Transformers

MUSTAFA HAJIJ

What is attention ?

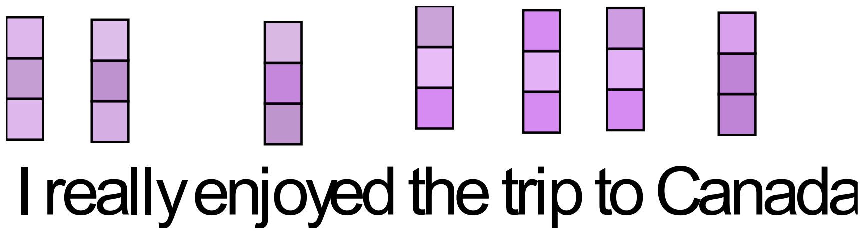
We like to model every word in the sentence in terms to other words around it.

I really enjoyed the trip to Canada

What is attention ?

We like to model every word in the sentence in terms to other words around it.

Assume every word has some embedding attached with it

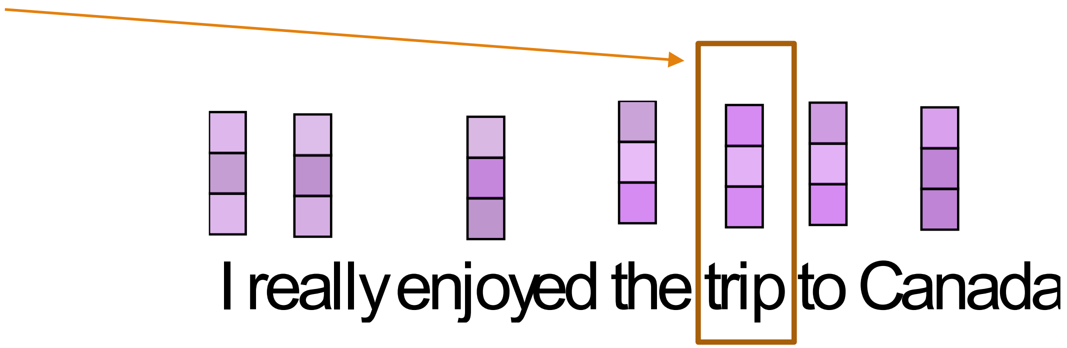


What is attention ?

We like to model every word in the sentence in terms to other words around it.

