Attention Is All You Need

NLP_논문 스터디 : NLP 발표자 : 이화정

00.Contents

01. Abstract

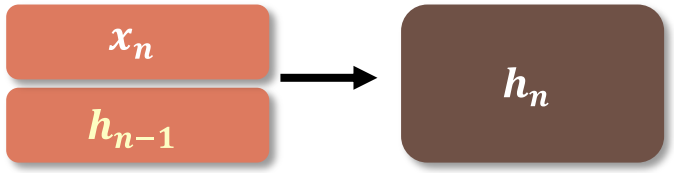
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

- ✔ Transfomer architecture 은 오직 'attention mechanisms' 을 사용한 것
- ✔ 병렬화를 통해 훈련속도를 감소시킴

02. Introduction & Background

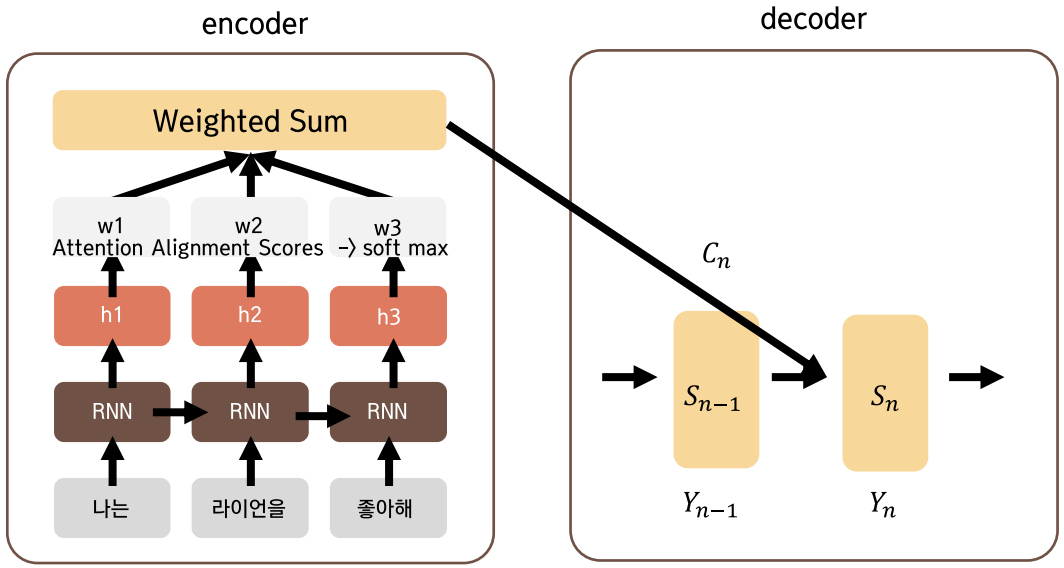
RNN, LSTM, GRU

순환 신경망 모델들은 단어의 Position 정보와 전 단계의 hidden states 를 더해서 현재의 hidden states 를 만든다.
이러한 방식 때문에, 병렬처리가 안된다는 문제가 발생함.



Seq2Seq with Attention

: Attention 은 디코더 내의 타겟 단어를 예측 할때마다, 인코더 내의 source 문장들 중 어느 입력단어들과 가장 유관한지를 탐색하는 것



03. Model Architecture

Dimension = 512

각각의 layer는 2개의 sub-layers
- 1ST sub-layer : multi-head Attention
- 2nd sub-layer : position-wise fully connected feed-forward network

