Attention is All you Need

2022. 03. 25

Joonhun Lee

Agenda

1. Introduction
2. Prerequisites
    - Positional Encoding
    - Self-Attention
    - Multi-head Attention
    - Masked Self-Attention
3. Model Architecture
4. Experiments

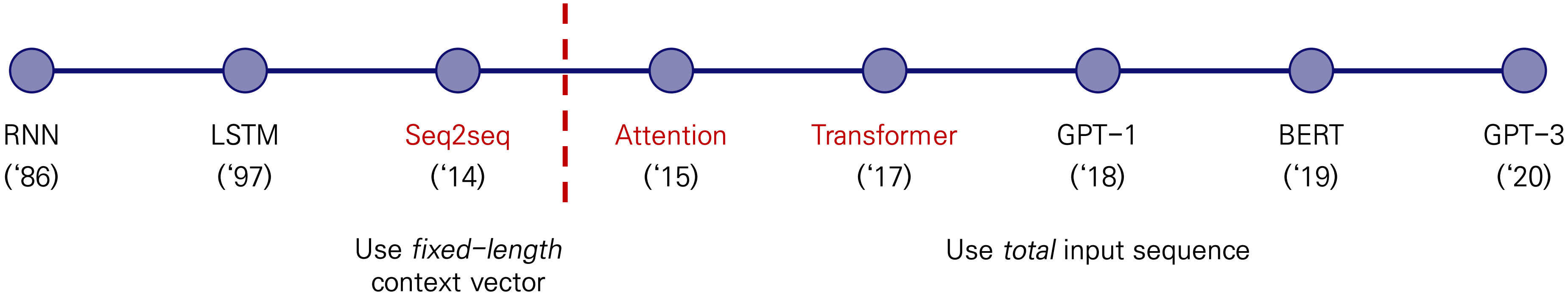
Agenda

1. Introduction
2. Prerequisites
   - Positional Encoding
   - Self-Attention
   - Multi-head Attention
   - Masked Self-Attention
3. Model Architecture
4. Experiments

Evolution of Machine Translation

SOTA models are based on Transformer architecture

- GPT leverage decoder architecture of Transformer
- BERT leverage encoder architecture of Transformer

