

# Transformers e BERT

Prof. Dr. Ricardo Primi



# Introdução

Slides de:

Afshine Amidi ( Stanford ICME, Workshop de verão de 2021)

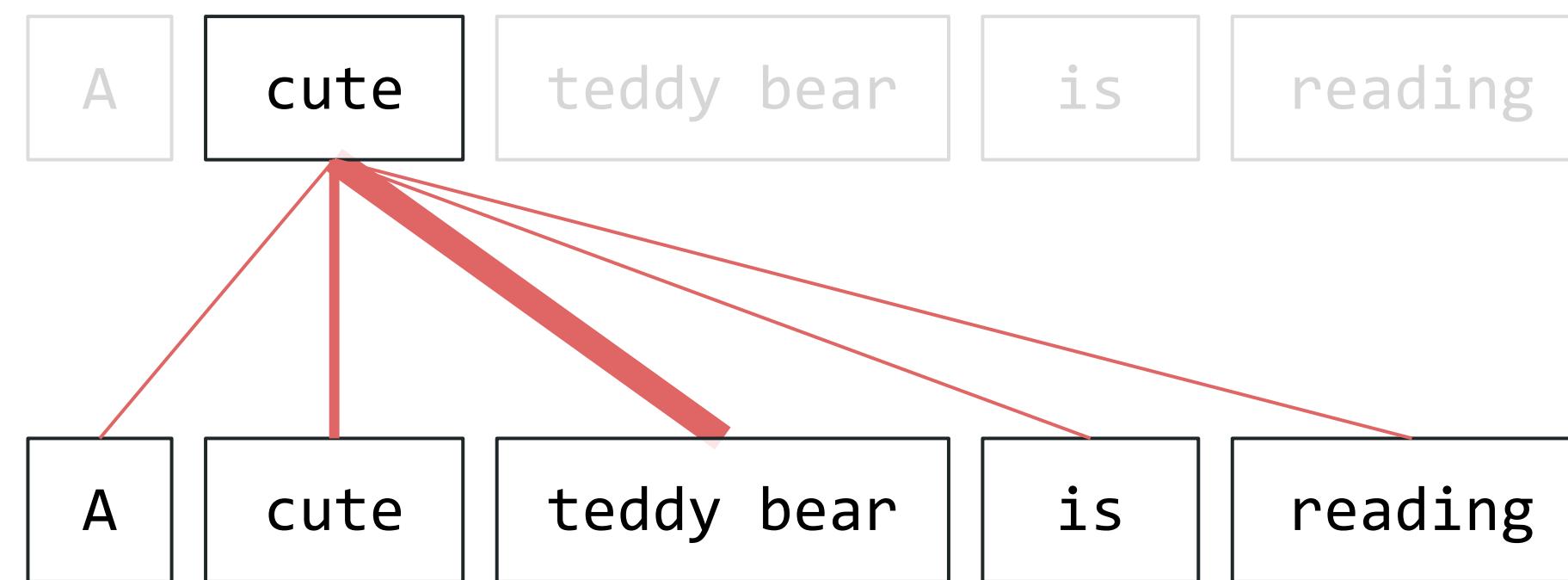
Martin Jocqueviel: Learn BERT - most powerful NLP algorithm by Google (Udemy)

- BERT: Bidirectional Encoder Representations from Transformers
- <https://peltarion.com/blog/data-science/self-attention-video>
- <https://towardsai.net/p/nlp/getting-meaning-from-text-self-attention-step-by-step-video>



# Overview of the Transformer

- Introduced in the 2017 paper "**Attention is All You Need**"
- Relies on the self-attention mechanism
- Encoder/decoder parts that are used in a lot of models
- State of the art results on machine translation tasks



# Standardized benchmark for NLP

## GLUE: General Language Understanding Evaluation

Grammatical  
correctness

*CoLA*

Paraphrase

*MRPC*

Similarity

*QQP, STS-B*

Common sense

*WNLI*

Entailment

*RTE, MNLI*

Sentiment  
Extraction

*SST-2*

Question  
Answering

*QNLI*

**Glue score**

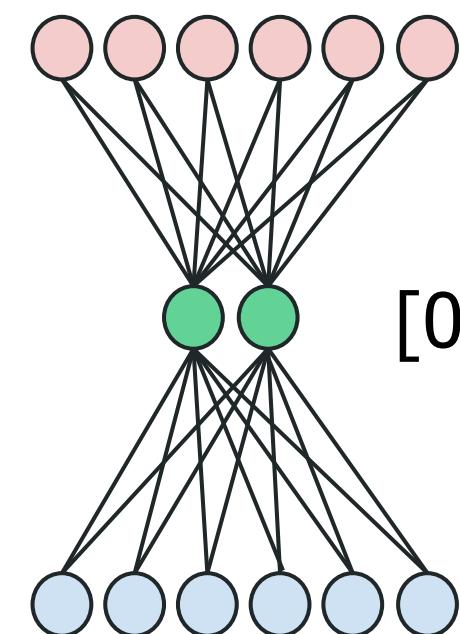
# Embeddings estáticos: word2vec

## Word2vec

**Example with left context window = 1**

A **cute** teddy bear is reading

[0.2, 0.4, 0.1, 0.1, 0.1, 0.1]



[0.2, 0.9]

[1,0,0,0,0,0]

A cute teddy bear is reading

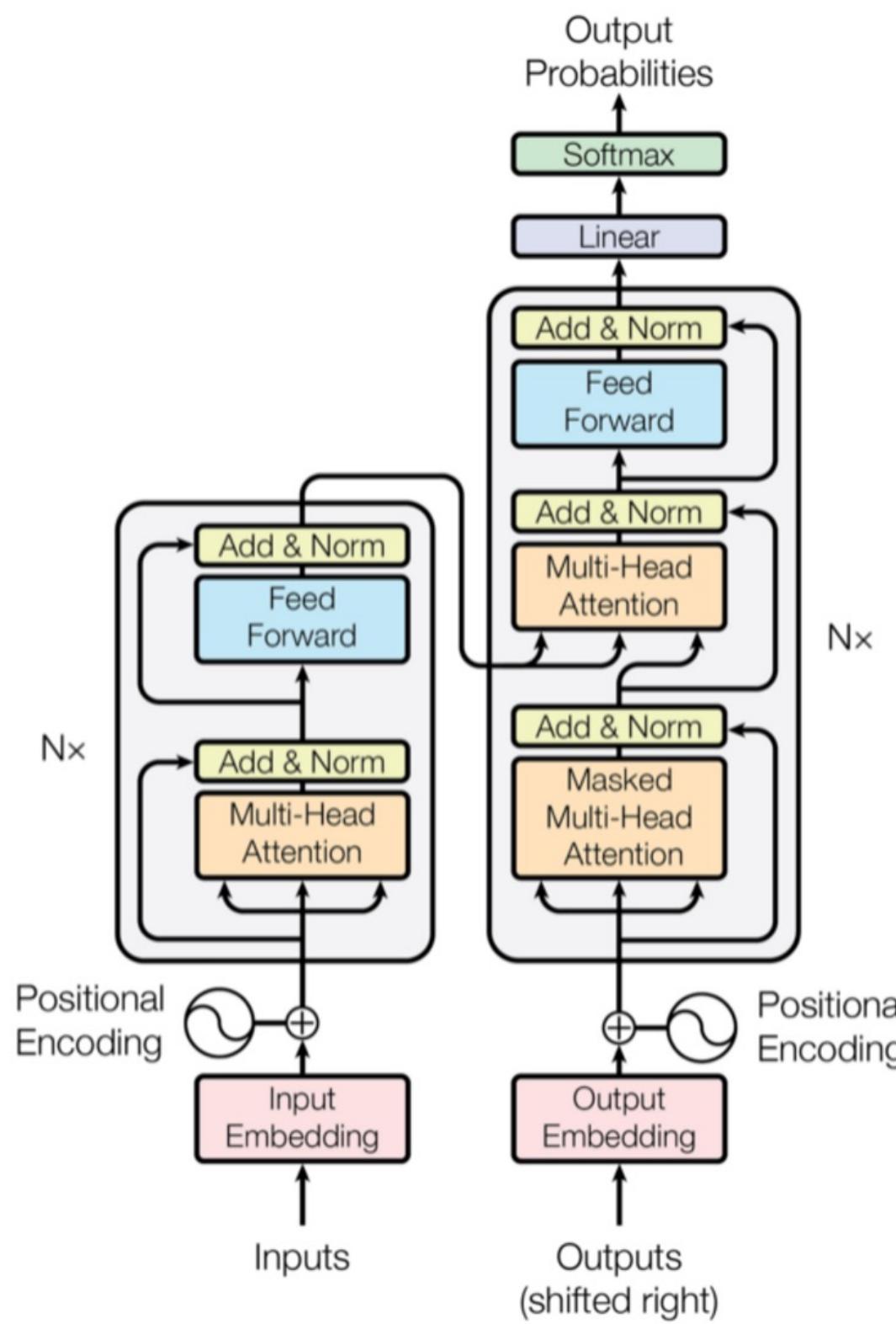
# Summary of main methods (non-exhaustive list)

Method	Pros	Cons
<b>Word2vec</b> e.g. CBOW, Skip-gram	<ul style="list-style-type: none"><li>• Very simple, yet powerful</li><li>• Intuitive embeddings</li></ul>	<ul style="list-style-type: none"><li>• Word order does not count</li><li>• Embeddings not context aware</li></ul>
<b>Recurrent Neural Networks</b> e.g. traditional RNN, LSTM	<ul style="list-style-type: none"><li>• Word order matters</li><li>• State-of-the-art results</li></ul>	<ul style="list-style-type: none"><li>• Vanishing gradient problem</li><li>• Embeddings not context aware</li><li>• Slow computations</li></ul>

## Tokenization summary

Method	Pros	Cons
<b>Word-level</b>	<ul style="list-style-type: none"><li>• Simple</li><li>• Interpretable</li></ul>	<ul style="list-style-type: none"><li>• Risk of OOV</li><li>• Does not leverage knowledge of root</li></ul>
<b>Subword-level</b> e.g. WordPiece, BPE	<ul style="list-style-type: none"><li>• Leverages prefix suffixes</li><li>• Learned from the data</li></ul>	<ul style="list-style-type: none"><li>• Risk of OOV, though less than word-level</li></ul>
<b>Character-level</b>	<ul style="list-style-type: none"><li>• Small chance of OOV</li><li>• RoBUsT tO CASInG anD MiSpeliNGs</li></ul>	<ul style="list-style-type: none"><li>• Makes computations slower</li></ul>