인코더와 디코더의 Layer는 6개

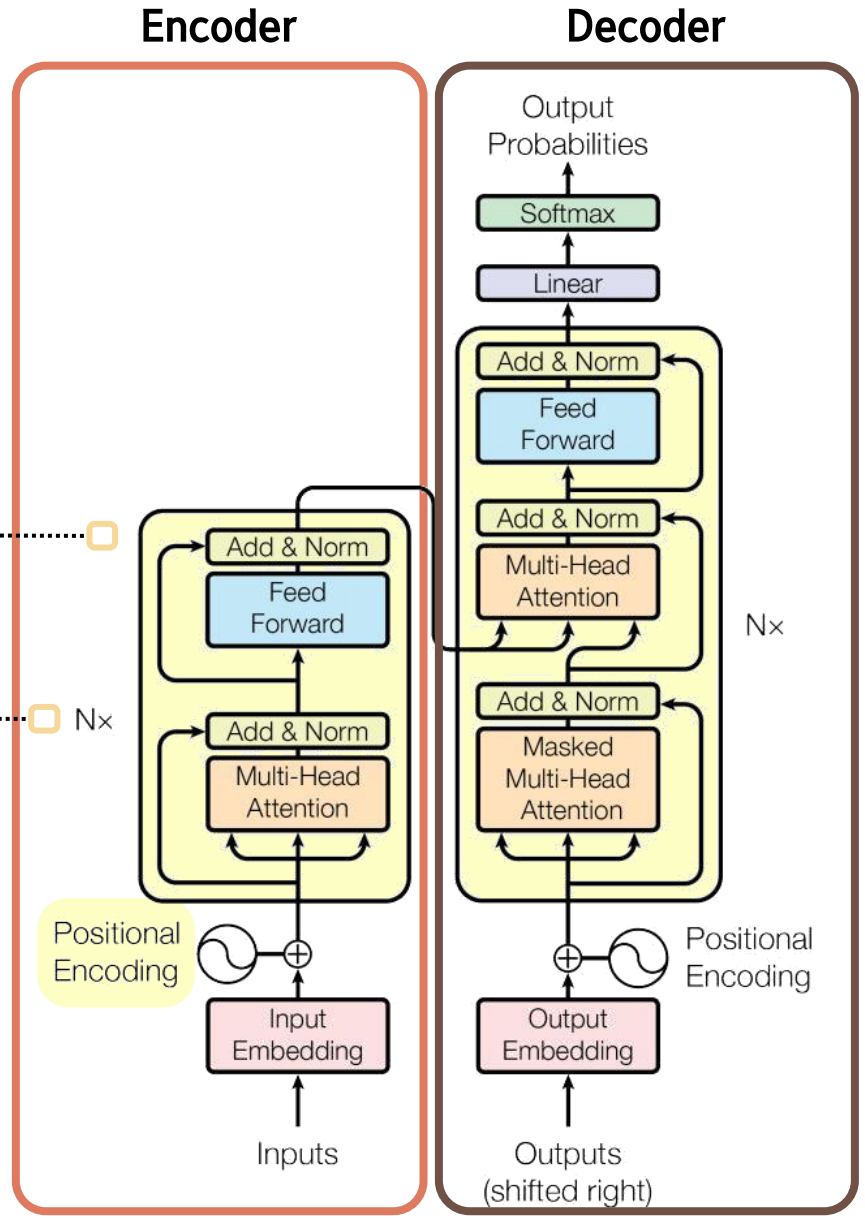
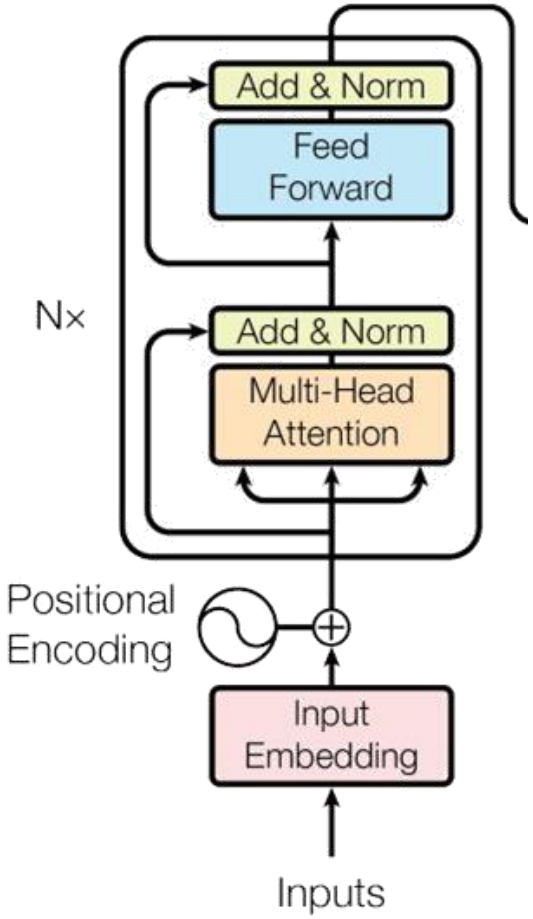


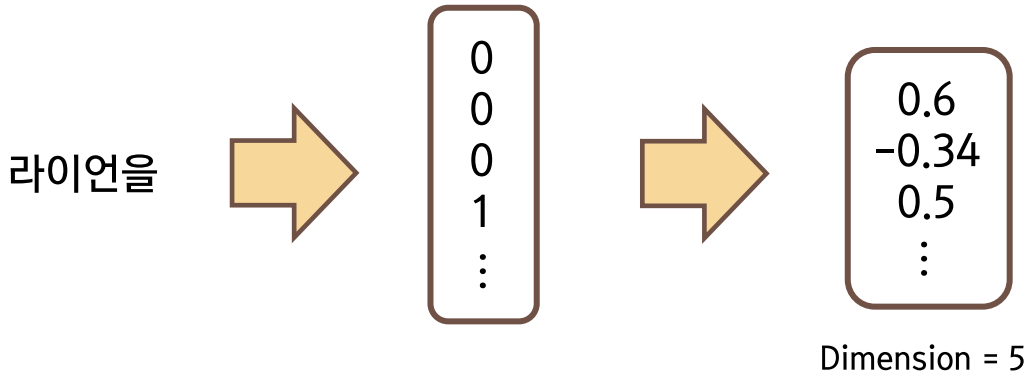
Figure 1: The Transformer - model architecture.

