Tutorial 5

Fei Gao & Yuliia Frund

Word Embeddings

Word Embedding Types

- Static
  - (e.g.: Word2Vec, GloVe)
- Contextual
  - (e.g.: ELMo, BERT)

Skip-gram embeddings

Train a model that predicts context words:

[CONTEXT TARGET CONTEXT]

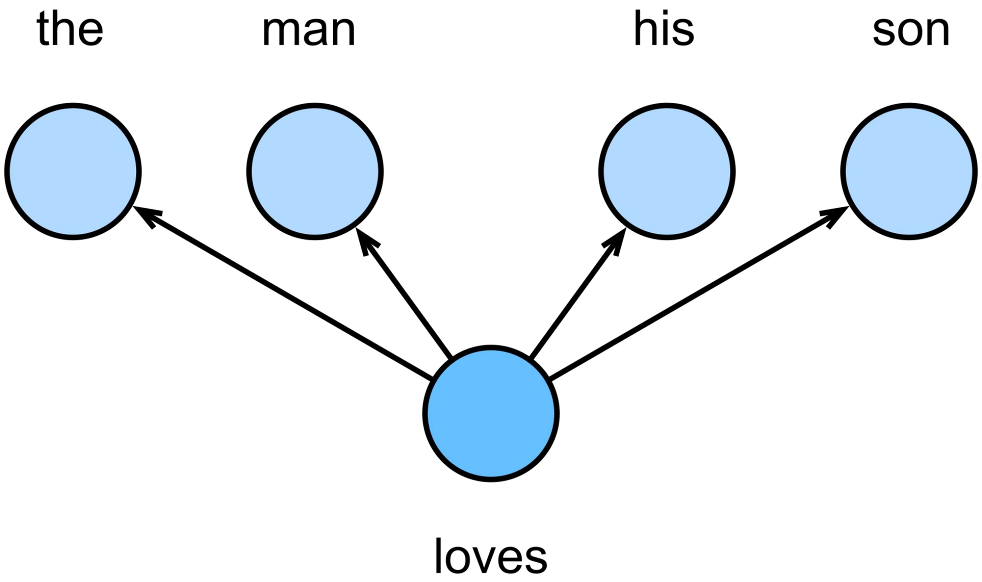


Image: https://en.wikipedia.org/wiki/Word2vec

Skip-gram embeddings

Train a model that predicts context words:

[CONTEXT TARGET CONTEXT]

These are **static** embeddings!

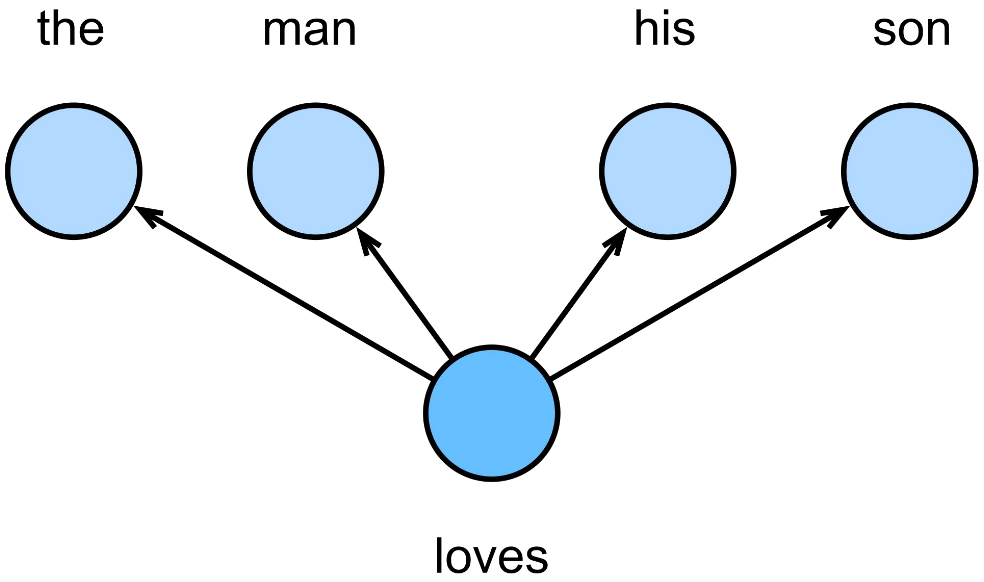


Image: https://en.wikipedia.org/wiki/Word2vec

Skip-gram embeddings

In other words...

Suppose this is our training data:

The quick brown fox jumps over the lazy dog.

