KU KONKUK UNIVERSITY

Transformer

본 챕터의 내용은 "Ta-Chun (Bgg/Gene) Su's blog (https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-cf81bf32c73d)" 자료를 참고하여 편집되었습니다.

Attention Models

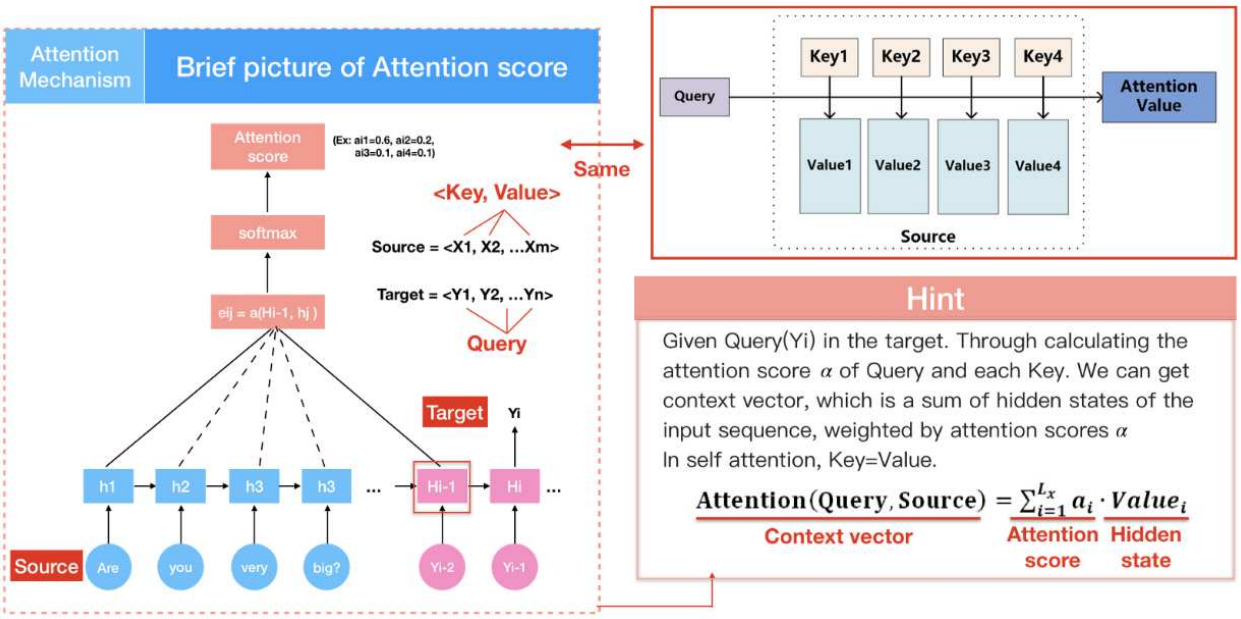


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Edited by Harksoo Kim

Attention Models in Detail

Decoder Formula

The context vector c_i is, then, computed as a weighted sum of these annotations h_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$ ↔ $\mathbf{Attention(Query, Source)} = \sum_{i=1}^{L_x} a_i \cdot Value_i$ (Attention vector; Attention score; Hidden state) — Stage 3

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$, ↔ $a_i = Softmax(Sim_i) = \frac{e^{Sim_i}}{\sum_{j=1}^{L_x} e^{Sim_j}}$ (Attention score) — Stage 2

where $e_{ij} = a(s_{i-1}, h_j)$ ↔ $\mathbf{Similarity(Query, Key_i) = Query \cdot Key_i}$ *dot* — Stage 1

is an *alignment model* which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} (just before emitting y_i, Eq. (4)) and the j-th annotation h_j of the input sentence.