03. Model Architecture

3.1. Input Embedding & Positional Encoding



Input Embedding



Dimension = 512

Positional Encoding

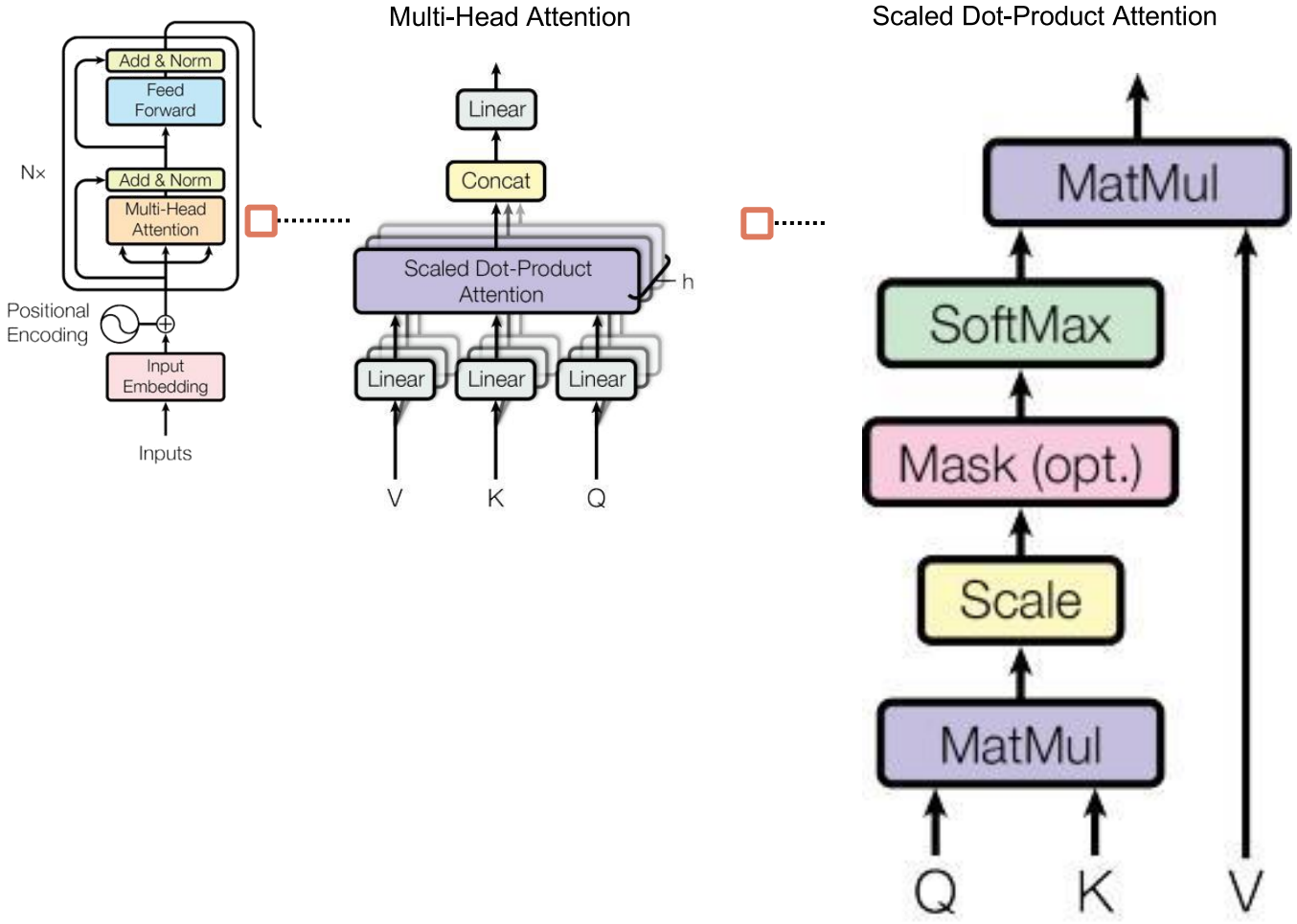
Transfomer 은 데이터를 한번에 입력하기 때문에, RNN 모델과 같이 토큰의 위치를 알 수 없다.
따라서, 위치를 기억해줄 수 있는 encoding 을 넣어준다.

이때, positional encoding의 길이는 임베딩 차원과 같다.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

