"Attention is all you need"

트랜스포머(Transformer)

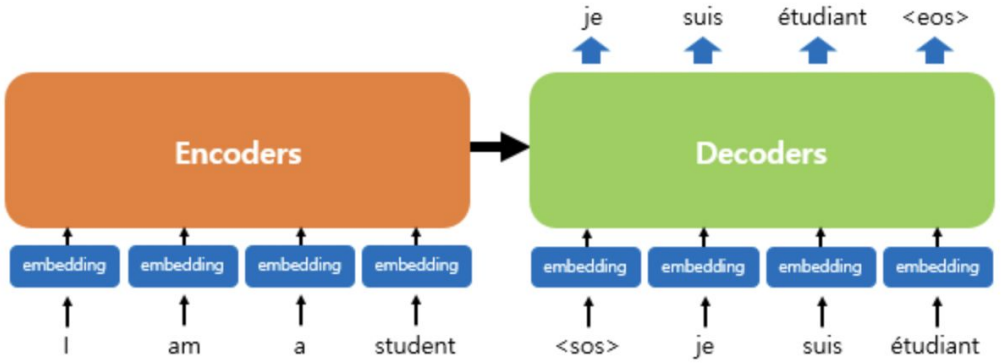
Transformer

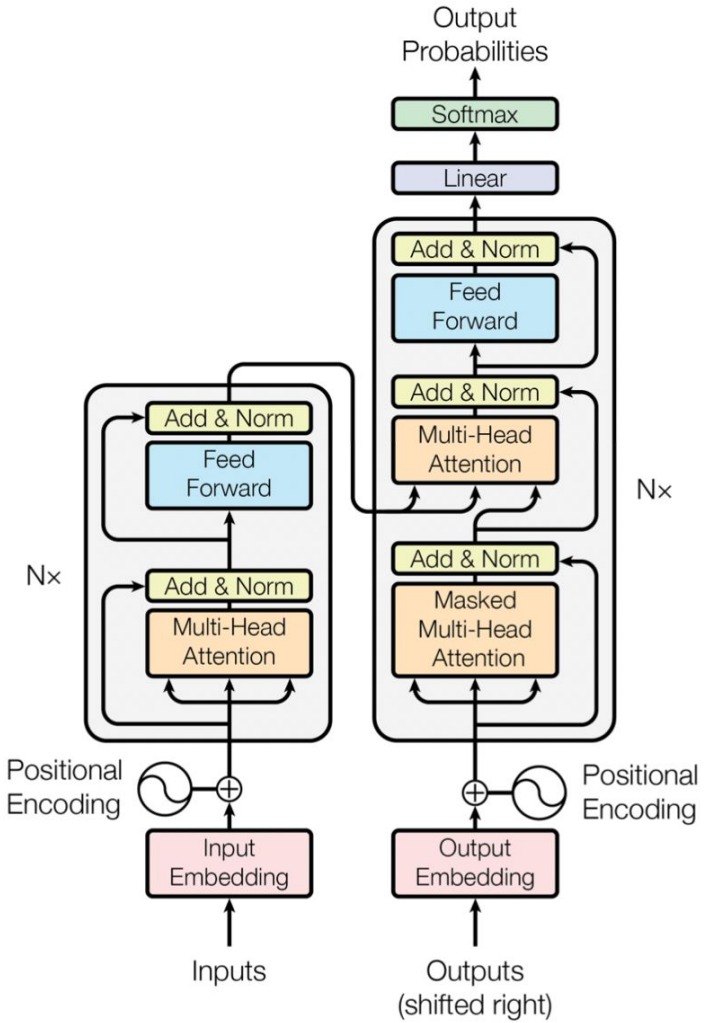
•기존의 seq2seq의 구조인 인코더-디코더를 따르면서,
어텐션(Attention)만으로 구현한 모델

