arXiv:2104.09864v5 [cs.CL] 8 Nov 2023

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`.

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 (RRNs) 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, 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].

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

arXiv:2104.09864v5 [cs.CL] 8 Nov 2023

RoFormer：增强型Transformer与旋转位置嵌入

苏建林 智艺科技有限公司
深圳
bojonesu@wezhuiyi.com

陆宇 智艺科技有限公司
深圳
julianlu@wezhuiyi.com

潘胜峰 智艺科技有限公司
深圳
nickpan@wezhuiyi.com

穆罕默德·穆尔塔达 智艺科技有限公司 深圳
mengjiayi@wezhuiyi.com

文波 智艺科技有限公司 深圳
brucewen@wezhuiyi.com

刘云峰 践行科技有限公司 深圳
glenliu@wezhuiyi.com

2023年11月9日

摘要

位置嵌入最近在Transformer架构中表现出有效性。它为序列中不同位置元素之间的依赖建模提供了有价值的监督。在本文中，我们首先研究了将位置信息集成到基于Transformer的语言模型学习过程中的各种方法。然后，我们提出了一种名为旋转位置嵌入（RoPE）的新方法，以有效地利用位置信息。具体来说，所提出的RoPE使用旋转矩阵对绝对位置进行编码，并同时将显式的相对位置依赖关系纳入自注意力公式中。值得注意的是，RoPE具有有价值的特性，包括序列长度的灵活性、随着相对距离的增加而衰减的词间依赖关系，以及为线性自注意力配备相对位置编码的能力。最后，我们在各种长文本分类基准数据集上评估了带有旋转位置嵌入的增强型Transformer，也称为RoFormer。我们的实验表明，它始终优于其替代方案。此外，我们提供了理论分析来解释一些实验结果。RoFormer已集成到Huggingface：
https://huggingface.co/docs/transformers/model_doc/roformer。

关键词 预训练语言模型 · 位置信息 编码 · 预训练 · 自然语言处理。

1 引言

词语的顺序对于自然语言理解非常重要。基于循环神经网络（RRNs）的模型通过沿着时间维度递归计算隐藏状态来编码标记的顺序。基于卷积神经网络（CNNs）的模型（CNNs）Gehring等人 [2017] 通常被认为是位置无关的，但最近的研究Islam等人 [2020] 表明，常用的填充操作可以隐式地学习位置信息。最近，基于Transformer Vaswani等人[2017], 构建的预训练语言模型（PLMs）在各种自然语言处理（NLP）任务（包括上下文表示学习Devlin等人 [2019], 机器翻译Vaswani等人 [2017], 和语言建模Radford等人 [2019], 等）中取得了最先进的性能。与RRNs和基于CNNs的模型不同，PLMs利用自注意力机制来语义地捕获给定语料库的上下文表示。因此，PLMs在并行化方面比RNNs有显著提升，并且与CNNs[1]相比，提高了对更长词内关系的建模能力。

[1]一个包含多个CNN层的堆叠也可以捕获更长的词内关系，这里我们仅考虑单层设置。

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.

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`.

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 $\mathbb{S}_N = \{w_i\}_{i=1}^N$ be a sequence of N input tokens with w_i being the i^{th} element. The corresponding word embedding of $\mathbb{S}_N$ 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 to the word embeddings and transforms them into queries, keys, and value representations.

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

where $\boldsymbol{q}_m, \boldsymbol{k}_n$ 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

值得注意的是，当前PLMs的自注意力架构表现出位置无关性Yun等人[2020]。基于这一观点，人们提出了各种方法将位置信息编码到学习过程中。一方面，通过预定义函数Vaswani等人[2017] 生成的绝对位置编码被添加到上下文表示中，同时还有可训练的绝对位置编码Gehring等人 [2017],Devlin等人 [2019], Lan等人 [2020], Clark等人 [2020],Radford等人 [2019], Radford和Narasimhan [2018]。另一方面，之前的工作 Parikh等人 [2016], Shaw等人 [2018], Huang等人 [2018], Dai等人 [2019], Yang等人 [2019], Raffel等人 [2020], Ke等人 [2020], He等人 [2020], Huang等人 [2020] 专注于相对位置编码，通常将相对位置信息编码到注意力机制中。除此之外，Liu等人 [2020] 的作者从神经常微分方程（Neural ODE）Chen等人 [2018a], 的角度提出了对位置编码依赖性的建模，Wang等人 [2020] 的作者则提出了在复数空间中对位置信息的建模。尽管这些方法很有效，但它们通常将位置信息添加到上下文表示中，因此不适合线性自注意力架构。

在本文中，我们介绍了一种名为旋转位置嵌入（RoPE）的新方法，将其用于将位置信息融入PLMS的学习过程中。具体而言，RoPE使用旋转矩阵对绝对位置进行编码，并同时将显式的相对位置依赖关系纳入自注意力公式中。请注意，所提出的RoPE通过其宝贵的特性优先于现有方法，包括序列长度灵活性、随着相对距离的增加而衰减的跨词依赖关系，以及为线性自注意力配备相对位置编码的能力。在各种长文本分类基准数据集上的实验结果表明，具有旋转位置嵌入的增强型Transformer，即RoFormer，与基线替代方案相比可以提供更好的性能，从而证明了所提出的RoPE的有效性。

