Transformers e BERT

Prof. Dr. Ricardo Primi

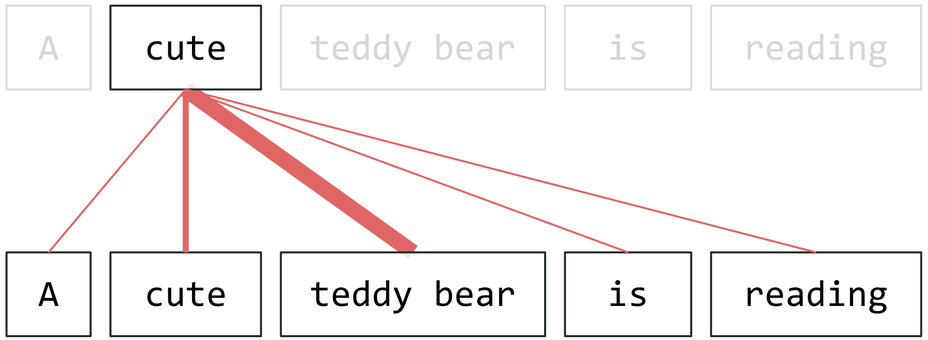
Introdução

Slides de:
Afshine Amidi (Stanford ICME, Workshop de verão de 2021)
Martin Jocqueviel: Learn BERT - most powerful NLP algorithm by Google (Udemy)

- BERT: **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- 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

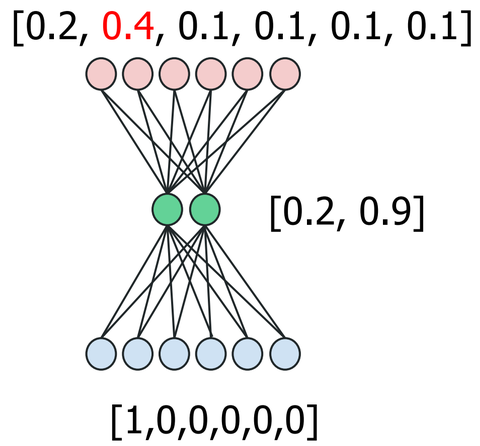


Embeddings estáticos: word2vec

Word2vec

Example with left context window = 1

A [cute] teddy bear is reading



[A] cute teddy bear is reading

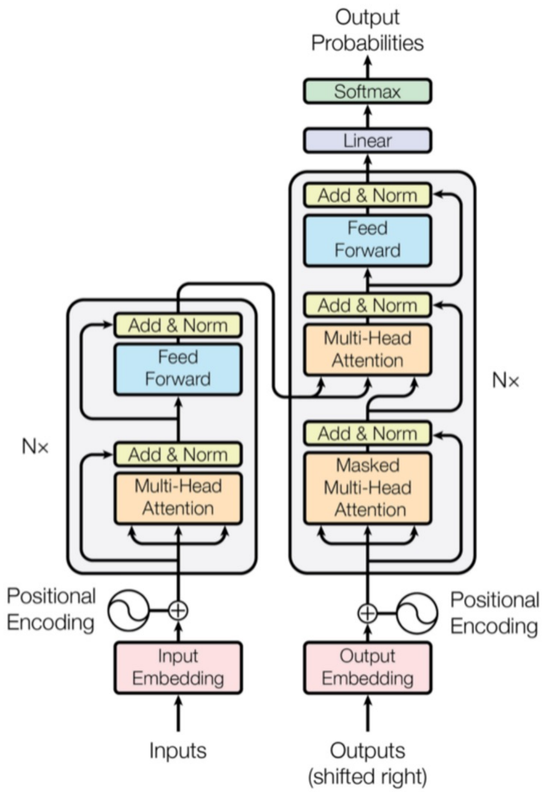
Summary of main methods (non-exhaustive list)

Method	Pros	Cons
Word2vec e.g. CBOW, Skip-gram	• Very simple, yet powerful • Intuitive embeddings	• Word order does not count • Embeddings not context aware
Recurrent Neural Networks e.g. traditional RNN, LSTM	• Word order matters • State-of-the-art results	• Vanishing gradient problem • Embeddings not context aware • Slow computations

Tokenization summary

Method	Pros	Cons
Word-level	• Simple • Interpretable	• Risk of OOV • Does not leverage knowledge of root
Subword-level e.g. WordPiece, BPE	• Leverages prefix suffixes • Learned from the data	• Risk of OOV, though less than word-level
Character-level	• Small chance of OOV • RoBUsT tO CASinG anD MIspeliNGs	• Makes computations slower