# Transformer architecture



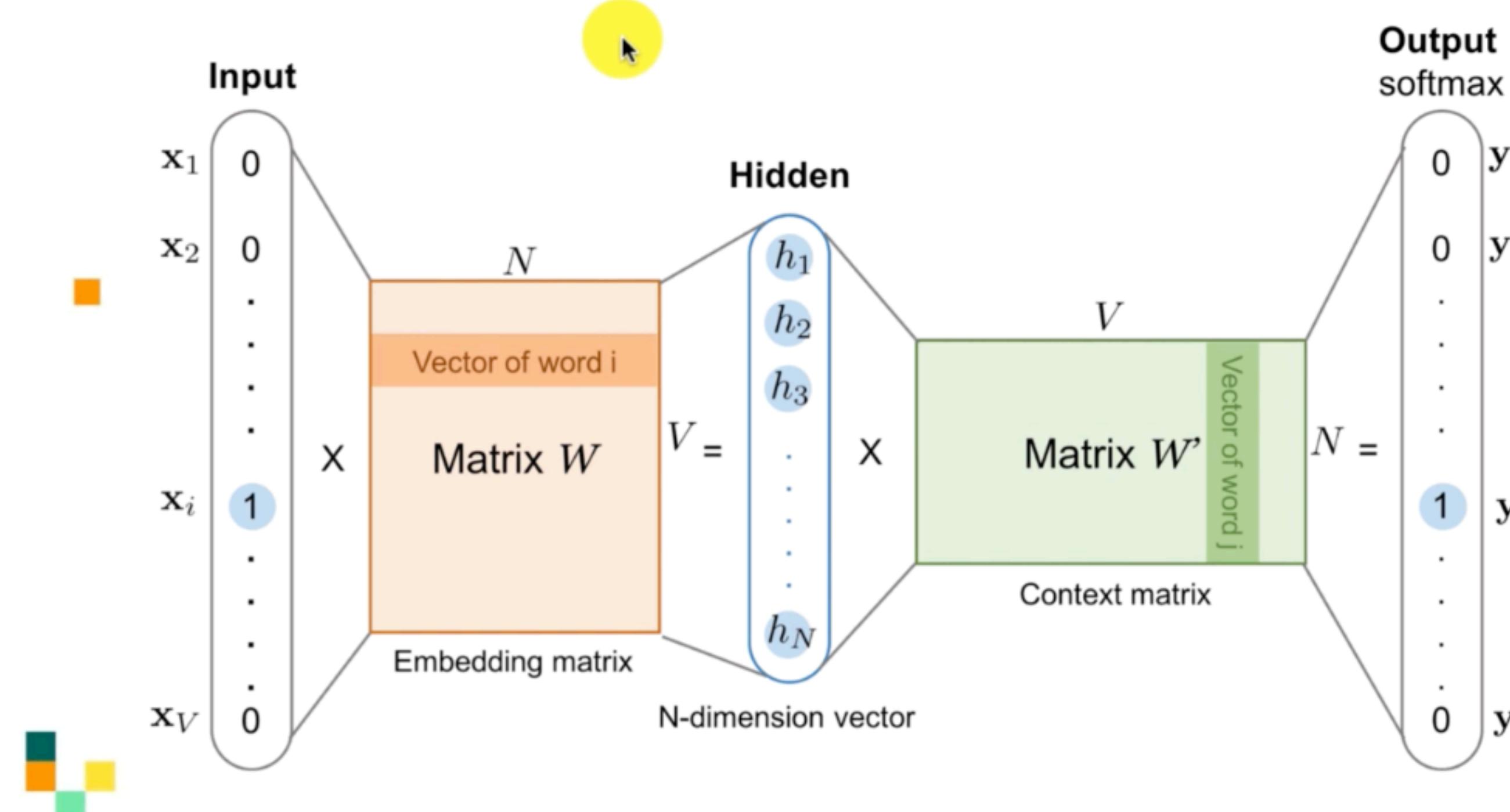
- **Attention layer (MHA)**
  - Self-attention (Encoder-Encoder, Decoder-Decoder)
  - Encoder-Decoder attention layer
- **Feed Forward Neural Network (FFNN)**
- **Positional Encoding (PE)**

Figure adapted from "[Attention Is All You Need](#)", Vaswani et al., 2017.

# Vizualização de atenção na tradução

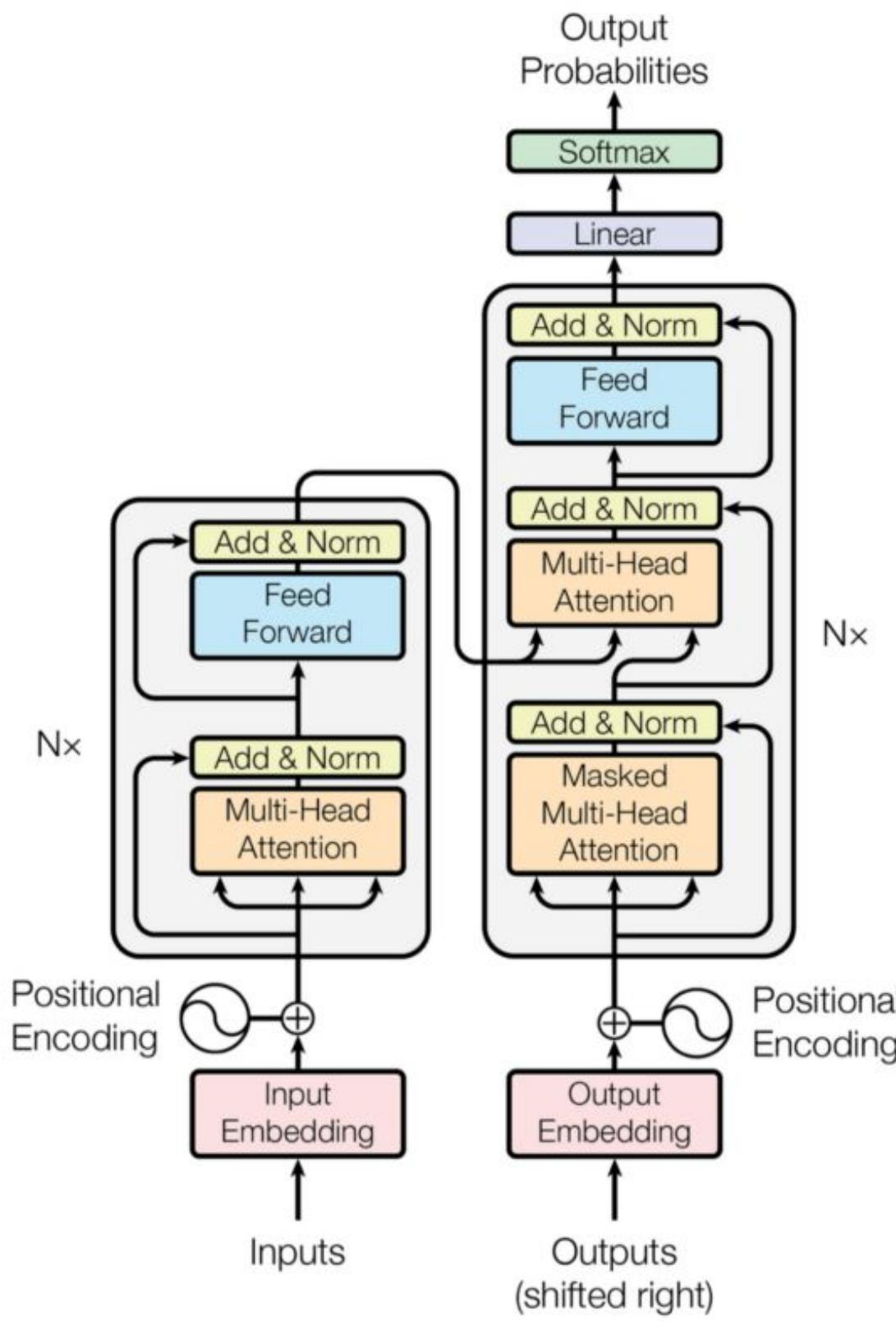
- <https://www.udemy.com/course/bert-nlp-algorithm/learn/lecture/17329584#notes>

# Word embedding

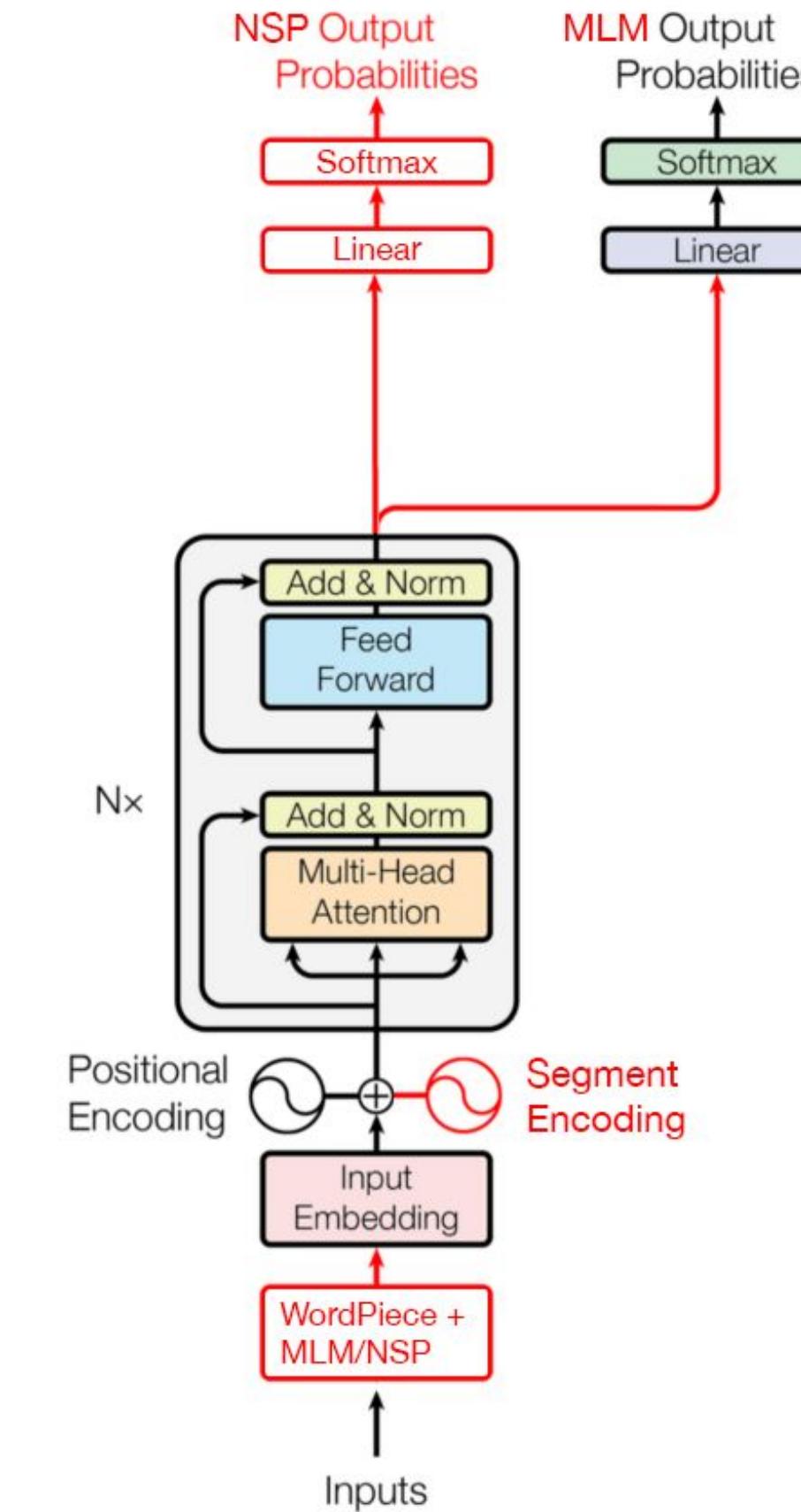


**Skip-gram model:** in sentence  
“In spite of everything, I still  
believe people are really good at  
heart”, word “good” produces  
pairs (“good”, “are”), (“good”,  
“really”), (“good”, “at”), (“good”,  
“heart”) as target/context.

