Neural Machine Translation with Reordering Embeddings

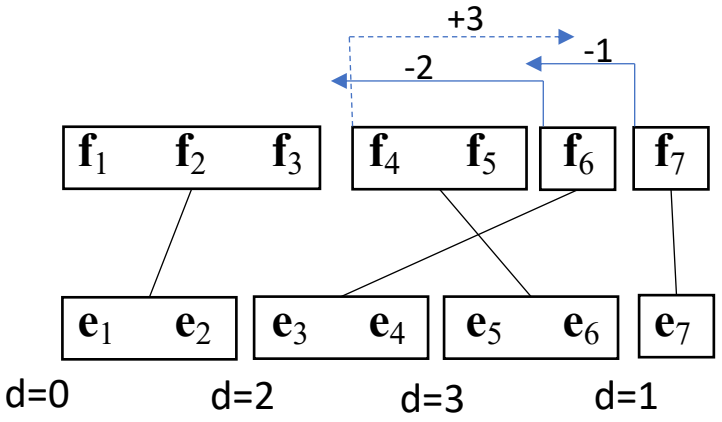
Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita

National Institute of Information and Communications Technology, Kyoto, Japan

2019.07.29

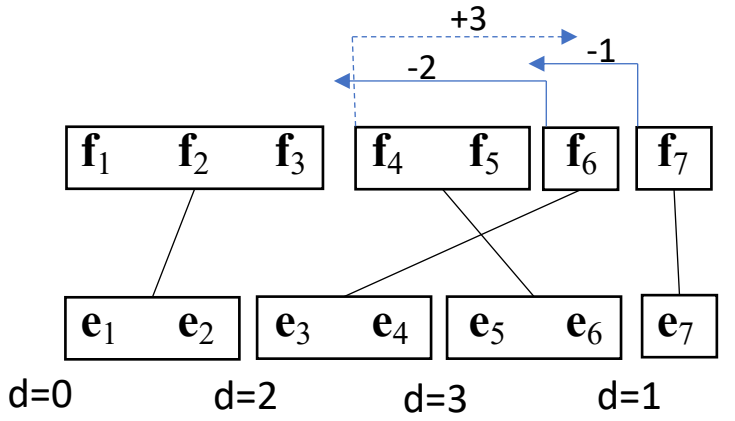
Introduction

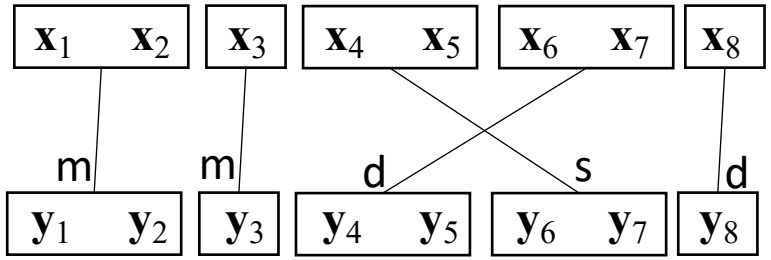
Reordering information in SMT



Introduction

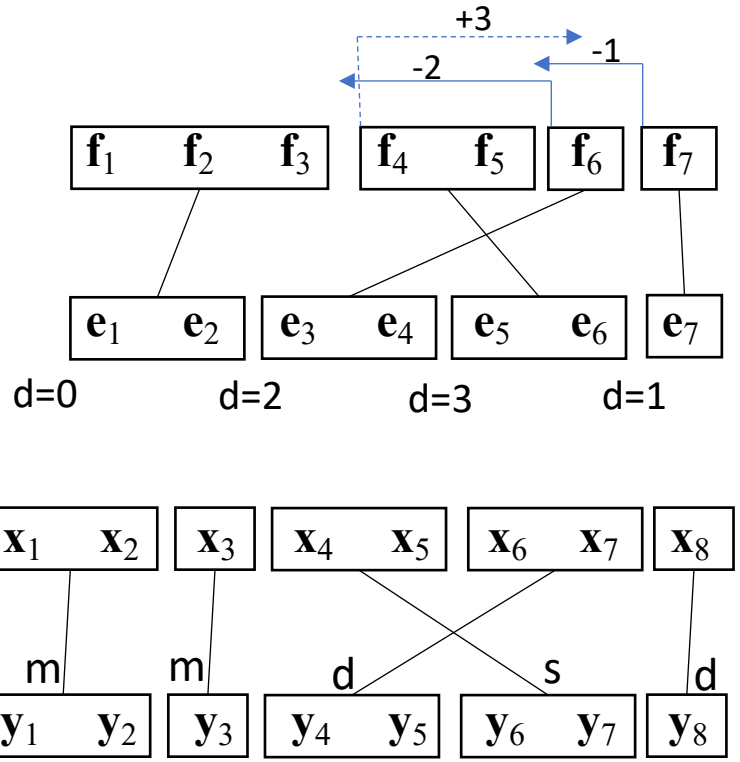
Reordering information in SMT





Introduction

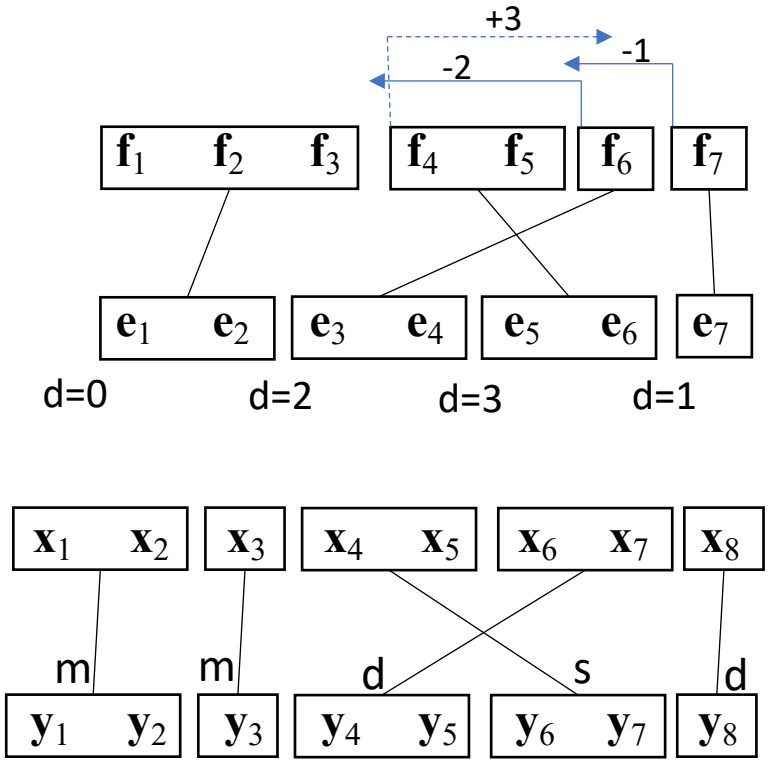
Reordering information in SMT



- Learning large-scale reordering rules in advance
- Depending on bilingual parallel sentence pair with “hard” word alignments

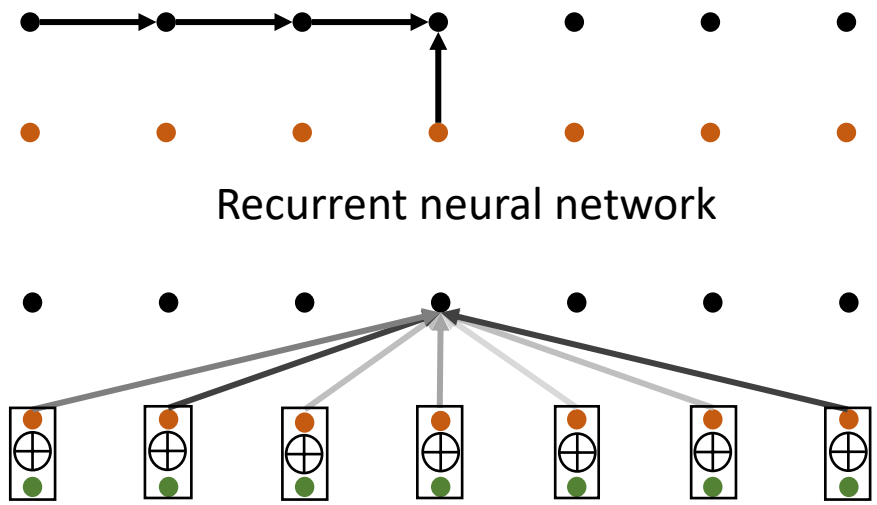
Introduction

Reordering information in SMT



- Learning large-scale reordering rules in advance
- Depending on bilingual parallel sentence pair with “hard” word alignments

Ordering information in NMT



Self-attention network with positional embeddings

$$\mathbf{pe}_{(j,2i)} = \sin(j/10000^{2i/d_{model}})$$

$$\mathbf{pe}_{(j,2i+1)} = \cos(j/10000^{2i/d_{model}})$$

- Depending on neural network itself or positional encoding mechanism to encode order sequentially
- Only “soft” attention alignment

Related Work

Pre-ordering source sentence (Goto et al., 2013)

Japanese: *kara wa kinou hon wo katta*

Pre-odering: *kara wa katta hon wo kinou*

English: *he bought books yesterday*

Pre-ordered each source sentence into the similar order as its target sentence

Related Work

Pre-ordering source sentence (Goto et al., 2013)

