (19)$$

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 \mathbf{v}_i in Equation (19) 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 $\mathbf{x}_q, \mathbf{x}_k$ corresponds to query and key and their position m and n , respectively. According to eq. (1), their position-encoded counterparts are:

$$\begin{aligned} \mathbf{q}_m &= f_q(\mathbf{x}_q, m), & \text{q_m, k_n은 단어임베딩인 x_q, x_k에} \\ \mathbf{k}_n &= f_k(\mathbf{x}_k, n), & \text{위치 인덱스 m, n이 통합된 단어 + 위치 인코딩} \end{aligned} \quad (20)$$

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

$$\mathbf{q}_m^\top \mathbf{k}_n = \langle f_q(\mathbf{x}_q, m), f_k(\mathbf{x}_k, n) \rangle = g(\mathbf{x}_q, \mathbf{x}_k, n - m), \quad \begin{array}{l} \text{x_m, x_n은 단어 임베딩} \\ \text{n-m은 상대 위치 인덱스} \end{array} \quad (21)$$

we further require below initial condition to be satisfied:

$$\begin{aligned} \mathbf{q} &= f_q(\mathbf{x}_q, 0), & \text{m=n=0이므로, 위치정보가 인코딩 되지 않은 상태} \\ \mathbf{k} &= f_k(\mathbf{x}_k, 0), \end{aligned} \quad (22)$$

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 Equations (20) and (21) into:

$$\begin{aligned} f_q(\mathbf{x}_q, m) &= R_q(\mathbf{x}_q, m) e^{i\Theta_q(\mathbf{x}_q, m)}, & \text{f_q, f_k은 2차원 공간에서 벡터의 기하학적 의미와} \\ f_k(\mathbf{x}_k, n) &= R_k(\mathbf{x}_k, n) e^{i\Theta_k(\mathbf{x}_k, n)}, & \text{복소수 표현을 활용하여 식 20, 21을 식 23처럼} \\ g(\mathbf{x}_q, \mathbf{x}_k, n - m) &= R_g(\mathbf{x}_q, \mathbf{x}_k, n - m) e^{i\Theta_g(\mathbf{x}_q, \mathbf{x}_k, n - m)}, & \text{극좌표 형식으로 분해함} \\ & & \text{내적을 함으로써, R의 인수에 n-m이 포함되므로,} \\ & & \text{상대적 위치값이 자동으로 반영됨} \end{aligned} \quad (23)$$

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 Equation (21), we get the relation:

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

with the corresponding initial condition as:

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

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

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

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

$$\Theta_k(\mathbf{x}_k, m) - \Theta_q(\mathbf{x}_q, m) = \Theta_g(\mathbf{x}_q, \mathbf{x}_k, 0) = \Theta_k(\mathbf{x}_k, 0) - \Theta_q(\mathbf{x}_q, 0) = \theta_k - \theta_q. \quad (26b)$$

On one hand, from, a straightforward solution of R_f could be formed from Equation (26a):

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

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

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

Further, by plugging $n = m + 1$ to Equation (24) and consider the above equation, we can get:

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

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

$$\phi(m) = m\theta + \gamma, \quad (30)$$

where $\theta, \gamma \in \mathbb{R}$ are constants and θ is non-zero. To summarize our solutions from Equations (27) to (30):

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

Note that we do not apply any constrains to f_q and f_k of Equation (22), thus $f_q(\mathbf{x}_m, 0)$ and $f_k(\mathbf{x}_n, 0)$ are left to choose freely. To make our results comparable to Equation (3), we define:

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

Then, we simply set $\gamma = 0$ in Equation (31) of the final solution:

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

RoPE는 쿼리와 키를 복소수 회전 형태로 인코딩하며, 회전 각도는 위치에 선형적으로 비례하는 산술수열 $m\theta$ 로 모델링된다. 반면 벡터 크기는 위치에 무관하다. 이러한 구조는 상대 위치 정보를 내재하면서도 계산 안정성과 해석 가능성을 동시에 보장한다.