# BERT: overview of the changes



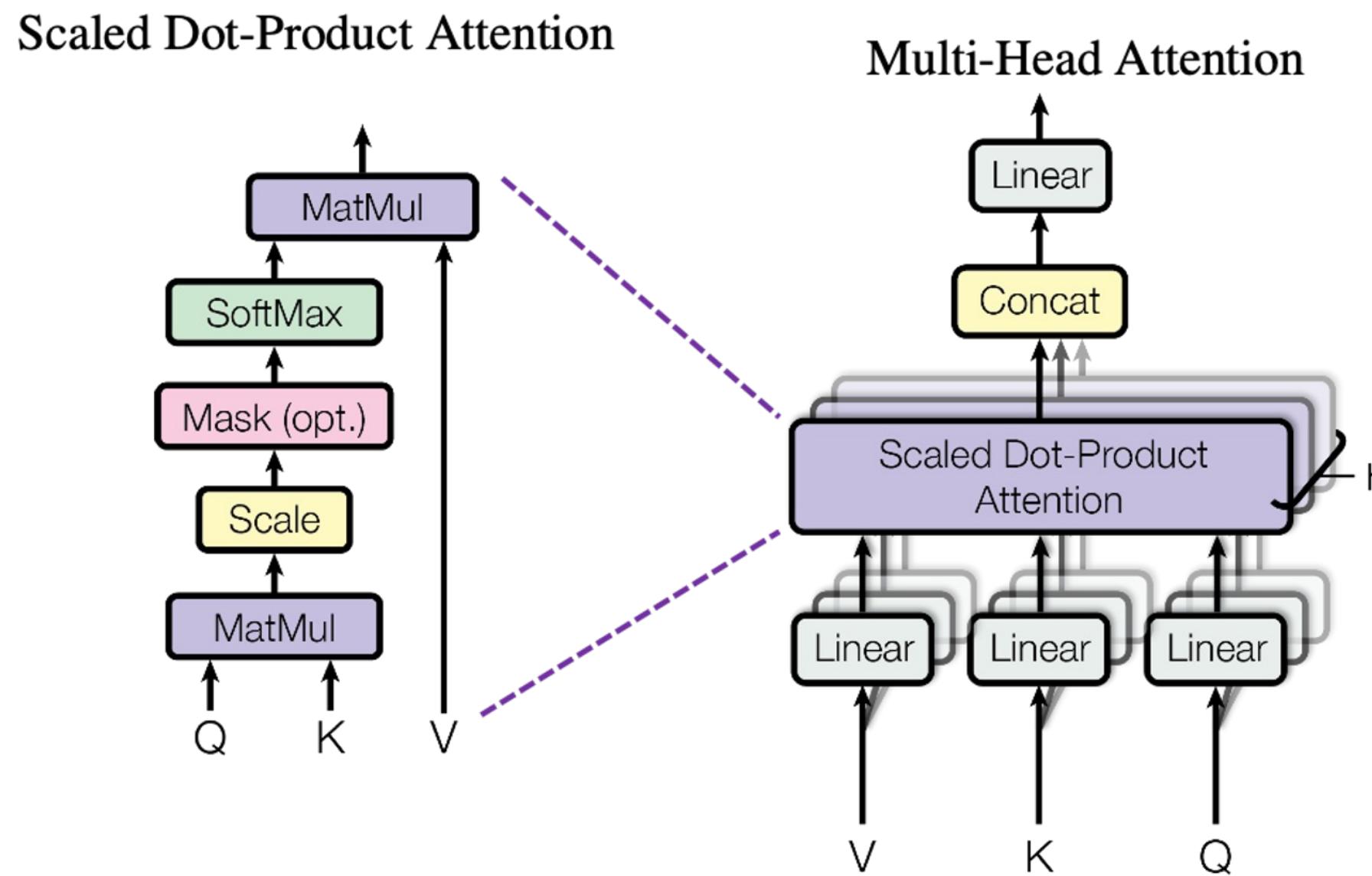
Original transformer (2017)



BERT (2018)