简而言之，我们的贡献分为三个方面如下：

- 我们研究了现有的相对位置编码方法，发现它们大多基于将位置编码添加到上下文表示的分解思想构建。我们引入了一种名为旋转位置嵌入（RoPE）的新方法，以将位置信息利用到 PLMS 的学习过程中。其核心思想是通过将上下文表示与具有明确理论解释的旋转矩阵相乘来编码相对位置。

- 我们研究了RoPE的特性，并证明其随相对距离的增加而衰减，这是所期望的对于自然语言编码而言。我们善意地指出，之前的基于相对位置编码的方法与线性自注意力不兼容不兼容

- 我们在各种长文本基准数据集上评估了提出的RoFormer。我们的实验表明，它始终比其替代方案表现更好。一些使用预训练语言模型的实验可在GitHub上找到：https://github.com/ZhuiyiTechnology/roformer。

本文的其余部分组织如下。我们在第(2)节中建立了自注意力架构中位置编码问题的正式描述，并回顾了之前的工作。我们在第(3)节中描述了旋转位置编码（RoPE），并研究了它的特性。我们在第(4)节报告实验。最后，我们在第(5)节总结本文。

2 背景与相关工作

2.1 初步

设t $\mathbb{S}_N = \{w_i\}_{i=1}^N$ 为 N 输入token序列，其中 w_i 是 i^{th} 元素。 $\mathbb{S}_N$ 对应的词嵌入表示为 $\mathbb{E}_N = \{\boldsymbol{x}_i\}_{i=1}^N$，其中 $\boldsymbol{x}_i \in \mathbb{R}^d$ 是token w_i 的d维词嵌入向量（不含位置信息）。自注意力机制首先将位置信息融入词嵌入，并将它们转换为查询、键和值表示。

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

在 $\boldsymbol{q}_m$、 $\boldsymbol{k}_n$ 和 $\boldsymbol{v}_n$ 分别通过 f_q、 f_k 和 f_v 整合 m^{th} 和 n^{th} 位置时，查询值和键值被用于计算注意力权重，而输出则计算为对值的加权和。

representation.

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

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

2.2 Absolute position embedding

A typical choice of Equation (1) is

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

where $\boldsymbol{p}_i \in \mathbb{R}^d$ is a d-dimensional vector depending of the position of token $\boldsymbol{x}_i$. 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 $\boldsymbol{p}_i \in \{\boldsymbol{p}_t\}_{t=1}^{L}$, where L is the maximum sequence length. The authors of Vaswani et al. [2017] have proposed to generate $\boldsymbol{p}_i$ using the sinusoidal function.

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

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.

2.3 Relative position embedding

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

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

where $\tilde{\boldsymbol{p}}_r^k, \tilde{\boldsymbol{p}}_r^v \in \mathbb{R}^d$ are trainable relative position embeddings. Note that $r = \text{clip}(m-n, r_{\text{min}}, r_{\text{max}})$ 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 Equation (3), the authors Dai et al. [2019] have proposed to decompose $\boldsymbol{q}_m^\intercal \boldsymbol{k}_n$ of Equation (2) as

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

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:

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

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 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:

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

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.

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

表示。

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

现有的基于Transformer的位置编码方法主要集中于选择一个合适的函数来形成公式(1)。

2.2 绝对位置嵌入

公式(1)的一个典型选择是

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

其中 $\boldsymbol{p}_i \in \mathbb{R}^d$ 是一个依赖于标记 $\boldsymbol{x}_i$ 位置的 d 维向量。先前的工作 Devlin等人. [2019],Lan等人 [2020], Clark等人 [2020], Radford等人 [2019], Radford和Narasimhan [2018] 引入了使用一组可训练的向量 $\boldsymbol{p}_i \in \{\boldsymbol{p}_t\}_{t=1}^{L}$，其中 L 是最大序列长度。Vaswani等人 [2017] 的作者提出了使用正弦函数生成 $\boldsymbol{p}_i$ 。

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

其中 $\boldsymbol{p}_{i,2t}$ 是 d 维向量 $\boldsymbol{p}_i$ 的 $2t^{th}$ 元素。在下一节中，我们从正弦函数的角度展示了我们提出的RoPE与这种直觉相关。然而，RoPE并不是直接将位置添加到上下文表示中，而是通过乘以正弦函数来整合相对位置信息。

2.3 相对位置嵌入

Shaw 等人的作者 [2018] 应用了不同的公式(1)设置，如下所示：

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

其中 $\tilde{\boldsymbol{p}}_r^k$ $\tilde{\boldsymbol{p}}_r^v \in \mathbb{R}^d$ 是可训练的相对位置嵌入。请注意，$r = \text{clip}(m-n, r_{\text{min}}, r_{\text{max}})$ 表示位置 m 和 n 之间的相对距离。他们通过假设精确的相对位置信息在超过一定距离后没有用，对相对距离进行了裁剪。保持公式(3)的形式，Dai 等人.[2019] 提出将公式(2) 中的 $\boldsymbol{q}_m^\intercal \boldsymbol{k}_n$ 分解为