Transformer architecture



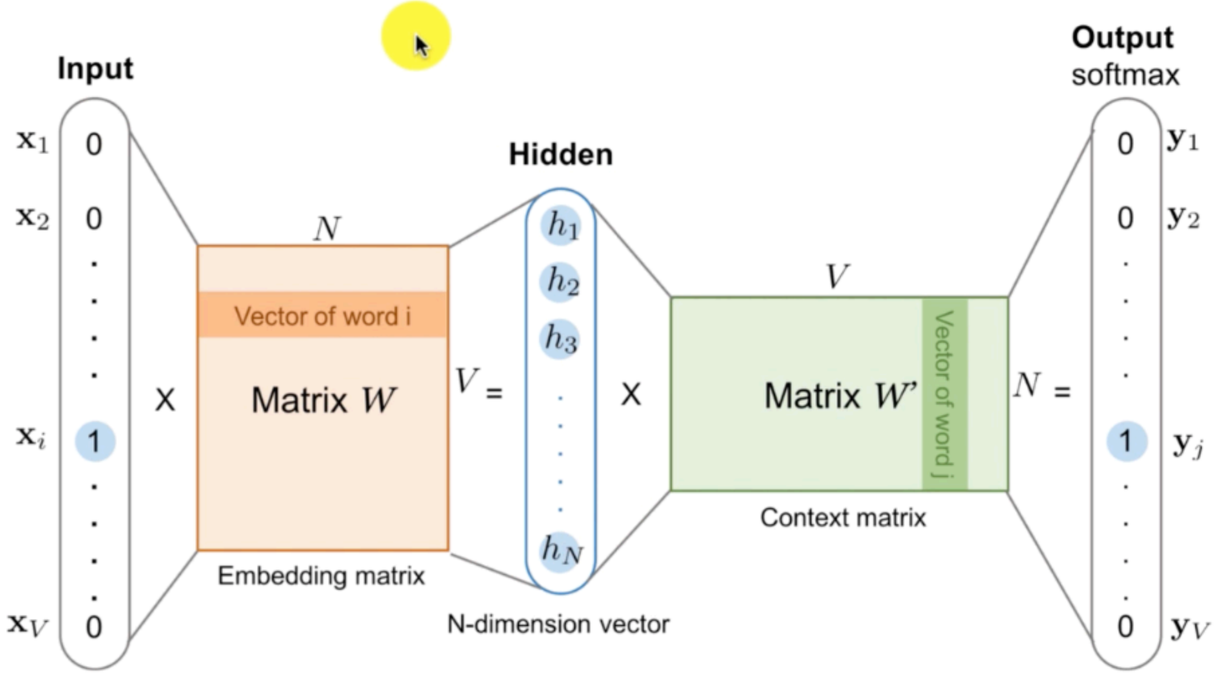
- **Attention layer** (MHA)
  - Self-attention (Encoder-Encoder, Decoder-Decoder)
  - Encoder-Decoder attention layer
- **Feed Forward Neural Network** (FFNN)
- **Positional Encoding** (PE)

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

Vizualização de atenção na tradução

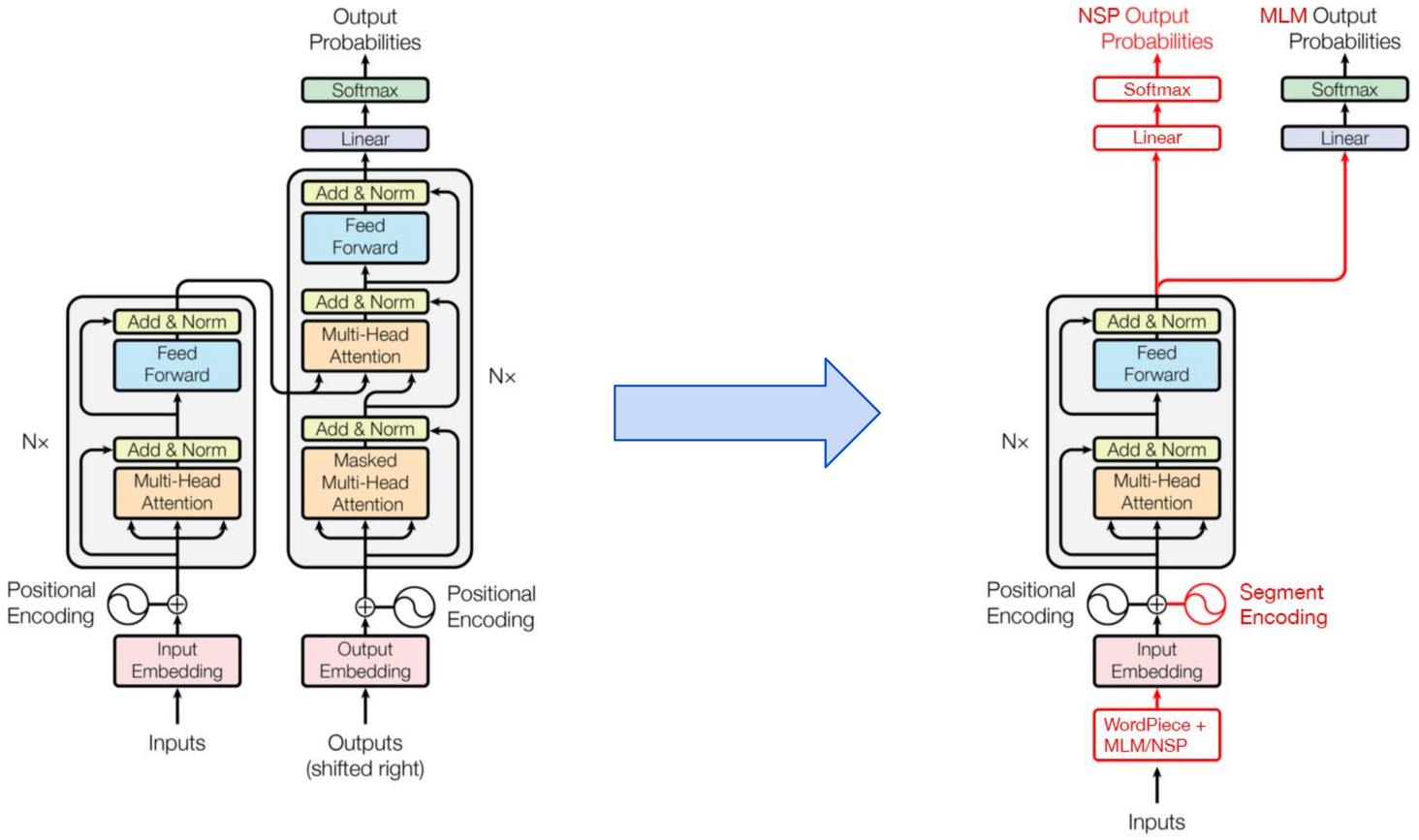
- https://www.udemy.com/course/bert-nlp-algorithm/learn/lecture/17329584#notes