# Attention mechanism

- **Query, Key, Value**

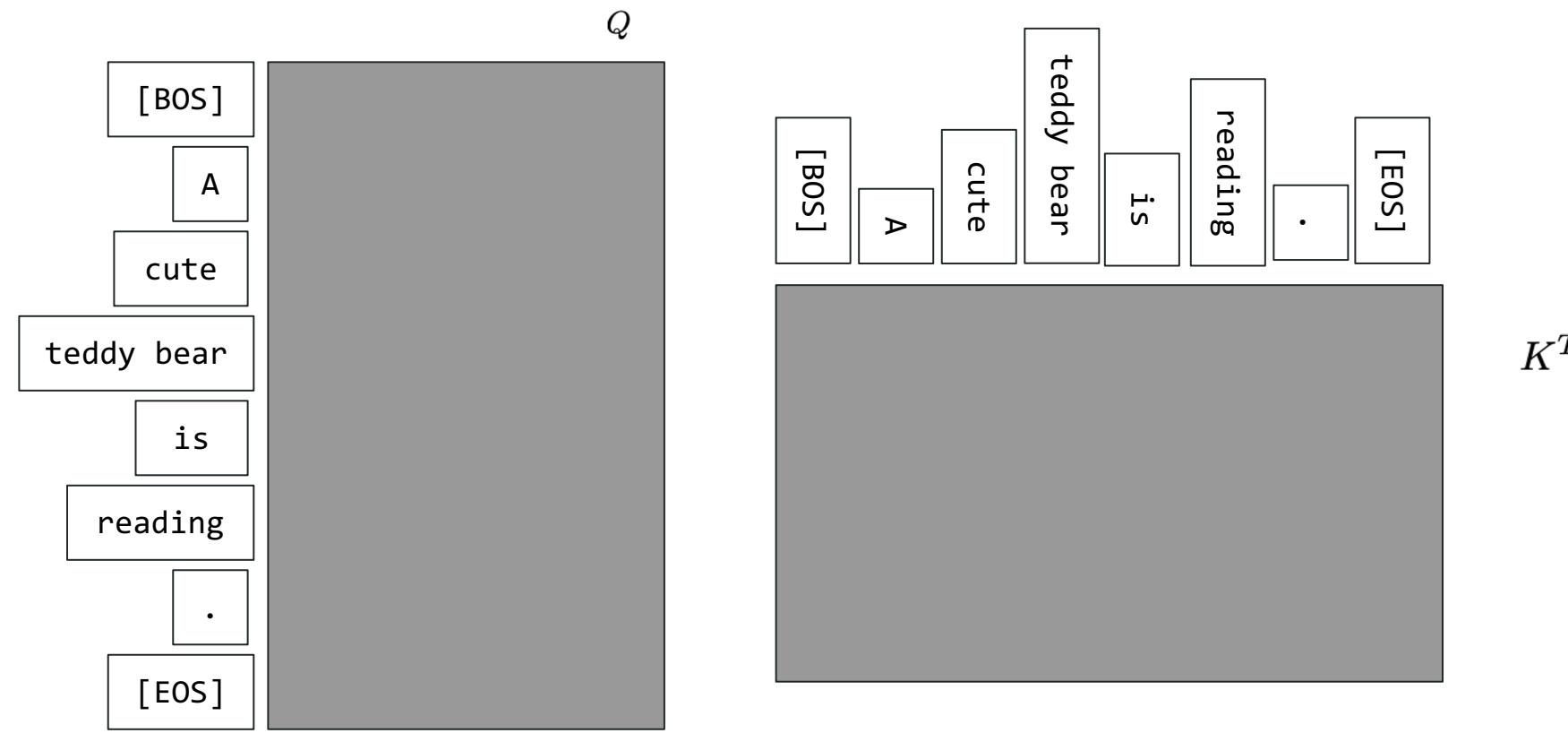


- Computationally efficient with matrices

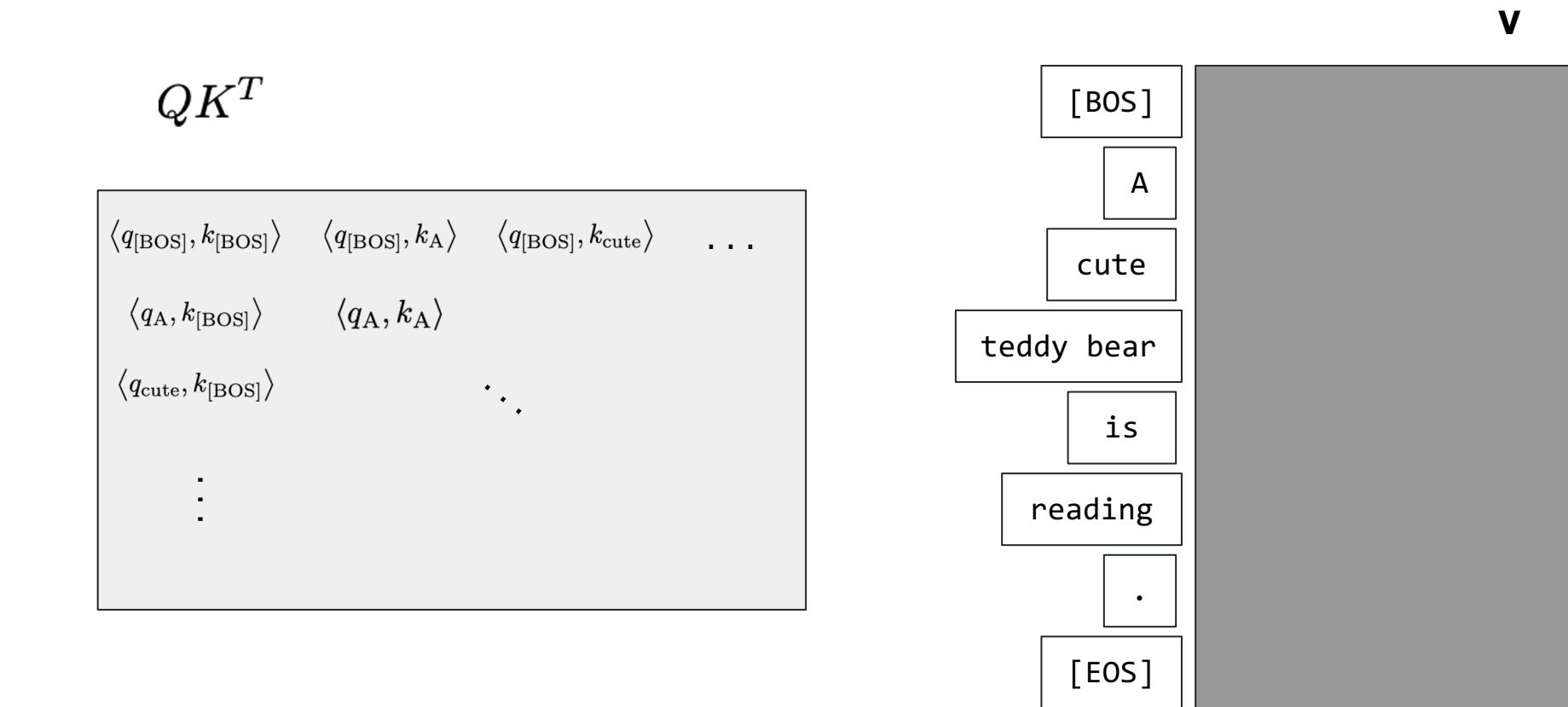
$$\text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

## Stitching all the pieces together with an example



## Stitching all the pieces together with an example



## Stitching all the pieces together with an example

$$\begin{aligned} & \langle q_{[BOS]}, k_{[BOS]} \rangle v_{[BOS]} + \langle q_{[BOS]}, k_A \rangle v_A + \langle q_{[BOS]}, k_{\text{cute}} \rangle v_{\text{cute}} + \dots \\ & \langle q_A, k_{[BOS]} \rangle v_{[BOS]} + \langle q_A, k_A \rangle v_A + \langle q_A, k_{\text{cute}} \rangle v_{\text{cute}} + \dots \\ & \quad \vdots \end{aligned}$$

$QK^TV$

## Stitching all the pieces together with an example

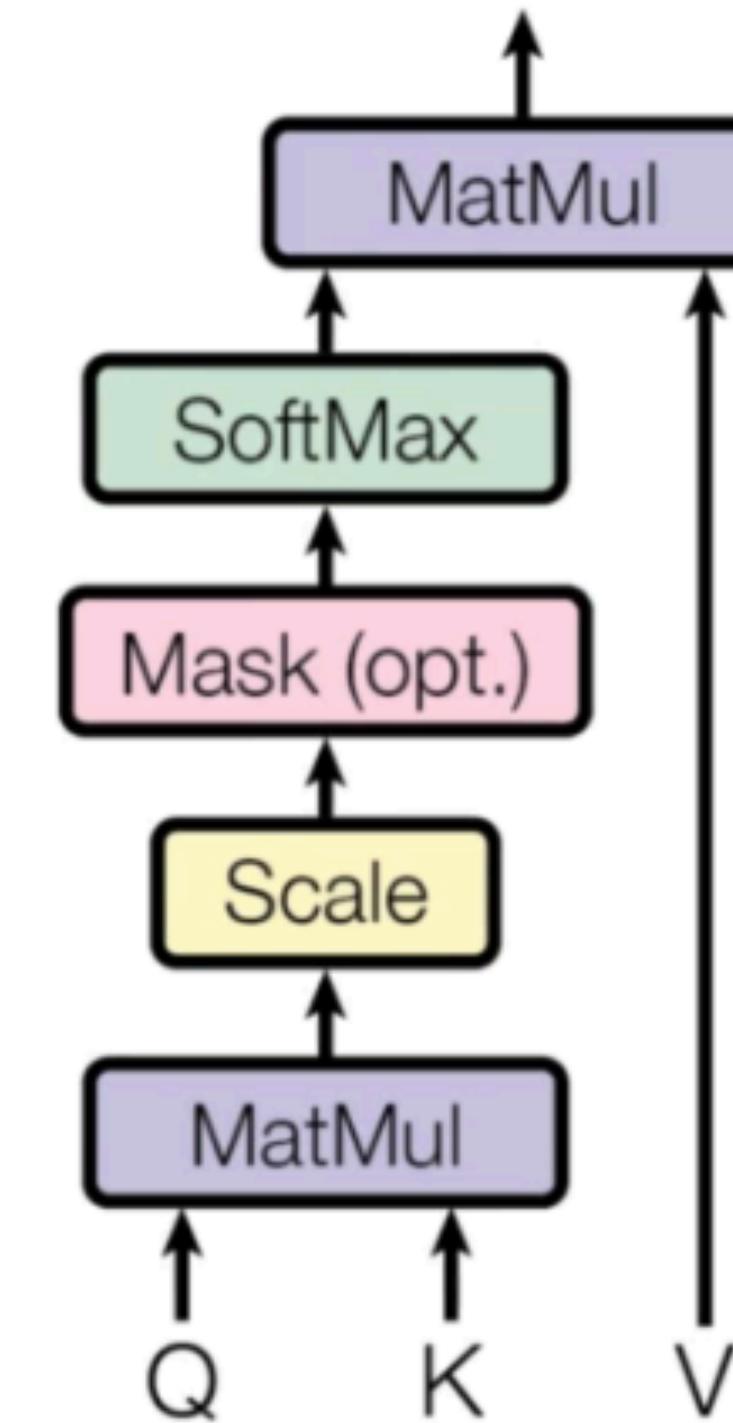
weighted average of values  
with weights being a function of  $\langle q, k \rangle$

softmax  $\left( \frac{QK^T}{\sqrt{d_k}} \right) V$

## Scaled-dot product

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

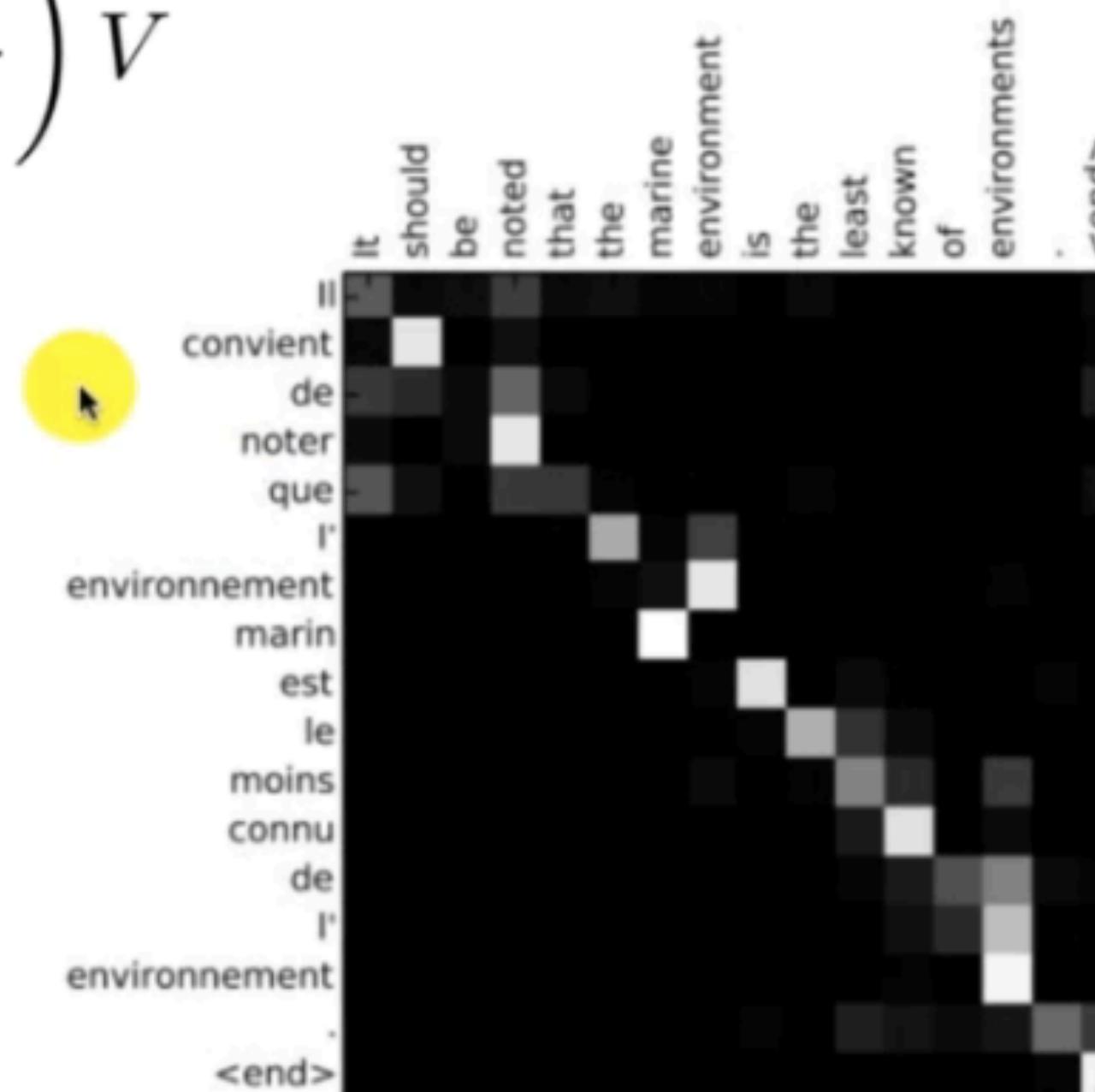
Q, K and V are matrices representing sentences/sequences (after embedding)



## Scaled-dot product

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

$QK^T$  example:

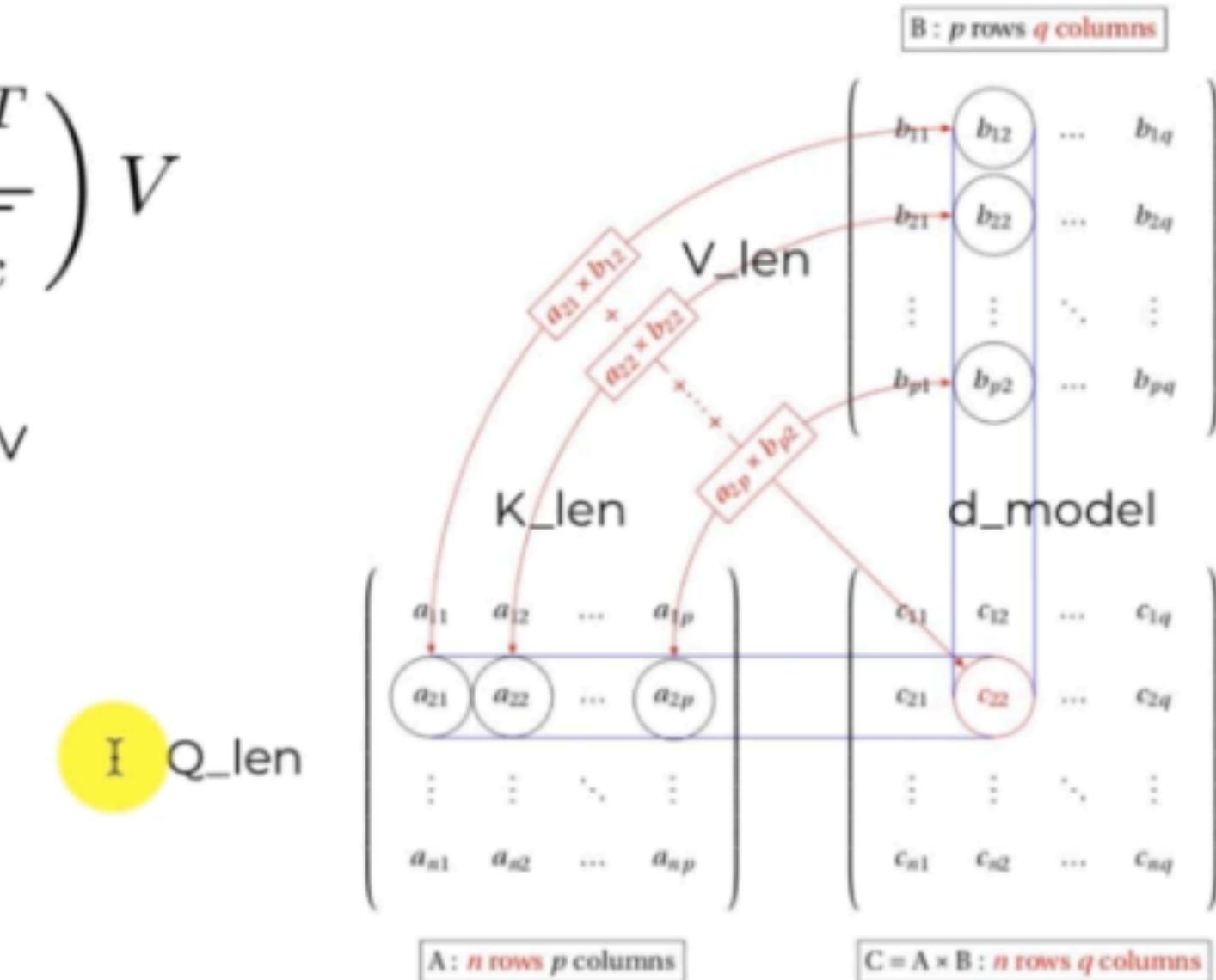


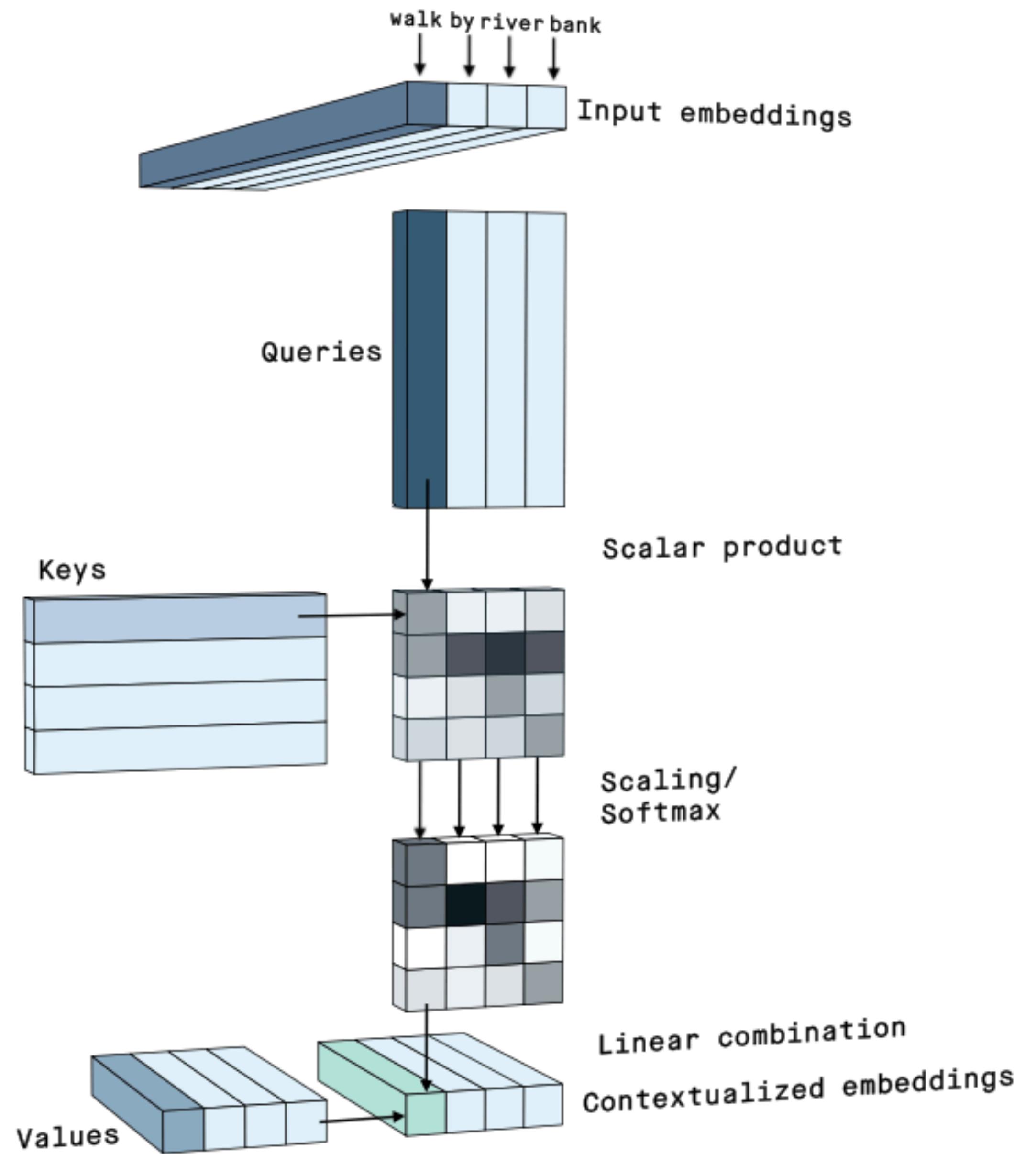
## Scaled-dot product

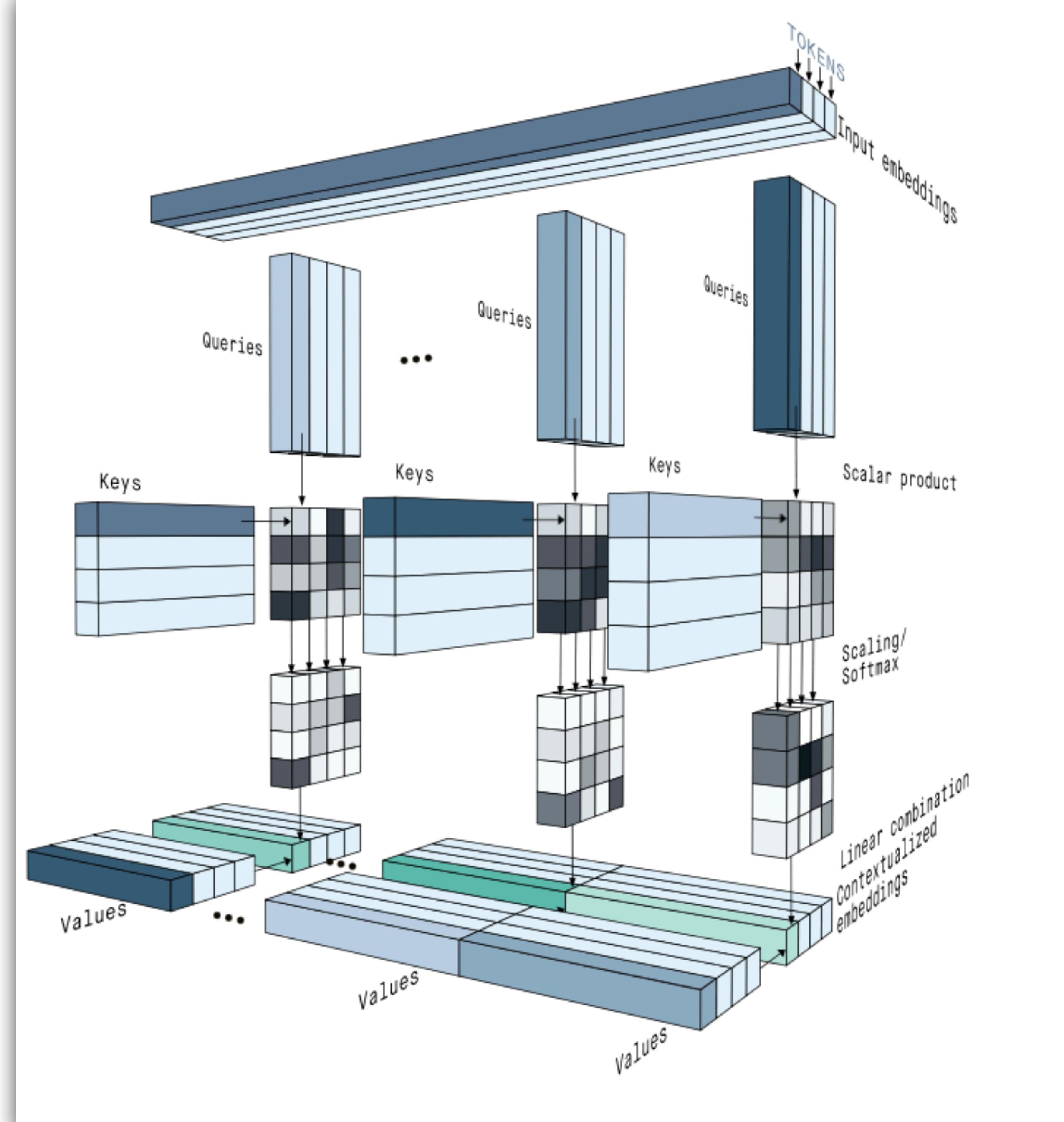
$QK^T$  says how Q is related to K, word by word. Let's combine  $V (= K)$  according to it.