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

关键思想是用其正弦编码的相对对应物 $\tilde{\boldsymbol{p}}_{m-n}$，替换绝对位置嵌入 $\boldsymbol{p}_n$，同时在第三和第四项中的绝对位置 $\boldsymbol{p}_m$ 使用两个与查询位置无关的可训练向量 u 和 v。此外，$\boldsymbol{W}_k$ 用于区分基于内容和基于位置的关键向量 $\boldsymbol{x}_n$ 和 $\boldsymbol{p}_n$，记为 $\sim\boldsymbol{W}_k$ 和 $\boldsymbol{W}_k$，结果为：

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

值得注意的是，通过设置 $f_v(\boldsymbol{x}_j) := \boldsymbol{W}_v \boldsymbol{x}_j$ 会移除值项中的位置信息。后续工作中 Raffel 等人 [2020], He 等人 [2020], Ke 等人 [2020], Huang 等人 [2020] 遵循这些设置，仅将相对位置信息编码到注意力权重中。然而，Raffel 等人 [2020] 改进了方程 (6) 如下：

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

$b_{i,j}$ 在哪里，是一个可训练的偏差。Ke 等人的作者 [2020] 研究了方程式(6)中间的两个项，发现绝对位置和词之间几乎没有相关性。Raffelet 等人的作者 [2020] 提出使用不同的投影矩阵来对词语或位置对进行建模。

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

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 $\boldsymbol{p}_m$ and $\boldsymbol{p}_n$ were simply replaced with the relative position embeddings $\tilde{\boldsymbol{p}}_{m-n}$:

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

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.

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), $\boldsymbol{q}_m^\intercal \boldsymbol{k}_n$ typically enables knowledge conveyance between tokens at different positions. In order to incorporate relative position information, we require the inner product of query $\boldsymbol{q}_m$ and key $\boldsymbol{k}_n$ to be formulated by a function g, which takes only the word embeddings $\boldsymbol{x}_m$, $\boldsymbol{x}_n$, and their relative position $m-n$ as input variables. In other words, we hope that the inner product encodes position information only in the relative form:

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

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 Equation (11) is:

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

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

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

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 amount of angle multiples of its position index and thus interprets the intuition behind *Rotary Position Embedding*.

He 等人的作者 [2020] 认为，两个标记的相对位置只能使用方程式(6)中间的两个项来完全建模。因此，绝对位置嵌入 $\boldsymbol{p}_m$ 和 $\boldsymbol{p}_n$ 被简单地替换为相对位置嵌入 $\tilde{\boldsymbol{p}}_{m-n}$:

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

Radford 和 Narasimhan [2018] 对相对位置嵌入的四种变体的比较表明，类似于公式(10)的变体在其余三种变体中效率最高。一般来说，所有这些方法都试图基于公式(3)在公式(2)中自注意力设置下的分解来修改公式(6)，该公式最初由 Vaswani 等人 [2017] 提出。它们通常引入直接将位置信息添加到上下文表示中。不同地，我们的方法旨在在某些约束下从公式(1)中推导出相对位置编码。接下来，我们通过结合相对位置信息与上下文表示的旋转，展示推导出的方法更具可解释性。

3 拟议方法

在本节中，我们讨论拟议的旋转位置嵌入（RoPE）。我们首先在 3.1 节中提出相对位置编码问题，然后在 3.2 节中推导出 RoPE，并在 3.3 节中研究其性质。

3.1 公式化

基于Transformer的语言建模通常通过自注意力机制利用单个token的位置信息。如式(2)所示， $\boldsymbol{q}_m^\intercal \boldsymbol{k}_n$ 通常能够实现不同位置token之间的知识传递。为了整合相对位置信息，我们需要将查询$\boldsymbol{q}_m$ 和键 $\boldsymbol{k}_n$ 的内积通过函数 g进行定义，该函数仅以词嵌入 $\boldsymbol{x}_m$、 $\boldsymbol{x}_n$及其相对位置 $m-n$ 作为输入变量。换句话说，我们希望内积仅以相对形式编码位置信息:

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

最终目标是在求解函数 $f_q(\boldsymbol{x}_m, m)$ 和 $f_k(\boldsymbol{x}_n, n)$ 时，使其符合上述关系，找到一个等效的编码机制。

3.2 旋转位置嵌入

3.2.1 二维情况

我们从维度为 $d = 2$的简单情况开始。在这些设置下，我们利用二维平面上向量的几何性质及其复数形式来证明（更多细节请参考第3.4.1节）我们的公式化方程(11)的解为:

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

其中 $\mathrm{Re}[\cdot]$ 是复数的实部， $(\boldsymbol{W}_k \boldsymbol{x}_n)^*$ 表示 $(\boldsymbol{W}_k \boldsymbol{x}_n)$ 的共轭复数。$\theta \in \mathbb{R}$ 是预设的非零常数。我们可以进一步将 $f_{\{q,k\}}$ 写成乘法矩阵:

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

$(x_m^{(1)}, x_m^{(2)})$ 在哪里以二维坐标表示。类似地， g 可以被视为一个矩阵，因此在二维情况下启用了第(3.1)节的公式求解。具体来说，结合相对位置嵌入很简单：只需将仿射变换后的词嵌入向量旋转其位置索引的角位移倍数，从而解释了旋转位置嵌入背后的直觉。

3.2.2 General form

In order to generalize our results in 2D to any $\boldsymbol{x}_i \in \mathbb{R}^d$ 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:

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

where

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

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

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

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 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.