Target word: FOX

A model with a context size 1 will predict:

FOX, brown

FOX, jumps

Byte Pair Encoding

Lexicon

- Let’s imagine that you’re training a machine translation system.
- Will it see **ALL** possible words in target and source language?
    - No! There will always be words your system hasn’t seen!
- How do you deal with **unseen words**?
    - Subword units!

Byte Pair Encoding

function BYTE-PAIR ENCODING(strings C, number of merges k) **returns** vocab V

$V \leftarrow$ all unique characters in C # initial set of tokens is characters
for $i = 1$ **to** k **do** # merge tokens k times
  $t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
  $t_{NEW} \leftarrow t_L + t_R$ # make new token by concatenating
  $V \leftarrow V + t_{NEW}$ # update the vocabulary
  Replace each occurrence of t_L, t_R in C with t_{NEW} # and update the corpus
return V

Figure 2.13 The token learner part of the BPE algorithm for taking a corpus broken up into individual characters or bytes, and learning a vocabulary by iteratively merging tokens. Figure adapted from Bostrom and Durrett (2020).

Sennrich, Rico; Birch, Alexandra; Haddow, Barry (2015-08-31). "Neural Machine Translation of Rare Words with Subword Units"
https://web.stanford.edu/~jurafsky/slp3/2.pdf

Byte Pair Encoding

In practice:

- Very frequent words are likely to be stored whole
- Rare and unseen words can still be handled
- Manageable vocabulary size

Attention

Class Slide 15

Attention Head (contd.)

- Now we have projected the inputs with three transformations
- We use the query and the key to compute attention

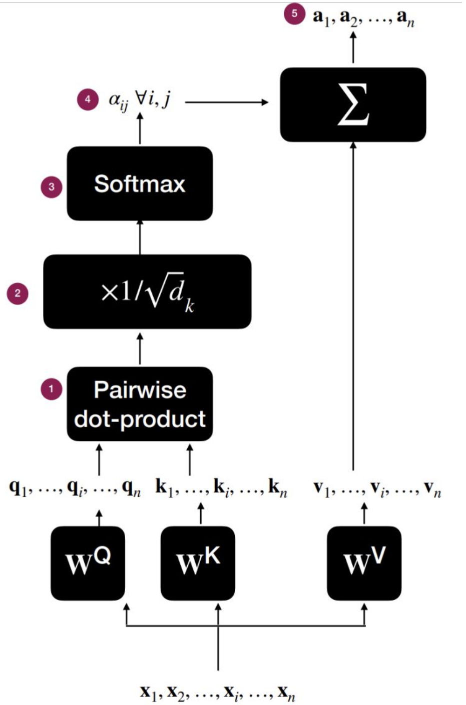
$$
\begin{aligned}
\text{score}(\mathbf{x}_i, \mathbf{x}_j) &= \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} && (9.11) \\
\alpha_{ij} &= \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \ \forall j \leq i && (9.12) \\
\mathbf{a}_i &= \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j && (9.13)
\end{aligned}
$$

Causal attention

Because we are only looking at the past

Full attention

Looks at both past and the future



Calculate α3

How to calculate the weight row $\boldsymbol{\alpha}_3$?

$\boldsymbol{\alpha}_3$ is a row. Think: What is its individual components?

Hint: what are j's that are smaller than i (i = 3, the query number), starting from 1?

Attention Head (contd.)

- Now we have projected the inputs with three transformations
- We use the query and the key to compute attention

$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\boxed{\mathbf{q}_i} \cdot \mathbf{k}_j}{\sqrt{d_k}} \quad (9.11)$$

$$\boxed{\alpha_{ij} = \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \ \forall j \leq i} \quad (9.12)$$

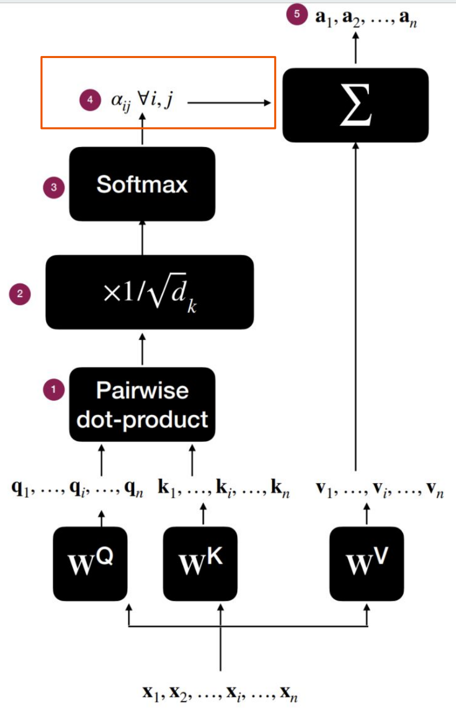
$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j \quad (9.13)$$

Causal attention

Because we are only looking at the past

Full attention

Looks at both past and the future



Calculate α3

How to calculate the weight row

α_3?

