Recurrent Positional Embedding for Neural Machine Translation

Kehai Chen Rui Wang Masao Utiyama Eiichiro Sumita
{khchen,wangrui,mutiyama,eiichiro.sumita}@nict.go.jp
National Institute of Information and Communications Technology (NICT), Kyoto, Japan

1. Motivation

Learning Sentence Representation

- Transformer translation systems rely on the position index of each word to learn a positional embedding to encode order information.
- Positional embeddings are then added to corresponding word embeddings to form a input representation.
- The input representation allows Transformer translation systems to parallel learn the sentence representation with order dependencies.

Several Observations

- The input representation only involves static order dependencies based on discrete numerical information. That is, any word in the vocabulary has the same positional embedding on the same position index.
- In addition, the order dependencies encoded by the existing positional embedding are independent of word contents, which may further hinder the improvement of translation capacity.

2. Recurrent Positional Embedding

- The sequence of word vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_J\}$ are splited into $\{\mathbf{x}^p_1, \cdots, \mathbf{x}^p_J\}$ and $\{\mathbf{x}^r_1, \cdots, \mathbf{x}^r_J\}$ and their dimensions are d_p and d_r (d_{model}=d_p+d_r), respectively.
- An RNN with an non-linear projection layer is used to learn its recurrent positional embedding (**RPE**) $\mathbf{r}_j$ for each word over $\{\mathbf{x}^r_1, \cdots, \mathbf{x}^r_J\}$, and gains a sequence of RPE $\boldsymbol{R}$ =$\{\mathbf{r}_1, \cdots, \mathbf{r}_J\}$:

$$\mathbf{r}_j = \mathrm{Tanh}(f_{\mathrm{RNN}}(\mathbf{x}^r_j, \mathbf{r}_{j-1})\mathbf{W}^r + \mathbf{b}^r). \quad (1)$$

- The traditional positional encoding is used to learn a reduced dimension input representation $\boldsymbol{P}$ =$\{\mathbf{p}_1, \cdots, \mathbf{p}_J\}$ for other sub-sequence $\{\mathbf{x}^p_1, \cdots, \mathbf{x}^p_J\}$:

$$\begin{aligned}\mathbf{pe}_{(j,2i)} &= \sin(j/10000^{2i/d_p}),\\ \mathbf{pe}_{(j,2i+1)} &= \cos(j/10000^{2i/d_p}).\end{aligned} \quad (2)$$