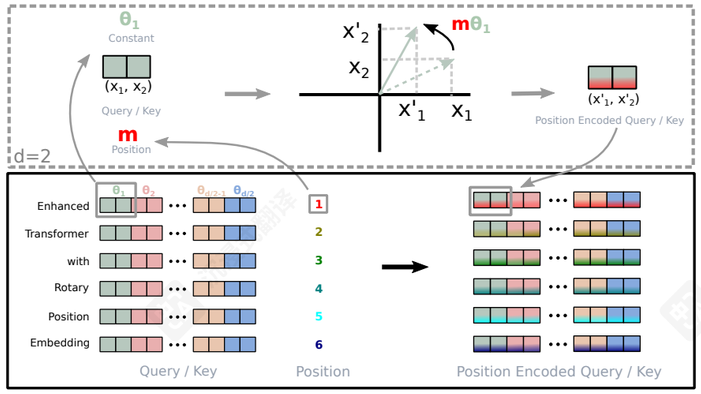

Figure 1: Implementation of Rotary Position Embedding(RoPE).

3.3 Properties of RoPE

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.

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

3.2.2 一般形式

为了将我们在二维中的结果推广到任何 $\boldsymbol{x}_i \in \mathbb{R}^d$，其中 d 是偶数，我们将d维空间划分为 $d/2$个子空间，并利用内积的线性特性将它们组合起来，将 $f_{\{q,k\}}$ 转换为：

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

其中

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

是预定义参数的旋转矩阵 $\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, ..., d/2]\}$。RoPE的图形说明如图(1)所示。将我们的RoPE应用于方程(2)中的自注意力，我们得到：

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

在$\boldsymbol{R}^d_{\Theta\ n-m} = (\boldsymbol{R}^d_{\Theta,m})^\intercal\boldsymbol{R}^d_{\Theta\ n}$，请注意， $\boldsymbol{R}^d_\Theta$ 是一个正交矩阵，这确保了在编码位置信息的过程中稳定性。此外，由于 R^d_Θ的稀疏性，直接按照公式(16)进行矩阵乘法在计算上并不高效；我们在理论解释中提供了另一种实现方式。

与先前工作中采用的加性位置嵌入方法（即公式(3)至(10)）不同，我们的方法具有乘性特性。此外，RoPE通过旋转矩阵乘积自然地包含了相对位置信息，而不是在加性位置编码的扩展公式中修改项，当与自注意力一起使用时。

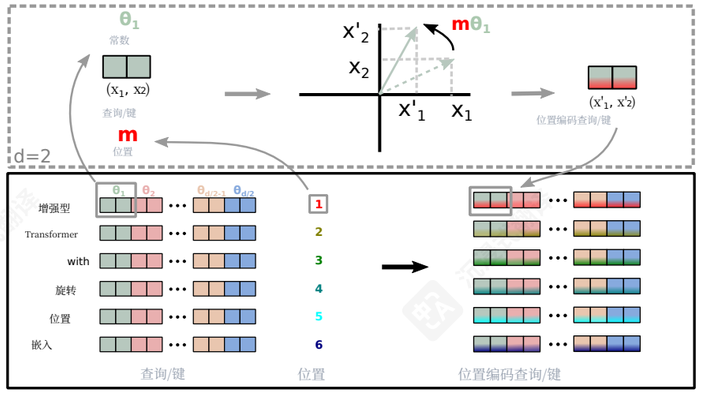

图1：旋转位置嵌入(RoPE)的实现。

3.3 RoPE的性质

长期衰减：遵循Vaswani等人 [2017]，，我们设置 $\theta_i = 10000^{-2i/d}$。可以证明，这种设置具有长期衰减特性（更多细节请参考第(3.4.3)节），这意味着当相对位置增加时，内积将衰减。这一特性与直观理解一致，即距离较远的两个token之间应该具有较弱的连接。

RoPE 线性注意力：自注意力可以表示为更一般的形式。

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

The original self-attention chooses $\text{sim}(\boldsymbol{q}_m,\boldsymbol{k}_n) = \exp(\boldsymbol{q}_m^\intercal \boldsymbol{k}_n/\sqrt{d})$. 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 $\mathbb{O}(N^2)$. Follow Katharopoulos et al. [2020], the linear attentions reformulate Equation (17) as

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

where $\phi(\cdot),\varphi(\cdot)$ are usually non-negative functions. 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 $\phi(\boldsymbol{q}_i) = \text{softmax}(\boldsymbol{q}_i)$ and $\phi(\boldsymbol{k}_j) = \exp(\boldsymbol{k}_j)$. 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.

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

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

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

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 $f_{\{q,k\}}$:

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

we further require below initial condition to be satisfied:

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

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:

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

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

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

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

原始自注意力选择 $\text{sim}(\boldsymbol{q}_m,\boldsymbol{k}_n) = \exp(\boldsymbol{q}_m^\intercal \boldsymbol{k}_n/\sqrt{d})$。注意原始自注意力需要计算查询和键对每个 token 对的 内积，这具有二次复杂度 $\mathbb{O}(N^2)$。遵循 Katharopoulos 等人 [2020], ，线性注意力将公式 (17) 重新表述为