α_3 is a combination of

α_{31}, α_{32}, and α_{33}

Attention Head (contd.)

- Now we have projected the inputs with three transformations
- We use the query and the key to compute attention

$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \quad (9.11)$$

$$\boxed{\alpha_{ij} = \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \;\; \forall j \le i} \quad (9.12)$$

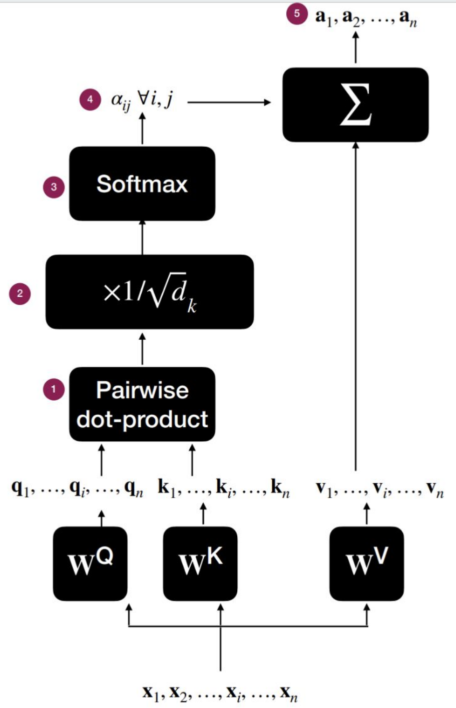
$$\mathbf{a}_i = \sum_{j \le i} \alpha_{ij} \mathbf{v}_j \quad (9.13)$$

Causal attention

Because we are only looking at the past

Full attention

Looks at both past and the future



Calculate α3

Now, calculate the score for each pair of q_i and k_j

Then, do softmax for these three together.

Then we get α_{31}, α_{32}, and α_{33}

Attention Head (contd.)

- Now we have projected the inputs with three transformations
- We use the query and the key to compute attention

$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \quad (9.11)$$

$$\alpha_{ij} = \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \ \forall j \leq i \quad (9.12)$$

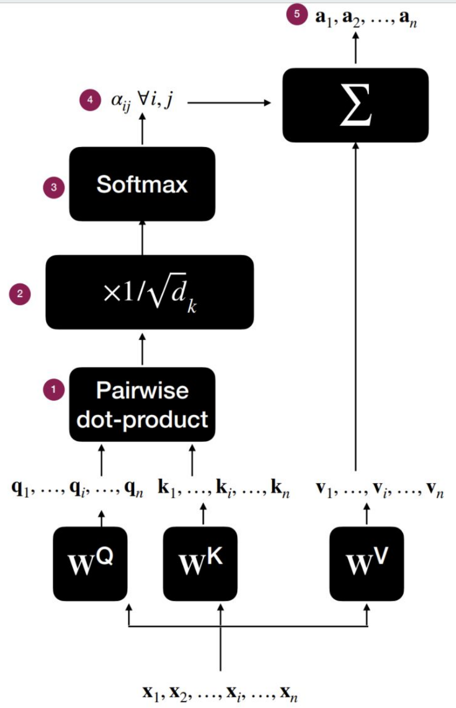
$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j \quad (9.13)$$

Causal attention

Because we are only looking at the past

Full attention

Looks at both past and the future



Calculate α3

Q: query
K: key

$$\frac{e^{\text{Score}(q_3,k_1)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

$$\frac{e^{\text{Score}(q_3,k_2)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

$$\frac{e^{\text{Score}(q_3,k_3)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

Attention Head (contd.)