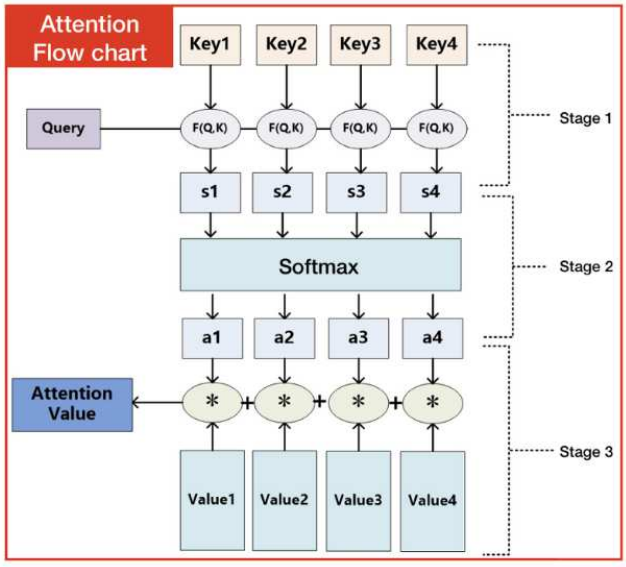


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Edited by Harksoo Kim

Problems of Attention-Based Models

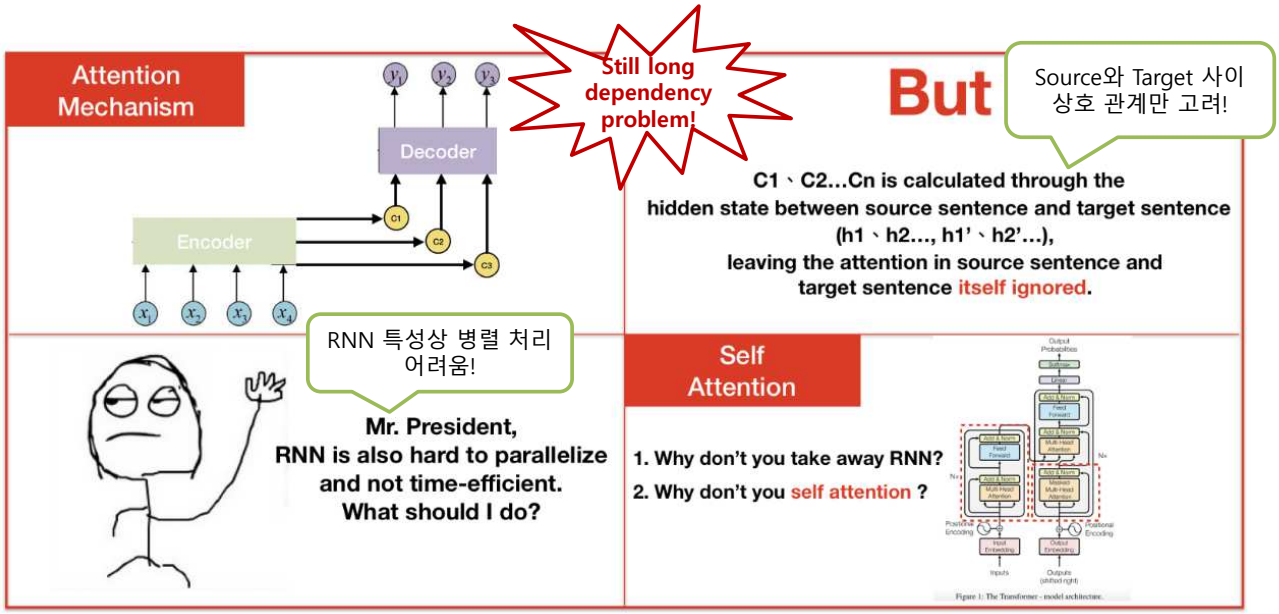


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Transformer