•RNN(순환 신경망)을 사용하지 않고, 인코더-디코더 구조를 설계하였음에도 번역 성능에서 RNN보다 우수한 성능

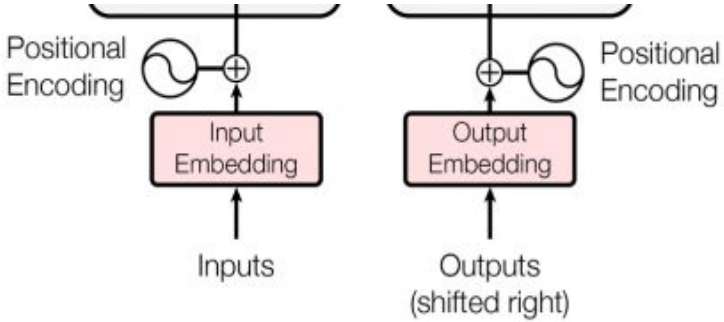
1. Model Architecture

- 인코더와 디코더가 각각 여러 개(N개) 쌓여있는 구조 (논문에서는 6개)
- 인코더에서 입력받고, 디코더에서 출력받는 인코더-디코더 구조





(1) Positional Encoding



- RNN이나 CNN을 사용하지 않기 때문에 위치 정보가 없음

- 임베딩 벡터가 인코더의 입력으로 사용되기 전 포지셔널 인코딩값이 더해지는 과정
- 사인 코사인 함수 사용

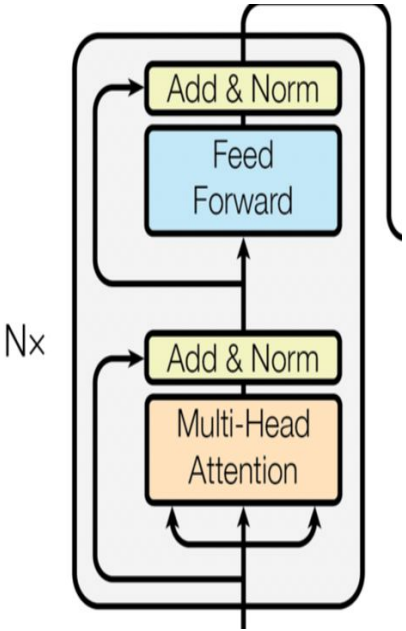
$$PE_{(pos,\,2i)} = sin(pos/10000^{2i/d_{model}})$$

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

=> 순서 정보 보존

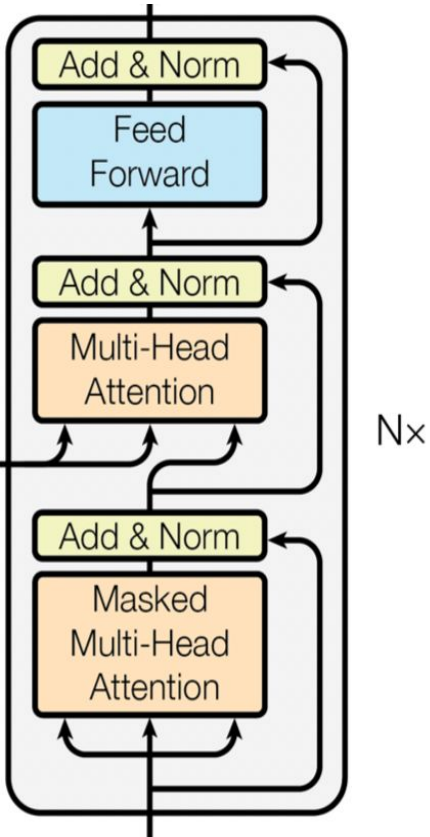
같은 단어이더라도 문장 내의 위치에 따라 입력으로 들어가는 임베딩 벡터의 값이 달라짐