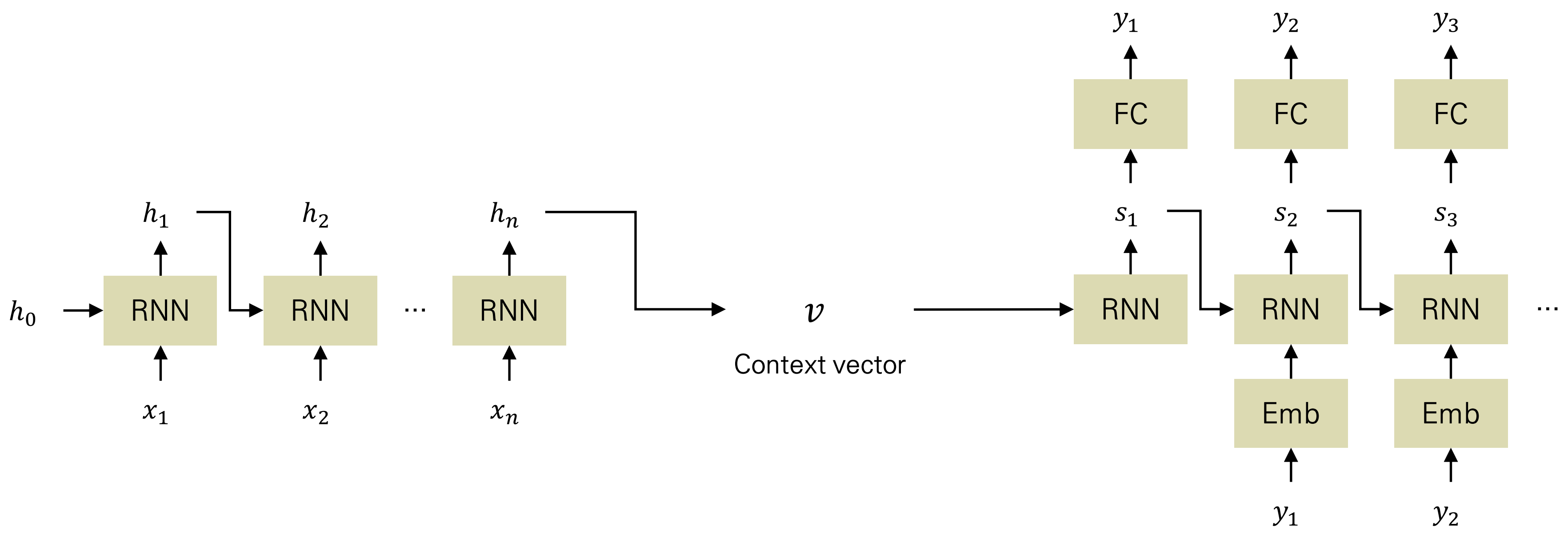


Sutskever et al. 2014

Drawback of Seq2seq

Seq2seq model scan the words in the source sentence one by one and compress the information into a context vector v

- Executing speed is limited due to bottleneck → ConvS2S
- The information of several words is compressed in a single fixed-length context vector → Seq2seq with Attention

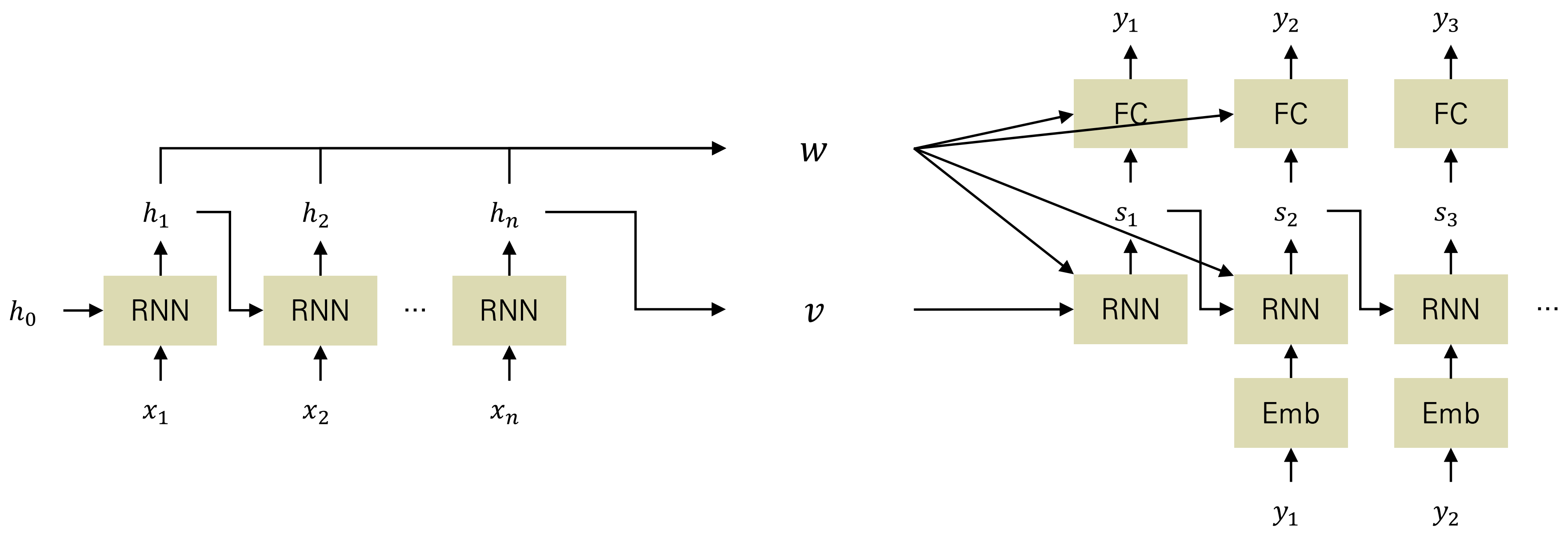


Sutskever et al. 2014

Seq2seq with Attention

With attention mechanism, decoder of Seq2seq model can refer to all outputs from encoder

- Decoder calculates the importance of each output of the encoder $\alpha_{ij} = \mathrm{Softmax}\left(\mathrm{Attention}(s_{i-1}, h_j)\right)$
- The information is still compressed in a single fixed−length context vector

