Transformer (Attention Is All You Need)

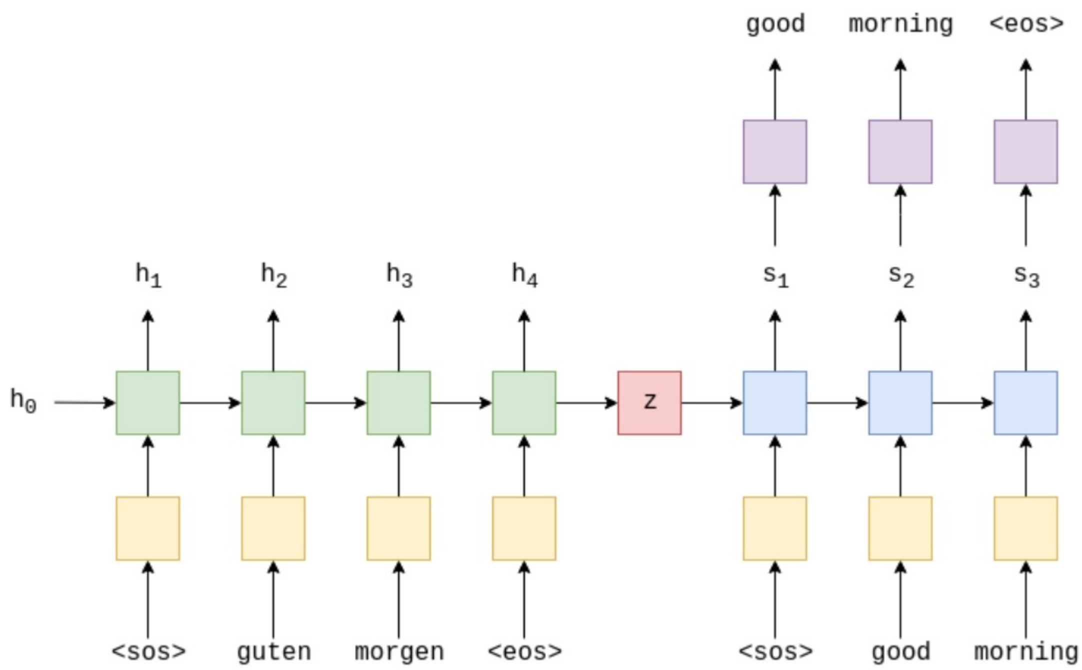
2024.3.26

Jaeho Yang

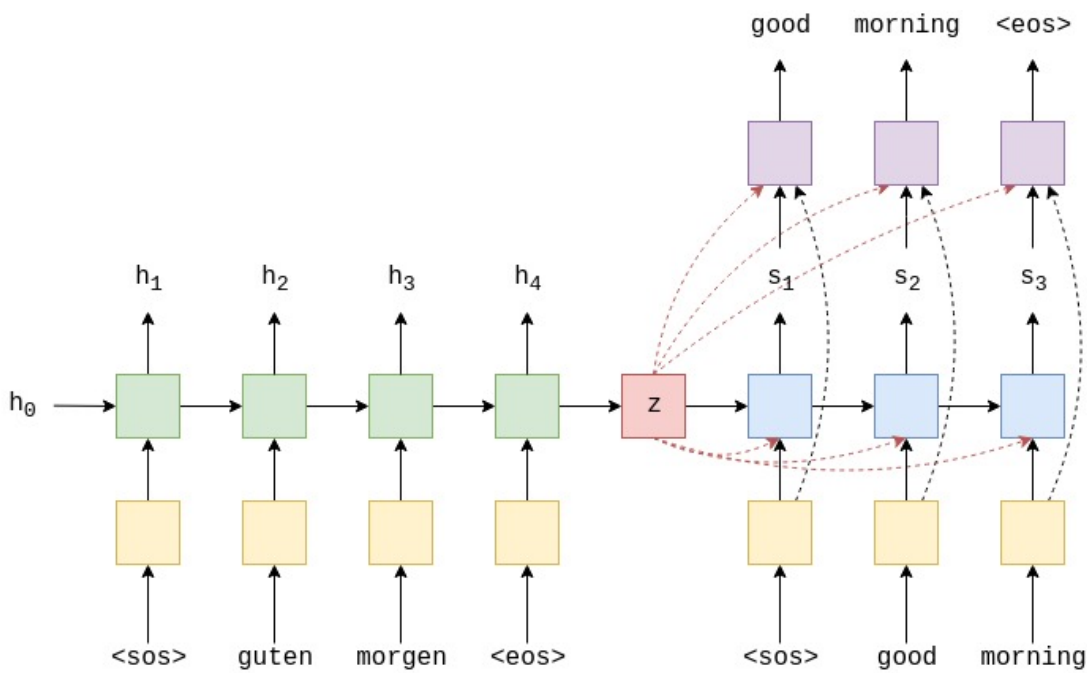
yangwogh@yonsei.ac.kr

Computer Engineering

- Disadvantage of seq2seq
  - The sentence is compression at Context vector
  - It have a problem because of bottleneck



- New concept of seq2seq
  - Every time check Context Vector