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

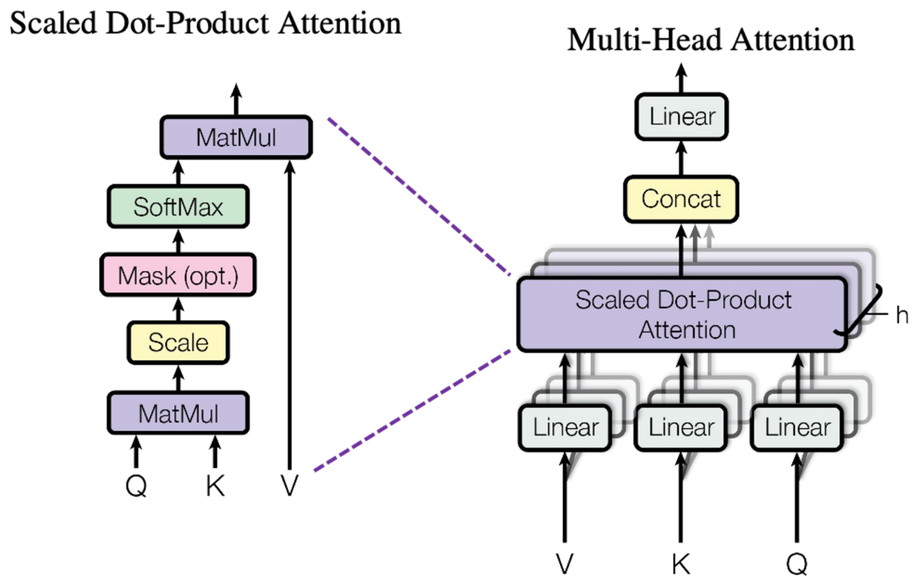


Original transformer (2017)

BERT (2018)

https://www.udemy.com/course/bert-nlp-algorithm/learn/lecture/17329584?start=574#notes

Attention mechanism

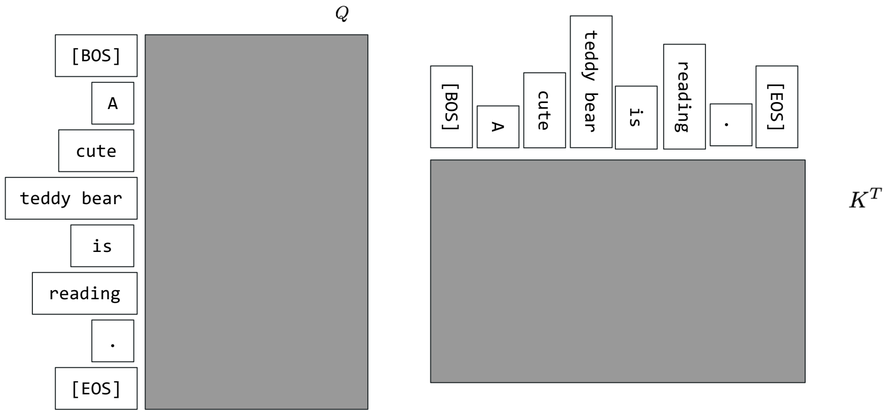


- **Q**uery, **K**ey, **V**alue
- Computationally efficient with matrices

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

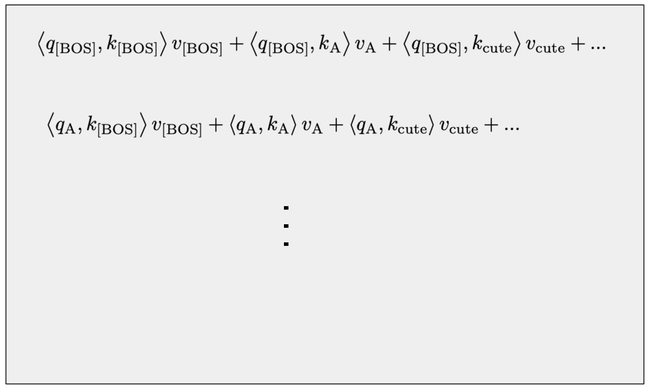
Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Stitching all the pieces together with an example



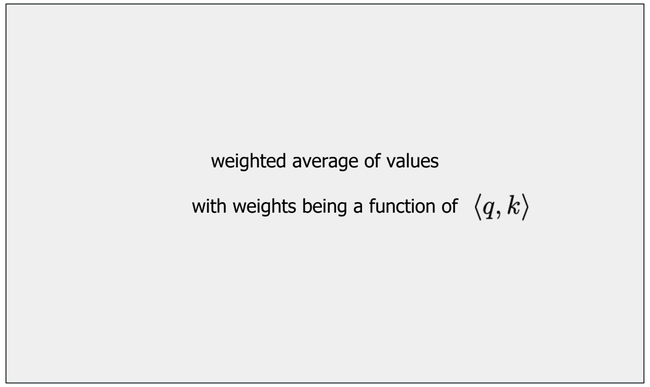
Stitching all the pieces together with an example