03. Model Architecture

3.2. Scaled Dot-Product Attention



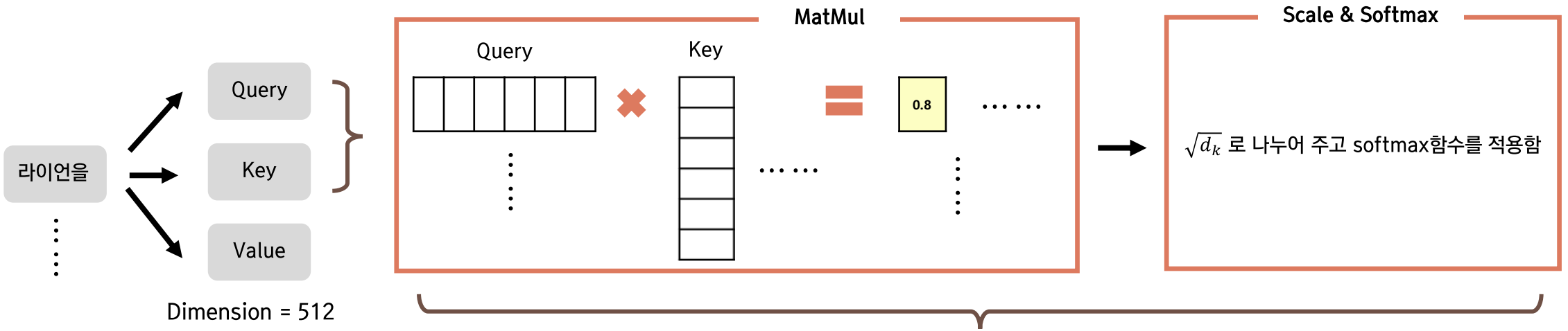
Query : 질문하는 값
Key : Query의 값을 기준으로 찾는 대상
Value : Key 가 갖고있는 값