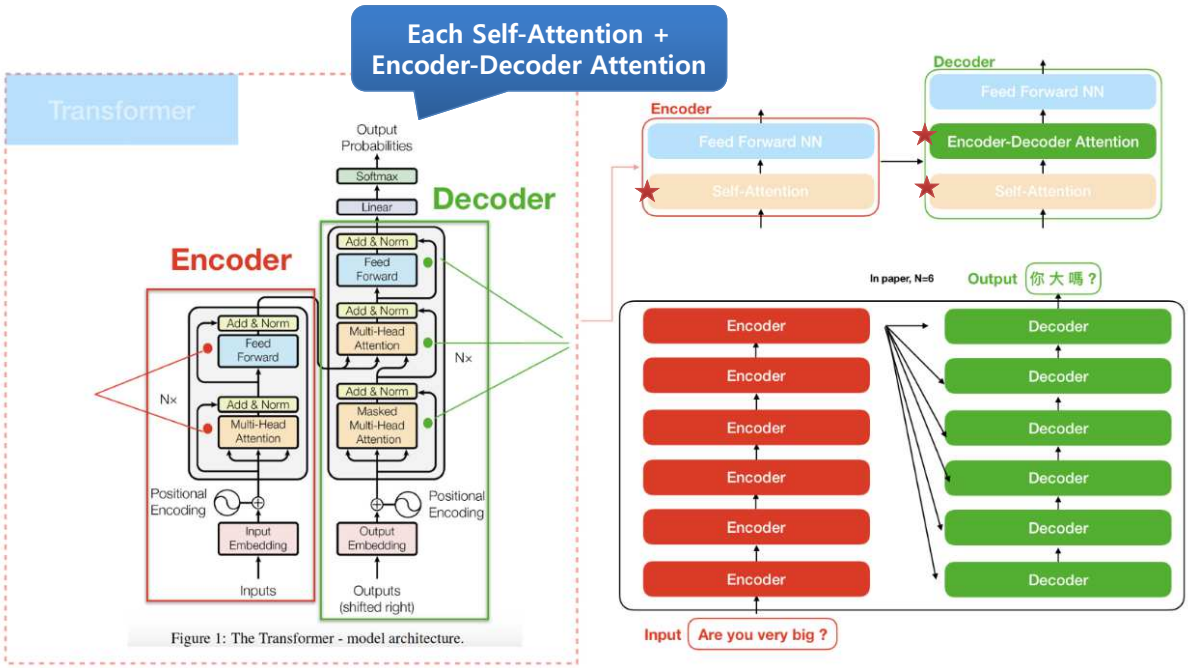


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Attentions in Transformer

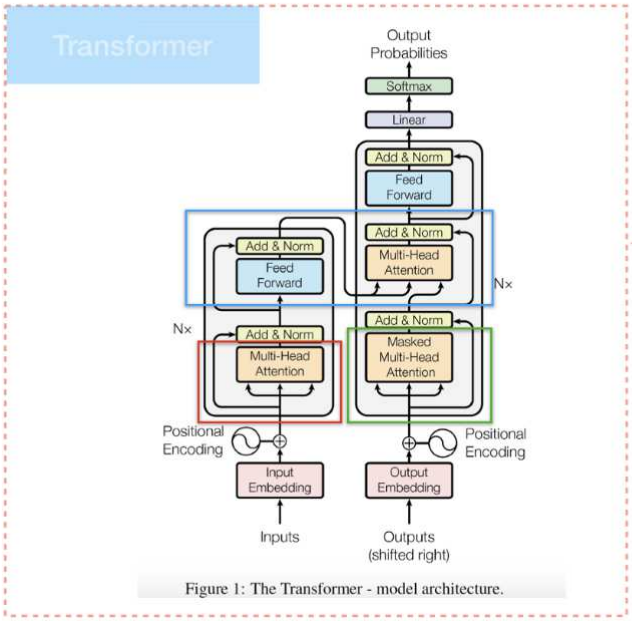


Figure 1: The Transformer - model architecture.