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

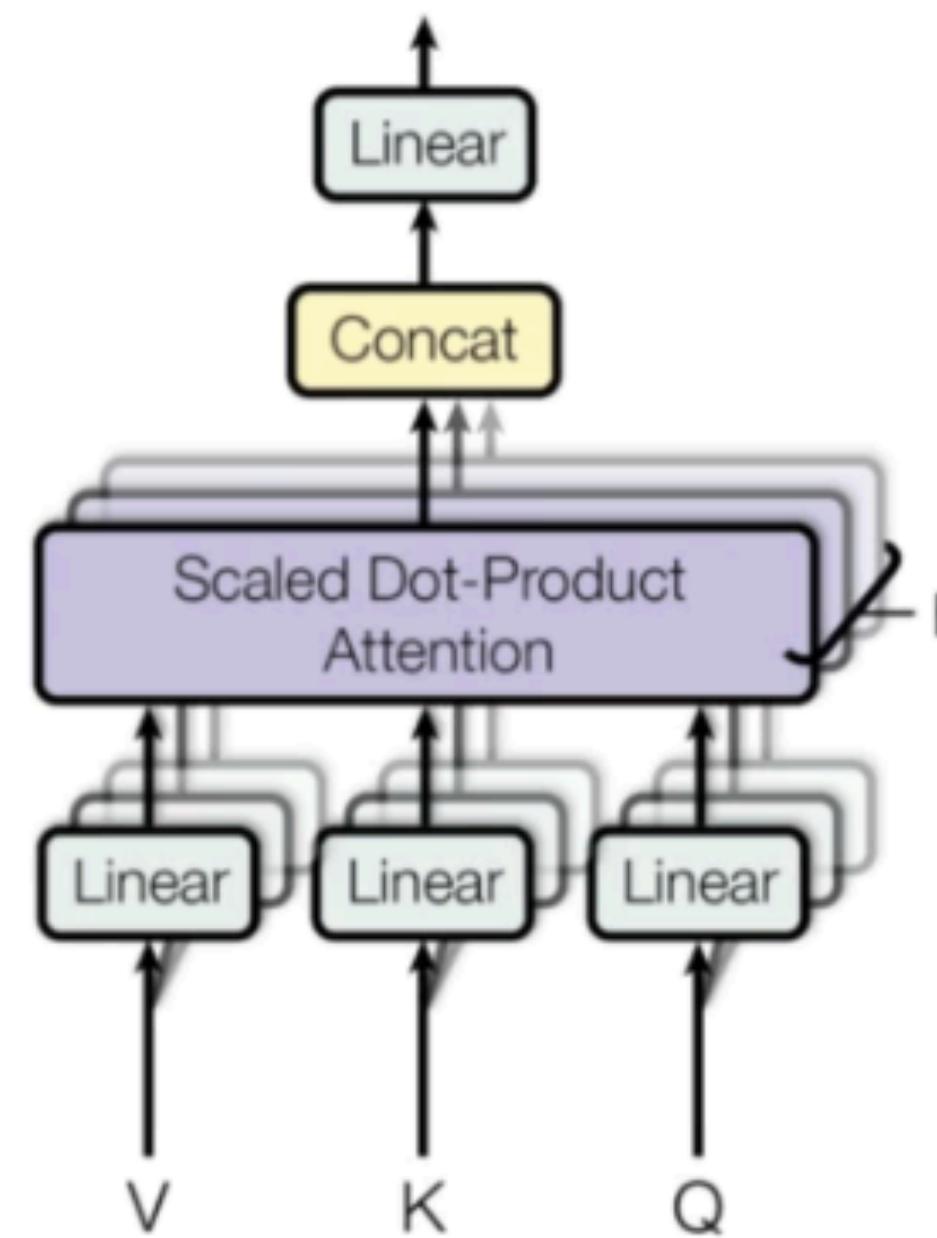
has same shape as Q, but is made with elements from V with respect to their correlations with Q.





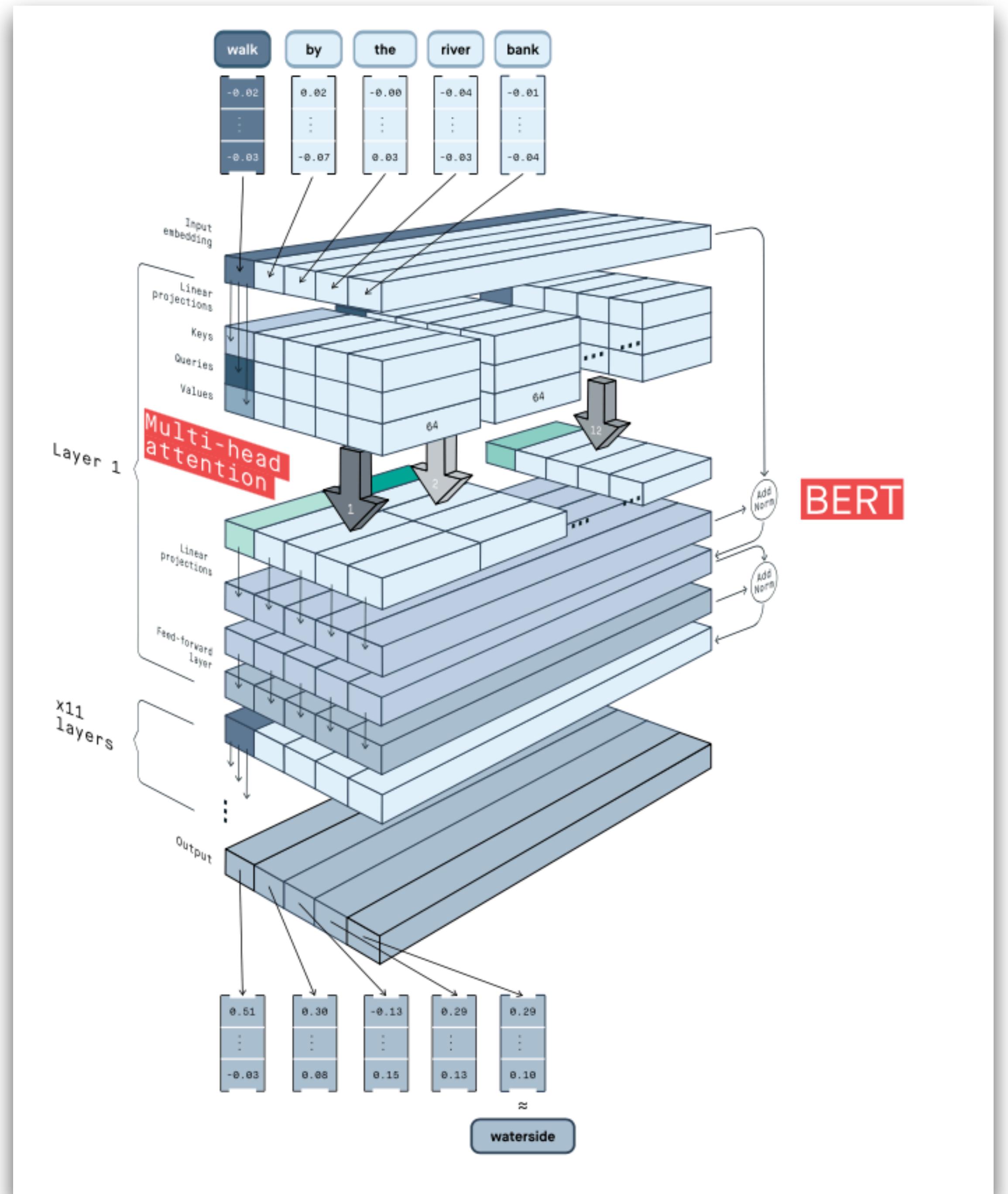


## Multi-head attention layer

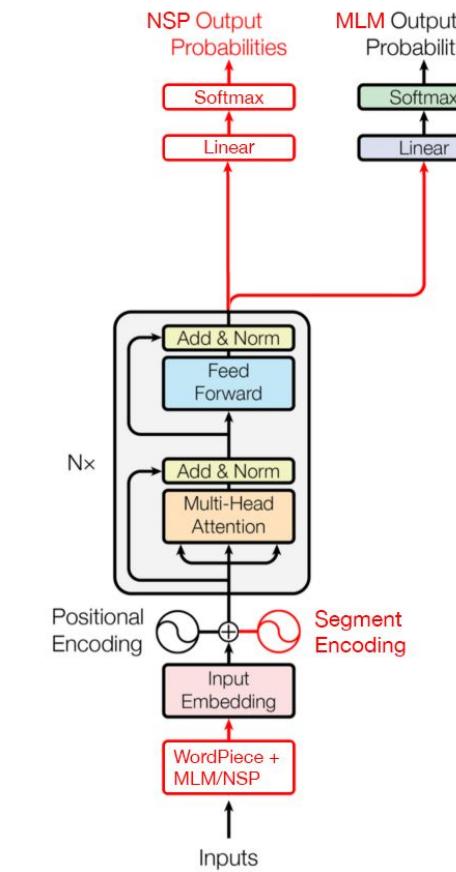


**Linear projections:** let's apply our attention mechanism to multiple learned subspaces.

"Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this"



# Overview



**Goal:** leverage general language representation for NLP tasks

# Pretraining

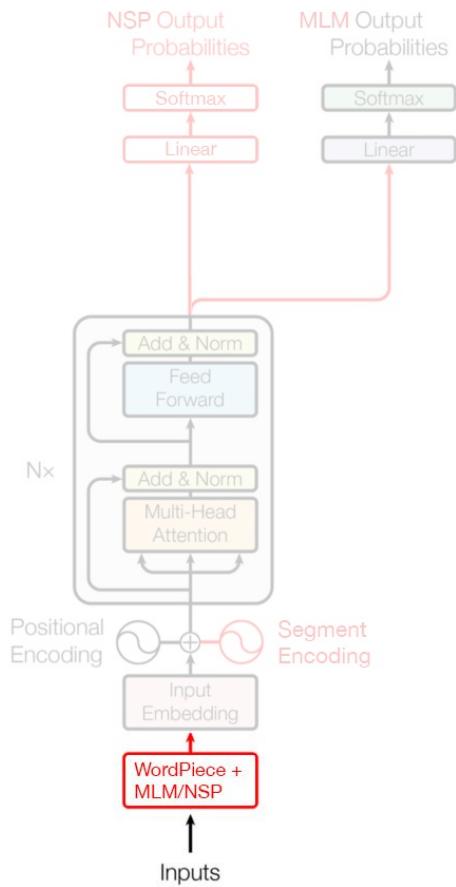
- Data: enormous unlabeled corpus of Books (800M words) and Wikipedia (2.5B words)
  - MLM task: predict 15% of input tokens
  - NSP task: predict whether sentences follow each other or not

# Fine-tuning

- Dataset: task-specific
  - Objective: tailored to end goal

Figure adapted from “Attention Is All You Need”, Vaswani et al., 2017.