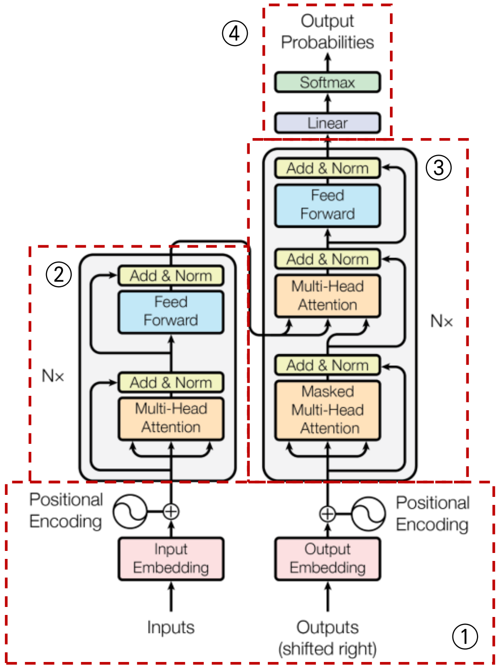


Bahdanau et al. 2015

Transformer

Transformer model operates in 4 steps as Seq2seq model

- Instead of RNN, Transformer consists of only attention



1. Embedding step
   - Data Embedding
   - Positional Encoding
2. Encoding step
   - Multi-Head Self-Attention
   - Feed Forward
   - Residual Connection
3. Decoding step
   - Masked Multi-Head Self-Attention
   - Multi-Head Self-Attention
   - Feed Forward
   - Residual Connection
4. Prediction step
   - Fully Connected and Softmax

Vaswani et al. 2017

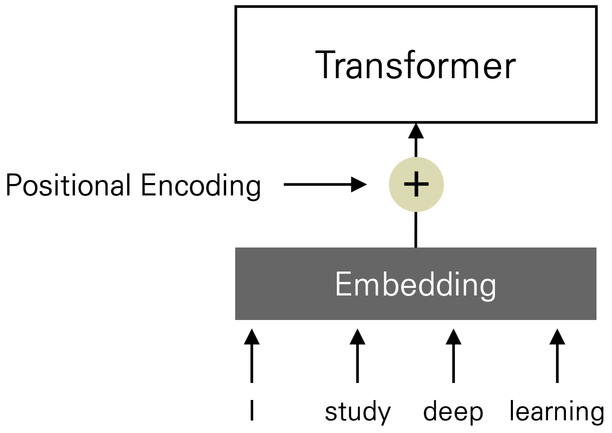
Agenda

1. Introduction
2. Prerequisites
    - Positional Encoding
    - Self-Attention
    - Multi-head Attention
    - Masked Self-Attention
3. Model Architecture
4. Experiments

Positional Encoding

Positional encoding is a representation of the position of items in a sequence

- As RNN–based models such as Seq2seq capture input order as the information on the order of words, simple embedding is needed
- Since Transformer contains no RNN or CNN, Positional encoding needed for the information on the position of each word



Positional Encoding

According to the paper, Sinusoidal Positional Encoding is used for Transformer

- Although the performance of Learned Positional Embedding is identical, Sinusoidal Positional Encoding is selected for the possibility of sequence longer than train dataset
- $PE_{(pos,\,2dim)} = \sin(pos/10^{8dim/d_{model}}),\ PE_{(pos,\,2dim+1)} = \cos(pos/10^{8dim/d_{model}})$

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)		positional embedding instead of sinusoids								4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

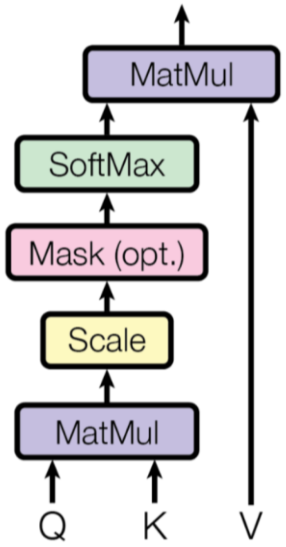
Vaswani et al. 2017

Self–Attention

Attention is a mapping a query to a set of key–value pairs which quantifies the interdependence