$$Attention(Q, K, V) = softmax\left(\frac{QK^T)}{\sqrt{d_k}}\right)V$$

03. Model Architecture

3.2. Scaled Dot-Product Attention

$$Attention(Q, K, V) = softmax\left(\frac{QK^T)}{\sqrt{d_k}}\right)V$$

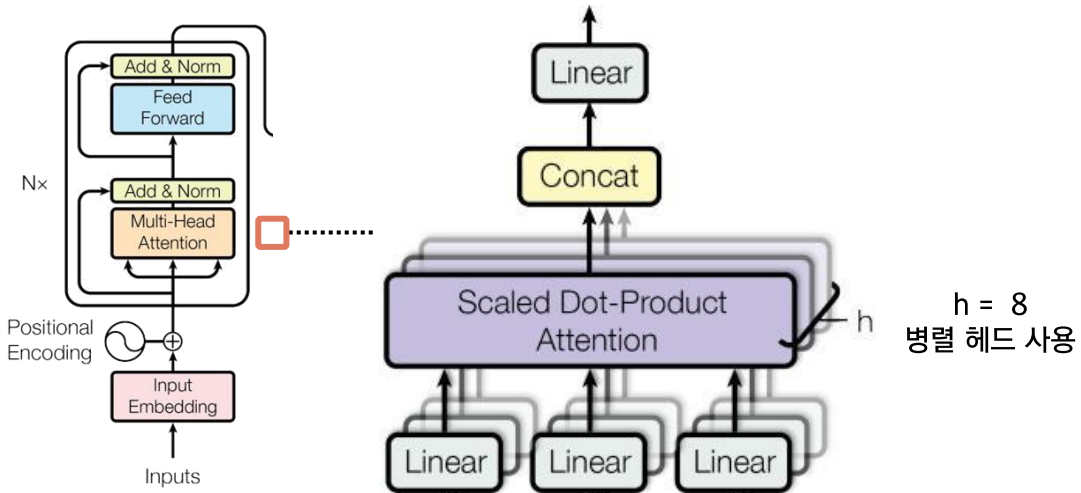


MatMul Softmax 를 거쳐서 나옴 attention weights 를 Value 와 내적을 해주면 Attention이 나옴

** Mask (옵션) : 토큰의 갯수가 부족할 때, 값을 0으로 채워서, 가중치 구하는 것을 막아줌

03. Model Architecture

3.3. Multi-Head Attention



h = 8
병렬 헤드 사용

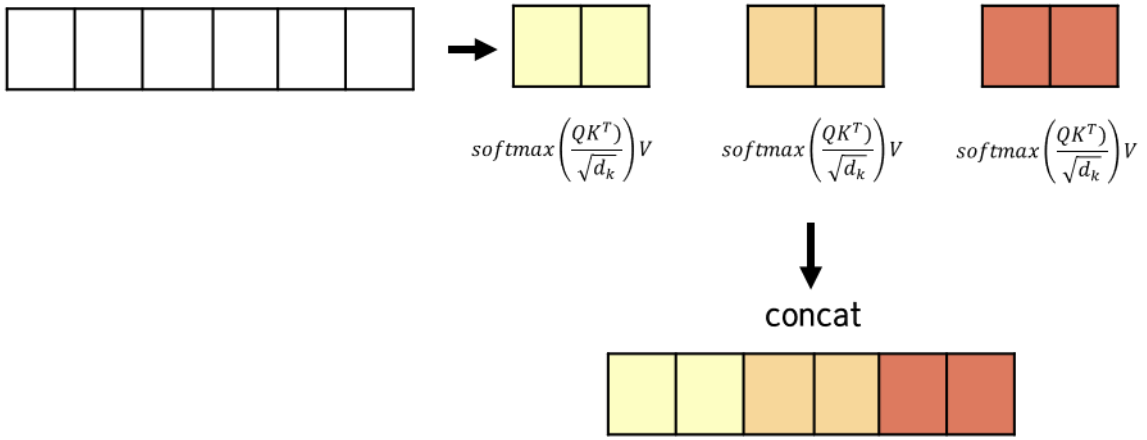
Multi-Head Attention

논문에서는 512의 차원을 8로 나누어주기 때문에, d_k = 64 로 설정됨

각 64차원으로 나누어진 값들에 Scaled Dot-Product Attention 수행

-> concat 을 하여 값들을 합침

예를 들어, 6차원을 num_head = 3 으로 병렬 처리해주면



$$MultiHead(Q,K,V) = Concat(head_1,\cdots,head_h)W^O$$

$$where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

03. Model Architecture

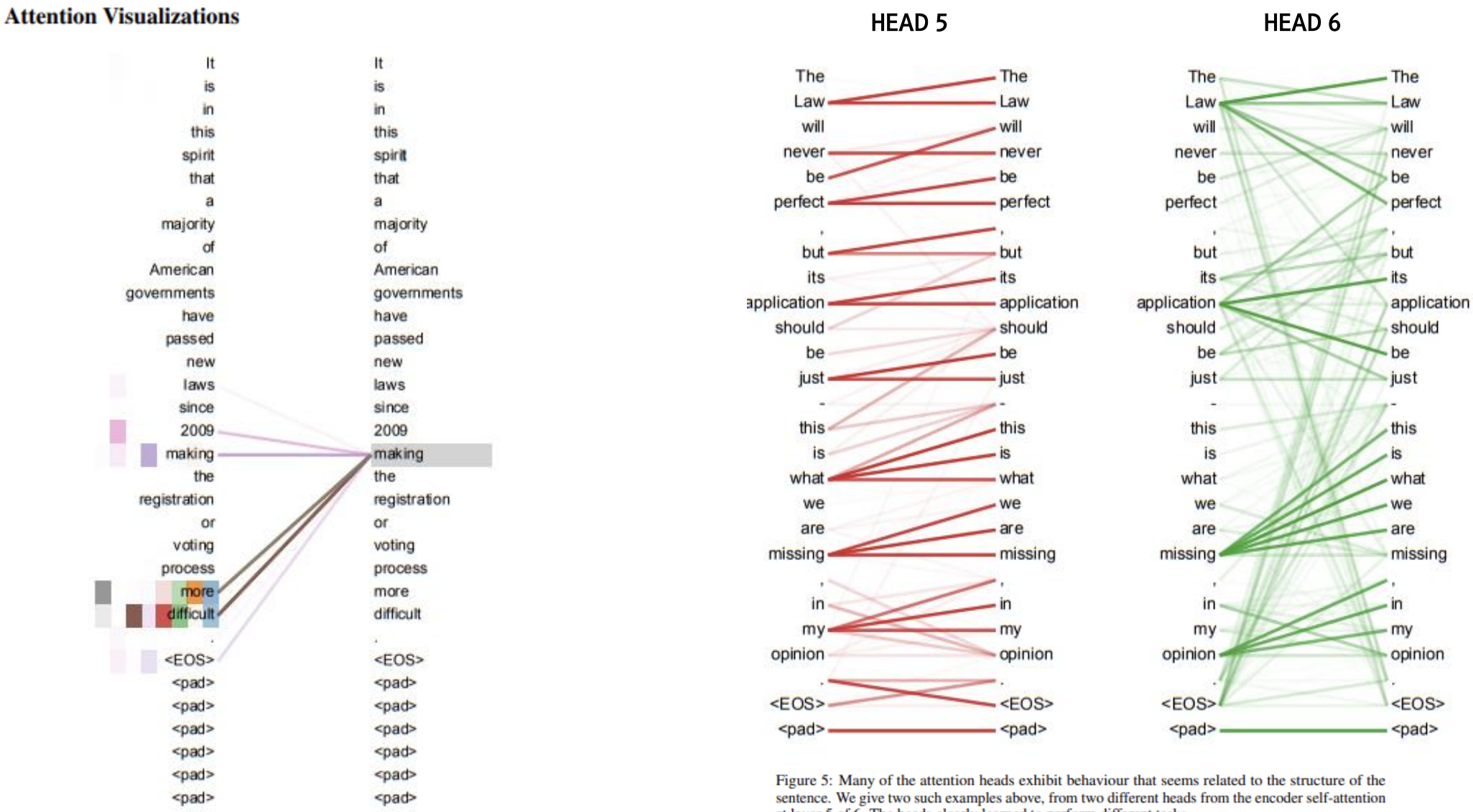
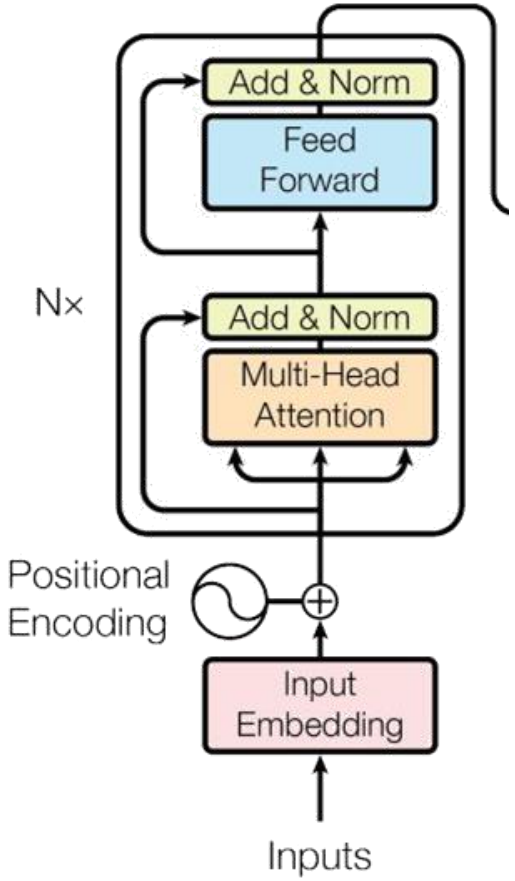
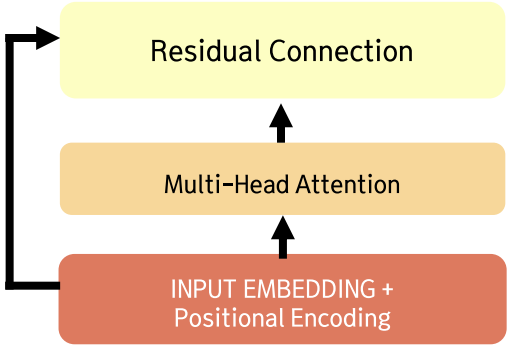