  - Therefore, even if sentence becomes longer, information of words are in the Context Vector can be added again.
  - It still have a problem about bottleneck

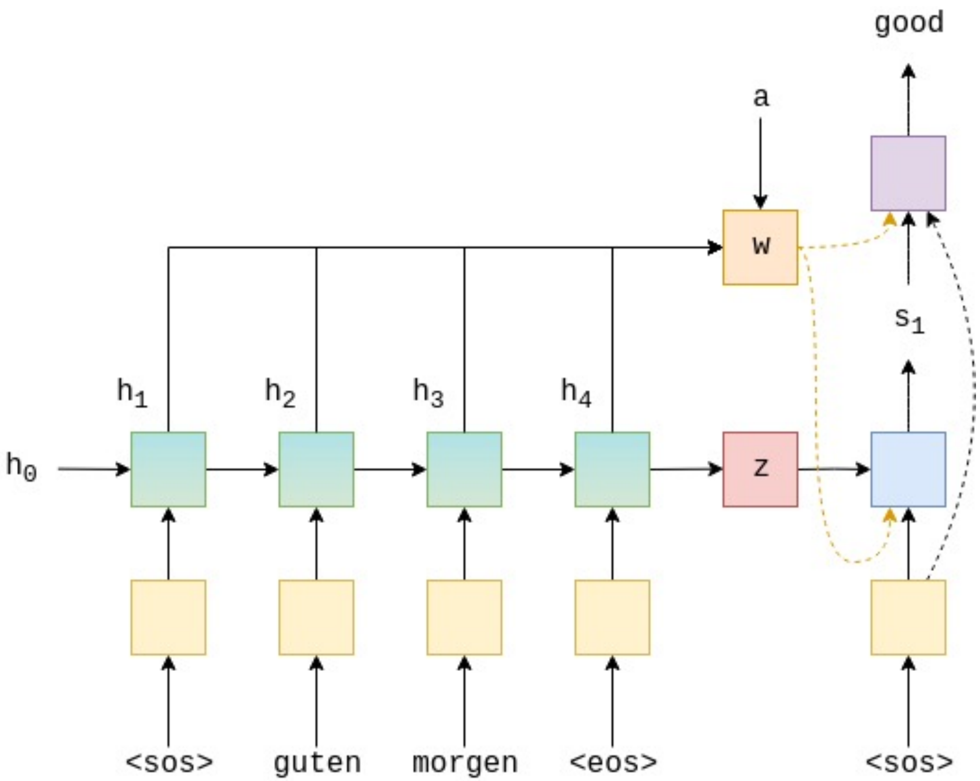


■ Seq2seq with Attention

- Problem : One of Context Vector must get all of sentence’s information.
- How about get all source sentence to output every time.
- It is possible because GPU performance is good enough.