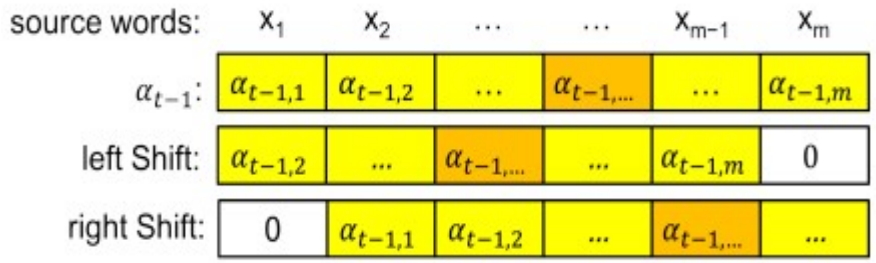
Japanese: *kara wa kinou hon wo katta*

Pre-odering: *kara wa katta hon wo kinou*

English: *he bought books yesterday*

Reordered each source sentence into the similar order as its target sentence

Additional position-based attention (Zhang et al., 2017)



$$\alpha_t = \lambda \cdot \mathbf{d}_t + (1-\lambda)\hat{\alpha}_t$$

$$\mathbf{d}_t = E[\Gamma(\alpha_{t-1})] = \sum_{k=-l}^{l} P(k|\Psi) \cdot \Gamma(\alpha_{t-1}, k)$$

$$\Gamma(\alpha_{t-1}, k) = \begin{cases} \{\alpha_{t-1,-k}, \ldots, \alpha_{t-1,m}, 0, \ldots, 0\}, & \text{k<0} \\ \alpha_{t-1}, & \text{k= 0} \\ \{0, \ldots, 0, \alpha_{t-1,1}, \ldots, \alpha_{t-1,m-k}\}, & \text{k>0} \end{cases}$$

Related Work

Pre-ordering source sentence (Goto et al., 2013)

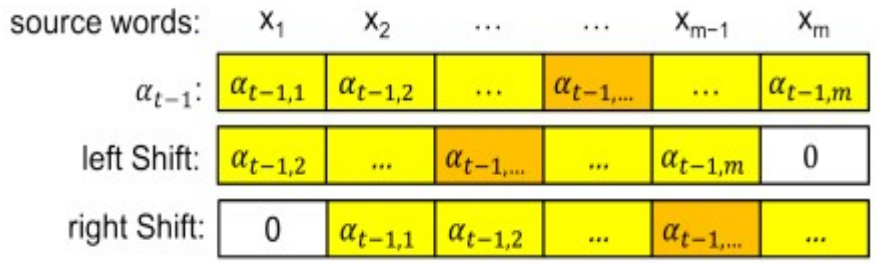
Japanese: *kara wa kinou hon wo katta*

Pre-odering: *kara wa katta hon wo kinou*

English: *he bought books yesterday*

Reordered each source sentence into the similar order as its target sentence

Additional position-based attention (Zhang et al., 2017)



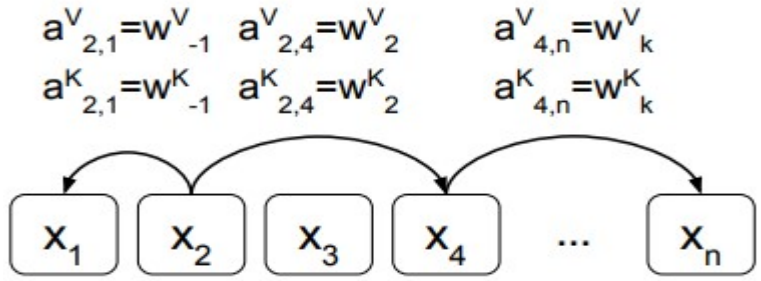
$$\alpha_t = \lambda \cdot \mathbf{d}_t + (1-\lambda)\hat{\alpha}_t$$

$$\mathbf{d}_t = E[\Gamma(\alpha_{t-1})] = \sum_{k=-l}^{l} P(k|\Psi) \cdot \Gamma(\alpha_{t-1}, k)$$

$$\Gamma(\alpha_{t-1}, k) = \begin{cases} \{\alpha_{t-1,-k}, \ldots, \alpha_{t-1,m}, 0, \ldots, 0\}, & \text{k<0} \\ \alpha_{t-1}, & \text{k= 0} \\ \{0, \ldots, 0, \alpha_{t-1,1}, \ldots, \alpha_{t-1,m-k}\}, & \text{k>0} \end{cases}$$

Relative position representation (Shaw et al., 2018)

2 <= k <= n − 4



$$a_{ij}^K = w_{\text{clip}(j-i,k)}^K$$

$$a_{ij}^V = w_{\text{clip}(j-i,k)}^V$$

$$\text{clip}(x,k) = \max(-k, \min(k,x))$$

$$w^K = (w_{-k}^K, \ldots, w_k^K) \text{ and } w^V = (w_{-k}^V, \ldots, w_k^V)$$

Motivation

In Transformer framework, the existing positional encoding focuses on sequentially encoded order relations between words, and does not explicitly consider reordering information in a sentence.

Motivation

In Transformer framework, the existing positional encoding focuses on sequentially encoded order relations between words, and does not explicitly consider reordering information in a sentence.

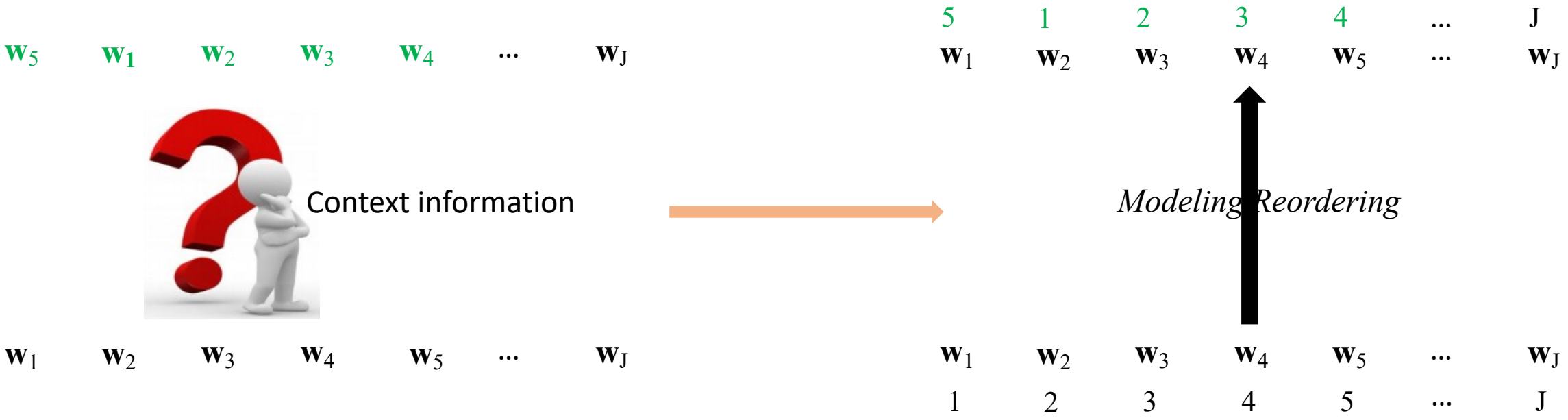
w_5 w_1 w_2 w_3 w_4 ... w_J

Context information

w_1 w_2 w_3 w_4 w_5 ... w_J

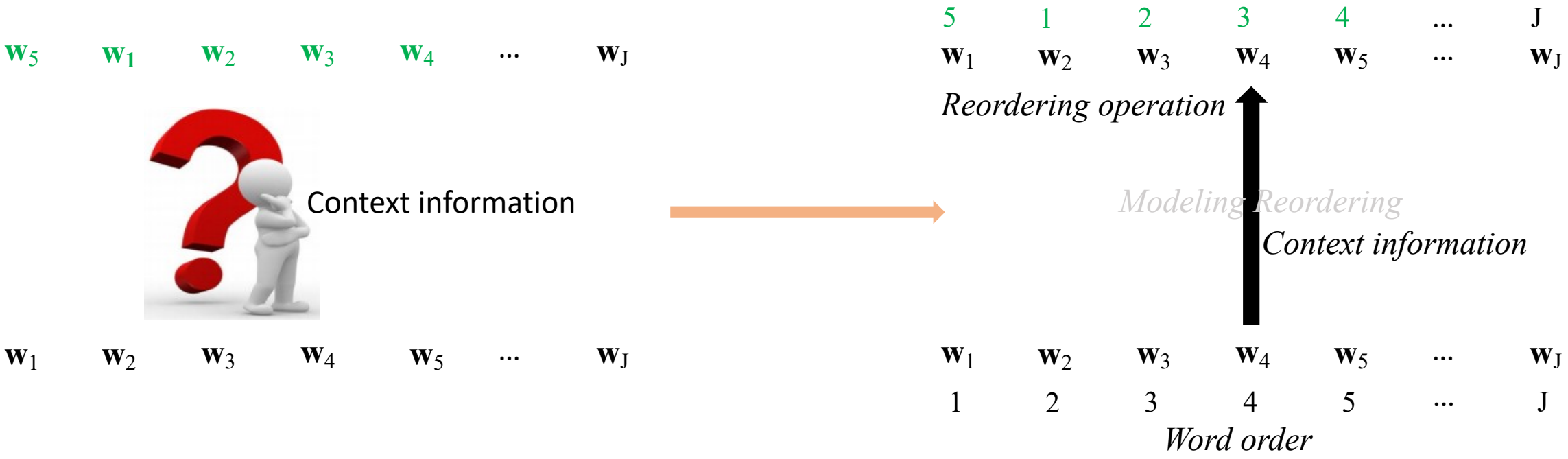
Motivation

In Transformer framework, the existing positional encoding focuses on sequentially encoded order relations between words, and does not explicitly consider reordering information in a sentence.



Motivation

In Transformer framework, the existing positional encoding focuses on sequentially encoded order relations between words, and does not explicitly consider reordering information in a sentence.