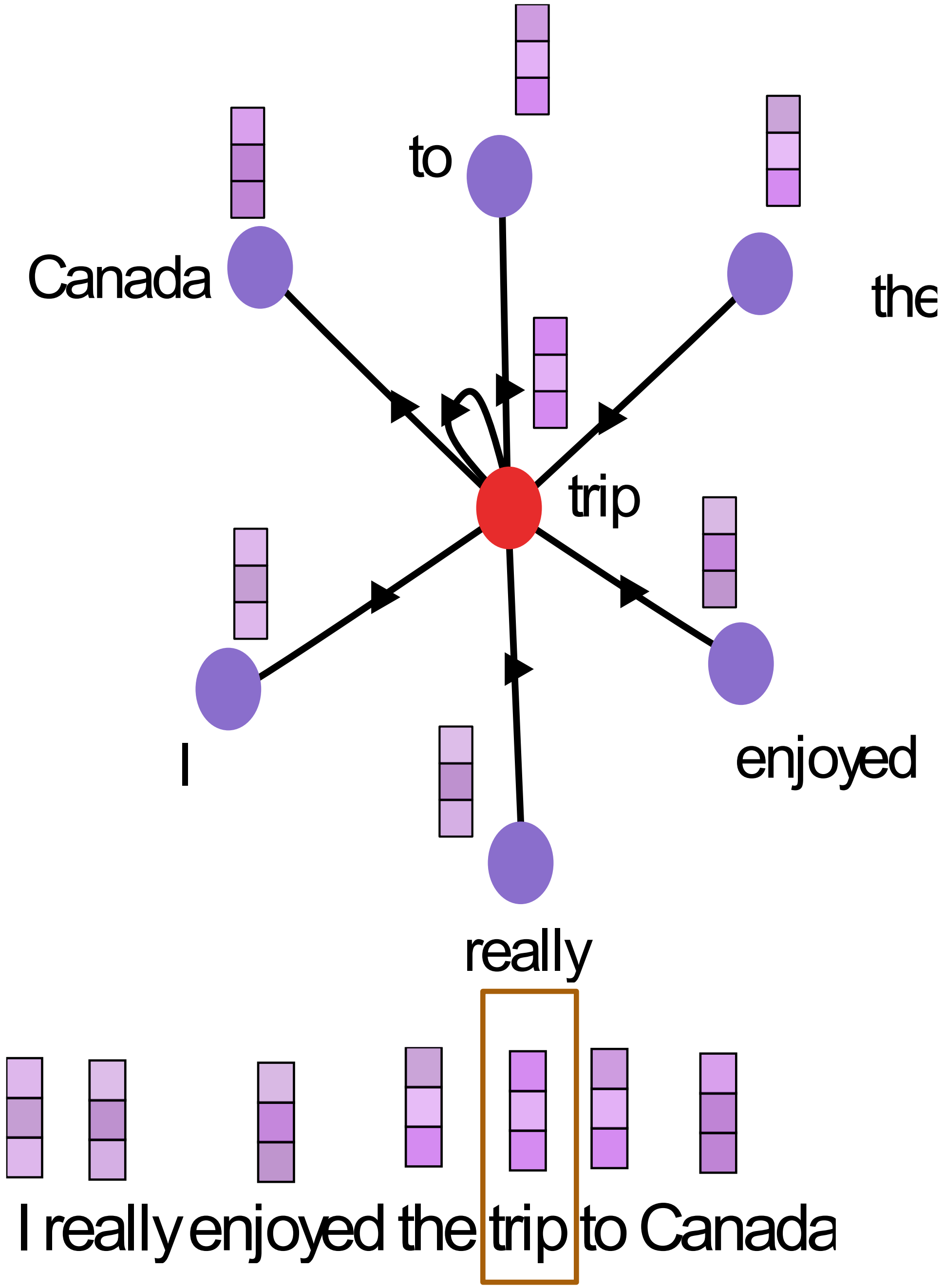
Assume every word has some embedding attached with it

Consider trip to be the target word



What is attention ?