$$\mathbf{p}_j = \mathbf{x}^p_j + \mathbf{pe}_j. \quad (3)$$

3. Methods

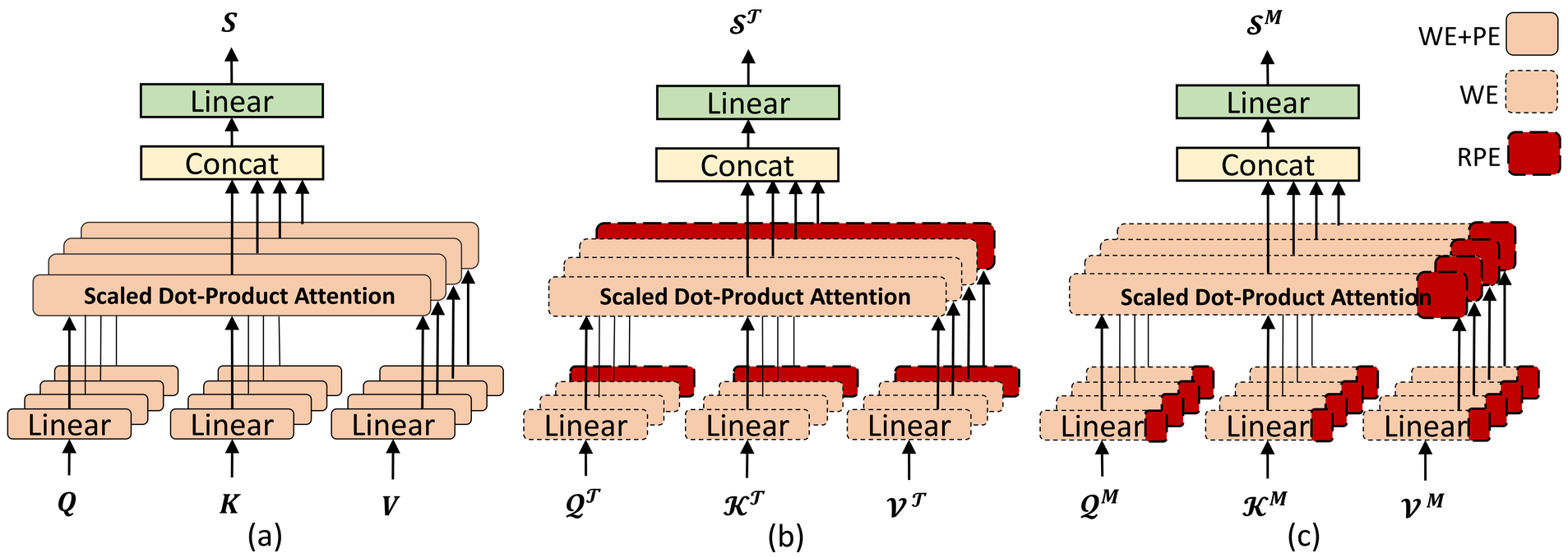


Fig 1: (a) Vanilla Multi-Head Self-Attention; (b) RPEHead Self-Attention; (c) MPRHead Self-Attention.

RPEHead Self-Attention

- These learned RPEs are integrated into multi-head self-attention (Fig 1(a)) as several independent heads to learn the sentence representation, as shown in Fig 1(b). To guarantee that there are two types of heads over the combined input representation $\boldsymbol{\mathcal{T}}$, d_r and d_p are set to t*d_{model}/H and d_{model}-d_r, respectively.

$$\begin{aligned}\boldsymbol{\mathcal{T}} &= \{\boldsymbol{P} : \boldsymbol{R}\},\\ \boldsymbol{\mathcal{O}}^{\mathcal{T}}_h &= \mathrm{Att}(\boldsymbol{\mathcal{Q}}^{\mathcal{T}}\boldsymbol{W}^{\mathcal{Q}}_h, \boldsymbol{\mathcal{K}}^{\mathcal{T}}\boldsymbol{W}^{\mathcal{K}}_h, \boldsymbol{\mathcal{V}}^{\mathcal{T}}\boldsymbol{W}^{\mathcal{V}}_h),\\ \boldsymbol{\mathcal{S}}^{\mathcal{T}} &= \mathrm{Concat}(\boldsymbol{\mathcal{O}}^{\mathcal{T}}_1, \cdots, \boldsymbol{\mathcal{O}}^{\mathcal{T}}_{H-1}, \boldsymbol{\mathcal{O}}^{\mathcal{T}}_H)\boldsymbol{\mathcal{W}}^O,\end{aligned} \quad (4)$$

where $\{\boldsymbol{\mathcal{Q}}^{\mathcal{T}}, \boldsymbol{\mathcal{K}}^{\mathcal{T}}, \boldsymbol{\mathcal{V}}^{\mathcal{T}}\}$ depends on $\boldsymbol{\mathcal{T}}$.

MPRHead Self-Attention

- Compared with the RPEHead with independent RPE Head, the MPRHead mixed $\boldsymbol{P}$ and $\boldsymbol{R}$ into a combined input representation $\boldsymbol{M}$ for learning the sentence representation $\boldsymbol{\mathcal{S}}^{\boldsymbol{M}}$, as shown in Fig 1(c).