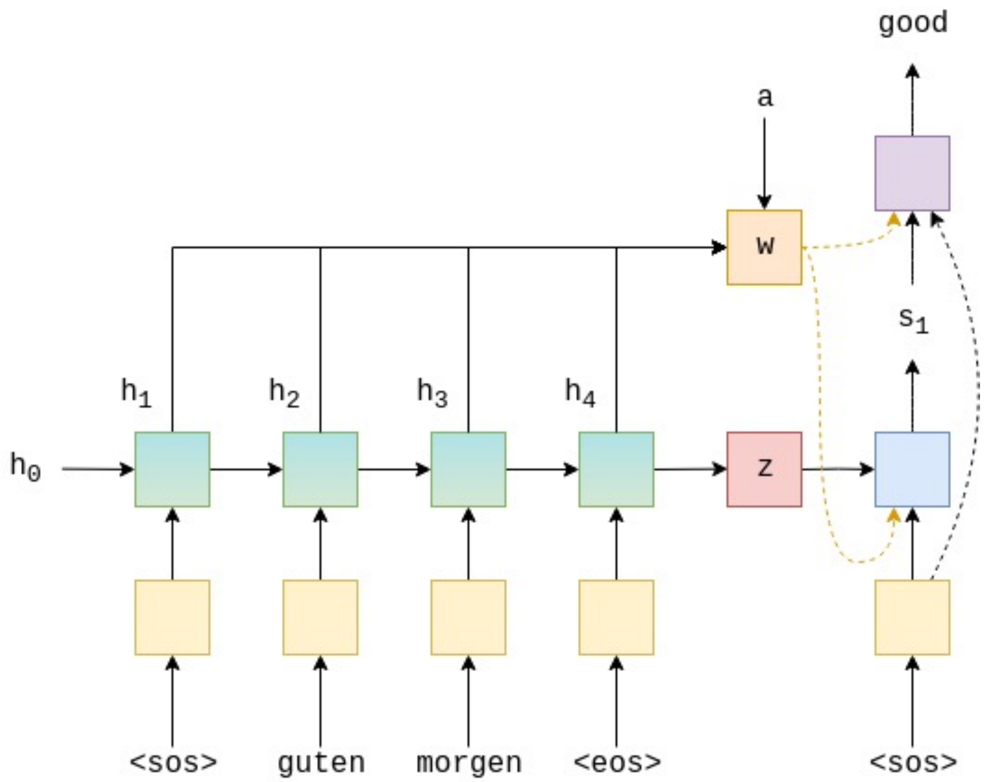
Attention

- Concept : At each time step when predicting an output from decoder, it checked input information again.
- However, it have not same rate in all input sentences, and it will pay more attention to part of the input word that id relates to the word to be predicted.