Stitching all the pieces together with an example



QK^TV

Stitching all the pieces together with an example

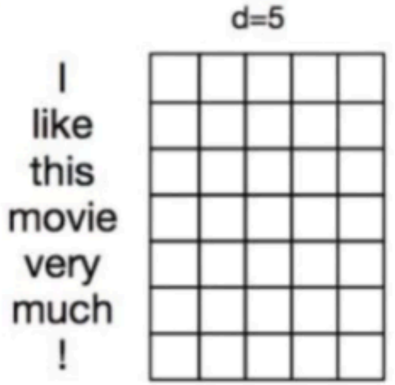


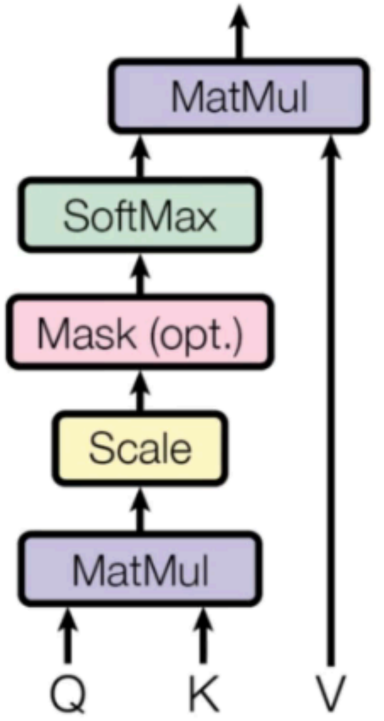
$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled-dot product

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

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

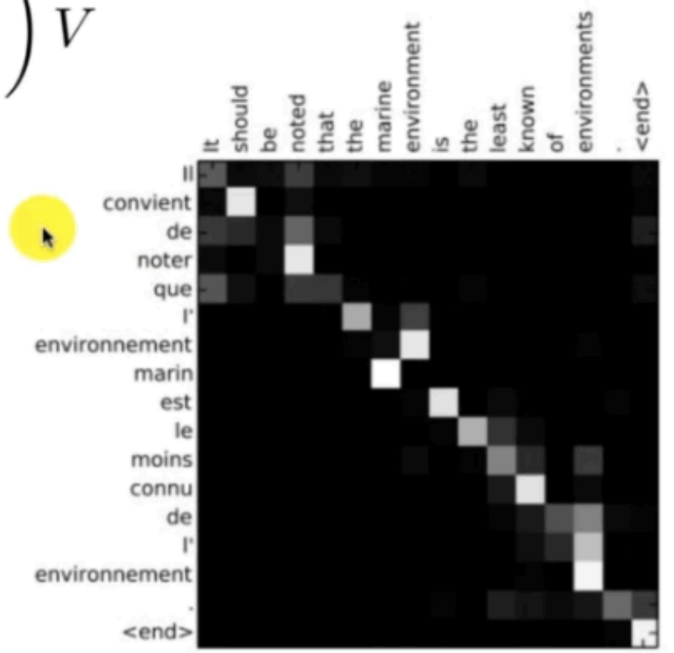




Scaled-dot product

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

QK^T example:

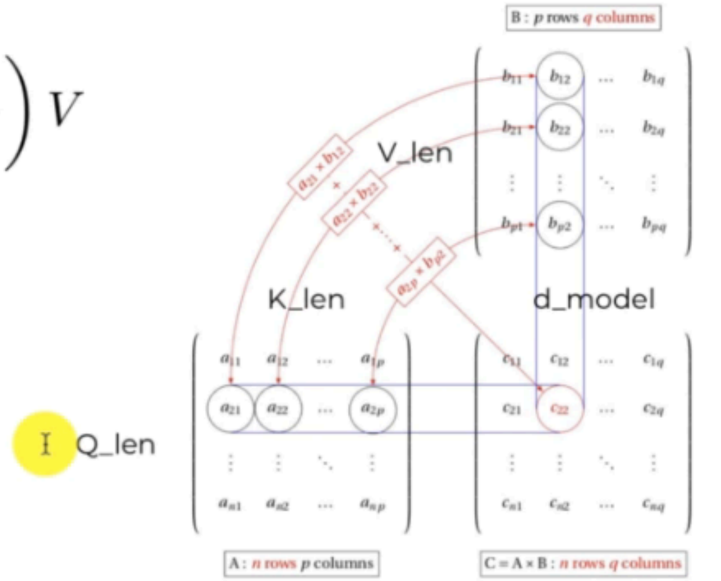


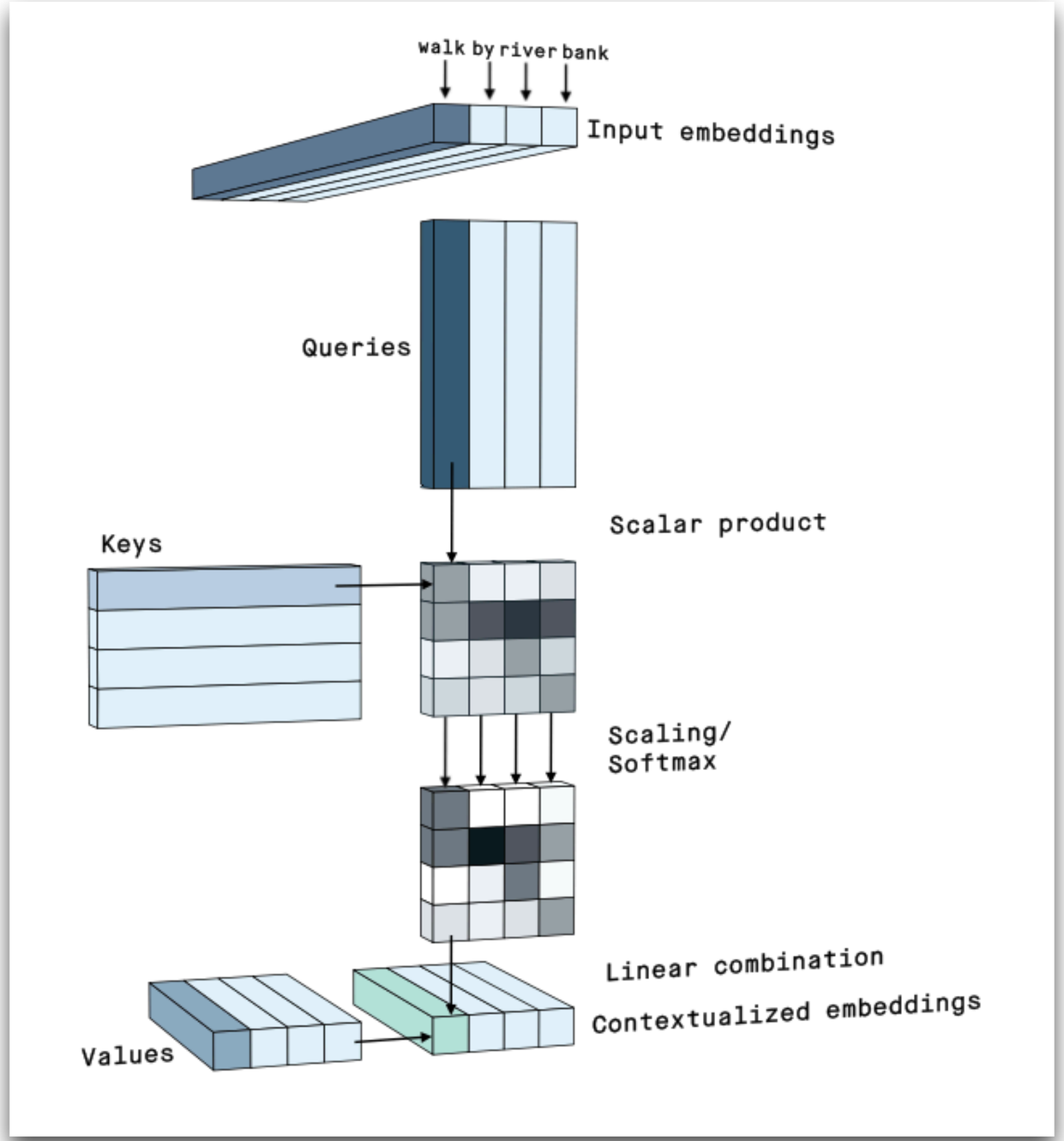
Scaled-dot product

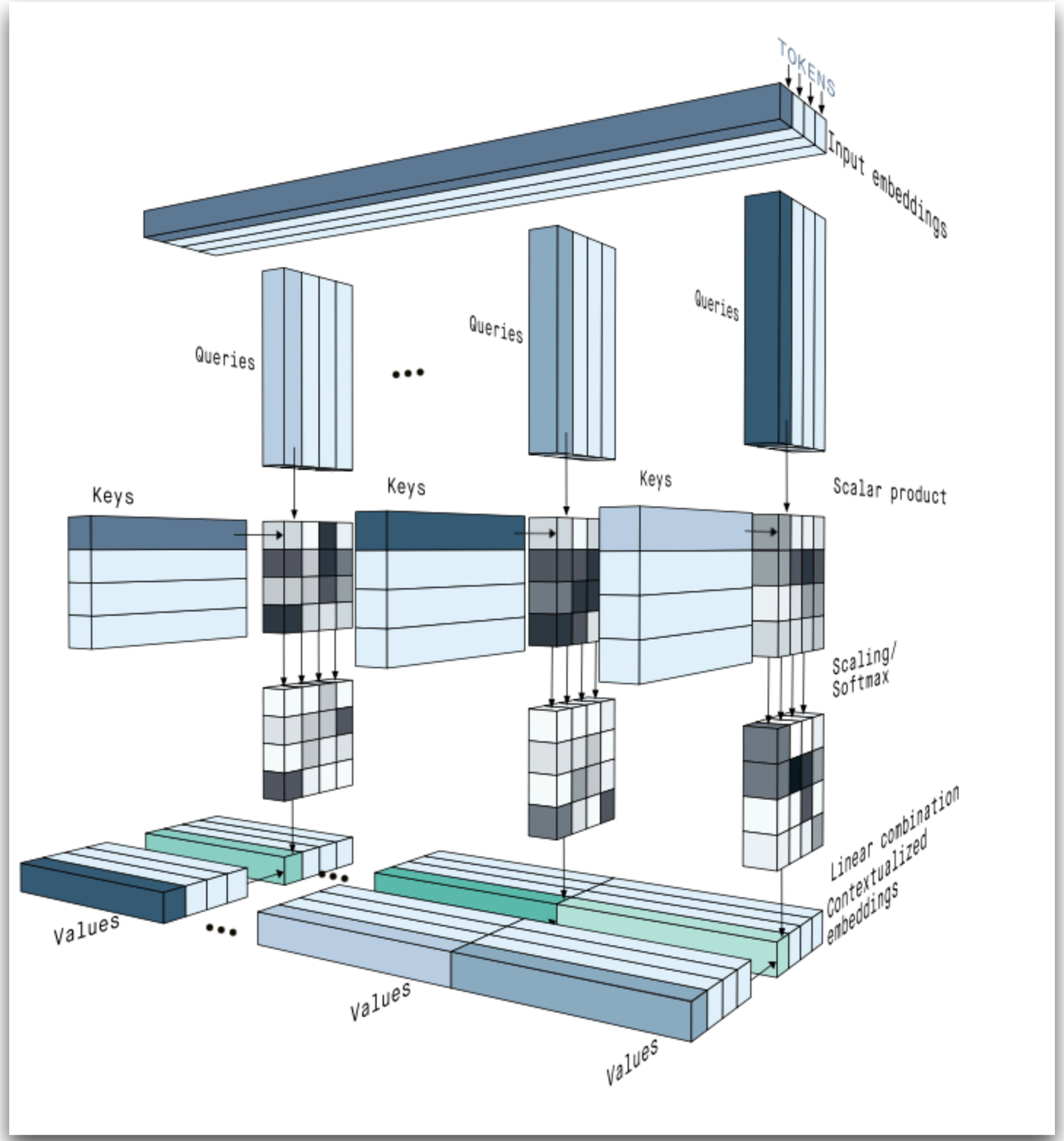
QK^T says how Q is related to K, word by word. Let's combine $V(=K)$ according to it.