Modeling the human reordering processing:

- Word order
- Context information
- Reordering operation

Reordering Embeddings

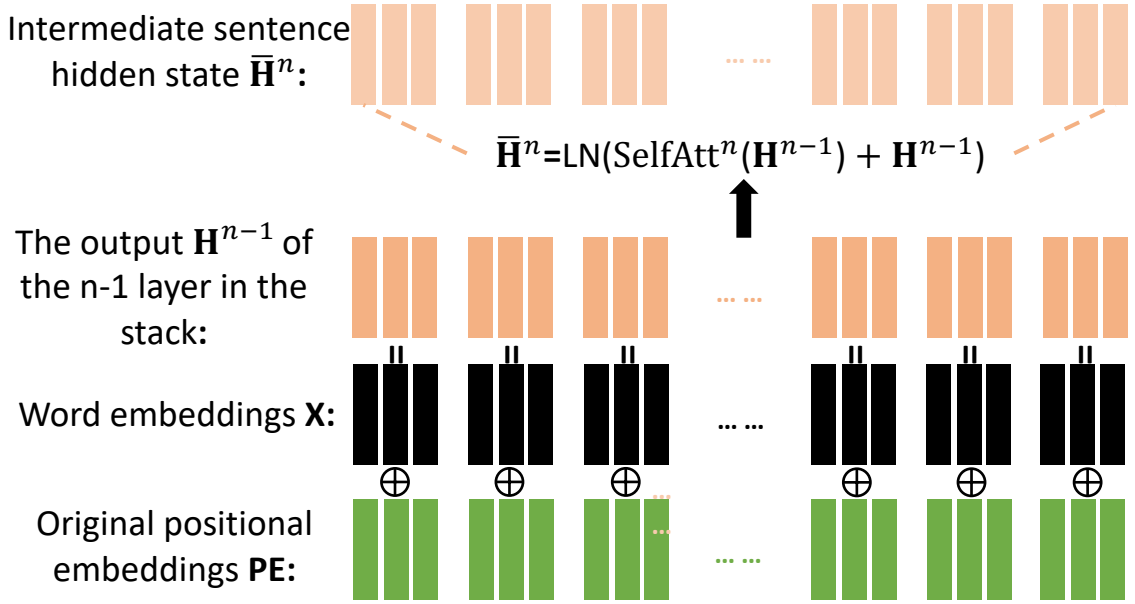
Reordering Embeddings

Original positional embeddings **PE:**

Original positional embeddings **PE**

Word order

Reordering Embeddings



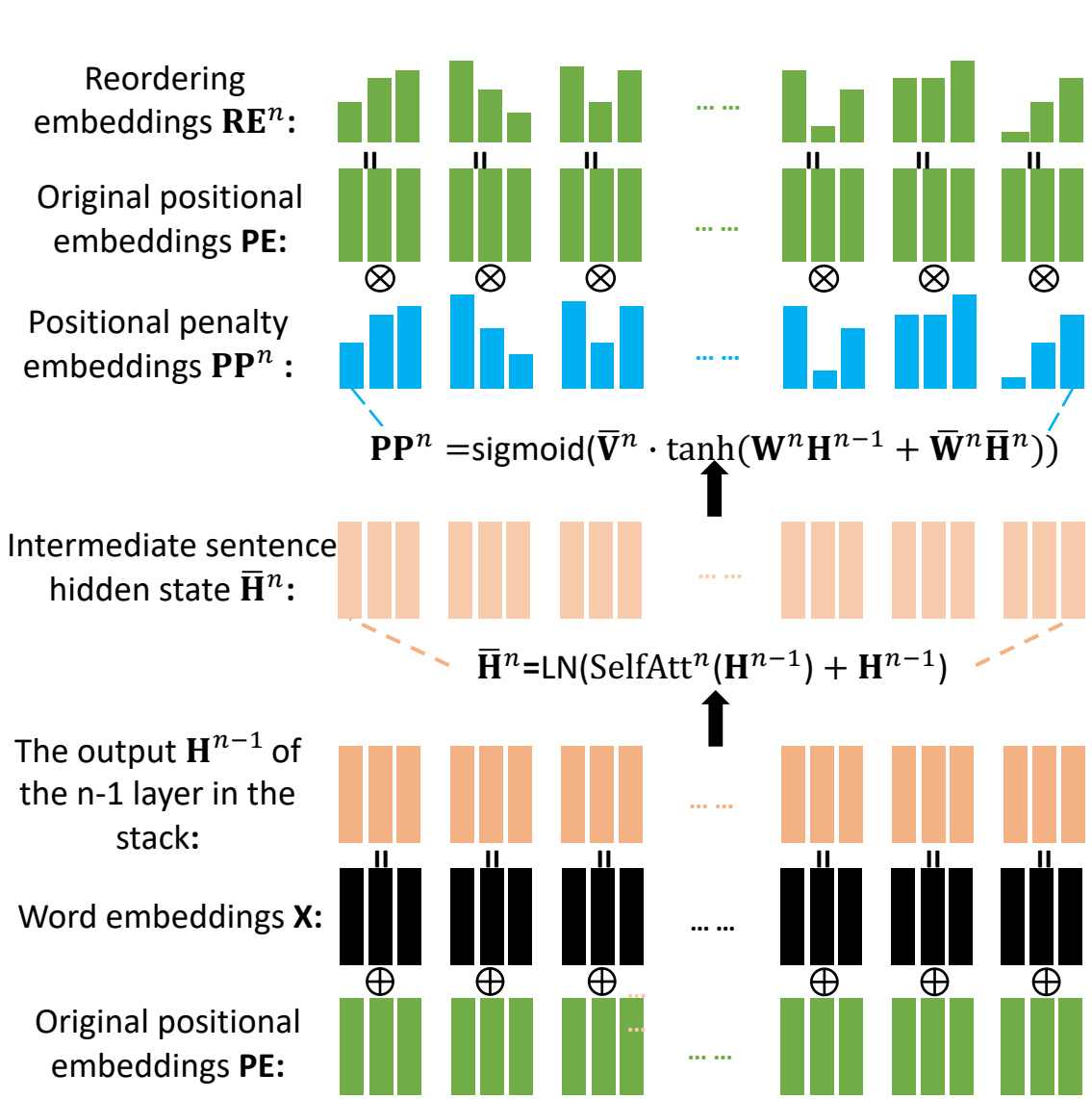
$\bar{\mathbf{H}}^n$=LN(SelfAtt^n($\mathbf{H}^{n-1}$) + $\mathbf{H}^{n-1}$)

Context information

Original positional embeddings **PE**

Word orders

Reordering Embeddings



$\mathbf{RE}^n$=$\mathbf{PE} \cdot \mathbf{PP}^n$

Model Reordering

$\mathbf{PP}^n$ =sigmoid($\bar{\mathbf{V}}^n \cdot \tanh(\mathbf{W}^n\mathbf{H}^{n-1} + \bar{\mathbf{W}}^n\bar{\mathbf{H}}^n)$)

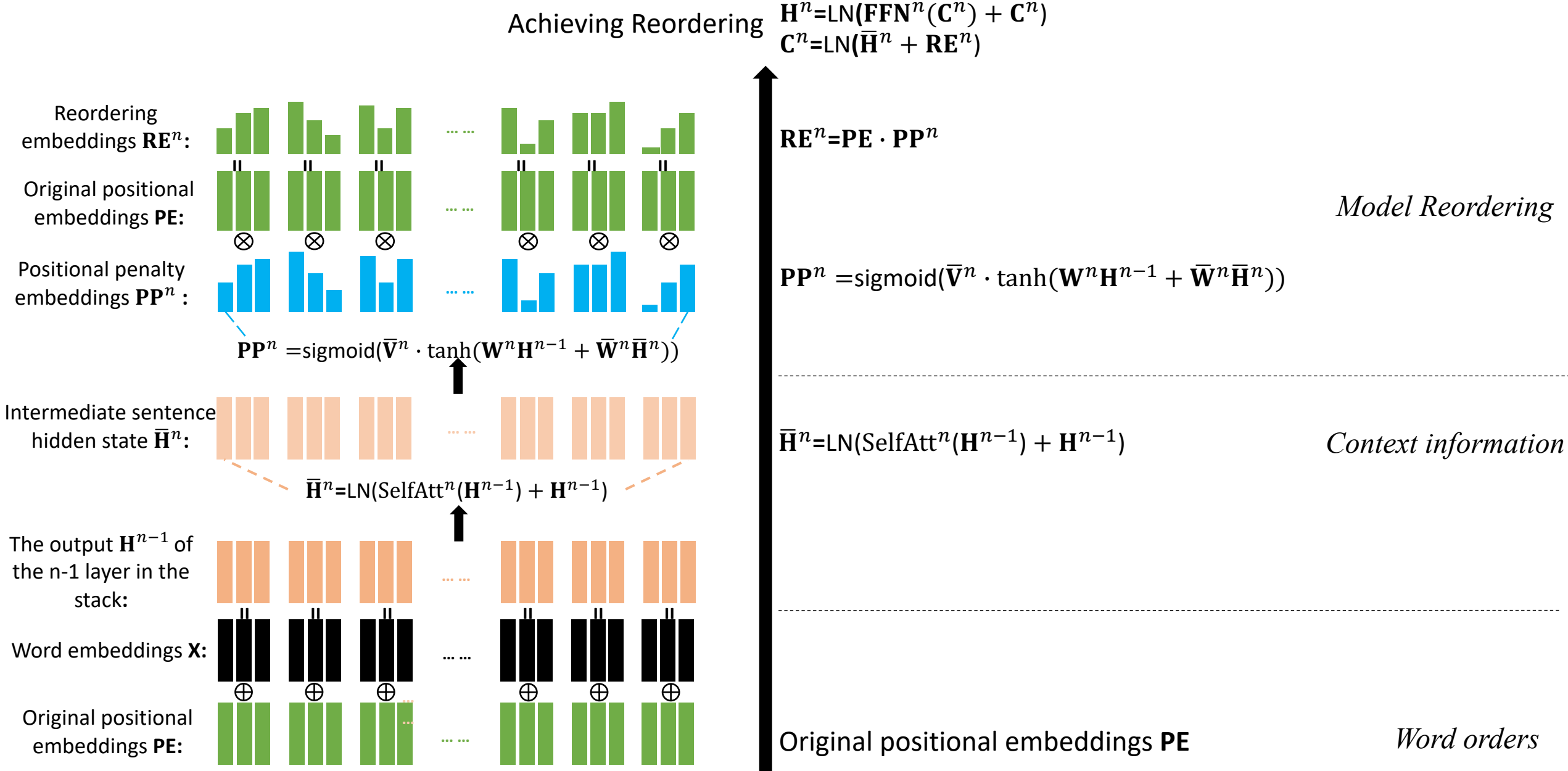
$\bar{\mathbf{H}}^n$=LN(SelfAtt^n($\mathbf{H}^{n-1}$) + $\mathbf{H}^{n-1}$)