encoder self attention

1. Multi-head Attention
2. Query=Key=Value

decoder self attention

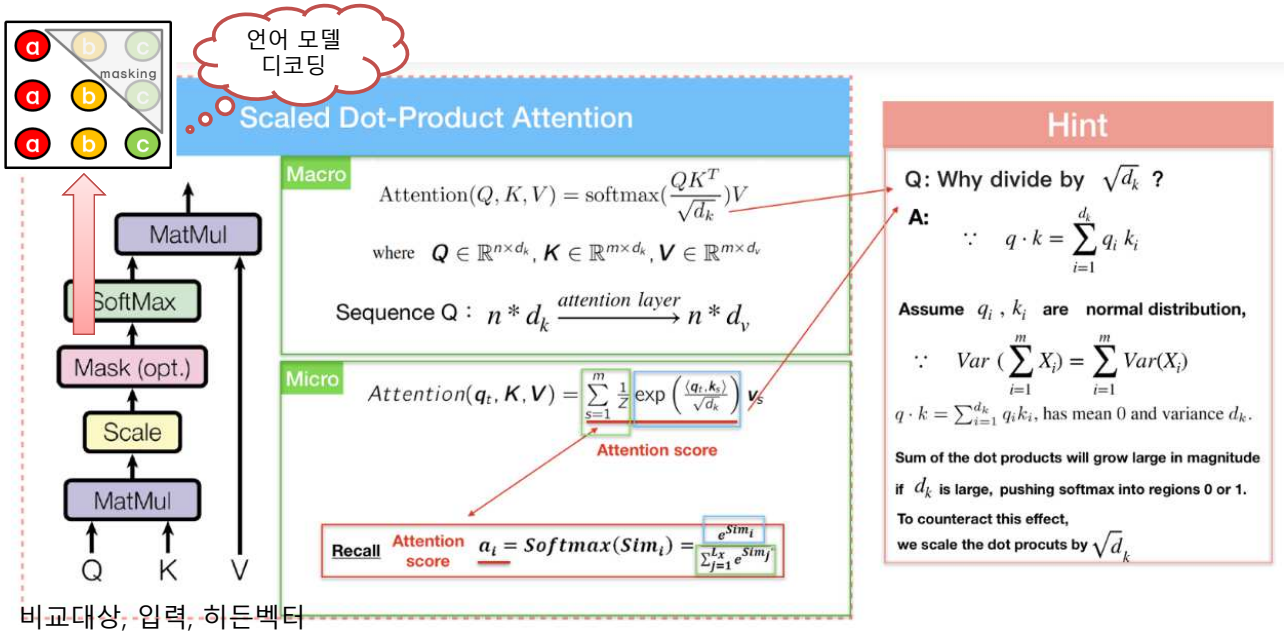
1. Masked Multi-head Attention
2. Query=Key=Value

encoder-decoder attention

1. Multi-head Attention
2. Encoder Self attention=Key=Value
3. Decoder Self attention=Query

그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Scaled Dot-Product Attention