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

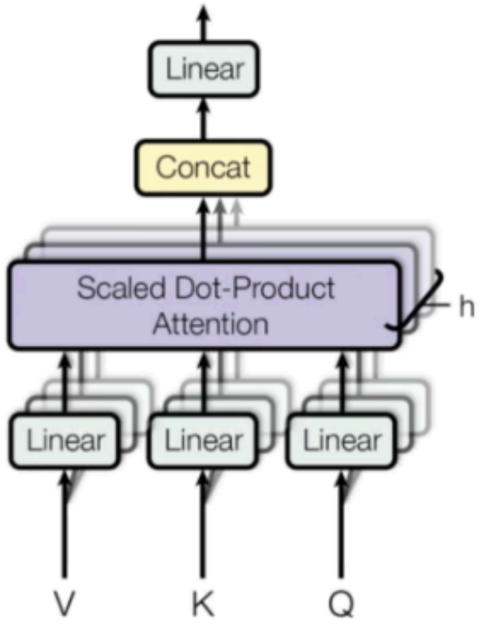
has same shape as Q, but is made with elements from V with respect to their correlations with Q.




walk by river bank
Input embeddings
Queries
Keys
Scalar product
Scaling/
Softmax
Linear combination
Contextualized embeddings
Values

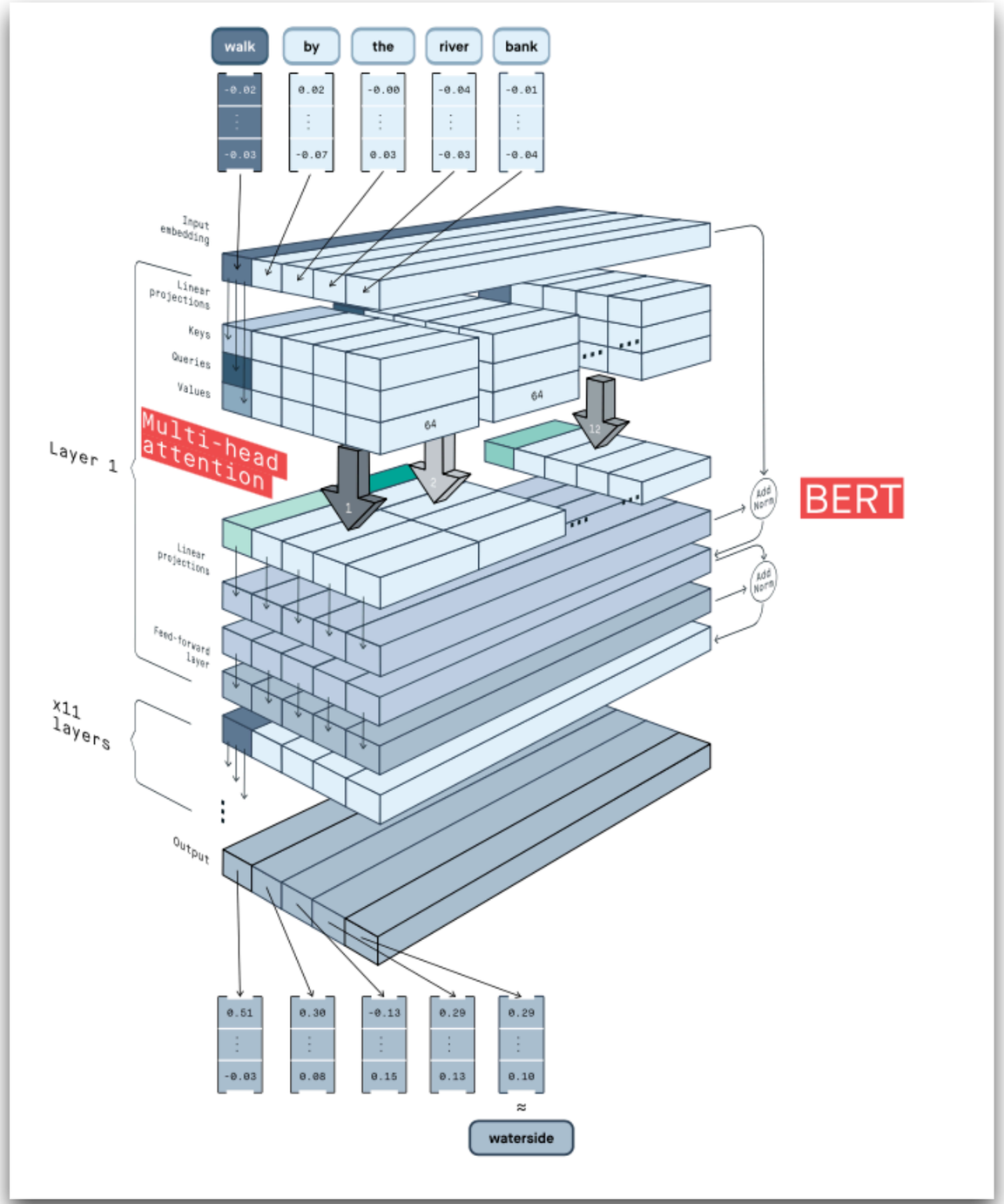

TOKENS
Input embeddings
Queries
Queries
Queries
Keys
Keys
Keys
Scalar product
Scaling/
Softmax
Linear combination
Contextualized
embeddings
Values
Values
Values

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"


walk
by
the
river
bank
-0.02
-0.03
0.02
-0.07
-0.00
0.03
-0.04
-0.03
-0.01
-0.04
Input embedding
Linear projections
Keys
Queries
Values
64
64
12
Multi-head attention
Layer 1
1
2
Add Norm
Add Norm
BERT
Linear projections
Feed-forward layer
x11 layers
Output
0.51
-0.03
0.30
0.08
-0.13
0.15
0.20