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

在 $\phi(\cdot)$ $\varphi(\cdot)$ 中， $\phi(\cdot)$ $\varphi(\cdot)$ 通常是非负函数。Katharopoulos 等人的作者 [2020] 提出了 $\phi(x) = \varphi(x) = \text{elu}(x)+1$，并首次使用矩阵乘法的结合律计算了键和值之间的乘积。Shen 等人 [2021] 使用 softmax 函数在内积之前分别归一化查询和键，这等效于 $\phi(\boldsymbol{q}_i) = \text{softmax}(\boldsymbol{q}_i)$ 和 $\phi(\boldsymbol{k}_j) = \exp(\boldsymbol{k}_j)$。有关线性注意力的更多详细信息，我们鼓励读者参考原始论文。在本节中，我们专注于讨论将 RoPE 与公式 (18) 结合。由于 RoPE 通过旋转注入位置信息，同时保持隐藏表示的范数不变，因此我们可以通过将旋转矩阵与非负函数的输出相乘，将 RoPE 与线性注意力相结合。

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

值得注意的是，我们保持分母不变以避免除零风险，分子中的求和可能包含负项。尽管公式 (19) 中每个值 $\boldsymbol{v}_i$ 的权重并非严格概率归一化，但我们恳切地认为，计算仍然可以模拟值的重要性。

3.4 理论解释

3.4.1 在二维下的 RoPE 推导

在 $d = 2$ 的情况下，我们考虑两个词嵌入向量 $\boldsymbol{x}_q$、 $\boldsymbol{x}_k$ 分别对应查询和键，以及它们的位置 m 和 n。根据等式 (1)，它们的位置编码形式为：

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

其中 $\boldsymbol{q}_m$ 和 $\boldsymbol{k}_n$ 的下标表示编码的位置信息。假设存在一个函数 g 定义了由 $f_{\{q,k\}}$: 生成的向量之间的内积。

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

我们进一步要求满足以下初始条件：

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

这可以理解为对具有空位置信息的向量进行编码。在给定这些设置的情况下，我们尝试找到 f_q、 f_k 的解。首先，我们利用二维中向量的几何意义及其复数对应部分，将等式 (20) 和 (21) 中的函数分解为：

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

其中 R_f、 R_g 和 Θ_f、 Θ_g 分别是 $f_{\{q,k\}}$ 和 g 的模长和角度分量。将它们代入公式(21)，我们得到该关系式：

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

with the corresponding initial condition as:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note that we do not apply any constrains to f_q and f_k of Equation (22), 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 Equation (3), we define:

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

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

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

3.4.2 Computational efficient realization of rotary matrix multiplication

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

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

其对应的初始条件为：

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

其中 $\|\boldsymbol{q}\|$、$\|\boldsymbol{k}\|$ 和 θ_q、θ_k 是 $\boldsymbol{q}$ 和 $\boldsymbol{k}$ 在二维平面上的径向和角度部分。

接下来，我们在公式(24)中设置 $m = n$，并考虑公式(25)中的初始条件：

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

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

一方面，从...，R_f 的直接解可以由公式(26a)形成：

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

这解释了径向函数 R_q、R_k 和 R_g 与位置信息无关。另一方面，如公式(26b)中所示，$\Theta_q(\boldsymbol{x}_q, m) - \theta_q = \Theta_k(\boldsymbol{x}_k, m) - \theta_k$ 表明角度函数不依赖于查询和键，我们将其设置为 $\Theta_f := \Theta_q = \Theta_k$，而项 $\Theta_f(\boldsymbol{x}_{\{q,k\}}, m) - \theta_{\{q,k\}}$ 是位置 m 的函数，且与词嵌入 $\boldsymbol{x}_{\{q,k\}}$ 无关，我们将其表示为 $\phi(m)$，得到：

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

此外，将 $n = m + 1$ 代入公式(24) 并考虑上述方程，我们可以得到：

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

由于右侧(RHS)是一个与 m 无关的常数，$\phi(m)$ 具有连续整数输入时会产生等差数列：

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

其中 θ $\gamma \in \mathbb{R}$、θ 是常数，且 θ 不为零。总结我们从公式(27) 到公式(30) 的解：

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

注意，我们没有对公式(22) 中的 f_q 和 f_k 施加任何约束，因此 $f_q(\boldsymbol{x}_m, 0)$ 和 $f_k(\boldsymbol{x}_n, 0)$ 可以自由选择。为了使我们的结果与公式(3) 相比，我们定义：

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

然后，我们在最终解的公式(31) 中简单地设置 $\gamma = 0$：

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

3.4.2 旋转矩阵乘法的计算高效实现

利用 $\boldsymbol{R}^d_{\Theta\ m}$ 的稀疏性，在公式 (15) 中，R^d_Θ 和 $\boldsymbol{x} \in \mathbb{R}^d$ 的乘积的一种更计算高效的实现是：

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

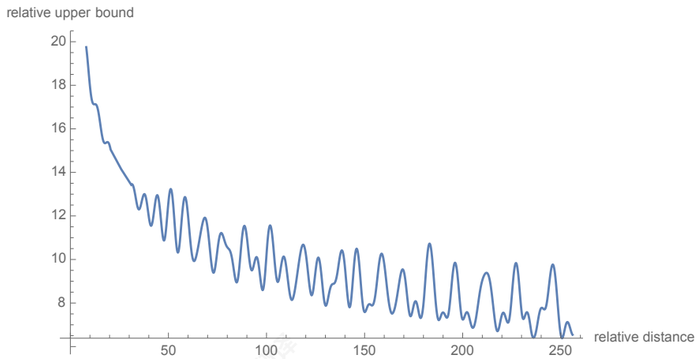

Figure 2: Long-term decay of RoPE.