$$\begin{aligned}\mathbf{m}_j &= \{\mathbf{p}^1_j : \mathbf{r}^1_j, \cdots, \mathbf{p}^H_j : \mathbf{r}^H_j\},\\ \boldsymbol{M} &= \{\mathbf{m}_1, \cdots, \mathbf{m}_J\},\\ \boldsymbol{\mathcal{O}}^{\boldsymbol{M}}_h &= \mathrm{Att}(\boldsymbol{\mathcal{Q}}^{\boldsymbol{M}}\boldsymbol{W}^{\mathcal{Q}}_h, \boldsymbol{\mathcal{K}}^{\boldsymbol{M}}\boldsymbol{W}^{\mathcal{K}}_h, \boldsymbol{\mathcal{V}}^{\boldsymbol{M}}\boldsymbol{W}^{\mathcal{V}}_h),\\ \boldsymbol{\mathcal{S}}^{\boldsymbol{M}} &= \mathrm{Concat}(\boldsymbol{\mathcal{O}}^{\boldsymbol{M}}_1, \cdots, \boldsymbol{\mathcal{O}}^{\boldsymbol{M}}_{H-1}, \boldsymbol{\mathcal{O}}^{\boldsymbol{M}}_H)\boldsymbol{\mathcal{W}}^O,\end{aligned} \quad (5)$$

where $\{\boldsymbol{\mathcal{Q}}^{\boldsymbol{M}}, \boldsymbol{\mathcal{K}}^{\boldsymbol{M}}, \boldsymbol{\mathcal{V}}^{\boldsymbol{M}}\}$ depends on $\boldsymbol{M}$.

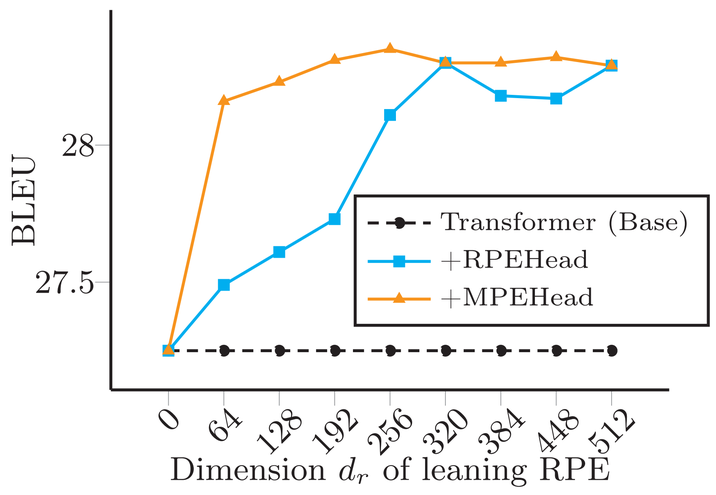
Experiments

Main Results

Model	NIST ZH-EN				WMT14 EN-DE	
	MT02	MT03	MT04	#Param	newstst14	#Param
Transformer(Base)	46.62	45.90	45.95	77.94	27.25	97.35M
+Relative PEs	46.91†	46.47†	45.96†	77.01M†	27.60†	97.42M
+RPEHead	47.06†	46.21†	46.32†	78.04M†	28.11†	97.84M
+MPRHead	47.69†	47.27†	47.31†	78.08†	28.35†	97.72M
Transformer(Big)	47.88	47.58	47.37	243.7M	28.22	272.6M
+MPRHead	48.73†	48.61†	48.21†	245.5M †	29.11†	289.1M

"†" indicates statistically better than Transformer(Base/Big) ($\rho < 0.01$).
"#Param" denotes the trainable parameter size of NMT model.

Efficiency Analysis for RPEs



Ablation Experiments

	newstest2014	#Speed
Transformer (Base)	27.25	11.7K
-PE of Dec	26.73	11.7K
-PE of Enc	10.71	11.7K
RPEHead (base)	28.11	10.5K
-RPE&PE of Dec	27.54	11.2K
-RPE&PE of Enc	11.16	10.7K
MPRHead (base)	28.35	9.9K
-RPE&PE of Dec	27.74	11.2K
-RPE/PE of Enc	10.89	10.7K
-RPE&PE of Enc&Dec	10.43	11.7K

"#Speed" denotes the training speed (tokens/second).