# Input processing



## WordPiece algorithm

- Tokenizer trained on a training set beforehand
  - Vocabulary size: ~30,000
  - Great at detecting common particles

# NSP task processing

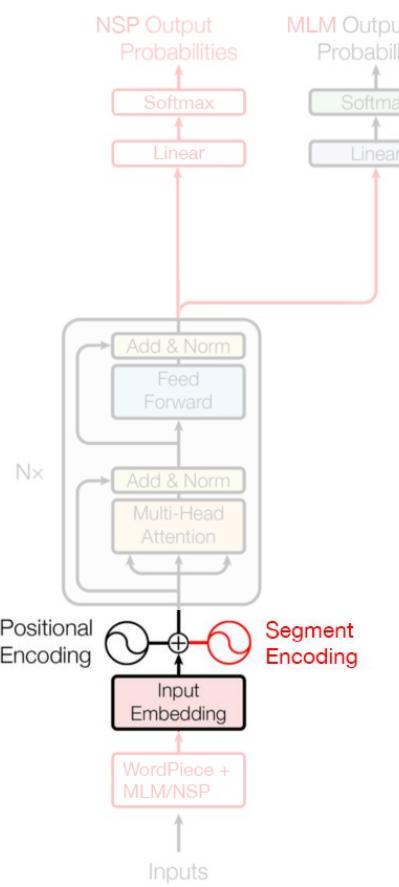
- Add [CLS] token at the beginning of the input

# MLM task processing

- Separate consecutive segments with the [SEP] token and put another one at the end

Figure adapted from “Attention Is All You Need” Vaswani et al. 2017

## Input embedding



### Input embeddings

- Gigantic lookup table
- Learns an embedding for each word of the vocabulary

### Positional encoding

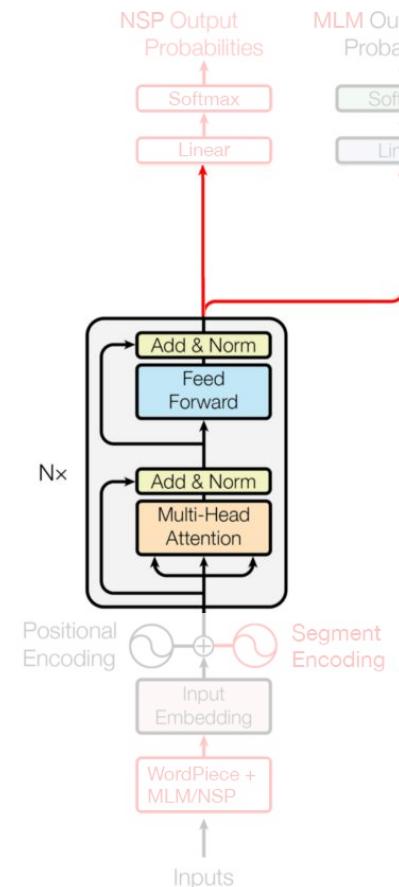
- Helps the network associate tokens with a position
- Encoding either learnt or fixed with cosines and sines

### (new!) Segment encoding

- Shared embedding for a segment

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

## Encoder-only model



### Model

Encoder part of the original transformers paper

### Goal

- Represent input data with features (hopefully) needed for NLP tasks
- Leverage the Transformer's self-attention mechanism
- Use learned embedding towards classification-oriented tasks

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

**Transfer learning**

**BERT's pre-training:**

- data: BooksCorpus (800M words) and English Wikipedia (2,5000M words). We need document-level corpus to have long contiguous sequences.
- 2 tasks: predicting a word, and next sentence prediction
- around three days on 16 TPU

**BERT - General idea**

## Proxy tasks

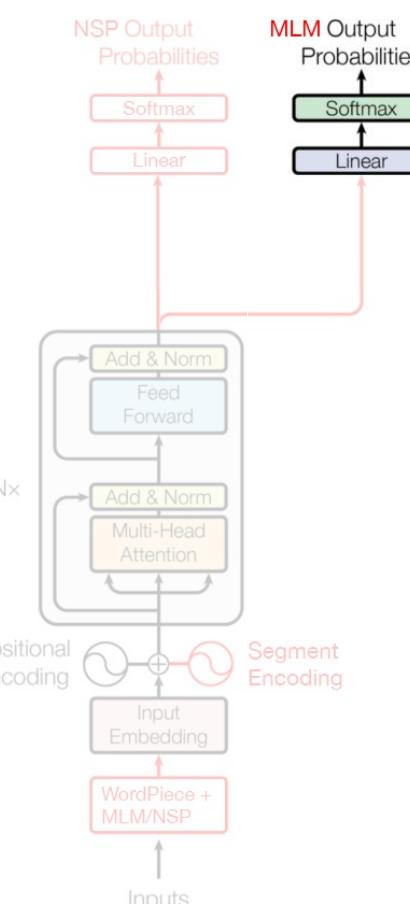


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

### Masked Language Modeling

Idea:

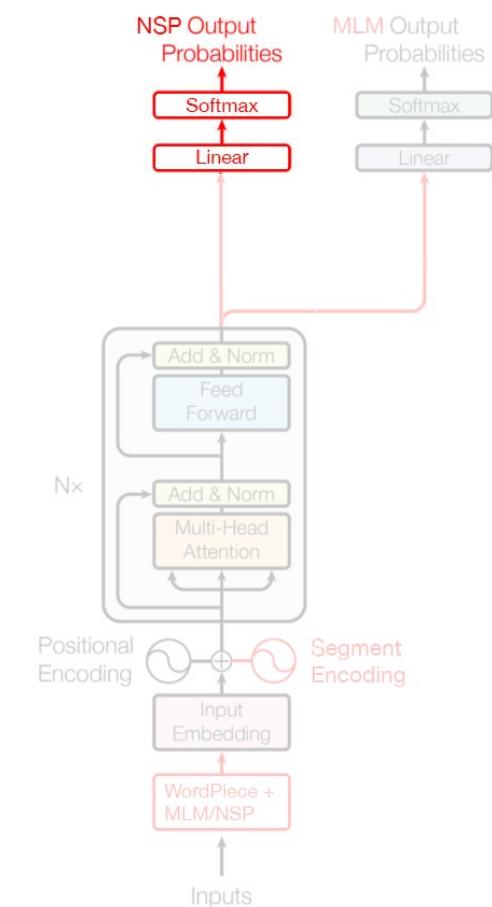
15% of input tokens are set up for prediction where

- 80% are masked
- 10% are changed to a random word
- 10% are unchanged

Benefits:

- Network learns language modeling based on contextual information
- Regularization reflects probabilistic nature of language

## Proxy tasks



### Next Sentence Prediction

Idea: pick two sentences from the corpus, where

- 50% of the time, they follow each other
- 50% of the time, they **do not** follow each other

Task: predict if they actually follow each other

Benefits:

- Network implicitly learns to detect useful contextual information
- Easy classification task that does not require any labels

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

## Outputs

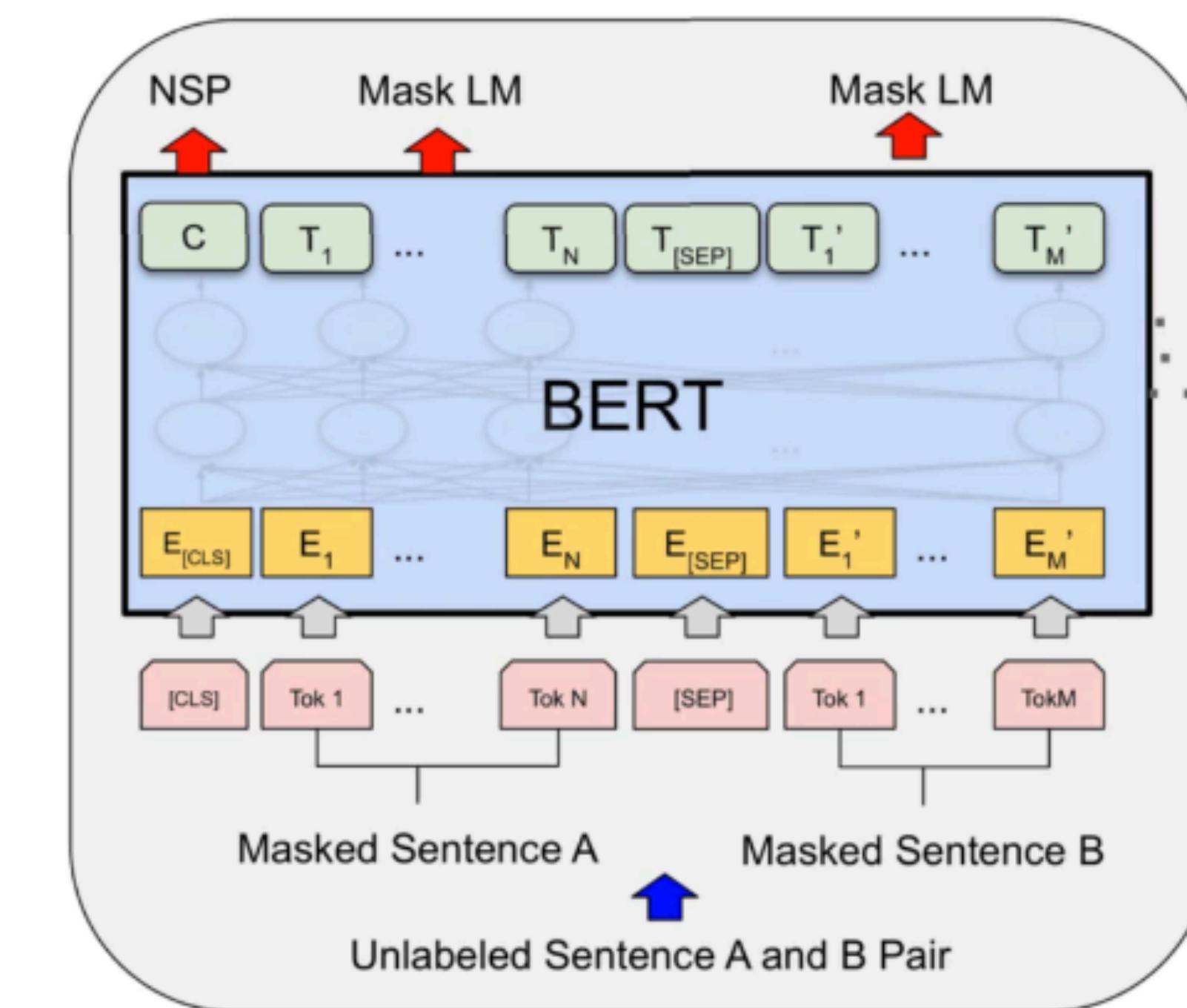
We get 2 types of outputs:

- Output for the [CLS] token
- Representation for each other token

C: used for classification. Trained during Next Sentence Prediction

$T_k$ : used for word-level task (embedding, Q&A). Trained during Mask Language Model.