Context information

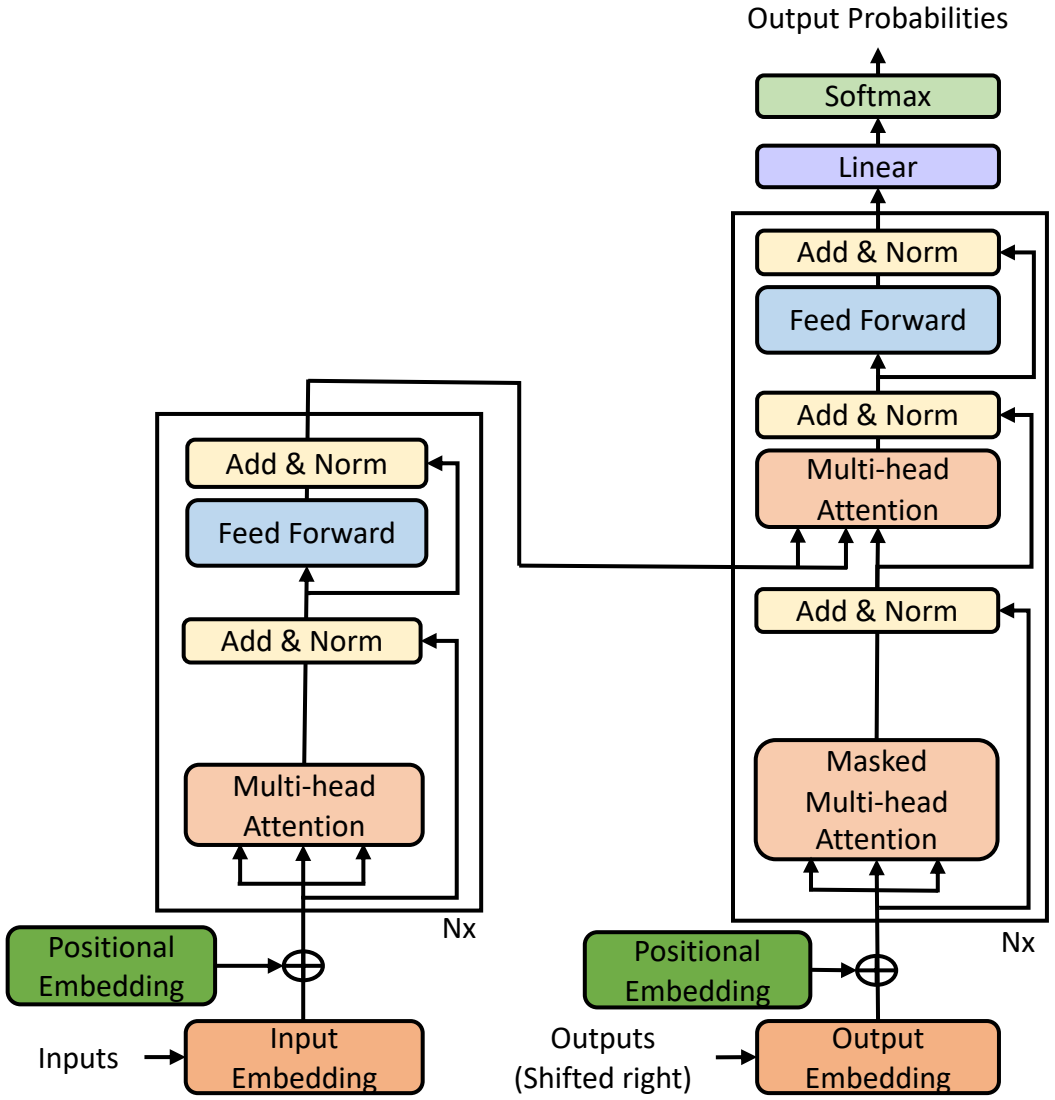
Original positional embeddings **PE**

Word orders

Reordering Embeddings



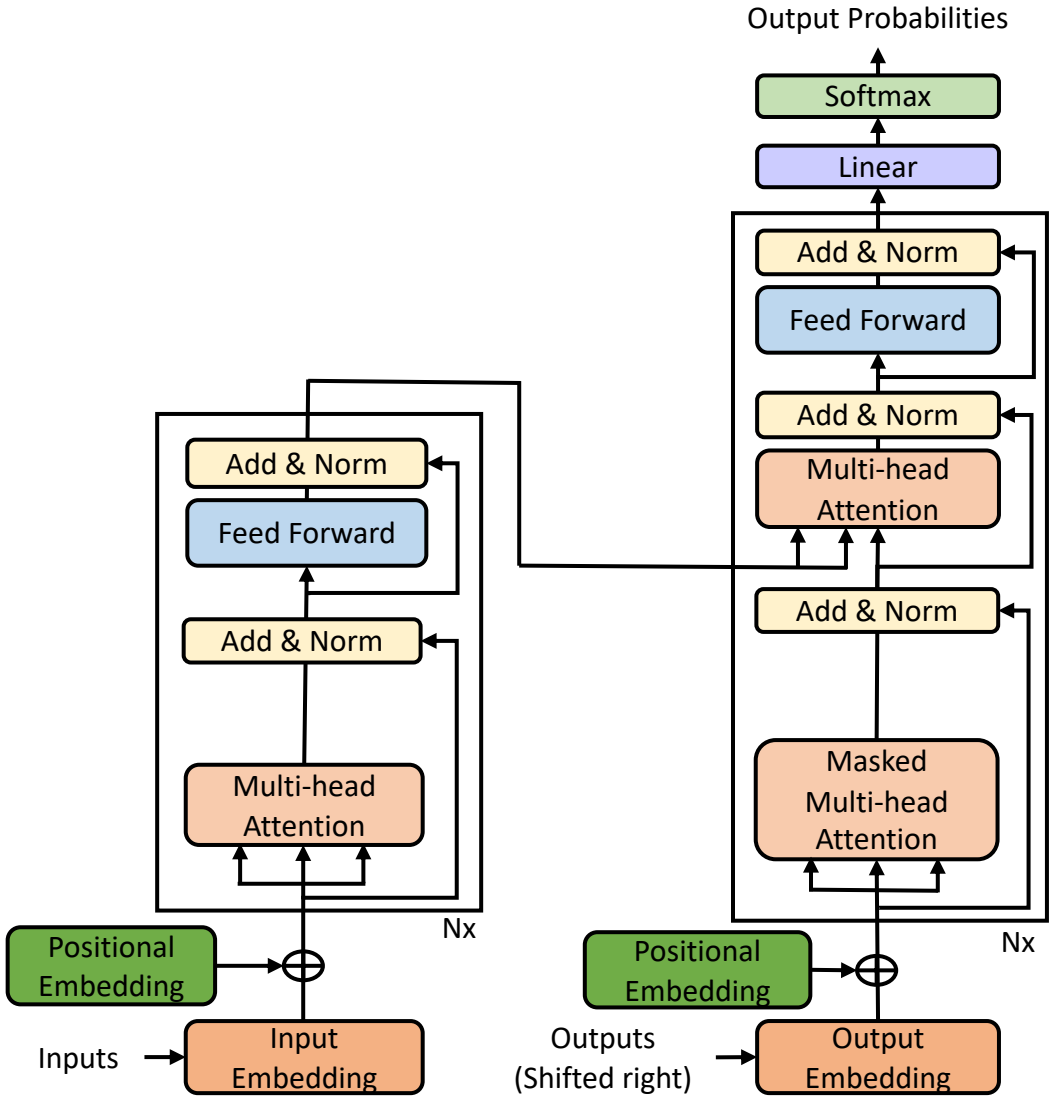
Model



Sentence representation for vanilla Transformer

$$\left[\begin{array}{l}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\bar{\mathbf{H}}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Model



Sentence representation for vanilla Transformer

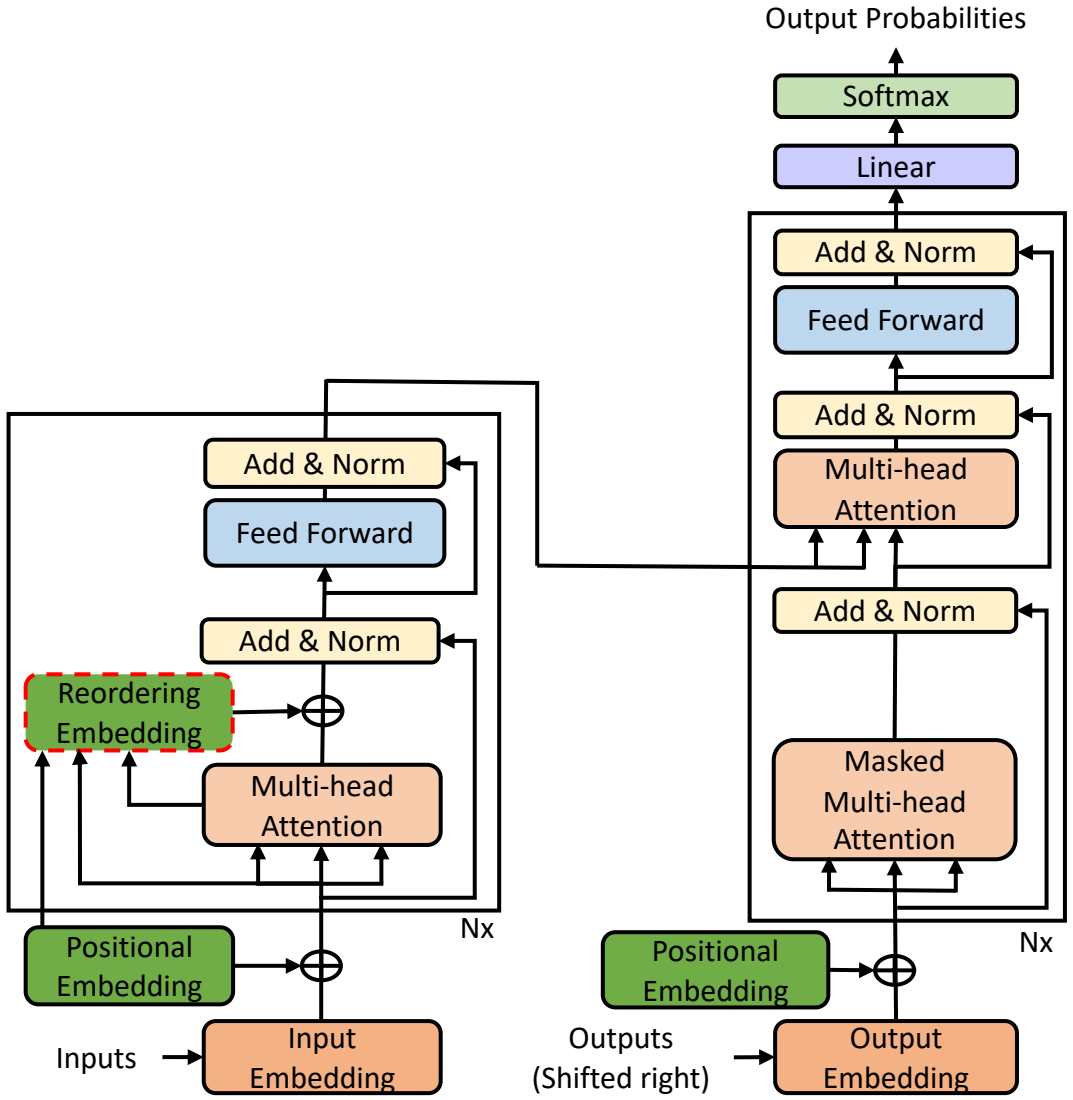
$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\bar{\mathbf{H}}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Sentence representation for our Transformer