- Scaled Dot–Product Attention addresses the vanishing gradient of softmax by scaling $1/\sqrt{d_k}$ where $Q, K \in \mathbb{R}^{d_{model} \times d_k}$
- $\text{Attention}(Q, K, V) = \text{softmax}\left(QK^T/\sqrt{d_k}\right)V$

Weighted sum of the values where the weight is computed by the query with the corresponding key

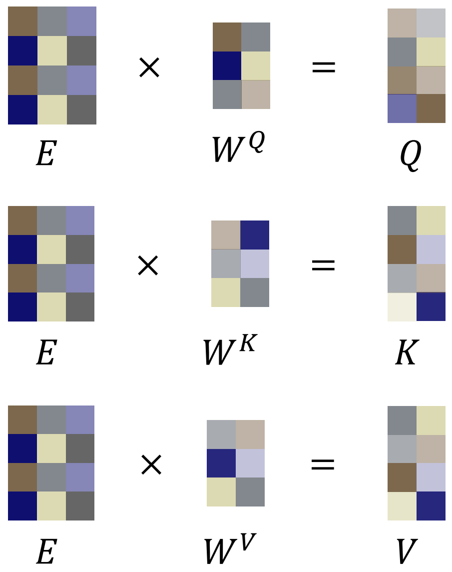


Vaswani et al. 2017

Self-Attention

Self-Attention relates different positions of a single sequence in order to compute a representation of the same sequence

- General Attention calculates query Q, key K, value V matrices with the input and output
  (e.g., Q is the hidden state of decoder at time t, and K & V are all hidden states of encoder in Seq2seq with Attention)
- In case of Self-Attention, $Q, K,$ and V are all calculated solely from the input, the embedding of sentence E

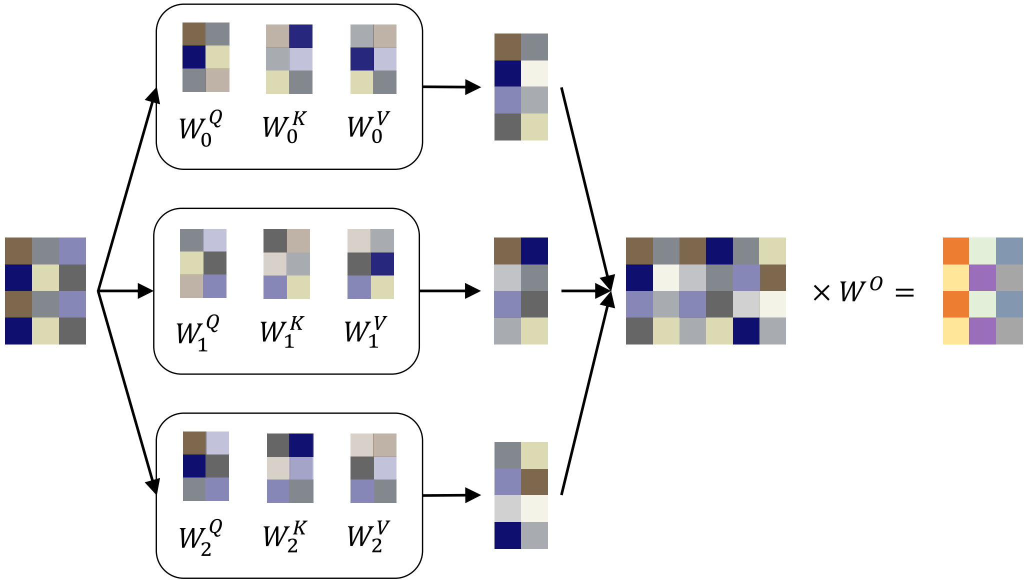


1) Luong et al. 2015
2) Vaswani et al. 2017

Multi-head Attention

Multi-head Attention allows the model to attend to information from different positions

- Instead of a single set of matrices $Q, K, V \in \mathbb{R}^{d_{model} \times d_{model}}$, Multi-head Attention projects different linear projections h times
- $\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(head_1, \dots, head_h) W^O$ where $head_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$
  where $W_i^Q, W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W_i^O \in \mathbb{R}^{d_{model} \times d_{model}}$