- Now we have projected the inputs with three transformations
- We use the query and the key to compute attention

$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \quad (9.11)$$

$$\alpha_{ij} = \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \;\; \forall j \leq i \quad (9.12)$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j \quad (9.13)$$

Causal attention

Because we are only looking at the past

Full attention

Looks at both past and the future

Calculate α3

3 4

$$\text{Softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right) = \begin{array}{c c} & \begin{array}{ccc} \text{I} & \text{am} & \text{good} \end{array} \\ \begin{array}{c} \text{I} \\ \text{am} \\ \text{good} \end{array} & \begin{bmatrix} 0.90 & 0.07 & 0.03 \\ 0.025 & 0.95 & 0.025 \\ \boxed{0.21 \quad 0.03 \quad 0.76} \end{bmatrix} \end{array}$$

K

Q

$$\frac{e^{\text{Score}(q_3,k_1)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

$$\frac{e^{\text{Score}(q_3,k_2)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

$$\frac{e^{\text{Score}(q_3,k_3)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

Query and Key

Note how it is written here:
For any j that is smaller than i…
This means

1. i is bigger than j
2. i is your current position, and j is the information before

Attention Head (contd.)

- Now we have projected the inputs with three transformations
- We use the query and the key to compute attention

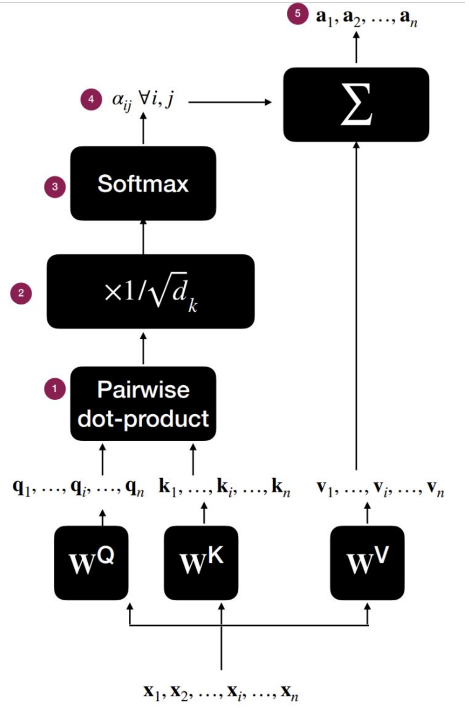
$$\mathrm{score}(\mathbf{x}_i,\mathbf{x}_j) = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \quad (9.11)$$

$$\alpha_{ij} = \mathrm{softmax}(\mathrm{score}(\mathbf{x}_i,\mathbf{x}_j)) \;\; \forall j \leq i \quad (9.12)$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j \quad (9.13)$$

Causal attention
Because we are only looking at the past

Full attention
Looks at both past and the future



Query and Key

Query (Q): a search request in a database.
You send a request - Find me all books about dragons - to the library database.

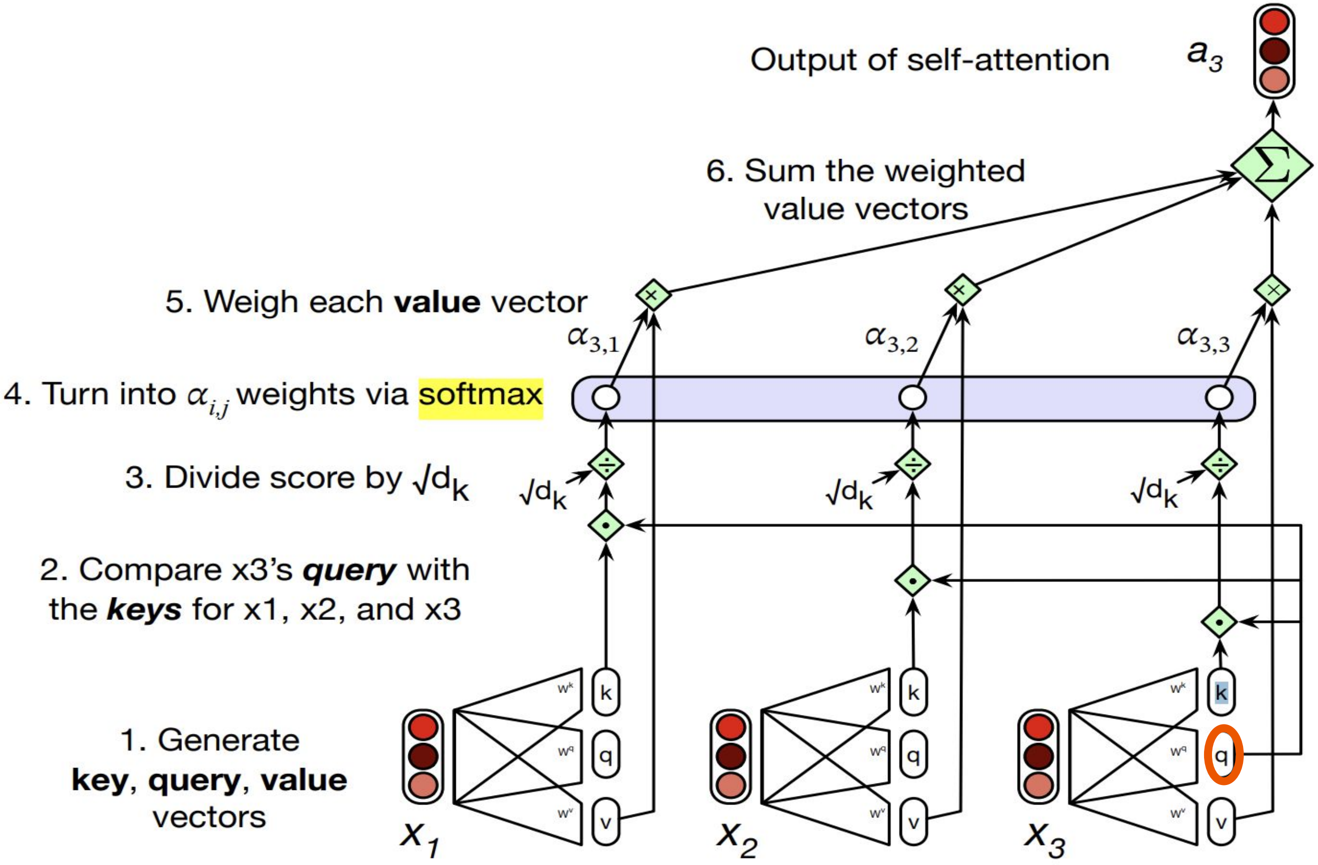
Key (K): labels or tags attached to items in the database.
In the library, each book might have keys like "genre", "author", "subject"
How to Kill a Dragon: ‘dragon’, ‘Indo-European linguistics’, ‘Calvert Watkins’