Figure 5: Many of the attention heads exhibit behaviour that seems related to the structure of the sentence. We give two such examples above, from two different heads from the encoder self-attention at layer 5 of 6. The heads clearly learned to perform different tasks.

03. Model Architecture



Residual Connection & Layer Norm



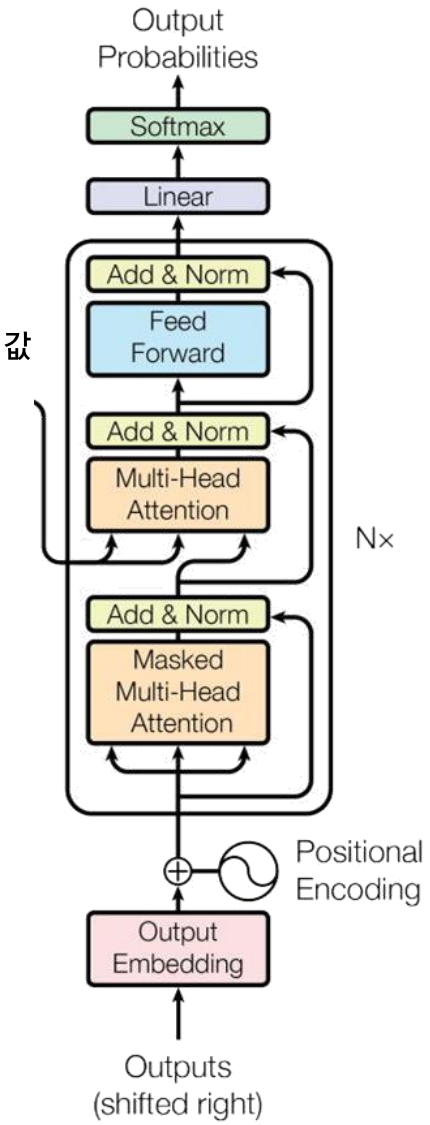
Feed Forward Network

$$FFN(x) = \max(0, xW_1 + b_1)\,W_2 + b_2$$

FFN 은 2개의 선형 변환으로 구성되어있으며, 활성함수는 ReLU를 이용한다.