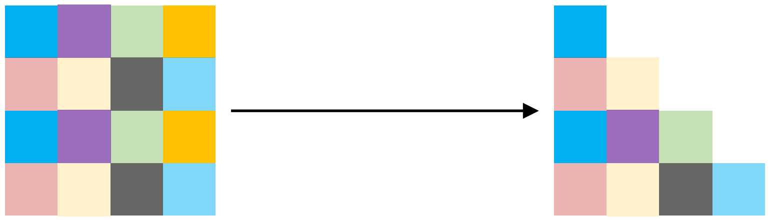


Vaswani et al. 2017

Masked Self-Attention

Masked Self-Attention masks the future tokens, so that the model attend only to the words in the past

- The next word while decoding should only refer to the word sequence already inferred
- $Scores_{ij} \coloneqq -\infty$ for $i < j$ where $Scores = QK^T$ to make $\mathrm{softmax}(Scores_{ij}) = 0$



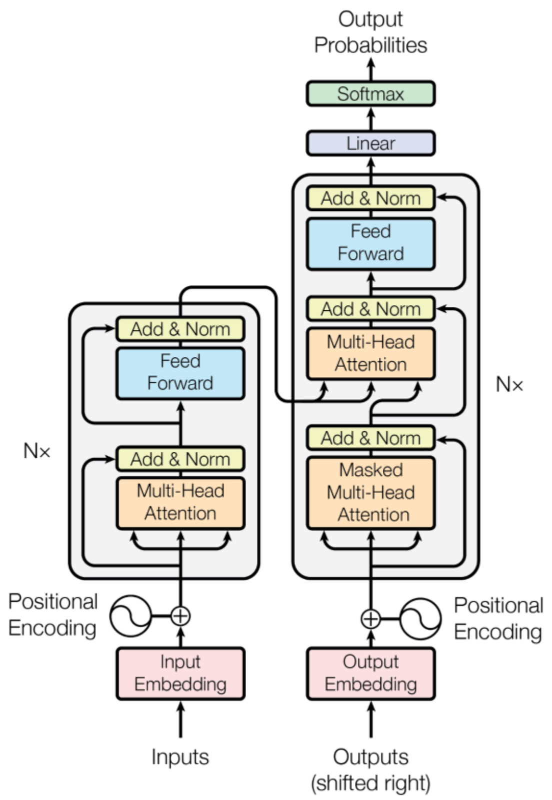
$Scores$ $\quad\quad$ $MaskedScores$

Vaswani et al. 2017

Agenda

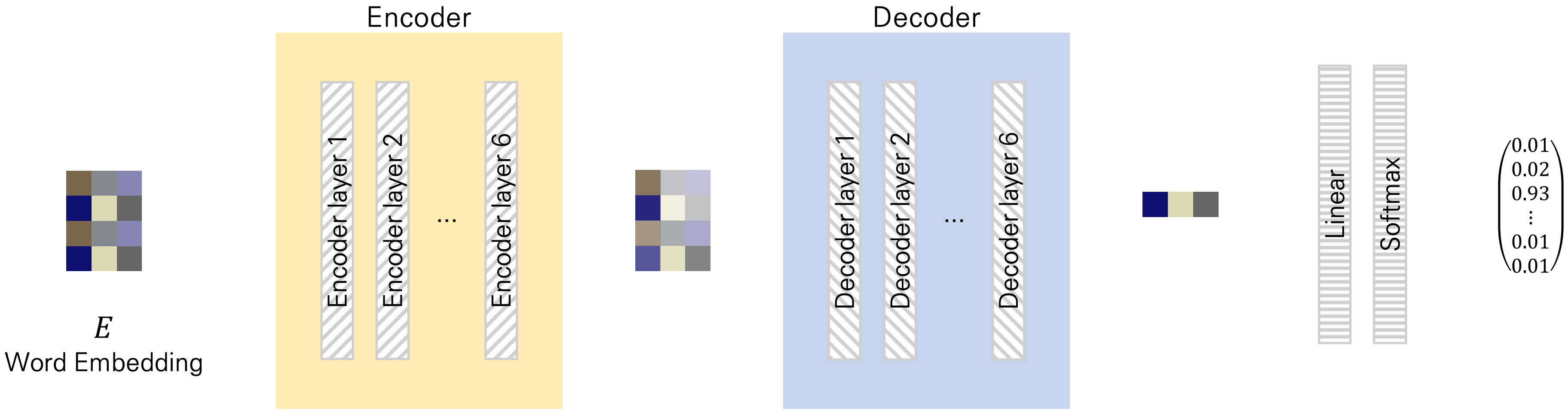
1. Introduction
2. Prerequisites
   - Positional Encoding
   - Self-Attention
   - Multi-head Attention
   - Masked Self-Attention
3. Model Architecture
4. Experiments