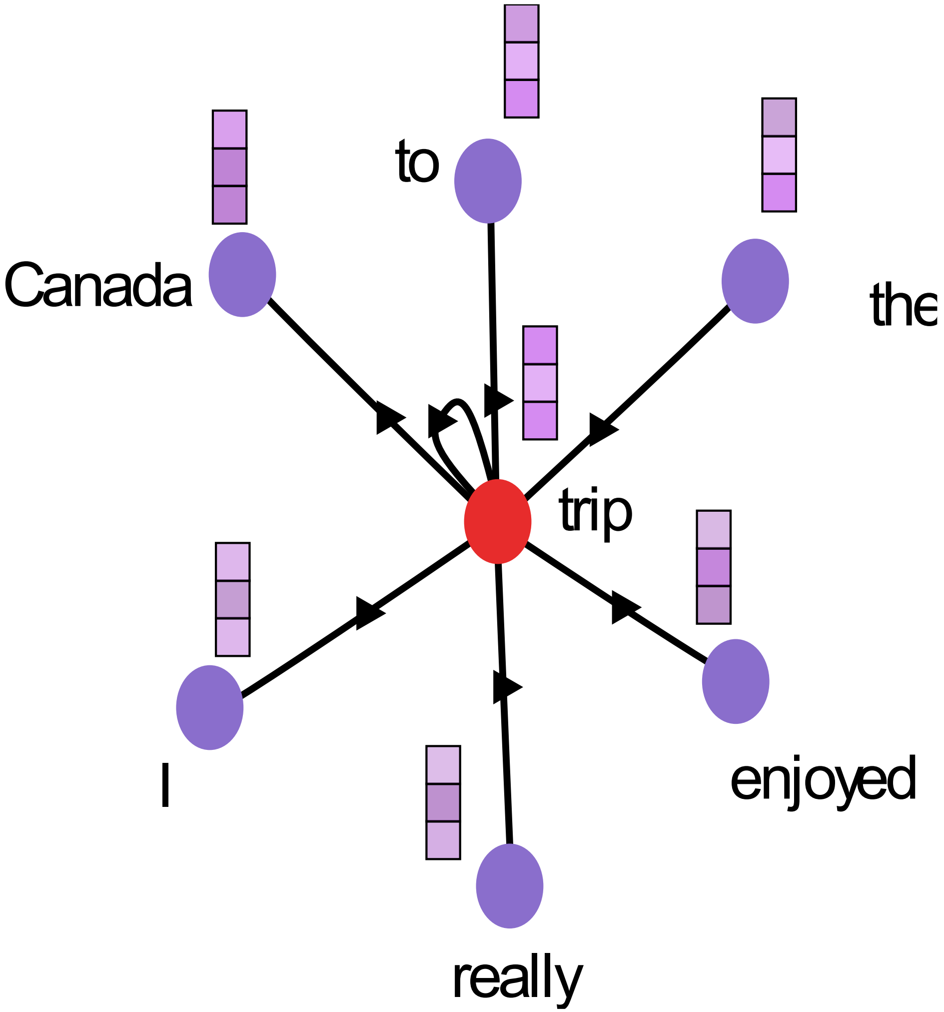
Lets put the sentence in a graph
With the target word in the middle
And connect all other words to it as follows:



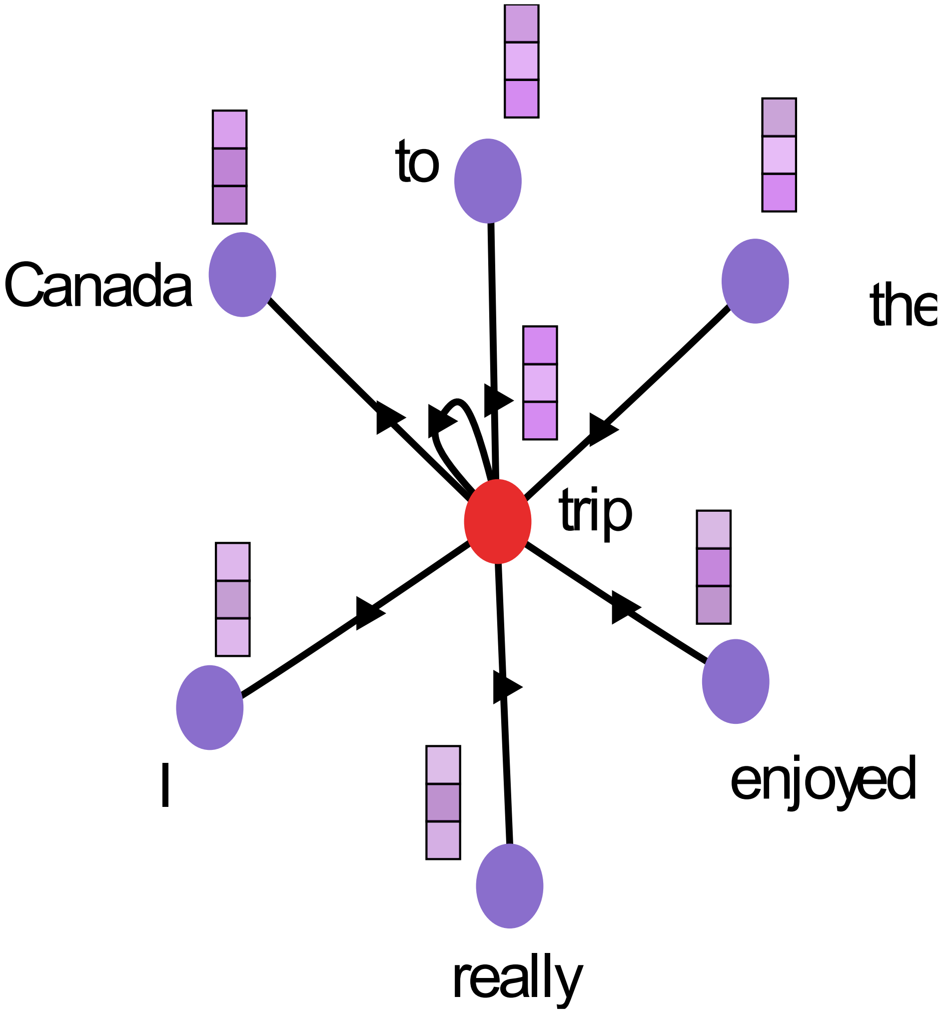
What is attention ?