3.4.2 Computational efficient realization of rotary matrix multiplication

Taking the advantage of the sparsity of $\mathbf{R}_{\Theta, m}^d$ in Equation (15), a more computational efficient realization of a multiplication of \mathbf{R}_{Θ}^d and $\mathbf{x} \in \mathbb{R}^d$ is:

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

- 2차원인 (x_{2i-1}, x_{2i}) Pair로 분리하여 처리
- 각 위치마다 위치 인덱스 m 에 비례하여 회전
- 벡터화된 연산으로 torch 등에서 빠르게 가능

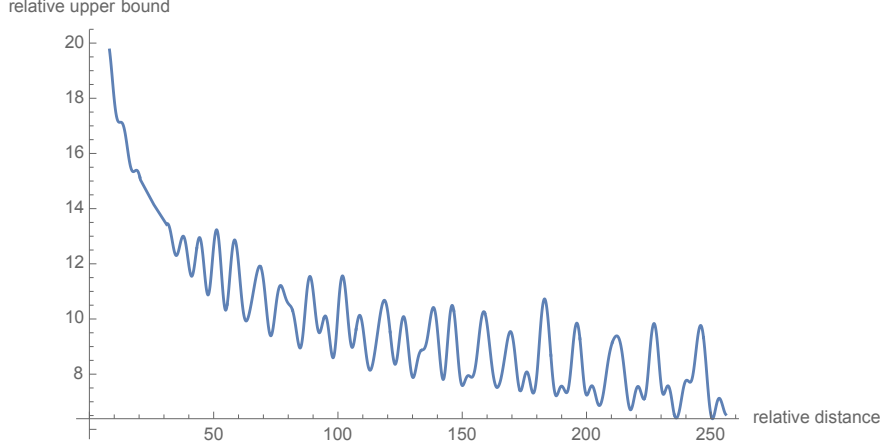


Figure 2: Long-term decay of RoPE.

3.4.3 Long-term decay of RoPE

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

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

where $\mathbf{q}_{[2i:2i+1]}$ represents the $2i^{\text{th}}$ to $(2i+1)^{\text{th}}$ entries of \mathbf{q} . Denote $h_i = \mathbf{q}_{[2i:2i+1]} \mathbf{k}_{[2i:2i+1]}^*$ 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

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

Thus,

$$\begin{aligned} \left| \sum_{i=0}^{d/2-1} \mathbf{q}_{[2i:2i+1]} \mathbf{k}_{[2i:2i+1]}^* e^{i(m-n)\theta_i} \right| &= \left| \sum_{i=0}^{d/2-1} S_{i+1} (h_{i+1} - h_i) \right| \\ &\leq \sum_{i=0}^{d/2-1} |S_{i+1}| |h_{i+1} - h_i| \\ &\leq \left(\max_i |h_{i+1} - h_i| \right) \sum_{i=0}^{d/2-1} |S_{i+1}| \end{aligned} \quad (37)$$

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 Figure (2)

쿼리와 키 두 토큰의 거리가 멀어질 수록
Attention의 값이 낮아져
멀리 떨어진 토큰간 연결성이 낮아지게 됨

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 Devlin et al. [2019] 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 Singh et al. [2018]. In Addition, we conduct experiments using the proposed RoPE with the linear attention of PerFormer Choromanski et al. [2020] in

Table 1: The proposed RoFormer gives better BLEU scores compared to its baseline alternative Vaswani et al. [2017] on the WMT 2014 English-to-German translation task Bojar et al. [2014].

Model	BLEU
Transformer-base Vaswani et al. [2017]	27.3
RoFormer	27.5

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 servers 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 Bojar et al. [2014], which consists of approximately 4.5 million sentence pairs. We compare to the transformer-based baseline alternative Vaswani et al. [2017].

4.1.2 Implementation details

We carry out some modifications on self-attention layer of the baseline model Vaswani et al. [2017] 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) Sennrich et al. [2015]. During the evaluation, a single model is obtained by averaging the last 5 checkpoints. The result uses beam search with a beam size of 4 and length penalty 0.6. We implement the experiment in PyTorch in the fairseq toolkit (MIT License) Ott et al. [2019]. Our model is optimized with the Adam optimizer using $\beta_1 = 0.9$, $\beta_2 = 0.98$, 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 Papineni et al. [2002] score on the test set as the final metric.

4.1.3 Results

We train the baseline model 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 the baseline Transformer.

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 sinusoidal position encoding of BERT with our RoPE during the pre-training step.