비교대상, 입력, 히든벡터

그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Calculation of Attentions

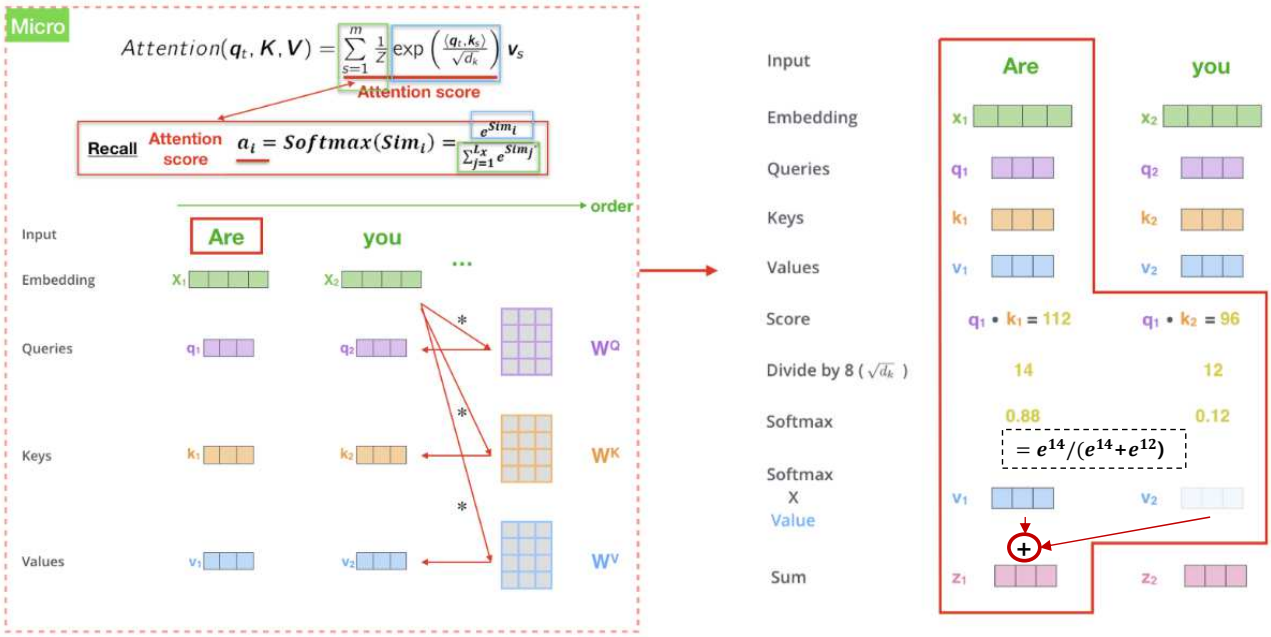


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Calculation of Attentions

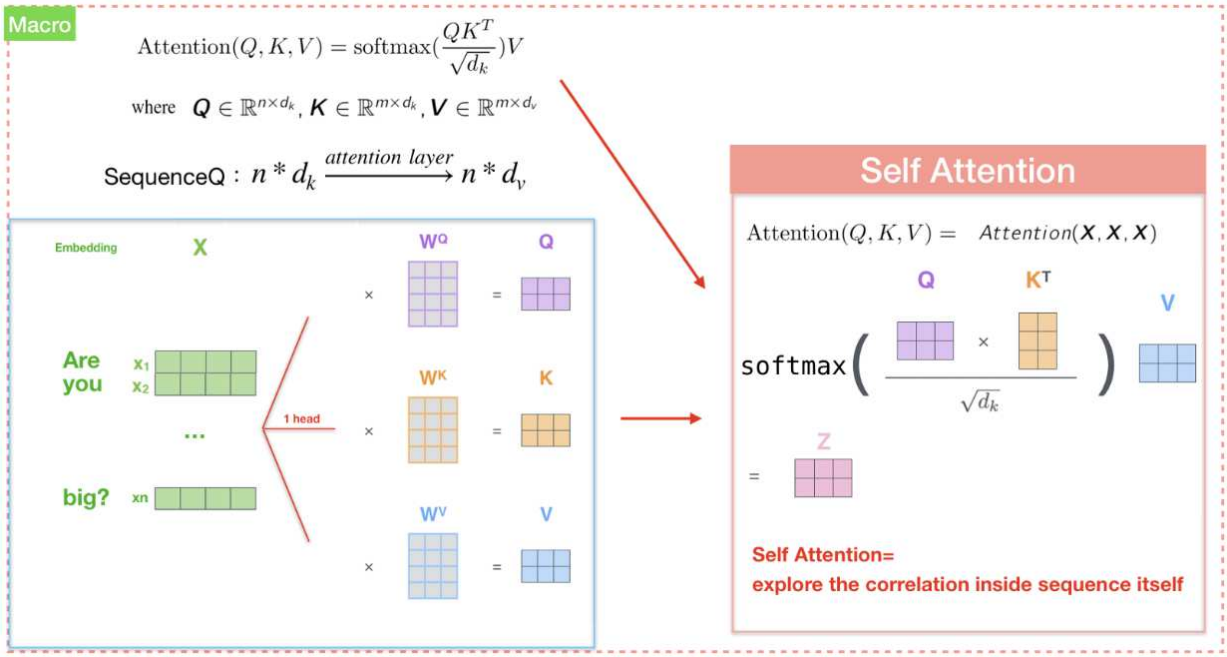


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

확인 문제

- 다음과 같이 단어 임베딩이 주어졌을 때, self-attention score를 계산하시오. (소수점 이하 두 자리에서 반올림)
  - are: [1,1], you: [2,1]