Lets put the sentence in a graph
With the target word in the middle
And connect all other words to it as follows:

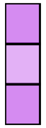
We want to update the vector attached with the 'trip' word in terms of the surrounding words



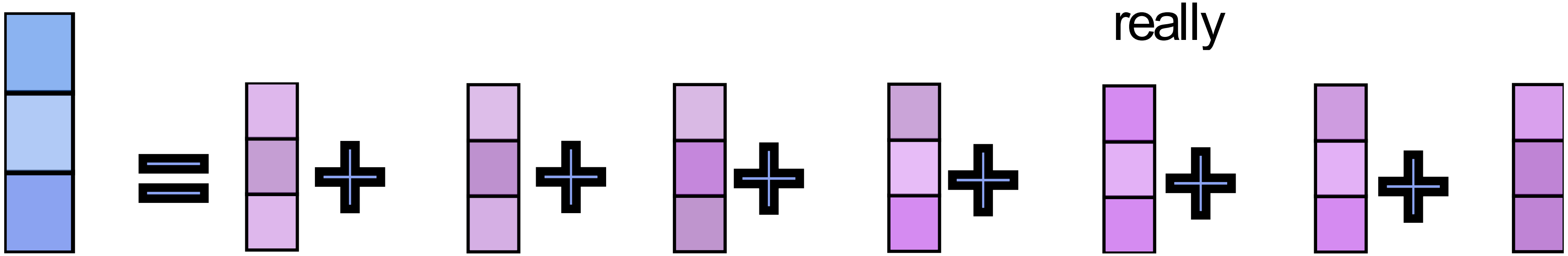
What is attention ?



Lets put the sentence in a graph
With the target word in the middle
And connect all other words to it as follows:

We want to update the vector 
attached with the ‘trip’ word
in terms of the surrounding words