$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{PP}^n = \mathrm{sigmoid}(\bar{\mathbf{V}}^n \cdot \tanh(\mathbf{W}^n \cdot \mathbf{H}^{n-1} + \bar{\mathbf{W}}^n \cdot \bar{\mathbf{H}}^n)) \\ \mathbf{C}^n = \mathrm{LN}(\bar{\mathbf{H}}^n + \mathbf{PE} \cdot \mathbf{PP}^n) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\mathbf{C}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Model

Encoder_RE



Sentence representation for vanilla Transformer

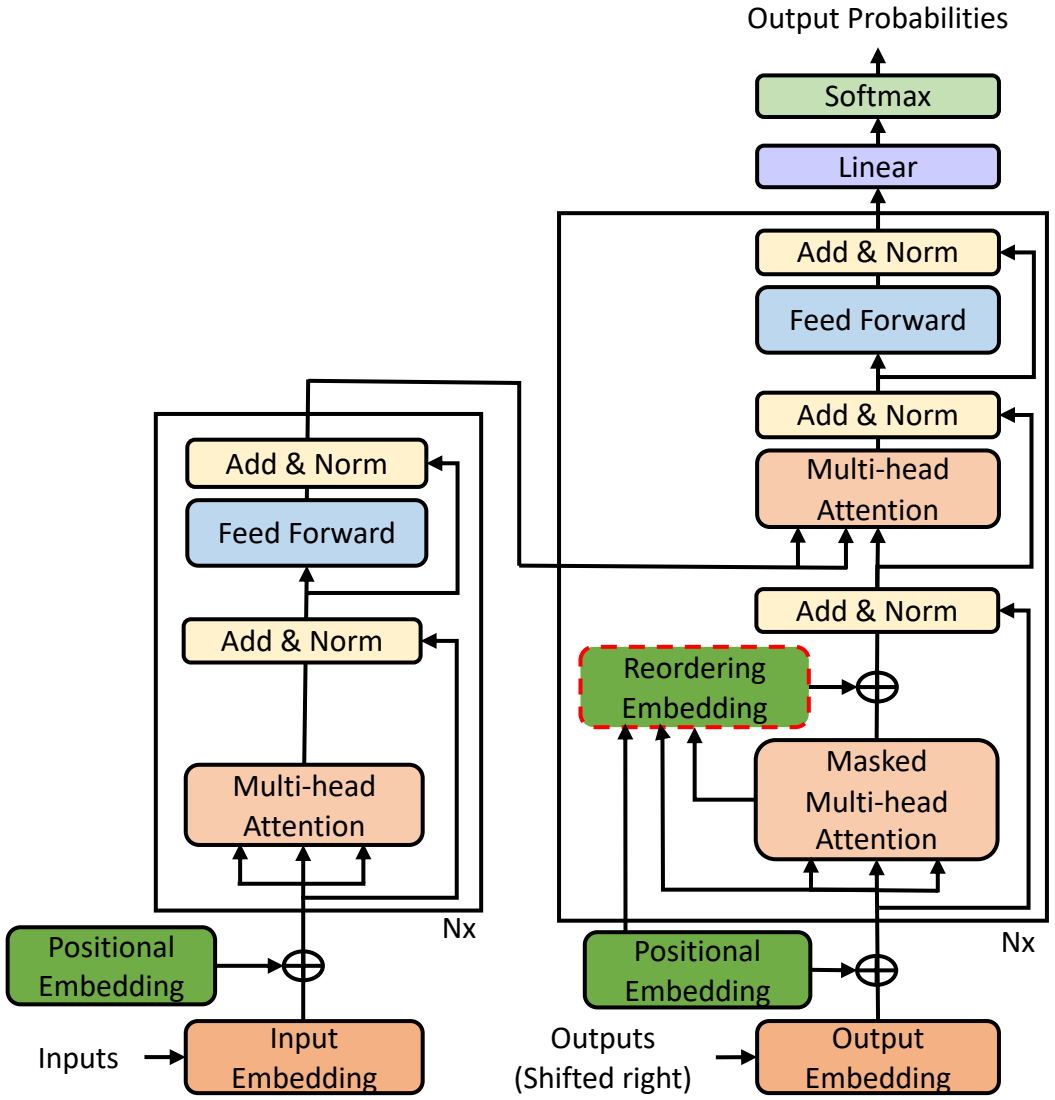
$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\bar{\mathbf{H}}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Sentence representation for our Transformer

$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{PP}^n = \mathrm{sigmoid}(\bar{\mathbf{V}}^n \cdot \tanh(\mathbf{W}^n \cdot \mathbf{H}^{n-1} + \bar{\mathbf{W}}^n \cdot \bar{\mathbf{H}}^n)) \\ \mathbf{C}^n = \mathrm{LN}(\bar{\mathbf{H}}^n + \mathbf{PE} \cdot \mathbf{PP}^n) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\mathbf{C}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Model

Decoder_RE



Sentence representation for vanilla Transformer

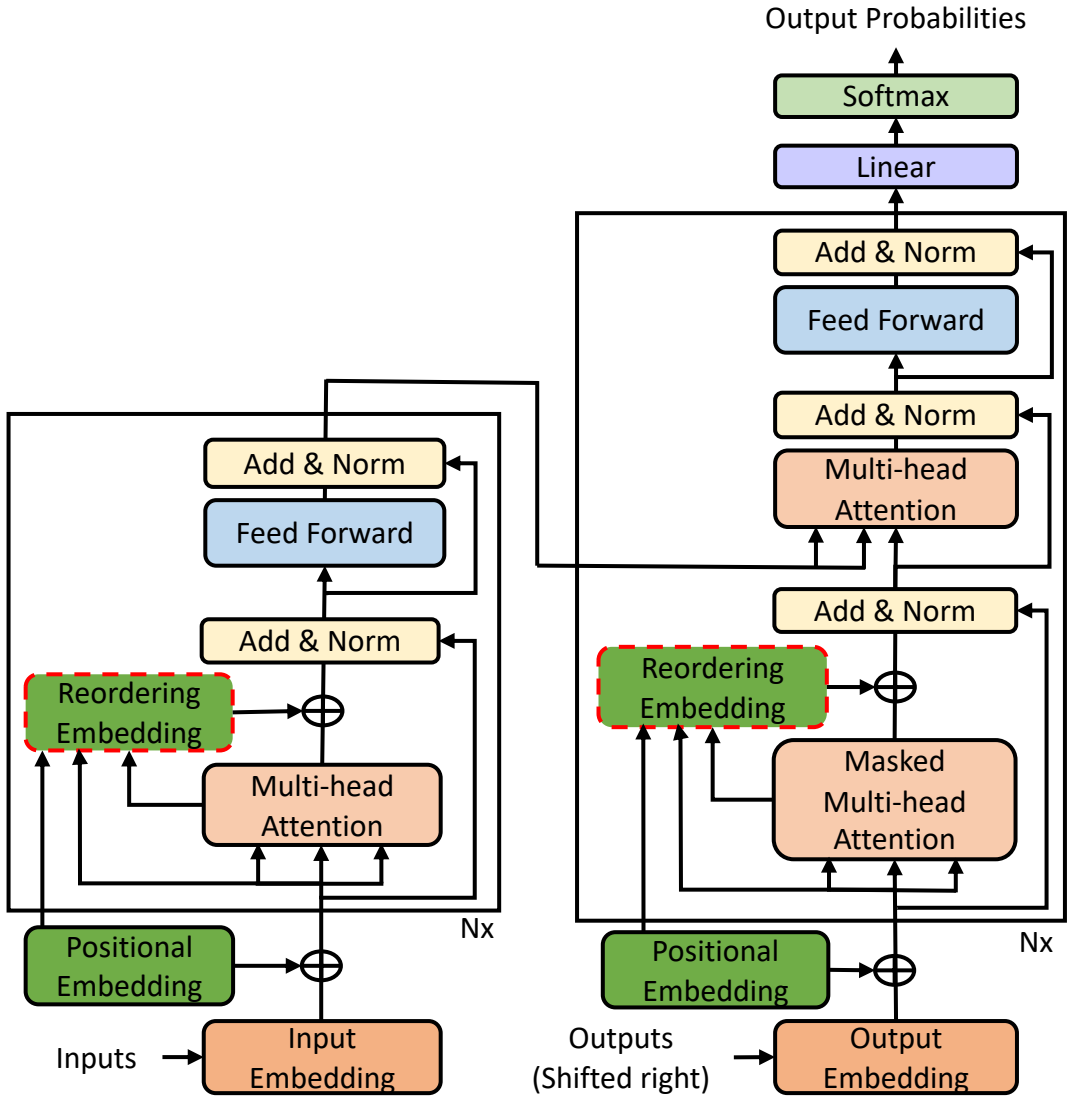
$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\bar{\mathbf{H}}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Sentence representation for our Transformer