03. Model Architecture

3.4. decoder 의 sub-layer



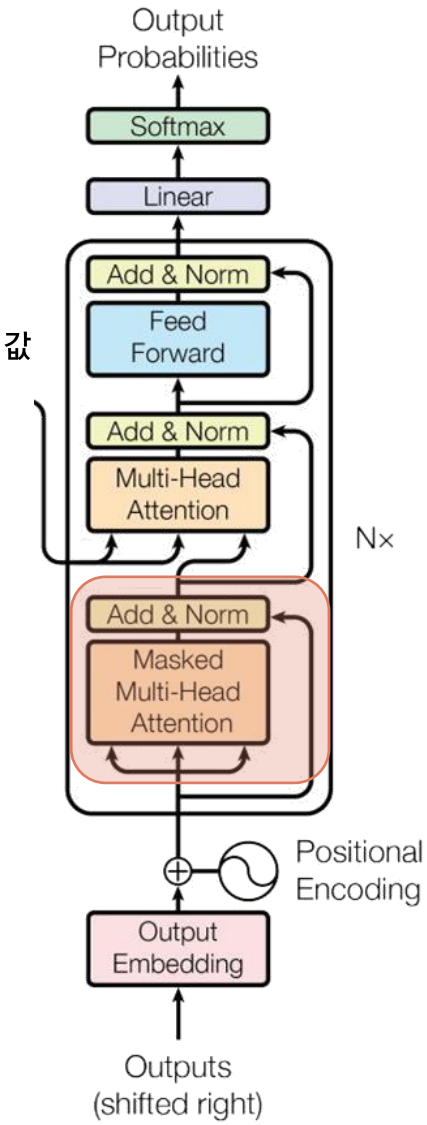
N layer 의 output 값

논문에서 layer 는 6개

- 1st sub-layer : Masked Multi-Head Attention [Masked Decoder Self - Attention]
- 2nd sub-layer : multi-head Attention [Encoder - Decoder Attention]
- 3rd sub-layer : position-wise fully connected feed-forward network

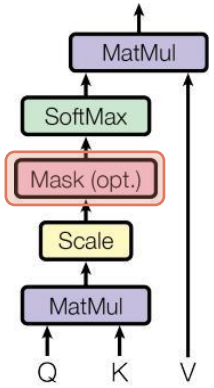
03. Model Architecture

3.4. decoder 의 sub-layer : Masked Multi-Head Attention



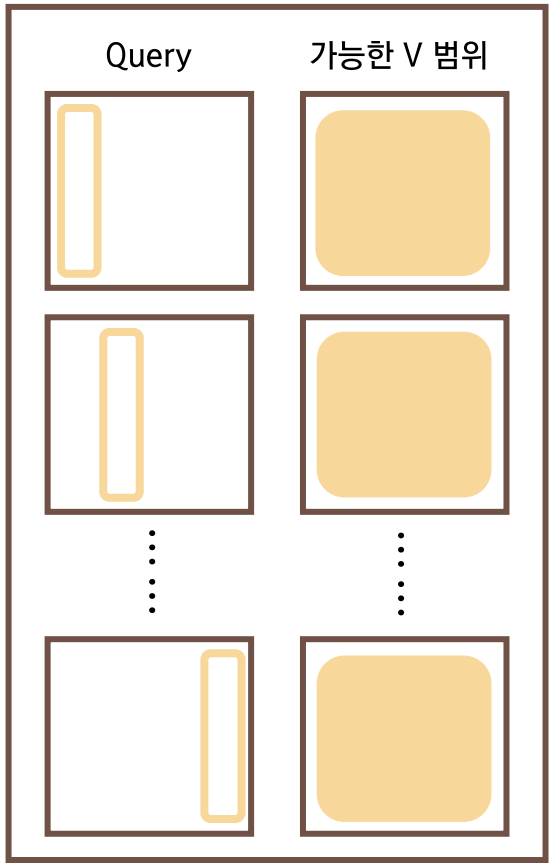
N layer 의 output 값

논문에서 layer 는 6개

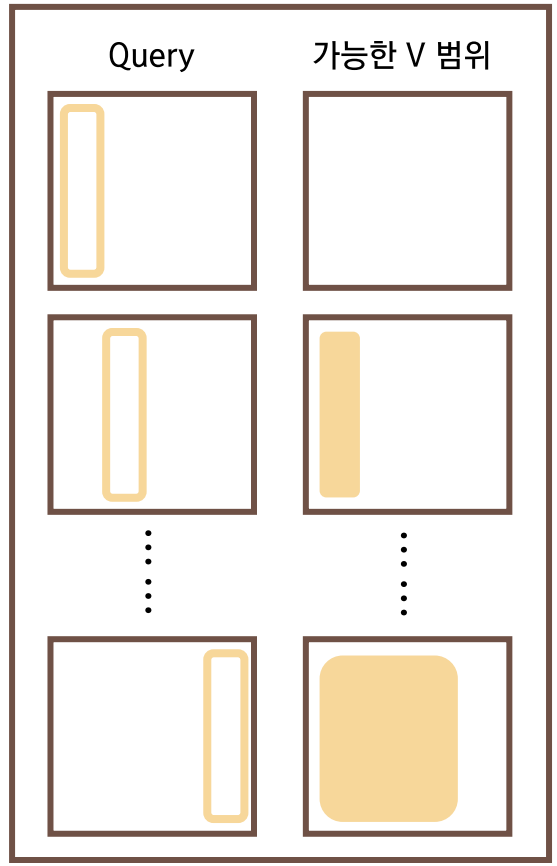


1st sub-layer : Masked Multi-Head Attention [Masked Decoder Self - Attention]