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 Equation (16) can be written as a complex number multiplication.

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

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

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

Thus,

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

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)**.

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

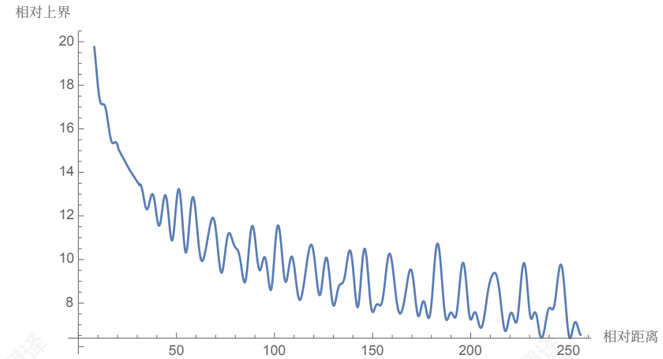

图2：RoPE的长期衰减

3.4.3 RoPE的长期衰减

我们可以将向量 $\boldsymbol{q} = \boldsymbol{W}_q \boldsymbol{x}_m$ 和 $\boldsymbol{k} = \boldsymbol{W}_k \boldsymbol{x}_n$ 的元素成对分组，公式 (16) 中 RoPE 的内积可以写成一个复数乘法。

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

其中$\boldsymbol{q}_{[2i:2i+1]}$ 表示 $\boldsymbol{q}$的 $2i^{th}$ 到$2i+1^{th}$ 的条目。记 $h_i = \boldsymbol{q}_{[2i:2i+1]} \boldsymbol{k}^*_{[2i:2i+1]}$ 和 $S_j = \sum_{i=0}^{j-1} e^{i(m-n)\theta_i}$，并令 $h_{d/2} = 0$和 $S_0 = 0$，我们可以使用阿贝尔变换重写求和

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

因此，

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

请注意，$\frac{1}{d/2} \sum_{i=1}^{d/2} |S_i|$ 衰减值随着相对距离 $m-n$ 的增加而通过设置 $\theta_i = 10000^{-2i/d}$ 而变化，如图 (2) 所示。

4 实验与评估

我们评估了提出的 RoFormer 在各种 NLP 任务上的表现。我们验证了该方案在机器翻译任务（第 4.1 节）上的性能。然后，我们在预训练阶段（第 4.2 节）将我们的 RoPE 实现与 BERTDevlin 等人 [2019] 进行比较。基于预训练模型，在第 4.3 节，我们从 GLUE 基准（Singh 等人 [2018]）的不同下游任务进行进一步评估。此外，我们在第 4.4 节使用提出的 RoPE 与 PerFormer Choromanski 等人 [2020] 的线性注意力进行了实验。

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 taskBojar et al. [2014].

Model	BLEU
Transformer-baseVaswani 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 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 datasetBojar 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 BLEUPapineni 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.

表1：所提出的RoFormer与基线替代方案Vaswani等人 [2017]相比，在WMT 2014英德翻译任务上给出了更好的BLEU分数 [2014]。

模型BLEU	
Transformer-baseVaswani等人 [2017]	27.3
RoFormer	**27.5**

最后，我们在第 4.5 节包含了针对中文数据的额外测试。所有实验均在两台云服务器上运行，配备 4 x V100GPU。

4.1 机器翻译

我们首先展示了RoFormer在序列到序列语言翻译任务上的性能。

4.1.1 实验设置

我们选择了标准的WMT 2014英德数据集Bojar等人 [2014], ，该数据集包含约450万句对。我们将RoFormer与基于Transformer的基线模型Vaswani等人 [2017]进行比较。

4.1.2 实现细节

我们对基线模型Vaswani等人 [2017] 的自注意力层进行了一些修改，以使RoPE能够融入其学习过程。我们基于联合源语言和目标语言字节对编码(BPE)Sennrich等人 [2015]，使用3.7k词汇量，复现了英语到德语的翻译设置。在评估期间，通过平均最后5个检查点来获得单个模型。结果使用beam size为4、长度惩罚为0.6的beam search。我们在fairseq工具包（MIT许可证）Ott等人 [2019]中用PyTorch实现了该实验。我们的模型使用Adam优化器进行优化，参数为 $\beta_1 = 0.9$, $\beta_2 = 0.98$，学习率从 $1e-7$ 线性增加到 $5e-4$，然后按步数的平方根倒数比例衰减。我们还采用了0.1的标签平滑。我们在测试集上报告了BLEU-Papineni等人 [2002] 分数作为最终指标。

4.1.3 结果

我们在相同设置下训练基线模型和我们的RoFormer，并在表(1)中报告结果。如图所示，我们的模型在BLEU分数上优于基线Transformer模型。

4.2 预训练语言建模

第二个实验是为了验证我们方案在学习上下文表示方面的性能。为此，我们在预训练步骤中将BERT的原始正弦位置编码替换为我们的RoPE。

4.2.1 实验设置

我们使用Huggingface数据集库（Apache License 2.0）中的BookCorpus Zhu等人 [2015] 和维基百科语料库Foundation [2021] 进行预训练。语料库进一步按8:2的比例分割为训练集和验证集。我们使用训练过程中的掩码语言模型（MLM）损失值作为评估指标。我们采用著名的BERT Devlin等人 [2019]作为我们的基线模型。请注意，我们在实验中使用bert-base-uncased。