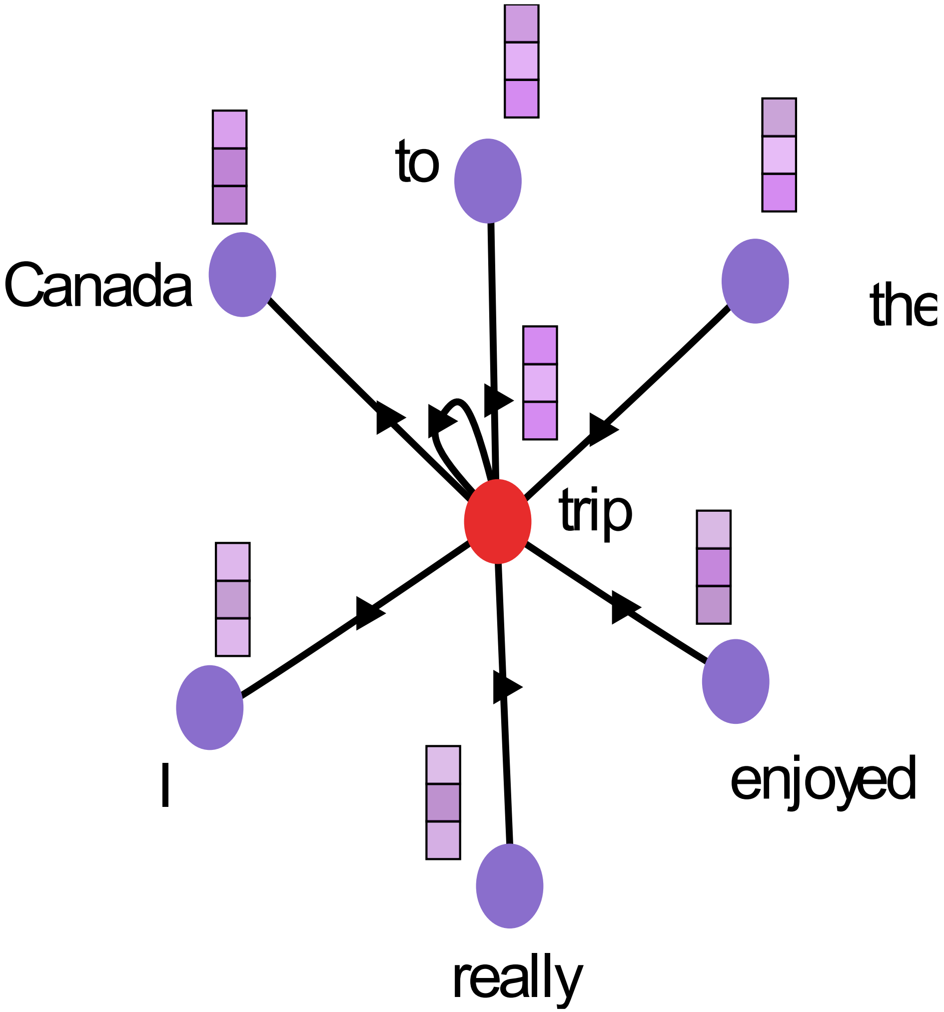
First attempt

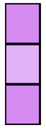


Updated rep of ‘trip’

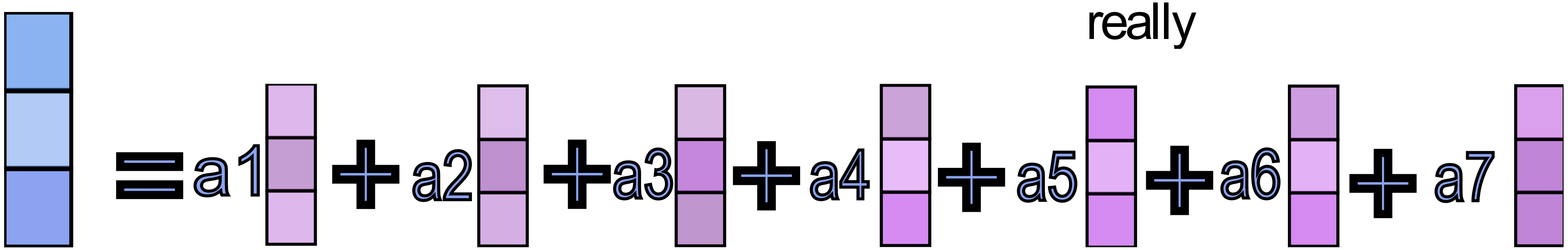
What is attention ?