  - root(2)=1.4로 계산

$$Attention(q_t, K, V) = \sum_{s=1}^{m} \frac{1}{Z} \exp\left(\frac{(q_t, k_s)}{\sqrt{d_k}}\right) v_s$$

Scaled dot-product

	are	you
are	?	
you		

Self-attention score

	are	you
are	?	
you		

Edited by Harksoo Kim

Multi-Head Attention

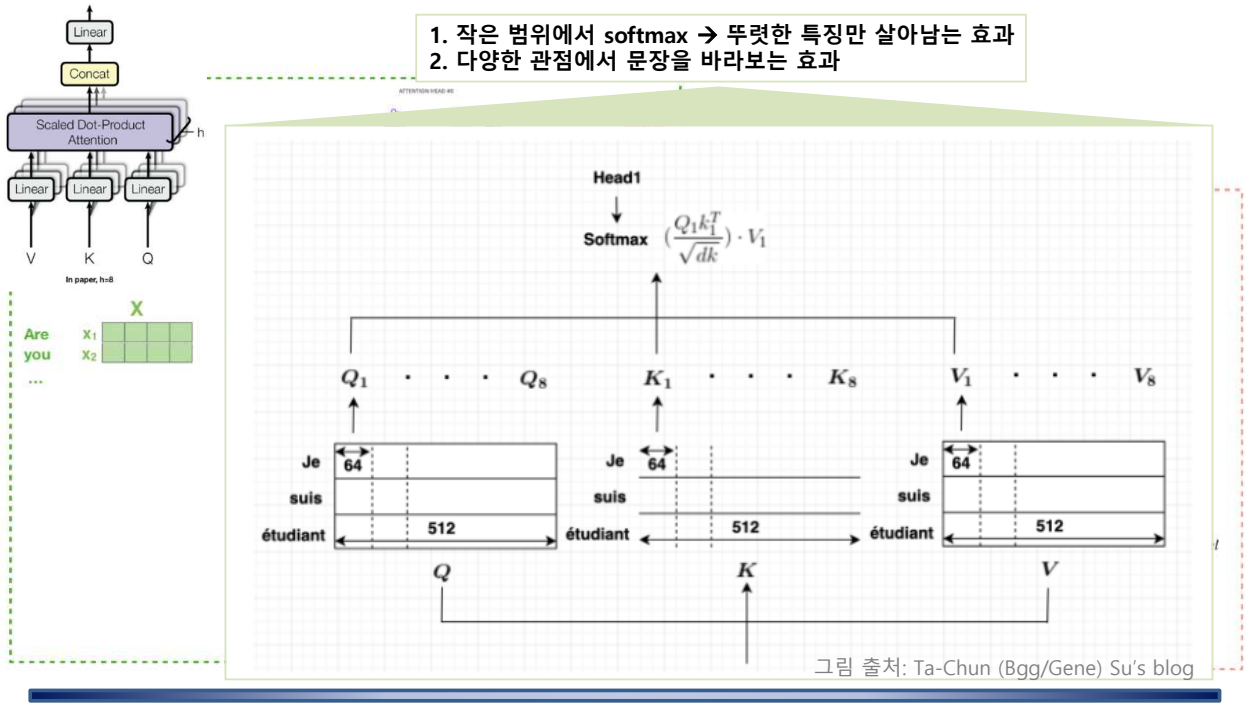


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Edited by Harksoo Kim

Layer Norm. & Residual Conn.