4.2.2 实现细节

对于 RoFormer，我们将基线模型中自注意力模块的正弦位置编码替换为我们的 RoPE，并根据公式 (16) 实现自注意力。我们使用批大小 64 和最大序列长度 512 对 BERT 和 RoFormer 进行 100k 步的训练。AdamW Loshchilov 和 Hutter [2017] 作为优化器，学习率为 1e-5。

4.2.3 结果

预训练过程中的MLM损失显示在图(3)的左图上。与vanilla BERT相比，RoFormer经历了更快的收敛。

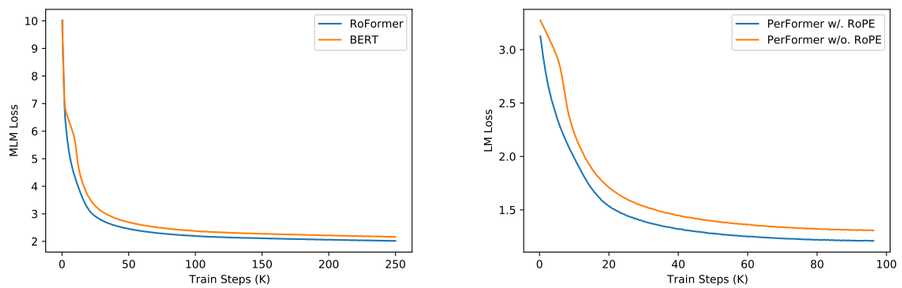

Figure 3: Evaluation of RoPE in language modeling pre-training. **Left**: training loss for BERT and RoFormer. **Right**: training loss for PerFormer with and without RoPE.

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)
BERTDevlin 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.

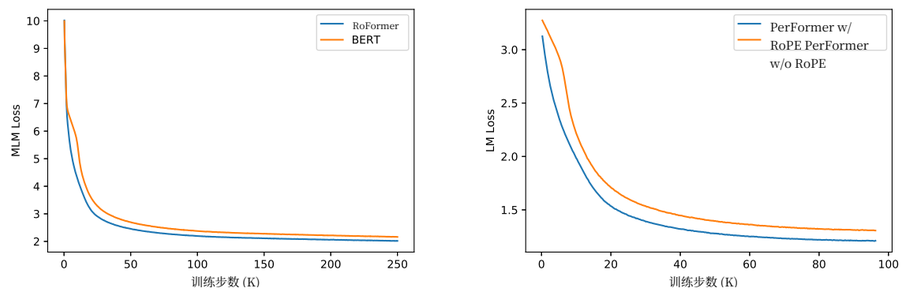

图 3：RoPE 在语言建模预训练中的评估。左：BERT 和 RoFormer 的训练损失。右：带和不带 RoPE 的 PerFormer 训练损失。

4.3 在GLUE任务上微调

与之前的实验一致，我们通过在各种GLUE任务上微调我们预训练的RoFormer的权重，以评估其在下游自然语言处理任务上的泛化能力。

4.3.1 实验设置

我们考察了GLUE中的多个数据集，即MRPC Dolan和Brockett [2005], SST-2 Socher等人 [2013], QNLI Rajpurkar等人 [2016], STS-B Al-Natsheh [2017],QQP Chen等人. [2018b]以及MNLI Williams等人 [2018]。我们使用F1分数作为MRPC和QQP数据集的评估指标，使用Spearman相关性作为STS-B的评估指标，使用准确率作为其余数据集的评估指标。

4.3.2 实现细节

我们使用Huggingface Transformers库（Apache License 2.0）Wolf等人 [2020] 对上述所有下游任务进行微调，每个任务微调3个epoch，最大序列长度为512，批大小为32，学习率为2,3,4,5e-5。遵循Devlin等人 [2019],的做法，我们在验证集上报告了最佳平均结果。

表2：通过在下游GLEU任务上微调比较RoFormer和BERT。

模型	MRPC	SST-2	QNLI	STS-B	QQP	MNLI(m/mm)
BERTDevlin等人 [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 结果

微调任务的评估结果报告在表(2)中。从表中可以看出，RoFormer在六个数据集中的三个上显著优于BERT，改进效果相当可观。

4.4 带有RoPE的Performer

Performer Choromanski等人 [2020] 提出了一种替代的注意力机制，线性注意力，旨在避免与输入序列长度成二次方增长的计算成本。如第(3.3)节所述，所提出的RoPE可以很容易地在PerFormer模型中实现，以实现相对位置编码，同时保持其自注意力中线性缩放的复杂度。我们通过语言建模的预训练任务展示了其性能。

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	BERTDevlin et al. [2019]	WoBERTSu [2020]	NEZHAWei 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

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

4.4.1 实现细节

我们对来自英语维基百科的 Enwik8数据集 Mahoney [2006]，进行测试，该数据集包含标记、特殊字符以及除英语文本之外的其他语言文本。我们将 RoPE 整合到具有 768 维度和 12 个头的 12 层字符级 PerFormer 中 [2]。为了更好地说明 RoPE 的有效性，我们在相同设置下报告了带有和没有 RoPE 的预训练过程的损失曲线，即学习率 1e-4、批大小 128 以及固定的最大序列长度为 1024 等。