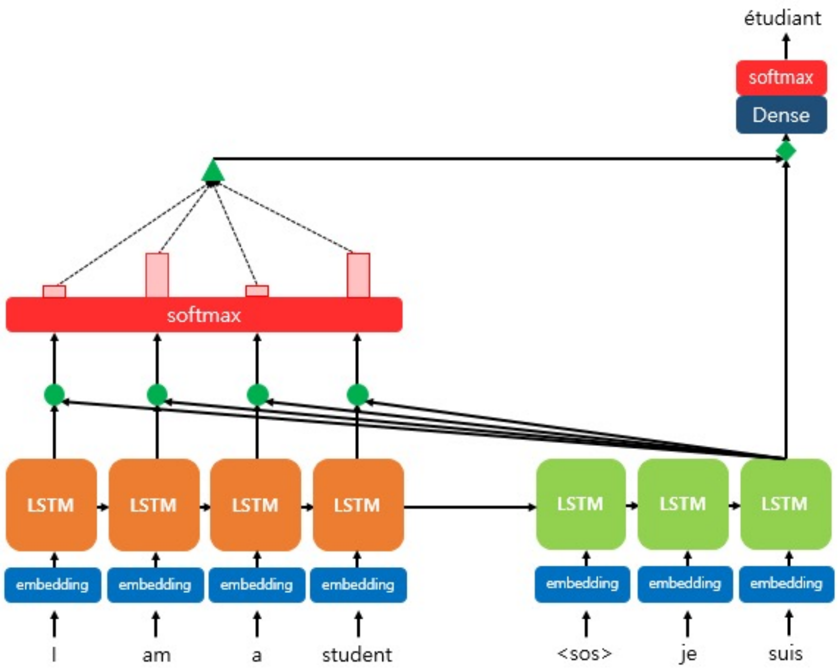
Attention

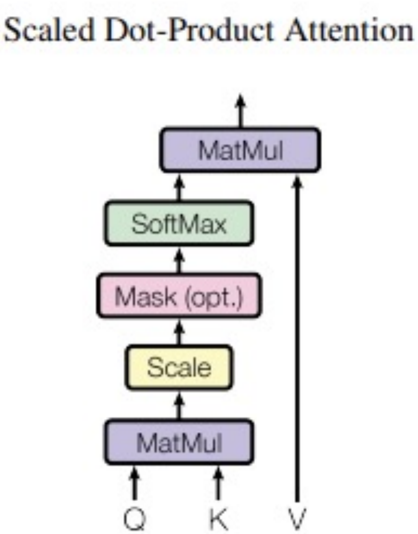
- Concept : At each time step when predicting an output from decoder, it checked input information again.
- However, it have not same rate in all input sentences, and it will pay more attention to part of the input word that id relates to the word to be predicted.



Dot-product Attention

- Q : Query - Hidden state in decoder cell at time t
  - A word representing the current output word.
- K : Key - Hidden state in the encoder cell at every point in time
- V : Value - Hidden state in the encoder cell at any time.
  - Vector corresponding to each word in the input sequence.





- Multi-Head Attention

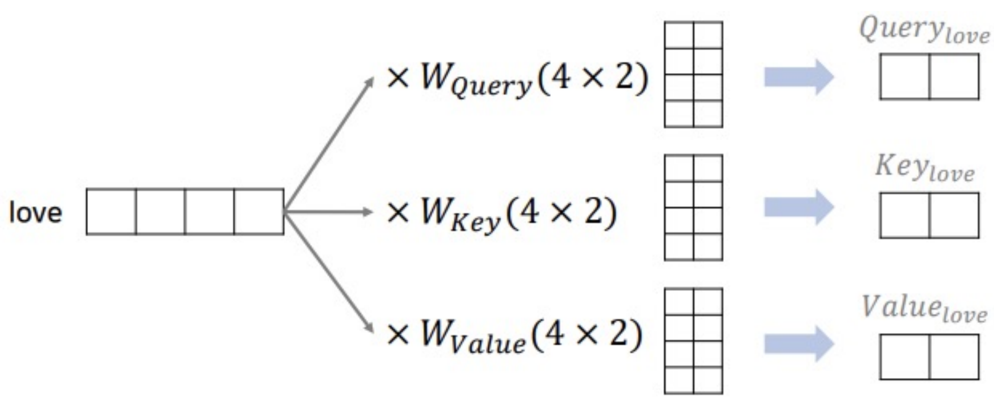
$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