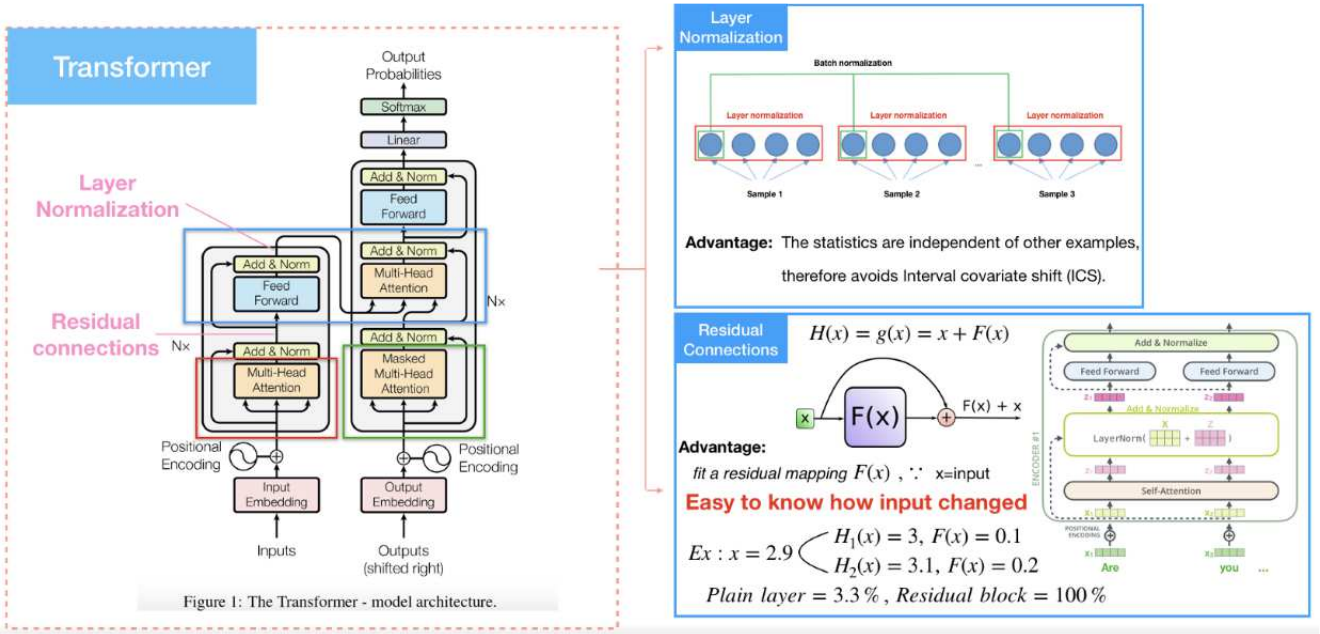


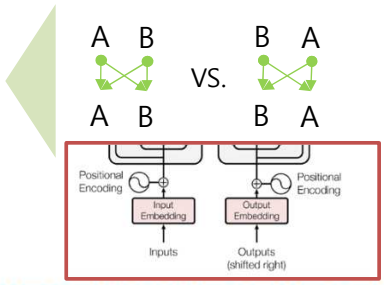
그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Edited by Harksoo Kim

Position Encoding

Problem

The multi-head attention network **cannot** naturally **make use of** the position of the words in the input sequence.
The output of the multi-head attention network would be **the same** for the same sentences in different order.