4.4.2 结果

如图3右侧所示，将RoPE替换到Performer中，在相同训练步数下可快速收敛并降低损失。这些改进，加上其线性复杂度，使Performer更具吸引力。

4.5 中文数据评估

除了在英文数据上的实验，我们还展示了中文数据上的额外结果。为了验证RoFormer在长文本上的性能，我们在长度超过512个字符的长文档上进行了实验。

4.5.1 实现

在这些实验中，我们对 WoBERT Su [2020] 进行了一些修改，将其绝对位置嵌入替换为我们的 RoPE。为了与其他中文预训练 Transformer 模型进行比较，即 BERT Devlin 等人 [2019], 、WoBERT Su [2020], 和 NEZHA Wei 等人 [2019], ，我们在表 (3) 中列出了它们的分词级别和位置嵌入信息。

表 3：我们的 RoFormer 与其他中文预训练模型之间的交叉比较。'abs' 和 'rel' 分别标注绝对位置嵌入和相对位置嵌入。

模型	BERTDevlin 等人 [2019]	WoBERTSu [2020]	NEZHAWei 等人. [2019]	RoFormer
分词级别	char	word	char	word
位置嵌入	abs.	abs.	rel.	RoPE

4.5.2 预训练

我们在中文维基百科、新闻和论坛收集的约 34GB 数据上预训练 RoFormer。预训练分多阶段进行，批大小和最大输入序列长度不断变化，以使模型适应各种场景。如表(4)所示，随着序列长度上限的增加，RoFormer 的准确率提升，这展示了 RoFormer 处理长文本的能力。我们认为，这正是 RoPE 具有出色泛化能力的属性。

表4：RoFormer在中文数据集上的预训练策略。训练过程被划分为多个连续阶段。在每个阶段中，我们使用特定的最大序列长度和批大小组合来训练模型。

阶段	最大序列长度	批大小	训练步数	Loss	准确率
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 下游任务 & 数据集

我们选择中文AI与法律2019相似案例匹配（CAIL2019-SCM）Xiao等人. [2019] 数据集来展示RoFormer处理长文本的能力，即语义文本匹配。CAIL2019-SCM包含8964个三元组

[2]对于此实验，我们采用来自 https://github.com/lucidrains/performer-pytorch 的代码（MIT 许可证）

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.
- 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.

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 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.

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*,

，这些案例是由最高人民法院发布的。输入三元组，记作(A, B和C)，是三个案例的事实描述。任务是预测在预定义的相似度度量下，对(A, B)是否比(A, C)更接近。请注意，由于文档长度（即大多超过512个字符），现有方法在CAIL2019-SCM数据集上表现大多不明显。我们根据众所周知的6:2:2比例划分训练、验证和测试集。

4.5.4 结果

我们将预训练的 RoFormer 模型应用于 CAIL2019-SCM，并使用不同的输入长度。该模型与在相同预训练数据上预训练的 BERT 和 WoBERT 模型进行了比较，如表5所示。在短文本截断的情况下，即512，RoFormer 的结果与 WoBERT 相当，并且略优于 BERT 实现。然而，当将最大输入文本长度增加到1024时，RoFormer 的表现优于 WoBERT，绝对提升为1.5%。

表5：CAIL2019-SCM 任务上的实验结果。第一列中的数字表示最大截断序列长度。结果以百分比准确率表示。

模型	验证	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 工作局限性

尽管我们提供了理论依据以及有前景的实验证明，但我们的方法受限于以下事实：

- 尽管我们从数学上以二维子空间下的旋转形式对相对位置关系进行格式化，但对于为什么它比包含其他位置编码策略的基线模型收敛更快，缺乏充分的解释。
- 尽管我们已经证明我们的模型对于内部词对产品具有长期衰减的有利特性（3.3节），这类似于现有的位置编码机制，但我们的模型在长文本上的表现优于同行模型，我们尚未提出一个可靠的解释。

我们提出的RoFormer基于Transformer基架构，这需要硬件资源用于预训练目的。

5 结论

在这项工作中，我们提出了一种新的位置嵌入方法，该方法在自注意力机制中融入了显式的相对位置依赖，以提升Transformer架构的性能。我们的理论分析表明，相对位置可以自然地通过自注意力中的向量积来表述，而绝对位置信息则通过旋转矩阵进行编码。此外，我们通过数学方法展示了所提出的方法在应用于Transformer时的优势特性。最后，在英语和中文基准数据集上的实验表明，我们的方法能够促进预训练过程中的更快收敛。实验结果还显示，我们提出的RoFormer在长文本任务上能够取得更好的性能。

参考文献

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=ByxRM0Ntvr.

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. 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.

RoFormer

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=ByxRM0Ntvr.

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. 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.

13

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 Likhosherstov, David Dohan, Xingyou Song, A. Gane, Tamás Sarlós, 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 Alevs Tamchyna. Findings of the 2014 workshop on statistical machine translation. pages 12–58, 06 2014. doi: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.

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.

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.

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.

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.mattmahoney.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.

RoFormer

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 Likhosherstov, David Dohan, Xingyou Song, A. Gane, Tamás Sarlós, 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 Alevs Tamchyna. Findings of the 2014 workshop on statistical machine translation. pages 12–58, 06 2014. doi: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.

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.

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.

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.

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.mattmahoney.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.

14