(2) Encoder and Decoder Stacks - Encoder



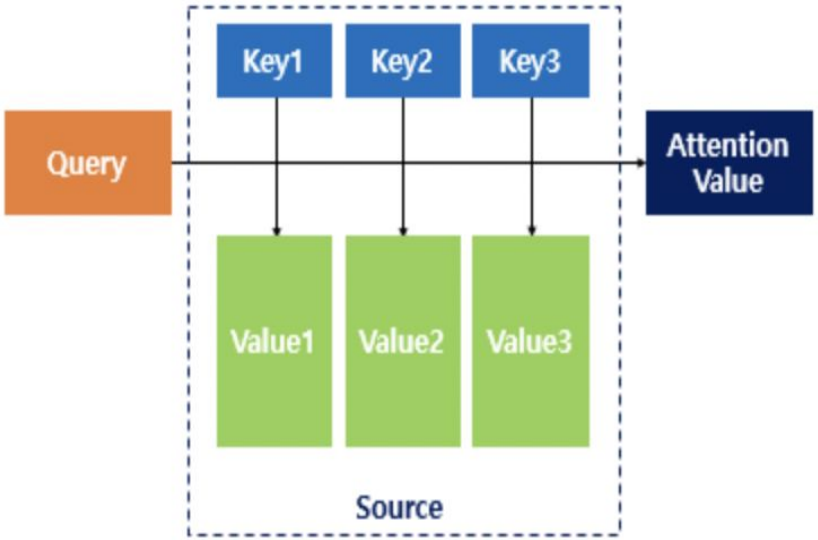
- **N** = 6 (6개의 인코더 층) (**하이퍼파라미터**)
- 하나의 인코더 - 2개의 서브층
  (멀티헤드셀프어텐션 / 포지션와이즈피드포워드 신경망)
- Add&Norm : residual connection(잔차연결)/
  layer normalization(층 정규화)
- Output of each sub-layer : LayerNorm(x + Sublayer(x))
- 모든 서브층 동일한 크기 : 512 (**dmodel-임베딩차원, 하이퍼파라미터**)

(3) Encoder and Decoder Stacks - Decoder



- N = 6 (6개의 디코더 층)
- 하나의 디코더 - 3개의 서브층(첫번째 서브층 추가)
- Add&Norm : residual connection/layer normalization
- 첫번째 서브층 - 인코더의 Multi-Head Attention과 다른점: **look-ahead Masking** (현재 시점의 예측에서 현재 시점보다 미래에 있는 단어들을 참고하지 못하도록)
- 모든 디코더의 layer의 두번째 sublayer
  (인코더와 Multi-Head attention인 것은 같지만 셀프 어텐션이 아님)
  -> 인코더의 마지막 layer출력값을 받아 입력
  (각각의 출력 단어가 입력 단어중에서 어떤정보와 가장 높은 연관성 가졌는지 계산)

2. Attention



Attention function :

주어진 '쿼리(Query)'에 대해서 모든 '키(Key)'와의 유사도를 각각 구한다.

그리고 이 유사도를 가중치로 하여 키와 맵핑되어있는 각각의 '값(Value)'에 반영한다.

그리고 유사도가 반영된 '값(Value)'을 모두 가중합하여 리턴한다.

(1) Attention

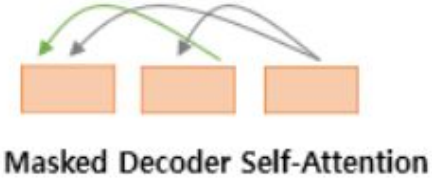
1. Encoder self-attention

   어텐션을 자기 자신에게 수행, 입력 문장 내의 단어들끼리 유사도 구함

2. Masked decoder self attention

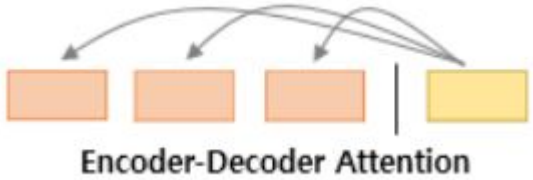
   출력하려는 단어 앞쪽에 나온 단어만 참고할 수 있도록 masking

   (나는 축구를 했다 - 축구를 -> '나는'만 참고 가능)

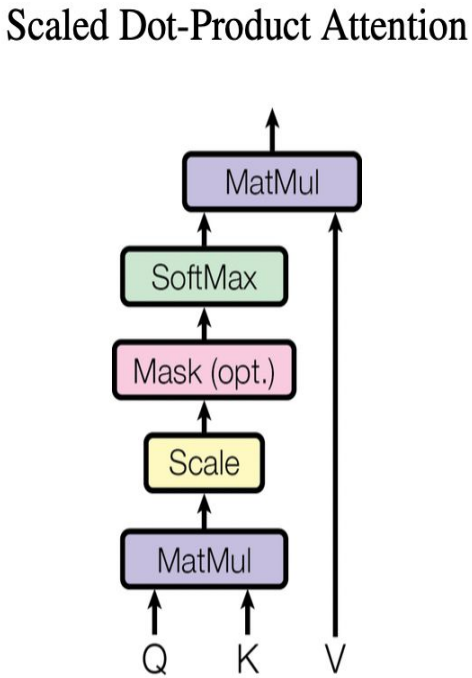


3. Encoder-decoder attention

Q만 디코더/ V,K는 인코더 (디코더에 있는 Q가 인코더의 V,K참고)