Encode Masking



Decoder Masking

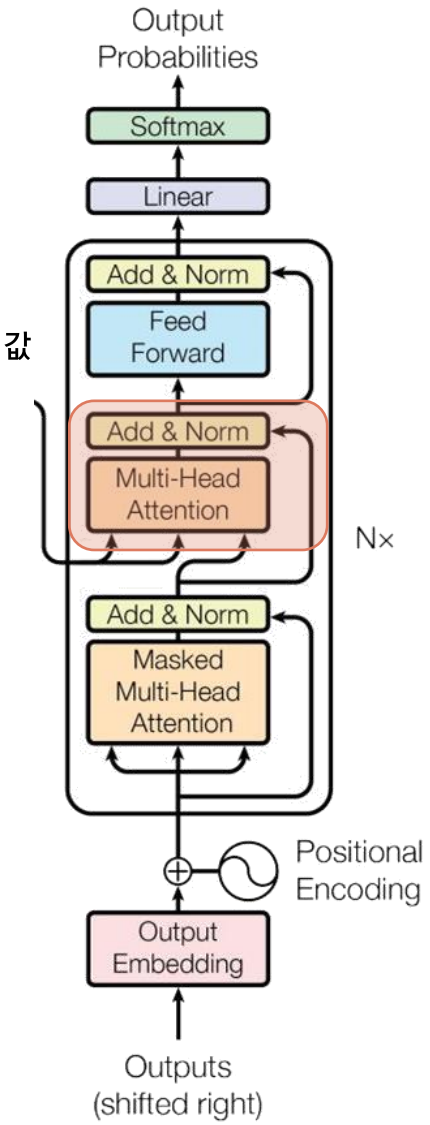


Decoder masking은 자기 자신을 포함한 미래값과는 Attention을 구하지 않음

Encoder / Decoder masking 모두, padding token 의 위치에 대한 making진행

03. Model Architecture

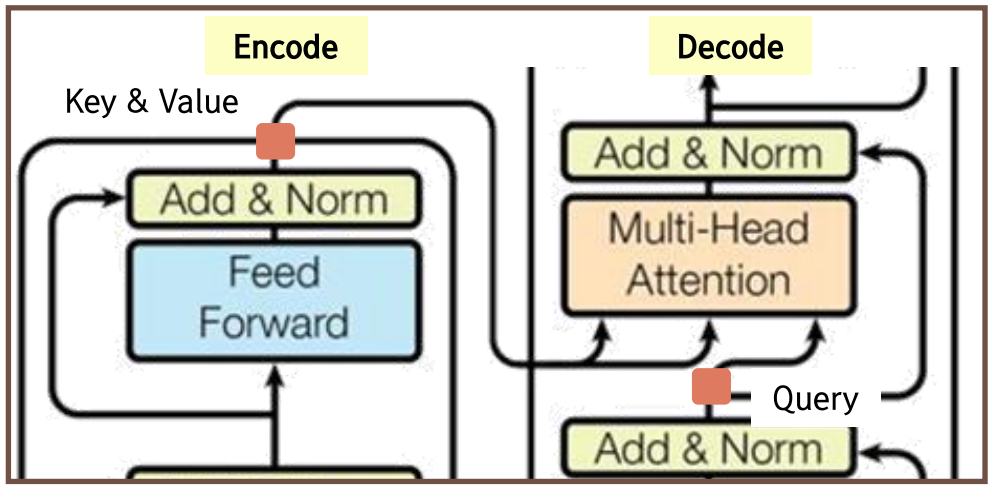
3.4. decoder 의 sub-layer : Encoder - Decoder Attention



N layer 의 output 값

논문에서 layer 는 6개

- 2nd sub-layer : multi-head Attention [Encoder - Decoder Attention]



Encoder 의 outpu이 decode의 2nd sub-layer의 key와 value 가 됨
Decoder의 self-attention 의 결과를 2nd sub-layer 의 query로 사용함

04. Why Self-Attention

1. 레이어당 총 계산 복잡도가 감소한다.

2. 병렬화 할 수 있는 계산량이 증가한다.

3. 장거리의 값들도 잘 학습이 된다.

4. 해석이 가능한 모델을 얻을 수 있다.

* Layer Type : 레이어당 총 계산 복잡도 / 병렬화할 수 있는 계산량 / 네트워크 장거리 경로 길이

** self-attention (restricted) : 매우 긴 시퀀스 관련 작업할 때, 계산 성능 향상을 위해 attention크기를 이웃 r 로 제한 할 수있다.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

05. Training & Results

Training data

- WMT 2014 English-German dataset 4.5 million sentence pairs
- WMT 2014 English-French dataset 26M sentence

Optimizer

- Adam Optimizer $lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$

Regularization

- Residual Dropout
- Label Smoothing

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	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

05. Training & Results

Model Variations

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

06. Conclusion

In this work, we presented the Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention.

For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers. On both WMT 2014 English-to-German and WMT 2014 English-to-French translation tasks, we achieve a new state of the art. In the former task our best model outperforms even all previously reported ensembles.

We are excited about the future of attention-based models and plan to apply them to other tasks. We plan to extend the Transformer to problems involving input and output modalities other than text and to investigate local, restricted attention mechanisms to efficiently handle large inputs and outputs such as images, audio and video. Making generation less sequential is another research goals of ours.

- ✔ Transfomer architecture 은 오직 'attention mechanisms' 을 사용한 것
- ✔ RNN, CNN을 기반으로한 모델보다 빠르고 좋은 성능을 가짐
- ✔ 다른 분야에도 적용될 것을 기대함