Lets put the sentence in a graph
With the target word in the middle
And connect all other words to it as follows:



We want to update the vector 
attached with the 'trip' word
in terms of the surrounding words

A better rep

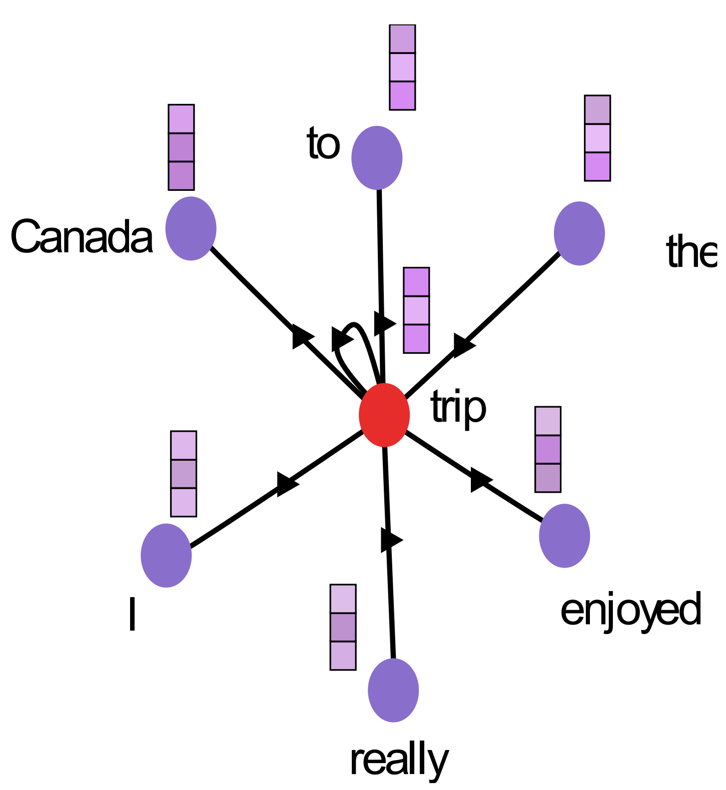


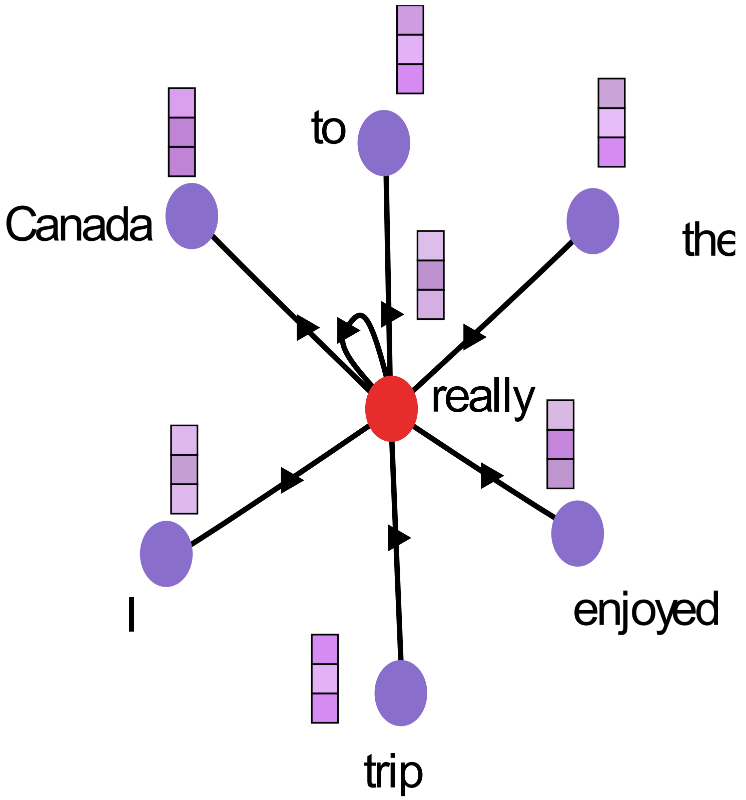
Updated rep of 'trip'

Where $\boldsymbol{a1} + \boldsymbol{a2} + \boldsymbol{a3} + \boldsymbol{a4} + \boldsymbol{a5} + \boldsymbol{a6} + \boldsymbol{a7} = \mathbf{1}$ **and** $\mathbf{0} \leq \boldsymbol{a_i} \leq \mathbf{1}$

What is attention ?