4.2.1 Experimental Settings

We use the BookCorpus Zhu et al. [2015] and the Wikipedia Corpus Foundation [2021] 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 Devlin et al. [2019] 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 Equation (16). We train both BERT and RoFormer with batch size 64 and maximum sequence length of 512 for 100k steps. AdamW Loshchilov and Hutter [2017] 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 Figure (3). Compare to the vanilla BERT, RoFormer experiences faster convergence.

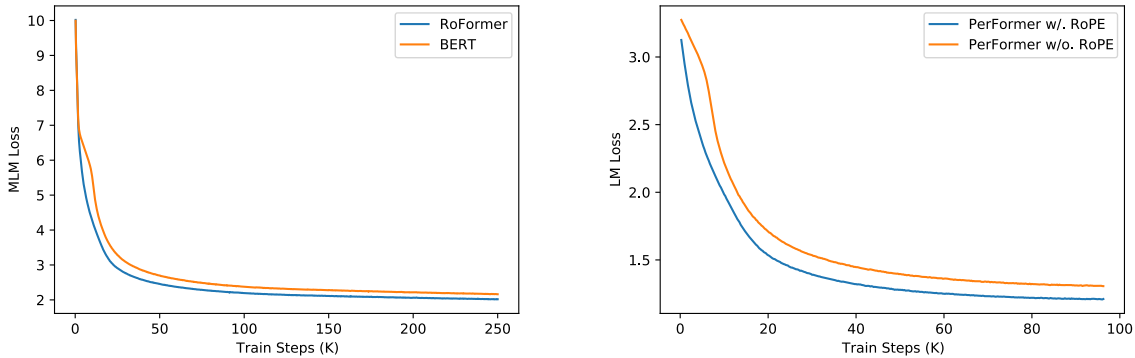


Figure 3: Evaluation of RoPE in language modeling pre-training. **Left:** training loss for BERT and RoFormer. **Right:** training loss for PerFormer with and without RoPE.

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 [Dolan and Brockett, 2005], SST-2 [Socher et al., 2013], QNLI [Rajpurkar et al., 2016], STS-B [Al-Natsheh, 2017], QQP [Chen et al., 2018b] and MNLI [Williams et al., 2018]. We use F1-score for MRPC and QQP dataset, spearman correlation for STS-B, and accuracy for the remaining as the evaluation metrics.

4.3.2 Implementation details

We use Huggingface Transformers library (Apache License 2.0 [Wolf et al., 2020]) 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 [Devlin et al., 2019], we report the best-averaged results on the validation set.

Table 2: Comparing RoFormer and BERT by fine tuning on downstream GLEU tasks.

Model	MRPC	SST-2	QNLI	STS-B	QQP	MNLI(m/mm)
BERT [Devlin et al., 2019]	88.9	93.5	90.5	85.8	71.2	84.6/83.4
RoFormer	89.5	90.7	88.0	87.0	86.4	80.2/79.8

4.3.3 Results

The evaluation results of the fine-tuning tasks are reported in Table 2. 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 [Choromanski et al., 2020] 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 [Mahoney 2006], 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 Figure (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.

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 [Su 2020] 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 [Devlin et al. 2019], WoBERT [Su 2020], and NEZHA [Wei et al. 2019], we tabulate their tokenization level and position embedding information in Table (3).

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 [Devlin et al. 2019]	WoBERT [Su 2020]	NEZHA [Wei et al. 2019]	RoFormer
Tokenization level	char	word	char	word
Position embedding	abs.	abs.	rel.	RoPE

4.5.2 Pre-training

We pre-train RoFormer on approximately 34GB 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.

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%

4.5.3 Downstream Tasks & Dataset

We choose Chinese AI and Law 2019 Similar Case Matching (CAIL2019-SCM) [Xiao et al. 2019] dataset to illustrate the ability of RoFormer in dealing with long texts, i.e., semantic text matching. CAIL2019-SCM contains 8964 triplets

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

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, the result from RoFormer is comparable to WoBERT and is slightly better than the BERT implementation. However, when increasing the maximum input text length to 1024, RoFormer outperforms WoBERT by an absolute improvement of 1.5%.