Each token is a vector of the same size (hidden size H, 768 for BERT<sub>base</sub>).



# Main transformer-based models

<b>Architecture</b>	<b>Models</b>
Encoder	BERT, DistilBERT, RoBERTa
Decoder	GPT-2, 3
Encoder - Decoder	T5, mT5, ByT5

## General parameters

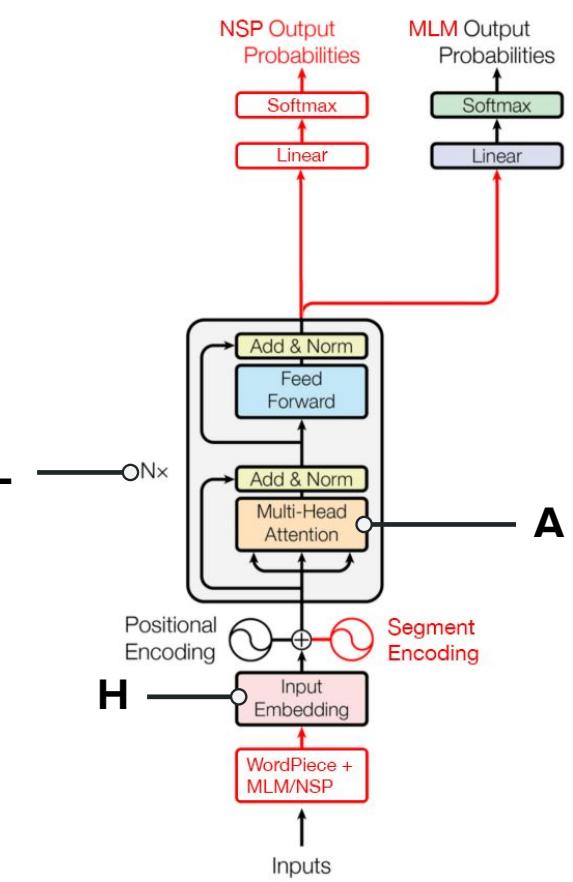


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

**Layers (L).** Number of layers. Corresponds to the "N" parameter in the original Transformers paper.

**Hidden (H).** Hidden layer size. Dimension of embeddings.

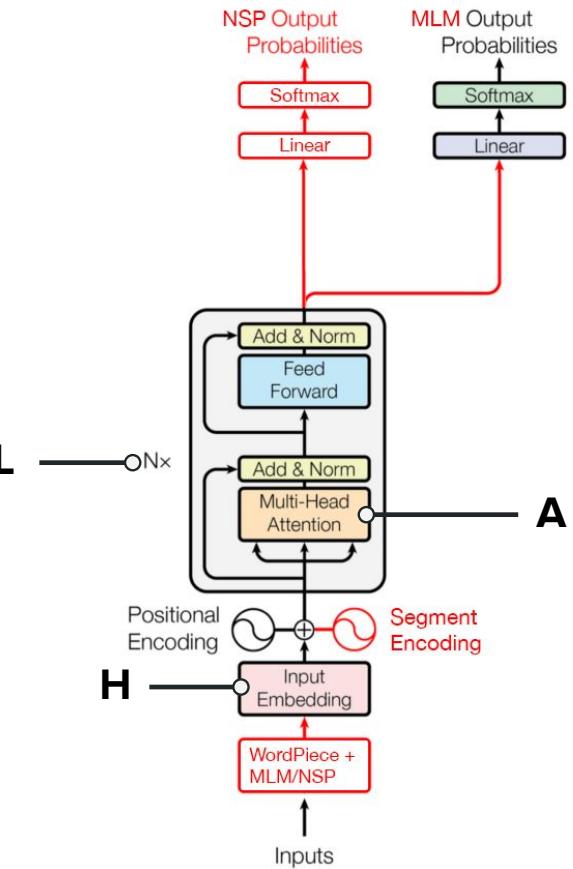
**Attention heads (A).** Number of attention heads operating in parallel.

**Original / whole word masking.** Whether separate tokens or entire words were masked during pre-training.

**Language-specific / multilingual.** Languages on which the model has been trained on.

**Cased / uncased.** Whether inputs are converted to lowercase or not.

## Some numbers



Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Parameters in the right table were computed in this paper.

	<b>L</b>	<b>H</b>	<b>A</b>	Parameters
<b>BERT-Tiny</b>	2	128	2	4M
<b>BERT-Mini</b>	4	256	4	11M
<b>BERT-Small</b>	4	512	8	30M
<b>BERT-Medium</b>	8	512	8	42M
<b>BERT-Base</b>	12	768	12	110M
<b>BERT-Large</b>	24	1024	16	330M

# Latest trends

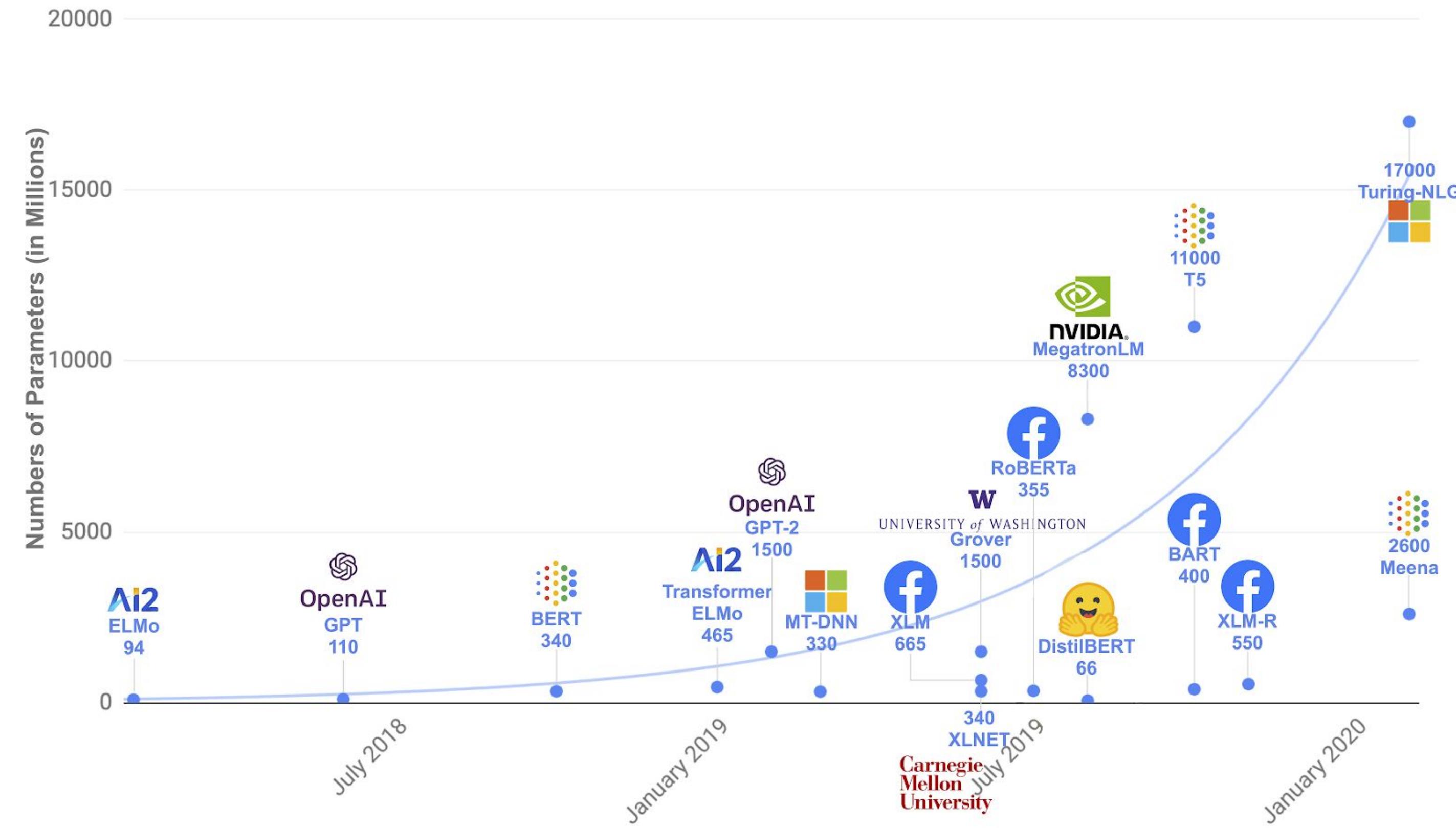


Figure adapted from this [Tensorflow blog post](#), initially designed by the HuggingFace team.

# Como usar o BERT?

- <http://www.r-text.org>

Kjell, O., & Schwartz, S. G. & A. (In progress). text: An R-package for analyzing and visualizing human language using natural language processing and deep learning (No. 0; Vol. 0, p. 0). <https://www.r-text.org/>

- **BERTimbau (Neuralmind)**

Souza, F., Nogueira, R., & Lotufo, R. (2020). BERTimbau: Pretrained BERT models for Brazilian Portuguese. 9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande Do Sul, Brazil, October 20-23 (to Appear).

<https://huggingface.co/neuralmind/bert-large-portuguese-cased>

<https://github.com/neuralmind-ai/portuguese-bert>

- <https://huggingface.co>