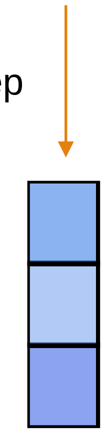
Repeat this for every word in the sentence

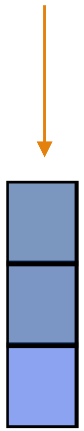




……

Gives the updated rep

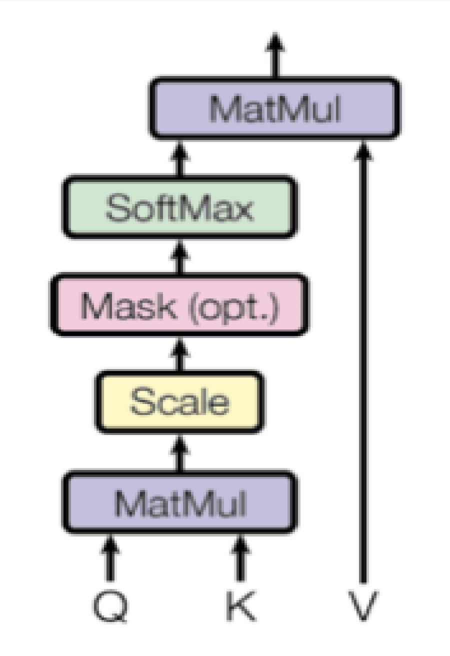




We obtain a 7X7 matrix of attention coefficients representing the attention of every word to every other word and 7 updated reps for the words.

The details

The transformer model, each attention unit learns three weight matrices: **query weights (W_Q), key weights (W_K), and value weights (W_V).**



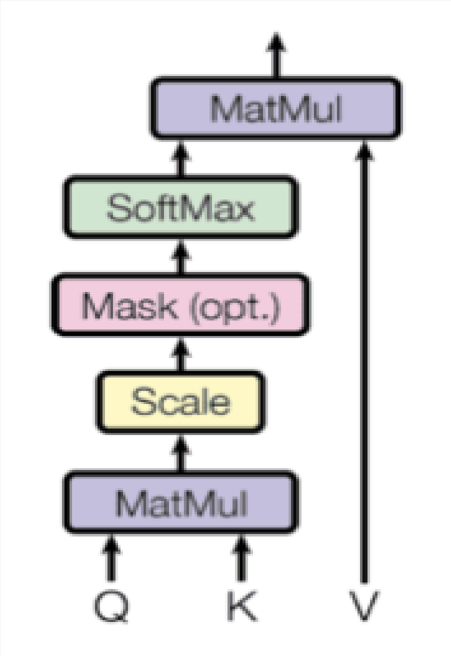
Scaled dot-product attention
Taken from "Attention Is All You Need"

The details

The transformer model, each attention unit learns three weight matrices: **query weights (W_Q), key weights (W_K), and value weights (W_V).**

For every token i, the input word embedding x_i is multiplied by these weight matrices to generate

- a query vector $(\boldsymbol{q_i} = \boldsymbol{x_i} * \boldsymbol{W_Q})$,
- a key vector $(\boldsymbol{k_i} = \boldsymbol{x_i} * \boldsymbol{W_K})$,
- and a value vector $(\boldsymbol{v_i} = \boldsymbol{x_i} * \boldsymbol{W_V})$