$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{PP}^n = \mathrm{sigmoid}(\bar{\mathbf{V}}^n \cdot \tanh(\mathbf{W}^n \cdot \mathbf{H}^{n-1} + \bar{\mathbf{W}}^n \cdot \bar{\mathbf{H}}^n)) \\ \mathbf{C}^n = \mathrm{LN}(\bar{\mathbf{H}}^n + \mathbf{PE} \cdot \mathbf{PP}^n) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\mathbf{C}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Model

Both_RE



Sentence representation for vanilla Transformer

$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\bar{\mathbf{H}}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Sentence representation for our Transformer

$$\left[\begin{array}{c}\bar{\mathbf{H}}^n = \mathrm{LN}(\mathrm{SelfAtt}^n(\mathbf{H}^{n-1}) + \mathbf{H}^{n-1}) \\ \mathbf{PP}^n = \mathrm{sigmoid}(\bar{\mathbf{V}}^n \cdot \tanh(\mathbf{W}^n \cdot \mathbf{H}^{n-1} + \bar{\mathbf{W}}^n \cdot \bar{\mathbf{H}}^n)) \\ \mathbf{C}^n = \mathrm{LN}(\bar{\mathbf{H}}^n + \mathbf{PE} \cdot \mathbf{PP}^n) \\ \mathbf{H}^n = \mathrm{LN}(\mathrm{FNN}^n(\mathbf{C}^n) + \bar{\mathbf{H}}^n)\end{array}\right]_N$$

Experiments

Data sets: WMT14 English-to-German (EN-DE), NIST Chinese-to-English (ZH-EN), and WAT ASPEC Japanese-to-English (JA-EN) benchmarks

Baselines: vanilla Transformer, Relative PE, Additional PE (control experiment), and pre-reordering (JA-EN)

Our models: Encoder_REs, Decoder_REs, and Both_REs

Main configure:

	N	d_{model}	d_{ff}	H	P_{drop}	e_{ls}
base	6	512	2048	8	0.1	0.1
big	6	1024	4096	16	0.3	0.2

System setting: BPE translation units, OpenNMT toolkit, batch-size 4096*4 tokens, single P100 GPU, and so on

Evaluating: SacreBLEU for EN-DE translation task, and multi-bleu.perl for ZH-EN and JA-EN translation tasks

Main Results

System	Architecture	newstest2014	#Speed1	#Speed2	#Params
Existing NMT systems					
Wu et al. (2016)	GNMT	26.3	N/A	N/A	N/A
Gehring et al. (2017b)	CONVS2S	26.36	N/A	N/A	N/A
Vaswani et al. (2017)	Transformer (base)	27.3	N/A	N/A	65.0M
Vaswani et al. (2017)	Transformer (big)	28.4	N/A	N/A	213.0M
Our NMT systems					
this work	Transformer (base)	27.24	9910	181	97.6M
	+Additional PEs	27.10	9202	179	97.6M
	+Relative PEs	27.63	4418	146	97.6M
	+Encoder_REs	28.03++	8816	179	102.1M
	+Decoder_REs	27.61+	9101	175	102.1M
	+Both_REs	28.22++	8605	174	106.8M
	Transformer (big)	28.34	4345	154	272.8M
	+Both_REs	29.11++	3434	146	308.2M

Table 1: Results of WMT 14 EN-DE translation task

Main Results

System	Architecture	newstest2014	#Speed1	#Speed2	#Params
Existing NMT systems					
Wu et al. (2016)	GNMT	26.3	N/A	N/A	N/A
Gehring et al. (2017b)	CONVS2S	26.36	N/A	N/A	N/A
Vaswani et al. (2017)	Transformer (base)	27.3	N/A	N/A	65.0M
Vaswani et al. (2017)	Transformer (big)	28.4	N/A	N/A	213.0M
Our NMT systems					
this work	Transformer (base)	27.24	9910	181	97.6M
	+Additional PEs	27.10	9202	179	97.6M
	+Relative PEs	27.63	4418	146	97.6M
	+Encoder_REs	28.03++	8816	179	102.1M
	+Decoder_REs	27.61+	9101	175	102.1M
	+Both_REs	28.22++	8605	174	106.8M
	Transformer (big)	28.34	4345	154	272.8M
	+Both_REs	29.11++	3434	146	308.2M

Table 1: Results of WMT 14 EN-DE translation task

Main Results

System	Architecture	newstest2014	#Speed1	#Speed2	#Params
Existing NMT systems					
Wu et al. (2016)	GNMT	26.3	N/A	N/A	N/A
Gehring et al. (2017b)	CONVS2S	26.36	N/A	N/A	N/A
Vaswani et al. (2017)	Transformer (base)	27.3	N/A	N/A	65.0M
Vaswani et al. (2017)	Transformer (big)	28.4	N/A	N/A	213.0M
Our NMT systems					
this work	Transformer (base)	27.24	9910	181	97.6M
	+Additional PEs	27.10	9202	179	97.6M
	+Relative PEs	27.63	4418	146	97.6M
	+Encoder_REs	28.03++	8816	179	102.1M
	+Decoder_REs	27.61+	9101	175	102.1M
	+Both_REs	28.22++	8605	174	106.8M
	Transformer (big)	28.34	4345	154	272.8M
	+Both_REs	29.11++	3434	146	308.2M

Table 1: Results of WMT 14 EN-DE translation task

Main Results

System	Architecture	newstest2014	#Speed1	#Speed2	#Params
Existing NMT systems					
Wu et al. (2016)	GNMT	26.3	N/A	N/A	N/A
Gehring et al. (2017b)	CONVS2S	26.36	N/A	N/A	N/A
Vaswani et al. (2017)	Transformer (base)	27.3	N/A	N/A	65.0M
Vaswani et al. (2017)	Transformer (big)	28.4	N/A	N/A	213.0M
Our NMT systems					
this work	Transformer (base)	27.24	9910	181	97.6M
	+Additional PEs	27.10	9202	179	97.6M
	+Relative PEs	27.63	4418	146	97.6M
	+Encoder_REs	28.03++	8816	179	102.1M
	+Decoder_REs	27.61+	9101	175	102.1M
	+Both_REs	28.22++	8605	174	106.8M
	Transformer (big)	28.34	4345	154	272.8M
	+Both_REs	29.11++	3434	146	308.2M

Table 1: Results of WMT 14 EN-DE translation task

Main Results

System	Architecture	Test Sets					#Param
		MT02	MT03	MT04	MT05	MT08	
Existing NMT systems							
Vaswani et al. (2017)	Transformer	N/A	N/A	N/A	N/A	N/A	N/A
Zhang et al. (2017)	RNNsearch+Distortion	N/A	38.33	40.40	36.81	N/A	N/A
Meng and Zhang (2018)	DTMT#1	46.90	45.85	46.78	45.96	36.58	170.5M
Meng and Zhang (2018)	DTMT#4	47.03	46.34	47.52	46.70	37.61	208.4M
Kong et al. (2018)	RNN-based NMT	N/A	38.62	41.98	37.42	N/A	87.9M
Zhao et al. (2018a)	RNN-based NMT+MEM	N/A	44.98	45.51	43.95	33.33	N/A
Our NMT systems							
this work	Transformer (base)	46.45	45.33	45.82	45.57	35.57	78.3M
	+Additional PEs	46.66	45.35	46.11	45.40	35.75	78.3M
	+Relative PEs	46.41	45.94	46.54	46.21	36.14	78.3M
	+Encoder_REs	47.47++	45.87++	46.82++	46.58++	36.42++	83.0M
	+Decoder_REs	46.80	45.43	46.23++	46.11++	36.02+	83.0M
	+Both_REs	47.54++	46.56++	47.27++	46.88++	36.77++	87.6M
	Transformer (Big)	47.76	46.66	47.51	47.71	37.73	244.7M
	+Both_REs	48.42++	47.32++	48.22++	48.56++	38.19+	269.7M

Table 2: Results of ZH-EN translation task

Systems	testset	#Param
Transformer (base)	30.33	73.9M
+Pre-Reordering	28.93	73.9M
+Additional PEs	30.16	73.9M
+Relative PEs	30.42	73.9M
+Encoder_REs	31.12++	78.6M
+Decoder_REs	30.78+	78.6M
+Both_REs	31.41++	84.4M
Transformer (big)	31.21	234.6M
+Both_REs	31.93++	273.7M

Table 3: Results of JA-EN translation task

Main Results

System	Architecture	Test Sets					#Param
		MT02	MT03	MT04	MT05	MT08	
Existing NMT systems							
Vaswani et al. (2017)	Transformer	N/A	N/A	N/A	N/A	N/A	N/A
Zhang et al. (2017)	RNNsearch+Distortion	N/A	38.33	40.40	36.81	N/A	N/A
Meng and Zhang (2018)	DTMT#1	46.90	45.85	46.78	45.96	36.58	170.5M
Meng and Zhang (2018)	DTMT#4	47.03	46.34	47.52	46.70	37.61	208.4M
Kong et al. (2018)	RNN-based NMT	N/A	38.62	41.98	37.42	N/A	87.9M
Zhao et al. (2018a)	RNN-based NMT+MEM	N/A	44.98	45.51	43.95	33.33	N/A
Our NMT systems							
this work	Transformer (base)	46.45	45.33	45.82	45.57	35.57	78.3M
	+Additional PEs	46.66	45.35	46.11	45.40	35.75	78.3M
	+Relative PEs	46.41	45.94	46.54	46.21	36.14	78.3M
	+Encoder_REs	47.47++	45.87++	46.82++	46.58++	36.42++	83.0M
	+Decoder_REs	46.80	45.43	46.23++	46.11++	36.02+	83.0M
	+Both_REs	47.54++	46.56++	47.27++	46.88++	36.77++	87.6M
	Transformer (Big)	47.76	46.66	47.51	47.71	37.73	244.7M
	+Both_REs	48.42++	47.32++	48.22++	48.56++	38.19+	269.7M