How does Q work with K？
The system compares the Q to the K.

Calculate α3

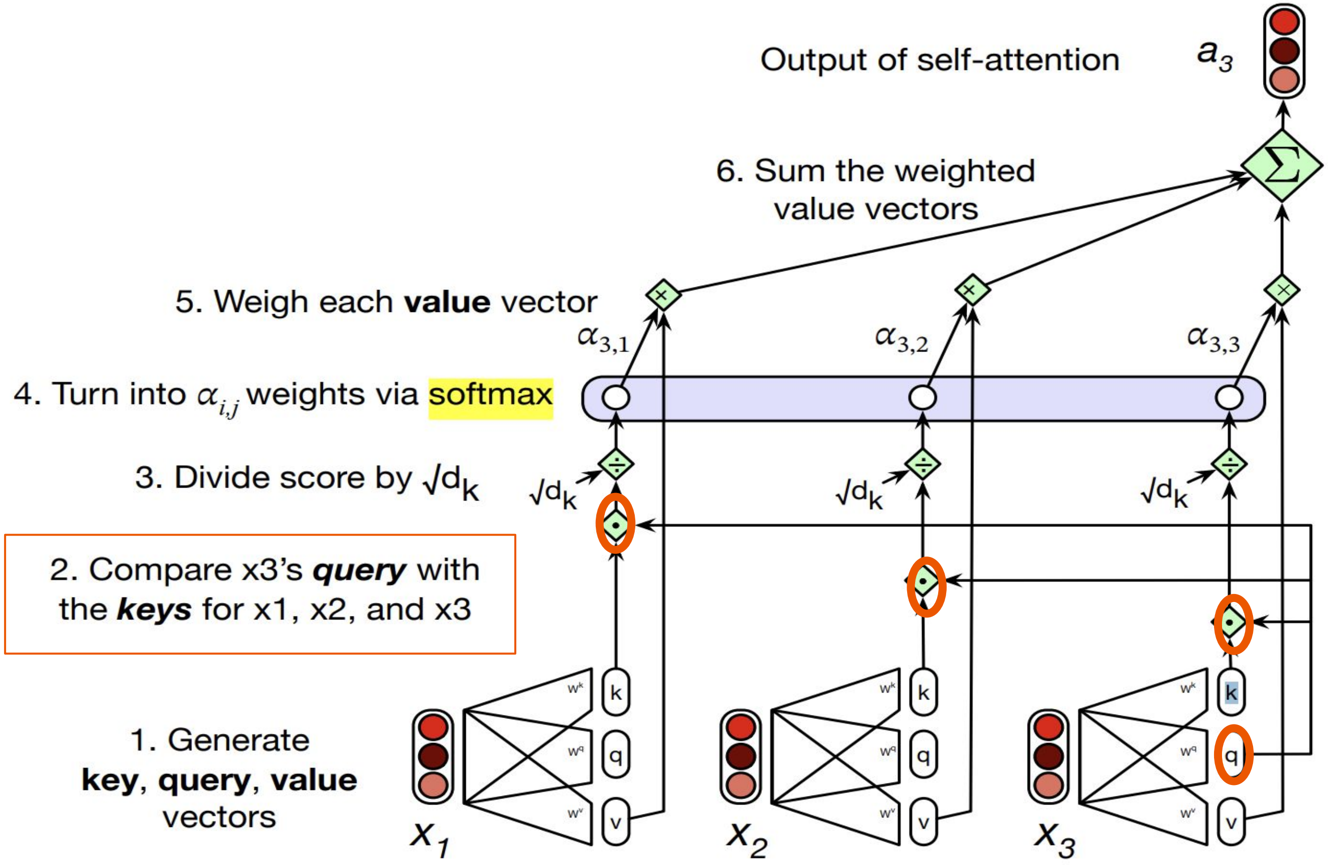
What this graph means:

You calculate for x3



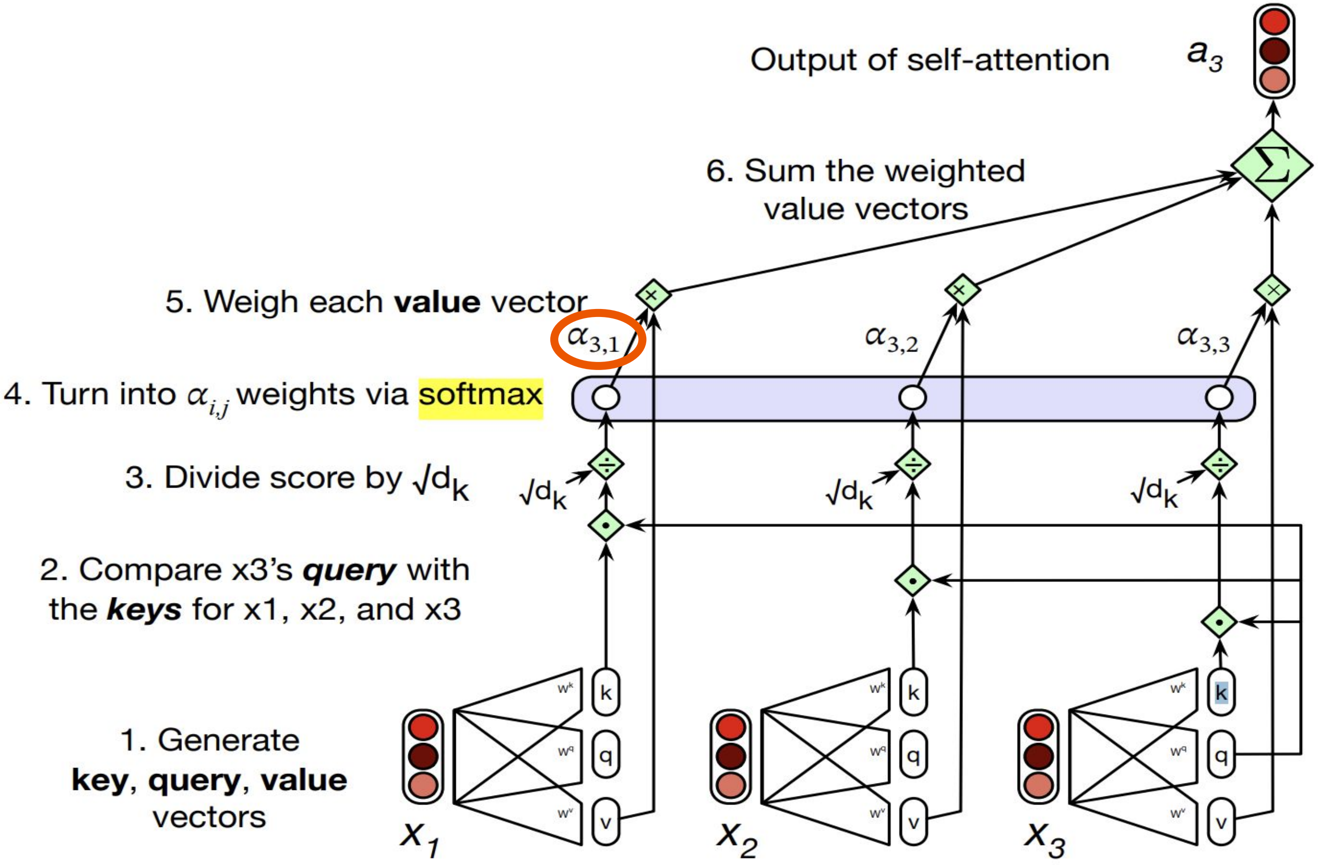
Source: https://web.stanford.edu/~jurafsky/slp3/slides/transformer24aug.pdf