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%
RoFormer-1024	66.07%	69.79%

4.5.5 Limitations of the work

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

- Despite the fact that we mathematically format the relative position relations as rotations under 2D sub-spaces, there lacks of thorough explanations on why it converges faster than baseline models that incorporates other position encoding strategies. 왜 다른 인코딩보다 RoPE가 빠르게 수렴하는지는 모르겠음
- Although we have proved that our model has favourable 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.

다른 인코딩도 Long term Decay를 보이는데, RoPE는 긴 텍스트에서 훨씬 우수함, 그 이유를 설명못하겠음

Our proposed RoFormer is built upon the Transformer-based infrastructure, which requires hardware resources for pre-training purpose.

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 indicates that relative position can be naturally formulated using vector production in self-attention, with absolute 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.

RoPE를 통해 상대적 위치 정보를 셀프어텐션 안에서 벡터 곱으로 공식화할 수 있음

RoPE는 절대위치정보도 회전 행렬을 통해 인코딩할 수 있음 (회전행렬에 위치 인덱스만큼 회전시키기 때문)

영어/중국어 Corpus에서 실험한 결과,
- Pretraining에서 빠른 수렴
- Long Context에서 우수함
- 다양한 Downstream에서 개선을 보임

References

- 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.
- Md. Amirul Islam, Sen Jia, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? *ArXiv*, abs/2001.08248, 2020.
- 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. URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.
- A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- 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. URL <https://openreview.net/forum?id=ByxRMONTvr>.
- 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. URL <https://openreview.net/forum?id=H1eA7AEtvS>.
- 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. URL <https://openreview.net/pdf?id=r1xMH1BtvB>.
- A. Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.
- Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, 2016.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL-HLT*, 2018.
- 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.
- 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.
- Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.
- 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.
- Guolin Ke, Di He, and T. Liu. Rethinking positional encoding in language pre-training. *ArXiv*, abs/2006.15595, 2020.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654, 2020.
- 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. doi [10.18653/v1/2020.findings-emnlp.298](https://doi.org/10.18653/v1/2020.findings-emnlp.298) URL <https://www.aclweb.org/anthology/2020.findings-emnlp.298>
- 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. URL <http://proceedings.mlr.press/v119/liu20n.html>
- 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, 2018a. URL <https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html>
- 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. URL <https://openreview.net/forum?id=Hke-WTVtwr>
- 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.
- 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.

- Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. 04 2018.
- Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, A. Gane, Tamás Szepesvári, Peter Hawkins, J. Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. *ArXiv*, abs/2009.14794, 2020.
- Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleksey Tamchyna. Findings of the 2014 workshop on statistical machine translation. pages 12–58, 06 2014. doi:[10.3115/v1/W14-3302](https://doi.org/10.3115/v1/W14-3302)
- Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 08 2015.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. pages 48–53, 01 2019. doi:[10.18653/v1/N19-4009](https://doi.org/10.18653/v1/N19-4009)
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. Bleu: a method for automatic evaluation of machine translation. 10 2002. doi:[10.3115/1073083.1073135](https://doi.org/10.3115/1073083.1073135)
- Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*, 2015.
- Wikimedia Foundation. Wikimedia downloads, <https://dumps.wikimedia.org/> 2021.
- Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv e-prints*, art. arXiv:1711.05101, November 2017.
- William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL <https://www.aclweb.org/anthology/I05-5002>
- Richard Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *EMNLP*, 1631:1631–1642, 01 2013.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. pages 2383–2392, 01 2016. doi:[10.18653/v1/D16-1264](https://doi.org/10.18653/v1/D16-1264)
- Hussein Al-Natsheh. Udl at semeval-2017 task 1: Semantic textual similarity estimation of english sentence pairs using regression model over pairwise features. 08 2017.
- Z. Chen, H. Zhang, and L. Zhang, X. and Zhao. Quora question pairs., 2018b. URL <https://www.kaggle.com/c/quora-question-pairs>
- Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. pages 1112–1122, 01 2018. doi:[10.18653/v1/N18-1101](https://doi.org/10.18653/v1/N18-1101)
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander 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*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
- Matt Mahoney. Large text compression benchmark, <http://www.matmahoney.net/dc/text.html> 2006.
- Jianlin Su. Wobert: Word-based chinese bert model - zhuiyiai. Technical report, 2020. URL <https://github.com/ZhuiyiTechnology/WoBERT>
- 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.
- 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.