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

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

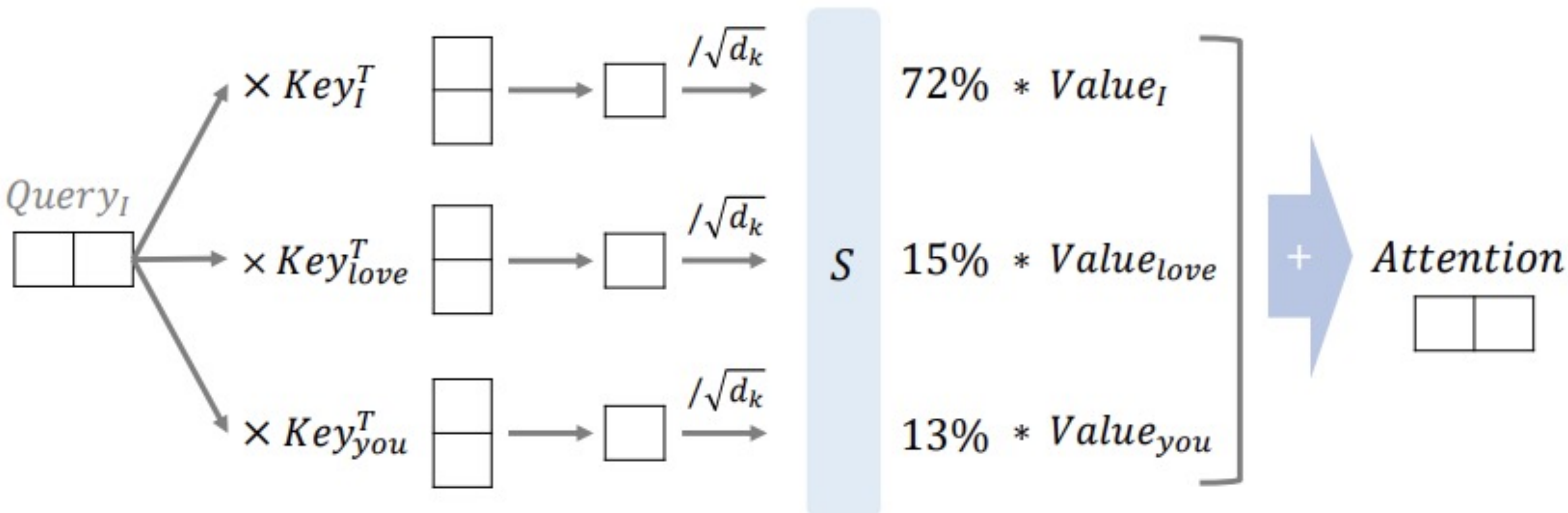
■ Scaled Dot-Product Attention

- Each word can use embedding



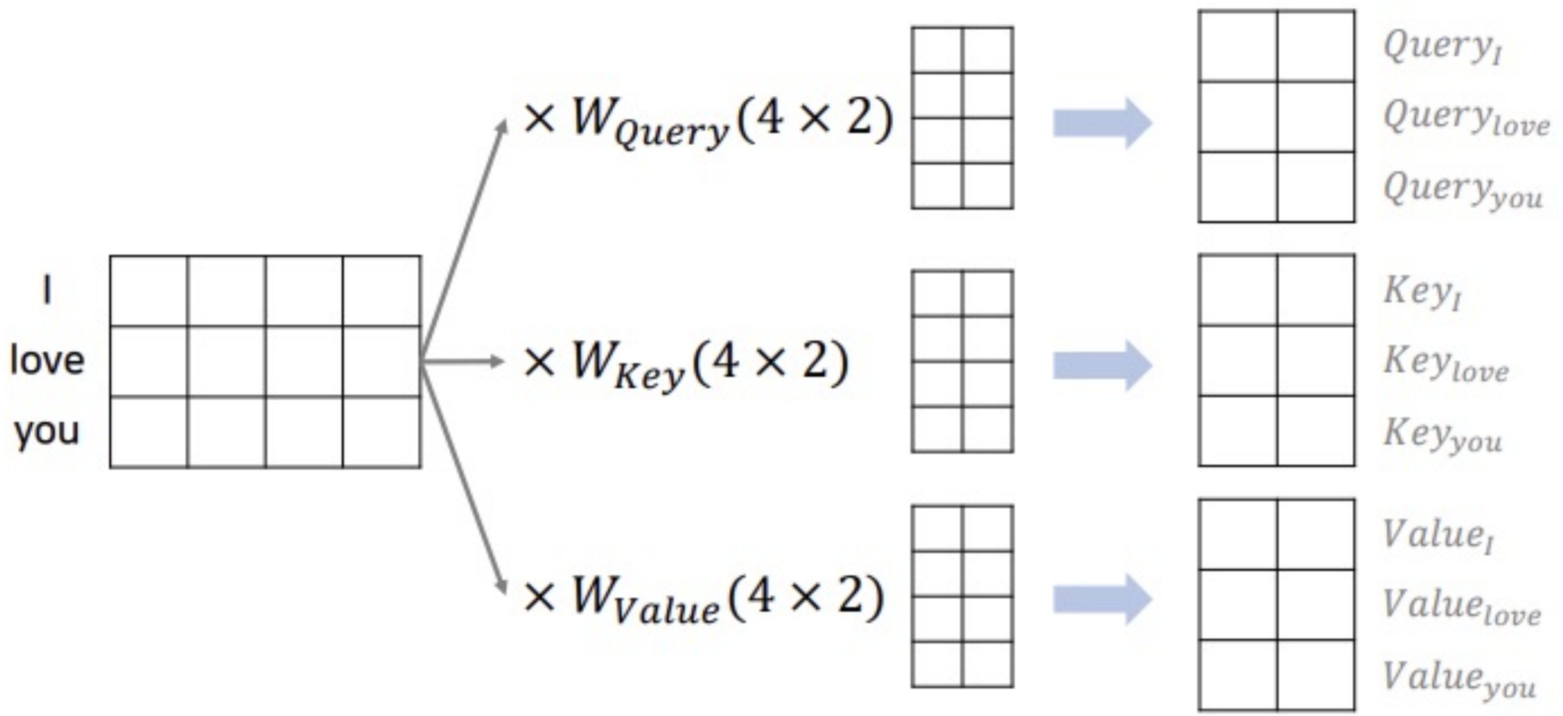