Table 2: Results of ZH-EN translation task

Systems	testset	#Param
Transformer (base)	30.33	73.9M
+Pre-Reordering	28.93	73.9M
+Additional PEs	30.16	73.9M
+Relative PEs	30.42	73.9M
+Encoder_REs	31.12++	78.6M
+Decoder_REs	30.78+	78.6M
+Both_REs	31.41++	84.4M
Transformer (big)	31.21	234.6M
+Both_REs	31.93++	273.7M

Table 3: Results of JA-EN translation task

Effect of REs

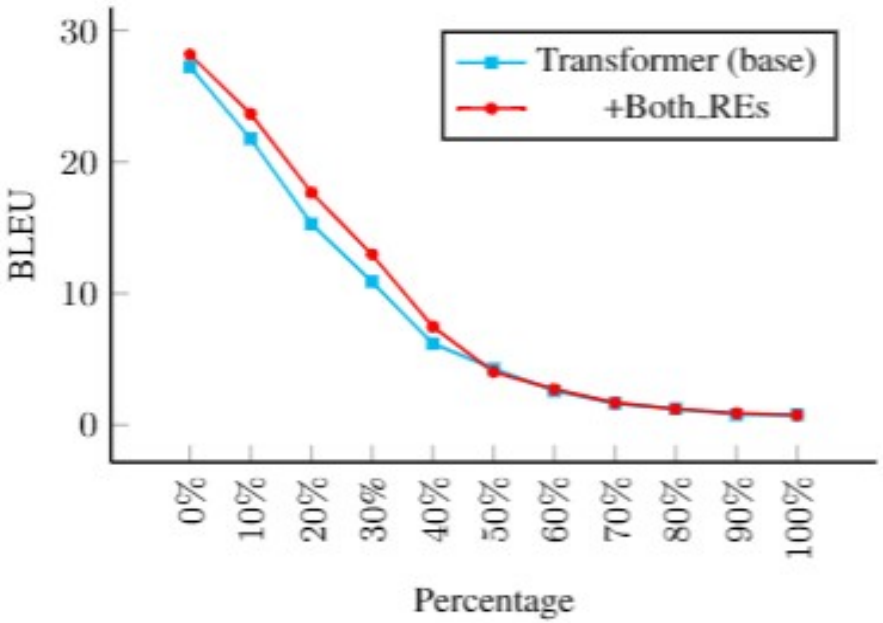


Figure 3: The effect of reordering in the test set where the word orders are partially wrong for test set of EN-DE.

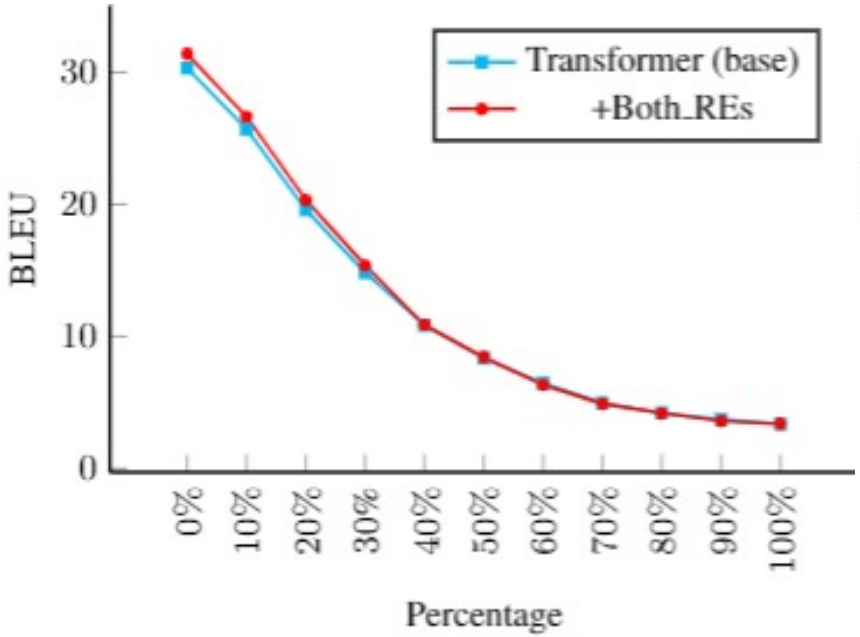


Figure 4: The effect of reordering in the test set where the word orders are partially wrong for test set of JA-EN.

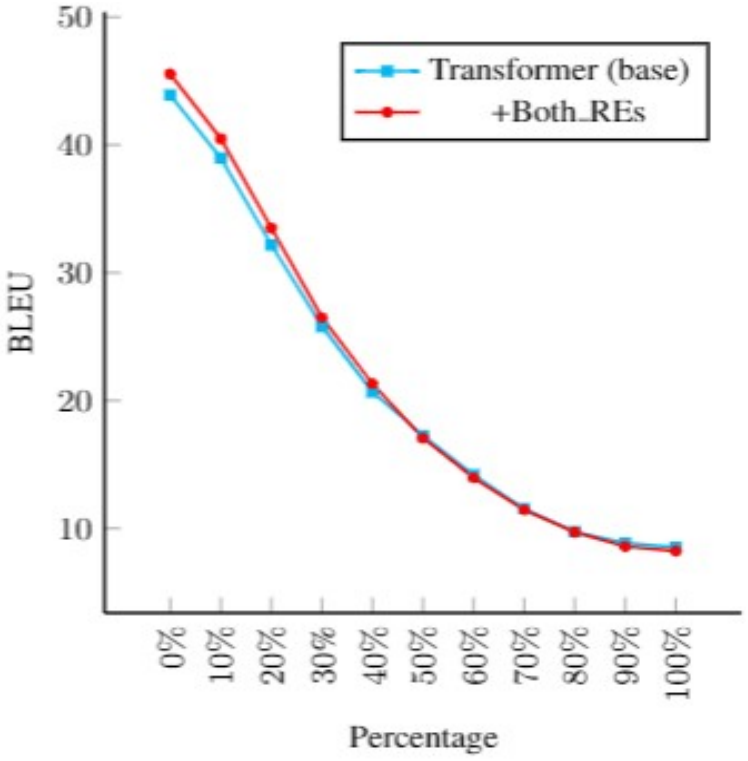


Figure 4: The effect of reordering in the test set where the word orders are partially wrong for test set of ZH-EN.

Effect of REs

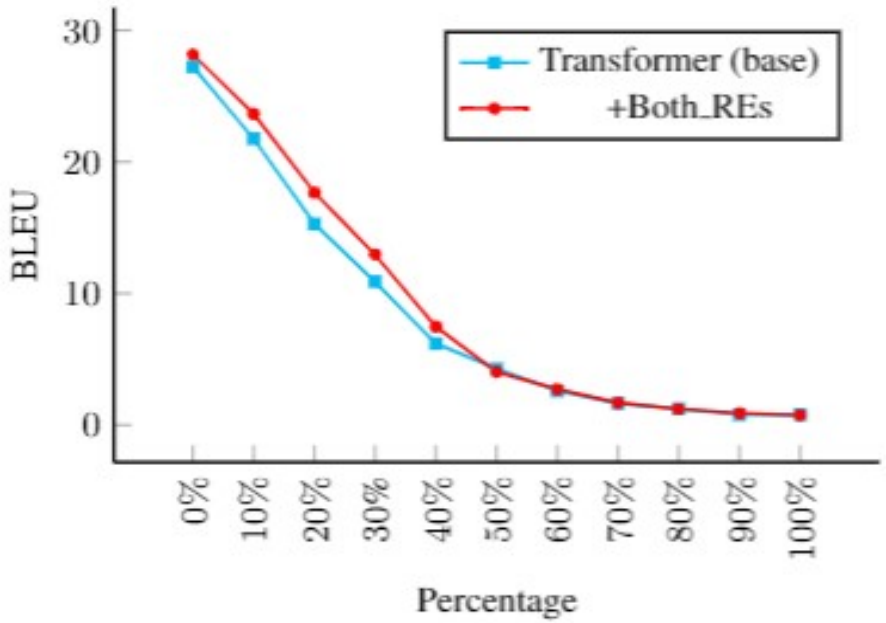


Figure 3: The effect of reordering in the test set where the word orders are partially wrong for test set of EN-DE.

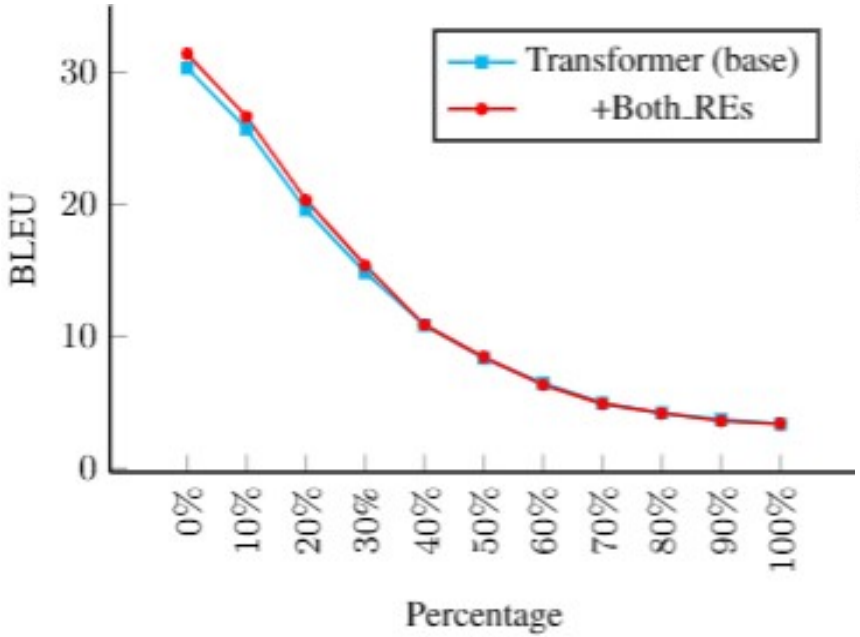


Figure 4: The effect of reordering in the test set where the word orders are partially wrong for test set of JA-EN.