Sol **Positional Encoding**

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

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

if shape of enc, dec $= [T, d_{model}]$ **pos: position of the word**

→ *then* $pos \in [0, T)$, $i \in [0, d_{model})$ **i: Element i in** d_{model}

Advantage:

PE[pos+k] can be represented as a linear function of **PE[pos]**, so the relative positon between different embeddings can be easily inferred

$sin(\alpha+\beta) = sin\alpha cos\beta + cos\alpha sin\beta$

$cos(\alpha+\beta) = cos\alpha cos\beta - sin\alpha sin\beta$

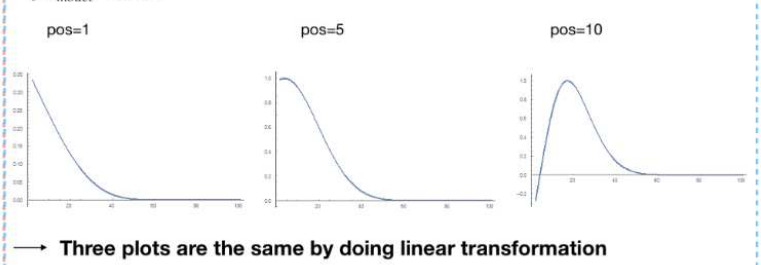
∵ **Trigonometric Periodicity**

∴ **Learned the relation between relative and absolute position**

Then Word embedding + positional encoding

(sum, concat...)

if $d_{model} = 512$:



→ **Three plots are the same by doing linear transformation**

그림 출처: Ta-Chun (Bgg/Gene) Su's blog

Edited by Harksoo Kim

Linear & Softmax

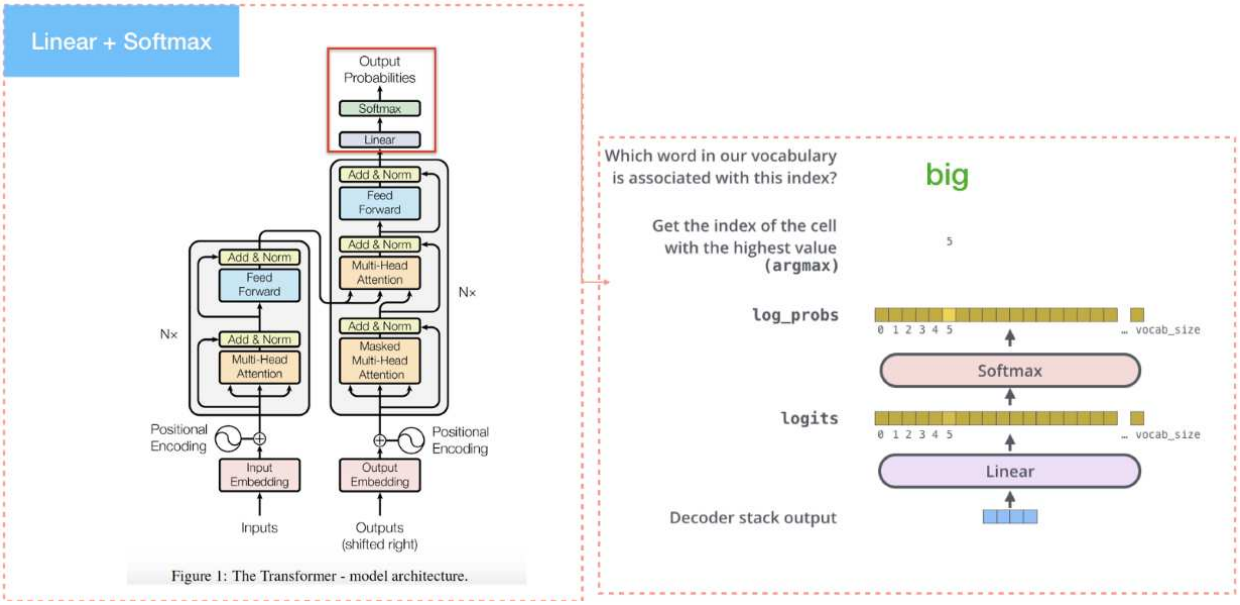


그림 출처: Ta-Chun (Bgg/Gene) Su's blog

질의응답

Q&A

Homepage: http://nlp.konkuk.ac.kr
E-mail: nlpdrkim@konkuk.ac.kr