■ Scaled Dot-Product Attention

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



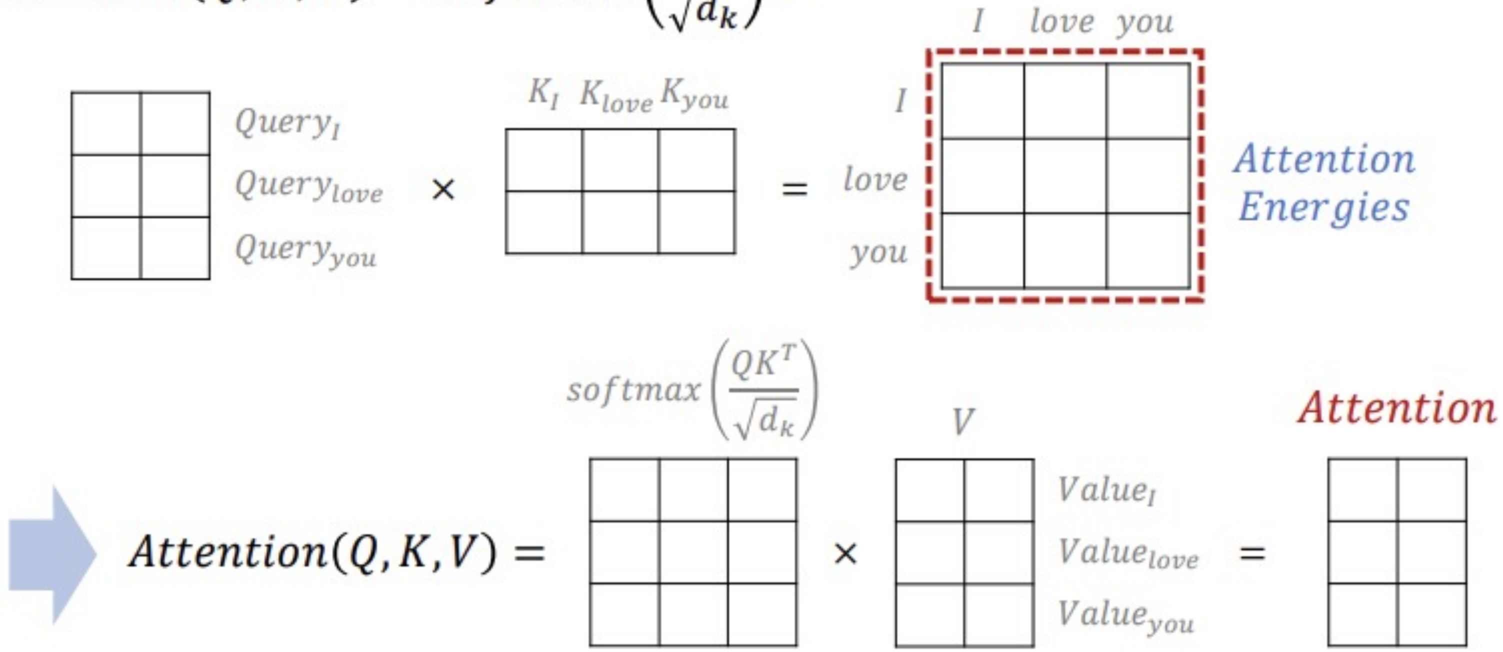
■ Scaled Dot-Product Attention

- It is possible to calculate them all at once using matrix multiplication operations.