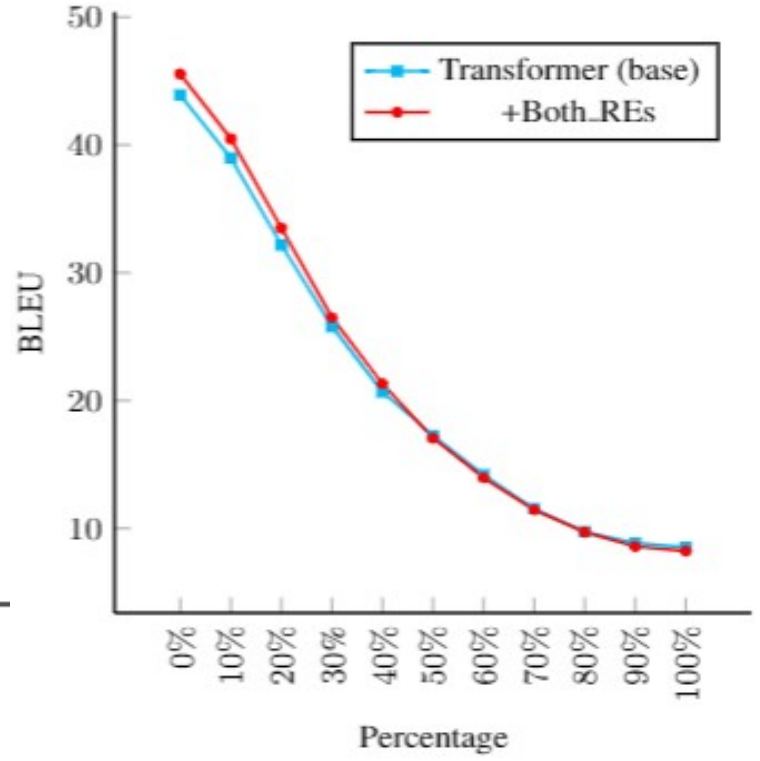


Figure 4: The effect of reordering in the test set where the word orders are partially wrong for test set of ZH-EN.

1. Reordering embedding can capture reordering information, especially, 0%~40%

Effect of REs

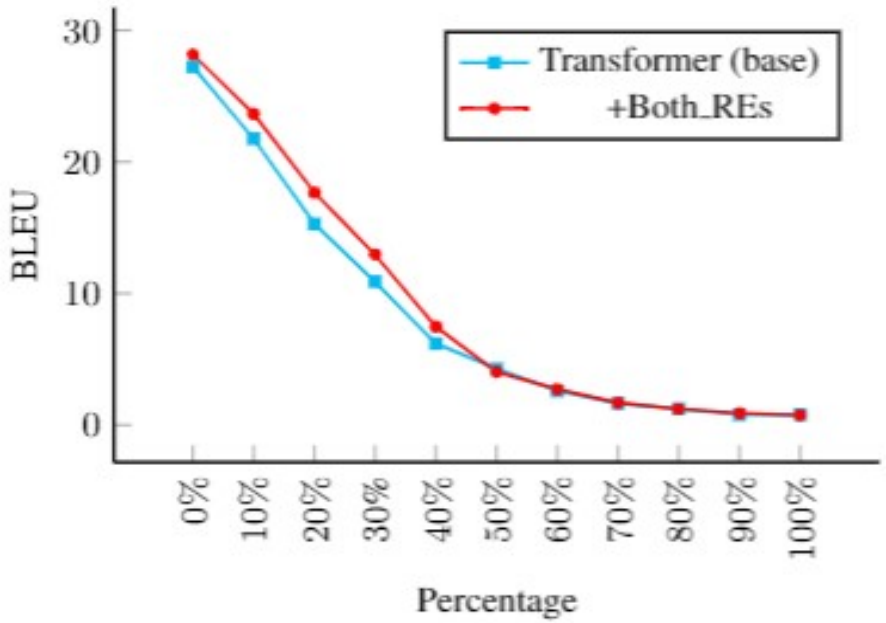


Figure 3: The effect of reordering in the test set where the word orders are partially wrong for test set of EN-DE.

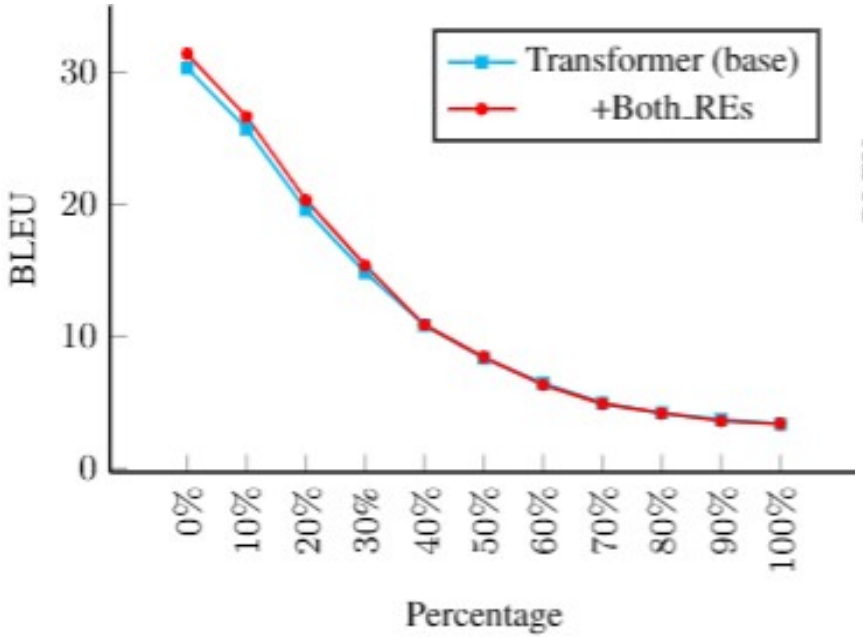


Figure 4: The effect of reordering in the test set where the word orders are partially wrong for test set of JA-EN.

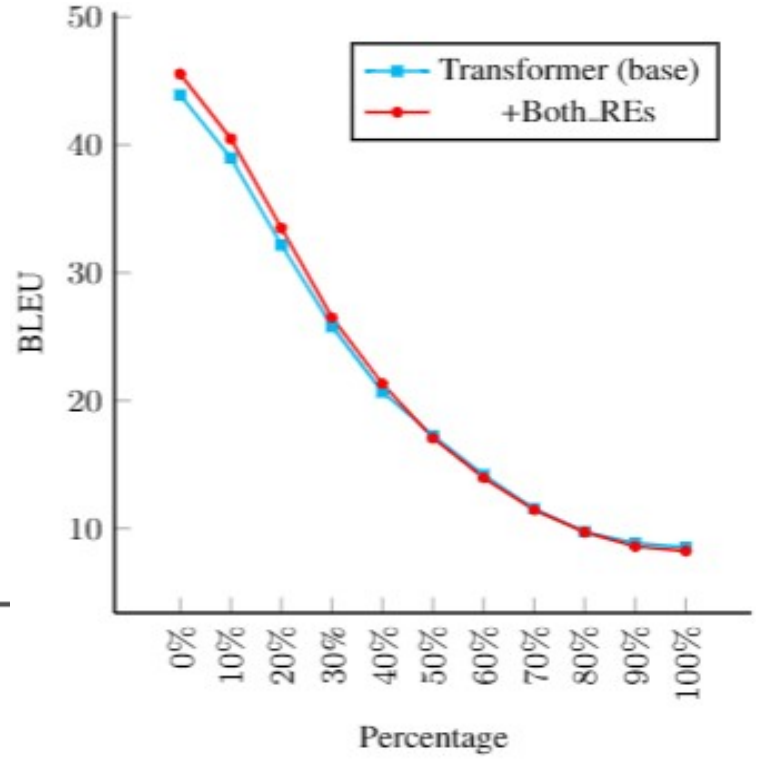


Figure 4: The effect of reordering in the test set where the word orders are partially wrong for test set of ZH-EN.

1. Reordering embedding can capture reordering information, especially, 0%~40%
2. Excessive exchanges of word orders are not conductive to learn reordering embedding, when more than 40%

Translation Cases

Src1: 继续 改革 的 努力 将 促成 经济 复苏
[continue] [reform] [to] [efforts] [will] [enhance] [economic] [recovery]

Transformer (base): continued reform efforts will bring about economic recovery

+Both_REs (base): the efforts to continue the reform will promote economic recovery

Ref1: the efforts to continue reform will enhance the economic recovery

Translation Cases

Src1: 继续 改革 的 努力 将 促成 经济 复苏
[continue] [reform] [to] [efforts] [will] [enhance] [economic] [recovery]

Transformer (base): continued reform efforts will bring about economic recovery

+Both_REs (base): the efforts to continue the reform will promote economic recovery

Ref1: the efforts to continue reform will enhance the economic recovery

---

Src2: 这 起 事件 造成 九 人 丧生
[the] [ ] [incident] [ ] [nine] [people] [killed]

Transformer (base): the incident killed nine people

+Both_REs (base): nine people were killed in the incident

Ref2: nine people were killed in the incident

Translation Cases

Src1: 继续 改革 的 努力 将 促成 经济 复苏
[continue] [reform] [to] [efforts] [will] [enhance] [economic] [recovery]

Transformer (base): continued reform efforts will bring about economic recovery

+Both_REs (base): the efforts to continue the reform will promote economic recovery

Ref1: the efforts to continue reform will enhance the economic recovery

---

Src2: 这 起 事件 造成 九 人 丧生
[the] [ ] [incident] [ ] [nine] [people] [killed]

Transformer (base): the incident killed nine people

+Both_REs (base): nine people were killed in the incident

Ref2: nine people were killed in the incident

*The proposed **reordering embedding** is beneficial to generate a translation in line with the target language word order.*

Conclusion and Future Work

- Introduce the reordering information to the Transformer initially
- Enable the Transformer translation system to explicitly model reordering information
- The proposed reordering mechanism can be easily integrated into the Transformer to improve the translation performance

Conclusion and Future Work

- Introduce the reordering information to the Transformer initially
- Enable the Transformer translation system to explicitly model reordering information
- The proposed reordering mechanism can be easily integrated into the Transformer to improve the translation performance

- In the future, we will further explore the effectiveness of reordering embeddings and try to apply reordering embeddings into other NLP tasks

Q&A