Model Architecture from the paper

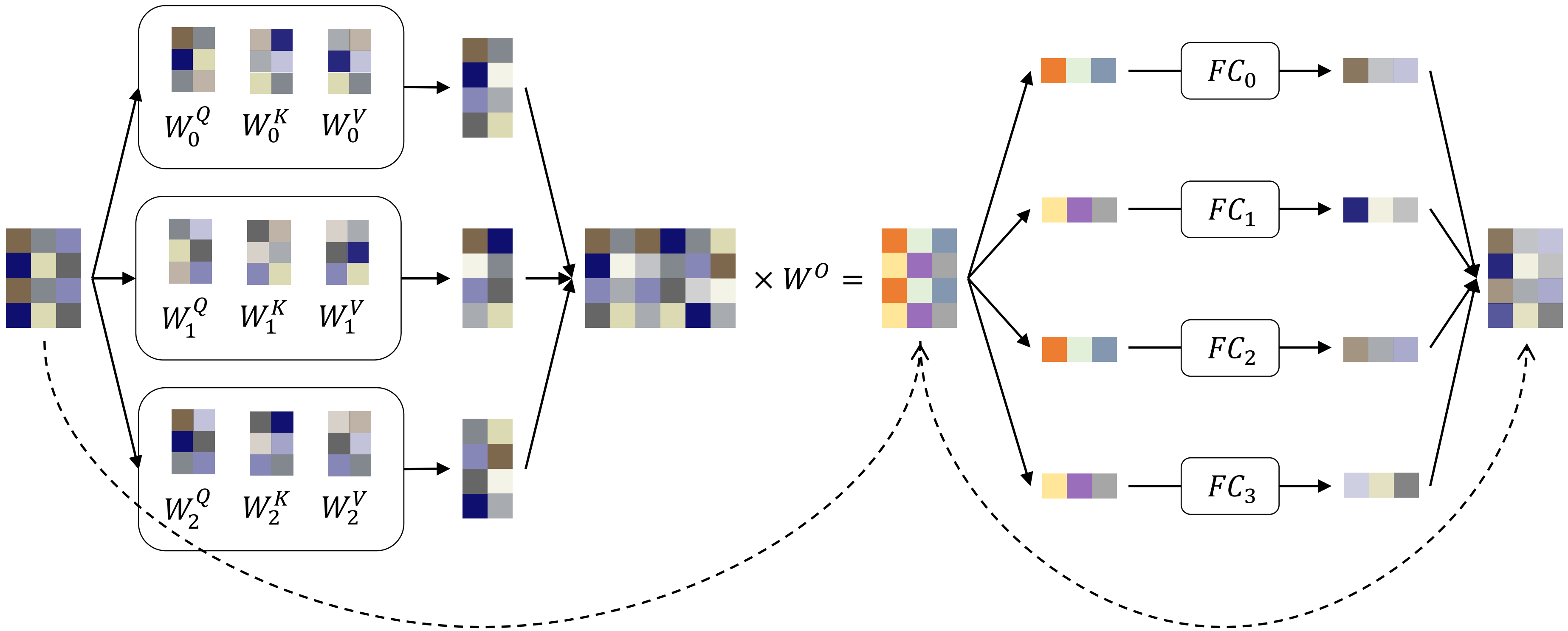


Vaswani et al. 2017

Simplified Model Architecture



Encoder layer



Agenda

1. Introduction
2. Prerequisites
    - Positional Encoding
    - Self-Attention
    - Multi-head Attention
    - Masked Self-Attention
3. Model Architecture
4. Experiments

Results

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ)

Parser	**Training**	**WSJ 23 F1**
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

Vaswani et al. 2017

E.O.D

Q & A