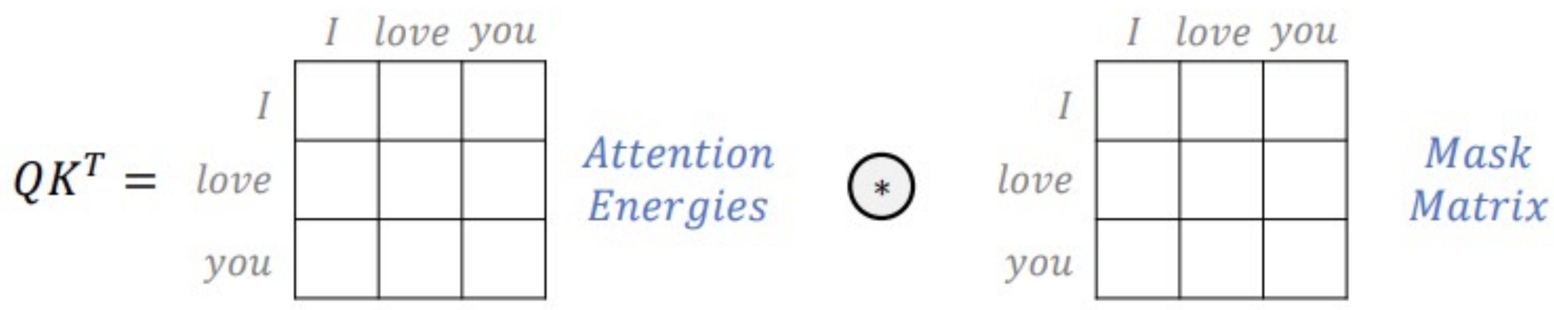
Scaled Dot-Product Attention

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



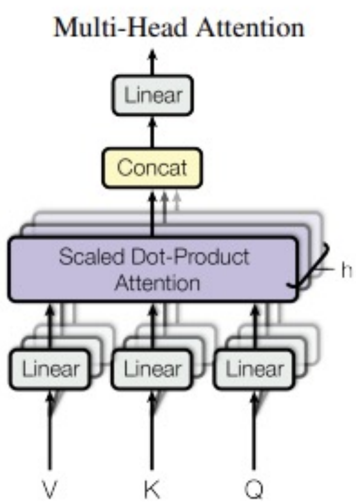
Scaled Dot-Product Attention

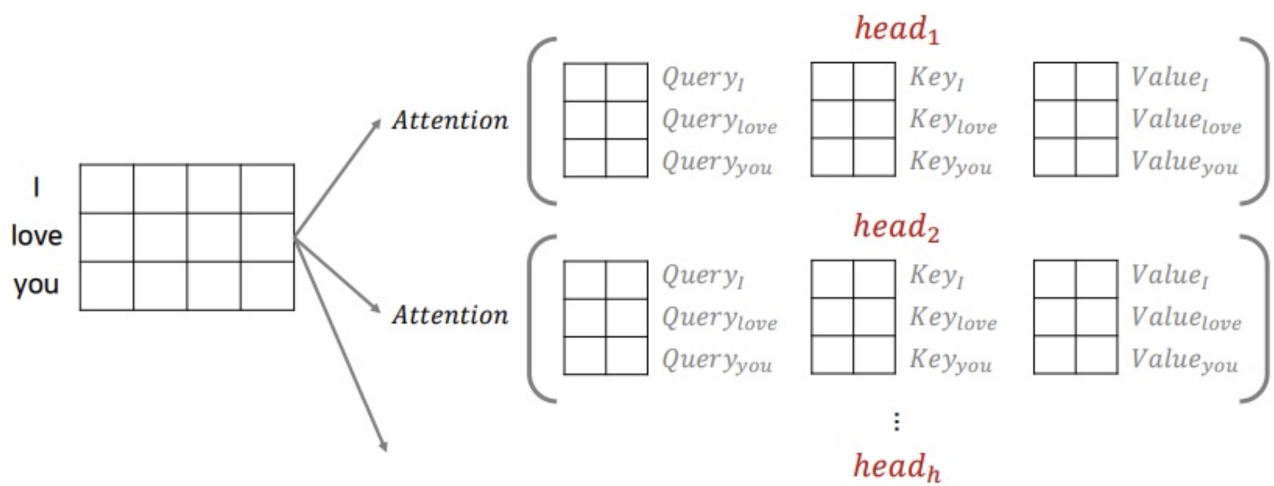
- Mask matrix can be used to ignore certain words.
- Enter a negative infinite value as the mask value so that the output of the softmax function approaches 0%.