(2) Attention - Scaled Dot-Product Attention



- 어텐션 함수로 score(q,k) = q*k/**루트dk** 사용

$\frac{1}{\sqrt{d_k}}$: **Scaling factor**

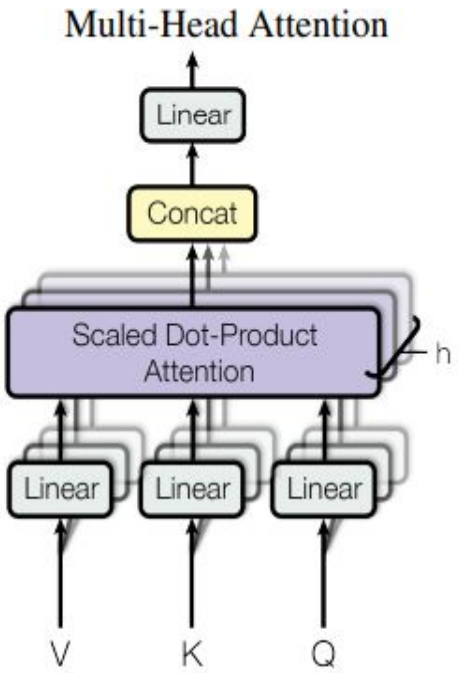
(dk: input of queries and keys of dimension**)**

- Matrix of outputs:

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

- Masking: 특정 단어를 무시하기 위해 매우 작은 음수값(-무한대에 가까운 수)를 넣어 스포트맥스 함수를 지나게 함

(3) Attention - Multi-Head Attention



- 여러번의 어텐션을 병렬로 사용- >각각 다른 학습내용
- dmodel의 차원을 h개로 나누어

  dmodel/h의 차원을 가지는 Q,K,V에 대해서

  h개의 병렬 어텐션 수행 (**h** = 8, 512/8 = 64)

  연결-> $\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1,...,\mathrm{head}_\mathrm{h})W^O$

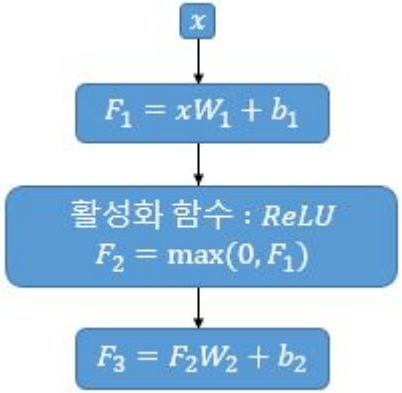
  $\mathrm{where\ head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

- MultiHead attention 수행 뒤에도 dimension 동일하게 유지

(4) Position-wise Feed-Forward Networks

- 인코더와 디코더에서 공통적으로 가지고 있는 서브층
- Fully connected feed-forward network

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



two linear transformations with a ReLU activation in between

- 매개변수 w,b는 하나의 인코더 층 내에서는 동일하게 사용,

  인코더 층 마다는 다른 값

- **dmodel** = 512 (입력층, 출력층 크기) / **dff** = 2048 (은닉층 크기)

2. Why Self-Attention

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Self-attention - 모든 위치를 일정한 수의 순차적으로 실행되는 작업으로 연결

Recurrent - O(n)개의 순차적 작업이 필요

n 이 d보다 작을 때 self-attention < Recurrent (최신모델이 사용하는 문장의 경우)

self-attention은 크기 r 의 이웃만 고려하도록 제한 - 향후 연구

=> Self-attention could yield more interpretable models

3. 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 [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	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**	**41.0**	$2.3 \cdot 10^{19}$	

- 영어-독일어 번역

기존 최고 BLEU 26.36 -> 28.4

cost (base model)도 가장 적음

- 영어-프랑스 번역

BLEU 41 달성, 모든 단일 모델 능가

이전 최소 cost의 1/4