Calculate α3



Source: https://web.stanford.edu/~jurafsky/slp3/slides/transformer24aug.pdf

Calculate α3



Source: https://web.stanford.edu/~jurafsky/slp3/slides/transformer24aug.pdf

$$\text{Softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right) = \begin{array}{c|ccc} & \text{I} & \text{am} & \text{good} \\ \hline \text{I} & 0.90 & 0.07 & 0.03 \\ \text{am} & 0.025 & 0.95 & 0.025 \\ \text{good} & \boxed{0.21} & 0.03 & 0.76 \end{array}$$

$$\frac{e^{\text{Score}(q_3,k_1)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

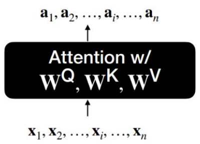
$$\frac{e^{\text{Score}(q_3,k_2)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

$$\frac{e^{\text{Score}(q_3,k_3)}}{e^{\text{Score}(q_3,k_1)} + e^{\text{Score}(q_3,k_2)} + e^{\text{Score}(q_3,k_3)}}$$

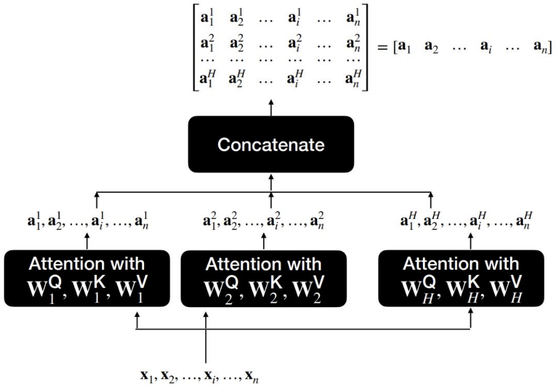
Attention

Multi-head attention (see class slides)
Masked-attention
Bidirectional Self Attention

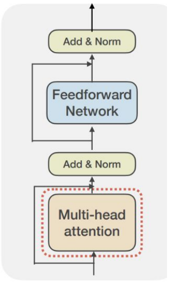
Multihead attention



Single head attention



Multihead attention



Transformer architecture

Attention

This formula in class is: masked self attention

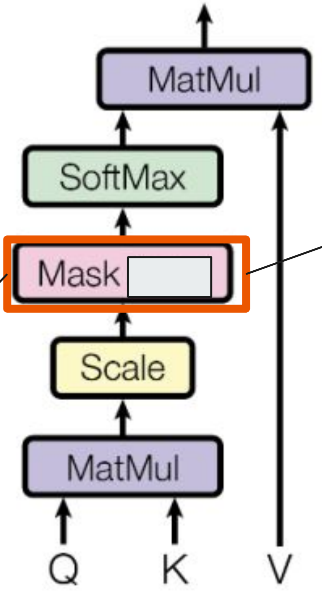
$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}}$$

$$\alpha_{ij} = \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \;\; \forall j \leq i$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j$$

Attention

Multi-head attention (see class slides)
Masked self-attention
Bidirectional self-attention



q1·k1	−∞	−∞	−∞
q2·k1	q2·k2	−∞	−∞
q3·k1	q3·k2	q3·k3	−∞
q4·k1	q4·k2	q4·k3	q4·k4

N

N

$$\text{Softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right) = \begin{array}{c} \\ I \\ am \\ good \end{array} \begin{array}{c} \begin{array}{ccc} I & am & good \end{array} \\ \left[\begin{array}{ccc} 0.90 & \boxed{} & \boxed{} \\ 0.025 & 0.95 & \boxed{} \\ 0.21 & 0.03 & 0.76 \end{array}\right] \end{array}$$