■ Multi-Head Attention

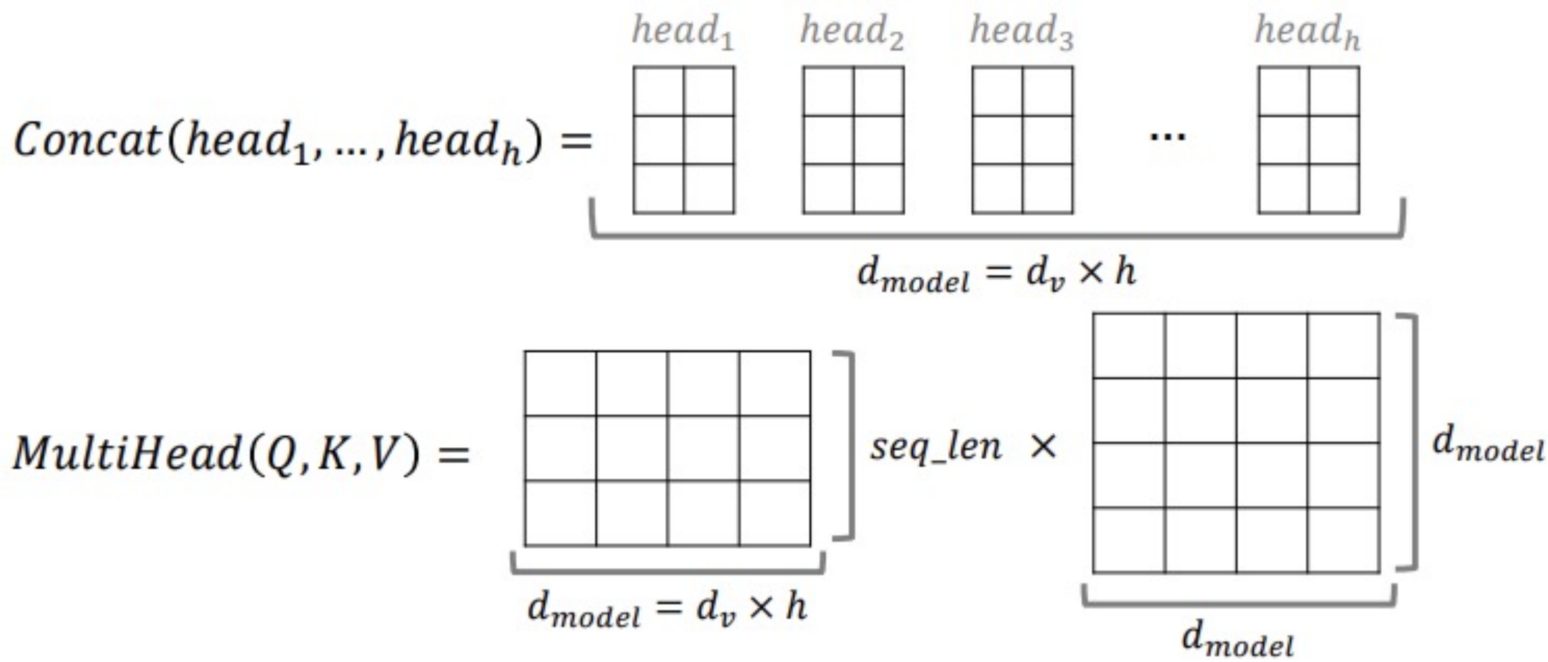
- $MultiHead(Q,K,V) = Concat(head_1, \ldots, head_h)W^O$





Multi-Head Attention

- The dimension remains the same even after performing MultiHead(Q, K, V)



- Transformer