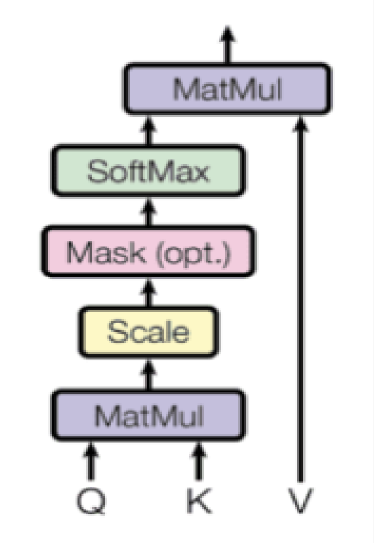


Scaled dot-product attention
Taken from "Attention Is All You Need"

The details

We define three matrices: $Q, K, and\ V$. Each row of these matrices corresponds to a vector q_i, k_i, and v_i respectively.

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

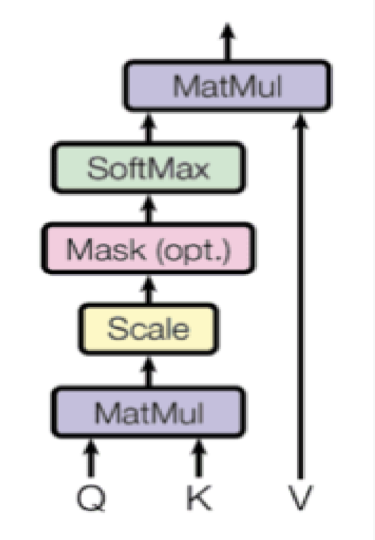


Scaled dot-product attention
Taken from "Attention Is All You Need"

The details

We define three matrices: $Q, K, and\ V$. Each row of these matrices corresponds to a vector q_i, k_i, and v_i respectively.

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

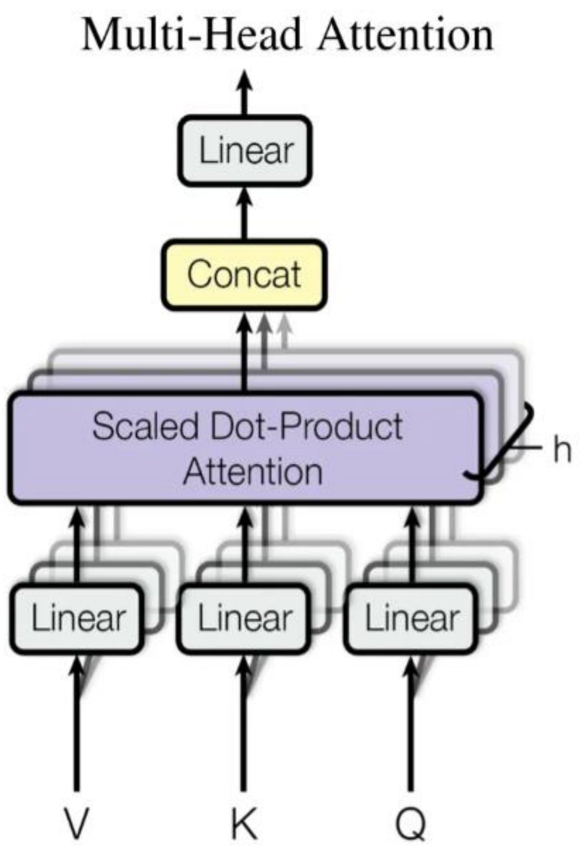


Scaled dot-product attention
Taken from "Attention Is All You Need"

- The attention weights a in the transformer model are determined by the influence of each word in the sequence (represented by Q) on all other words in the sequence (represented by K).

- The SoftMax function is applied to these weights to ensure they form a distribution between 0 and 1.
- These weights are then used to scale the corresponding word vectors in V, which are the same vectors as Q for the encoder and decoder, but different for the module that combines encoder and decoder inputs.

Multihead attention



$$\text{MultiheadedAttention}(Q, K, V) = \text{Concat}_{i \in [\#heads]}(\text{Attention}(XW_i^Q, XW_i^K, XW_i^V))W^O$$

Refs