Attention

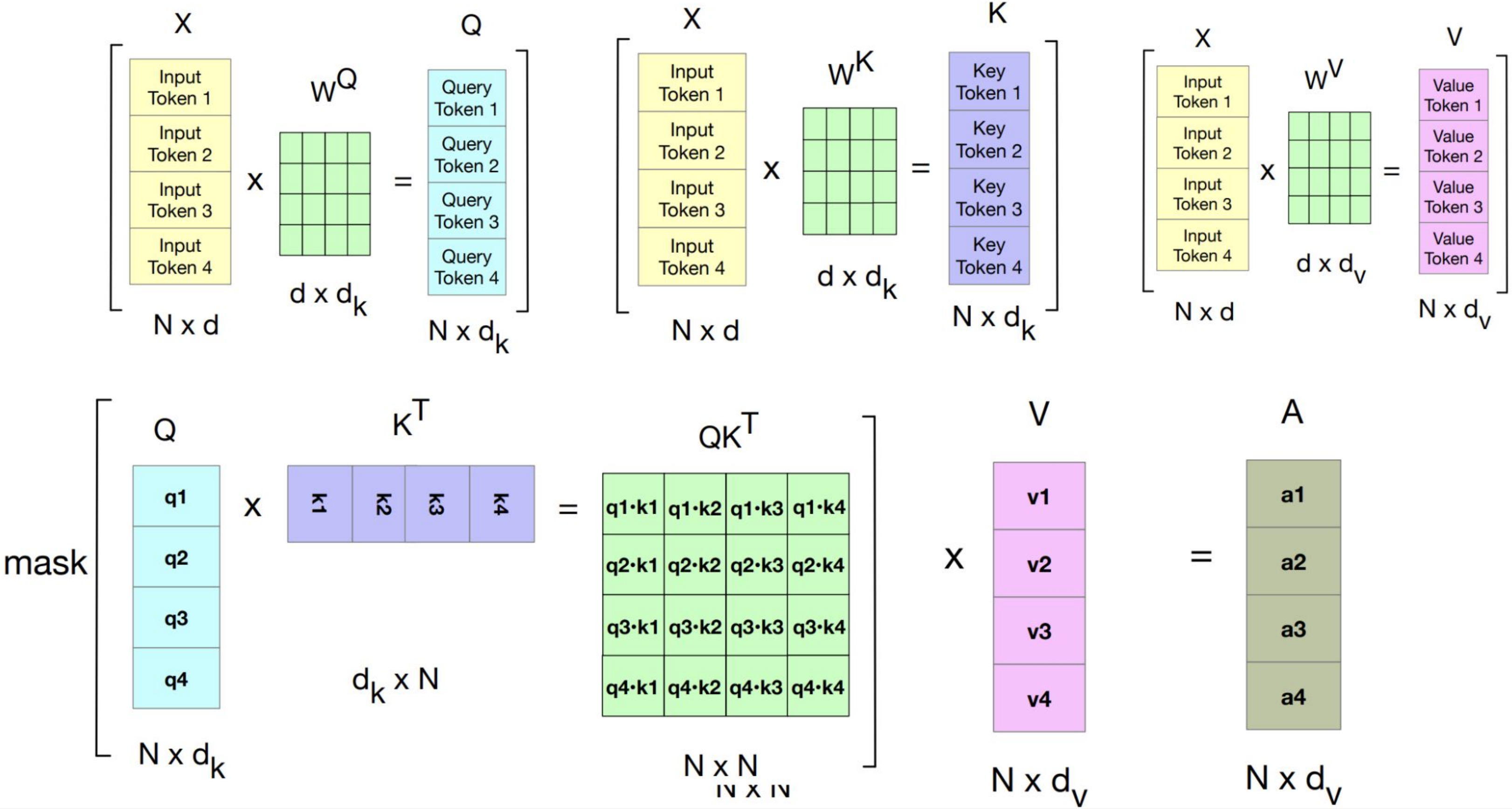
Multi-head attention (see class slides)
Masked self-attention
Bidirectional self-attention

$$\text{Softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right) = \begin{array}{c} \\ \text{I} \\ \text{am} \\ \text{good} \end{array} \begin{array}{c} \begin{array}{ccc} \text{I} & \text{am} & \text{good} \end{array} \\ \begin{bmatrix} 0.90 & 0.07 & 0.03 \\ 0.025 & 0.95 & 0.025 \\ 0.21 & 0.03 & 0.76 \end{bmatrix} \end{array}$$

Attention

Multi-head attention (see class slides)
Masked self-attention
Bidirectional self-attention
Encoder-decoder attention (not self-attention)

Attention again



Source: https://web.stanford.edu/~jurafsky/slp3/slides/transformer24aug.pdf

More resources?

https://www.youtube.com/watch?v=iDulhoQ2pro (Yannic Kilcher)

Or search for “attention calculation” “self-attention math” etc. on Youtube or medium

Source: https://web.stanford.edu/~jurafsky/slp3/slides/transformer24aug.pdf

End