0.13
0.29
0.10
≈
waterside

Overview

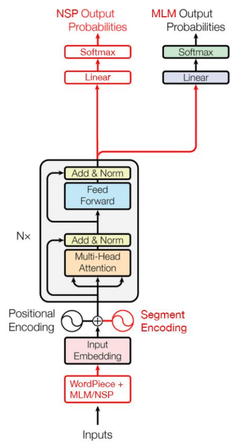


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

Input processing

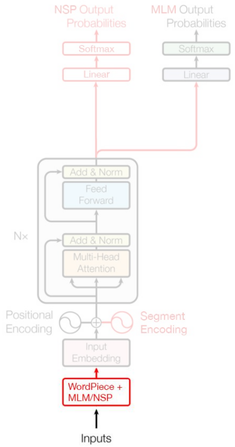


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

Input embedding

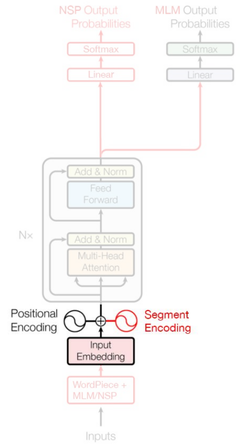


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

Encoder-only model

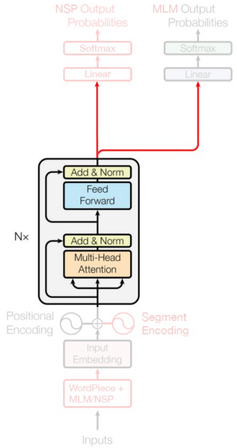


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

udemy | Learn BERT - most powerful NLP algorithm by Google

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

Proxy tasks

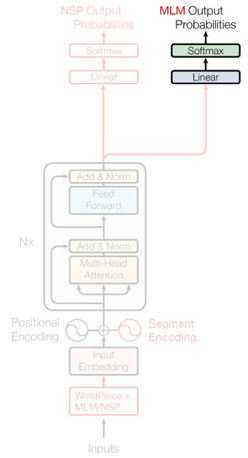


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

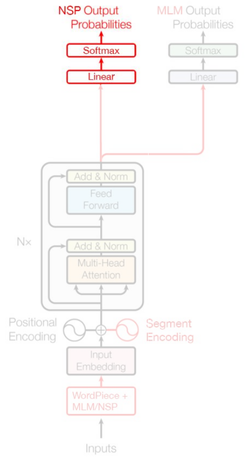


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

Outputs

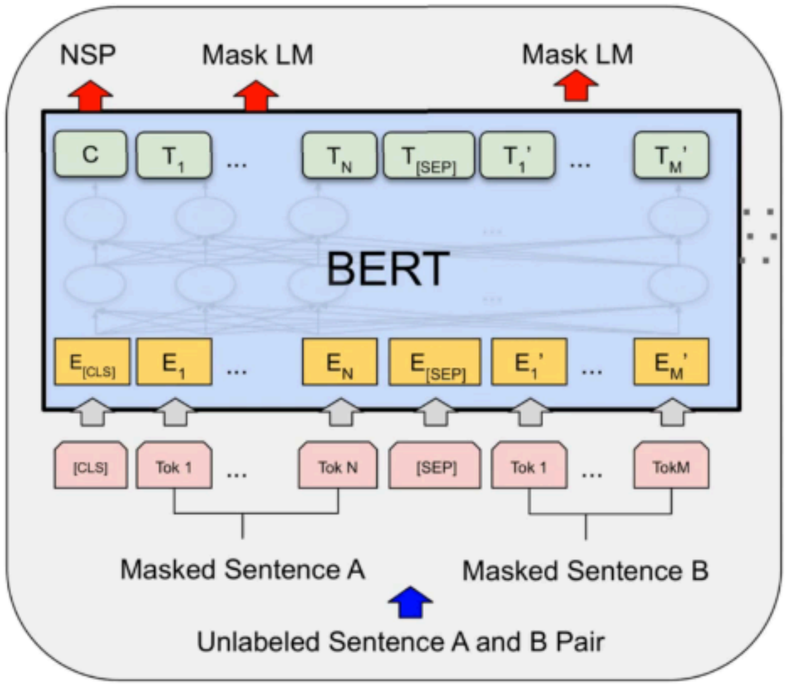
We get 2 types of outputs:

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

C: used for classification. Trained during Next Sentence Prediction

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

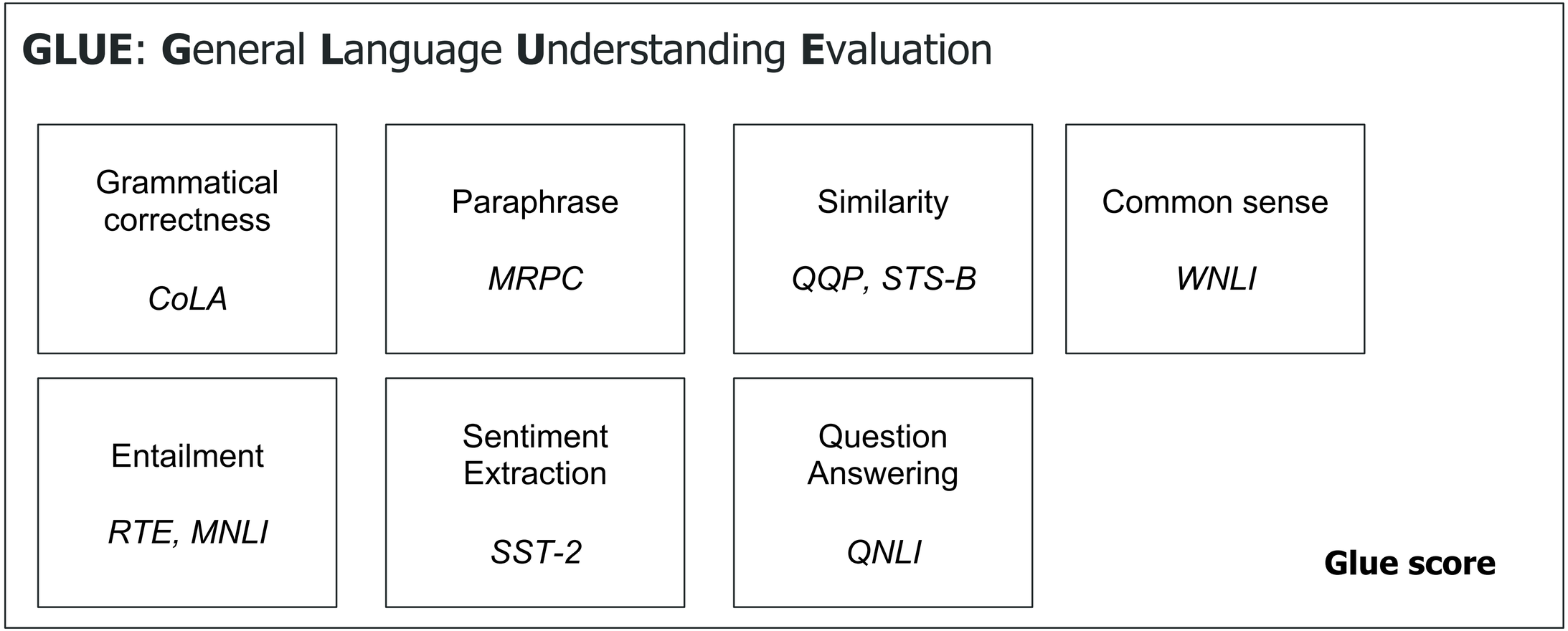
Each token is a vector of the same size (hidden size H, 768 for $BERT_{base}$).



Main transformer-based models

Architecture	Models
Encoder	BERT, DistilBERT, RoBERTa
Decoder	GPT-2, 3
Encoder - Decoder	T5, mT5, ByT5

Standardized benchmark for NLP



More information about the GLUE tasks can be found here: https://gluebenchmark.com/tasks

General parameters

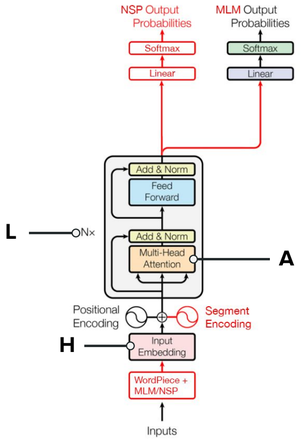


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

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

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

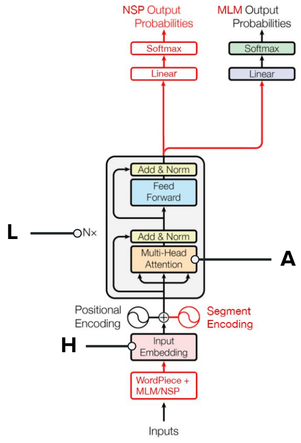
Attention heads (A). Number of attention heads operating in parallel.

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



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

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

Latest trends

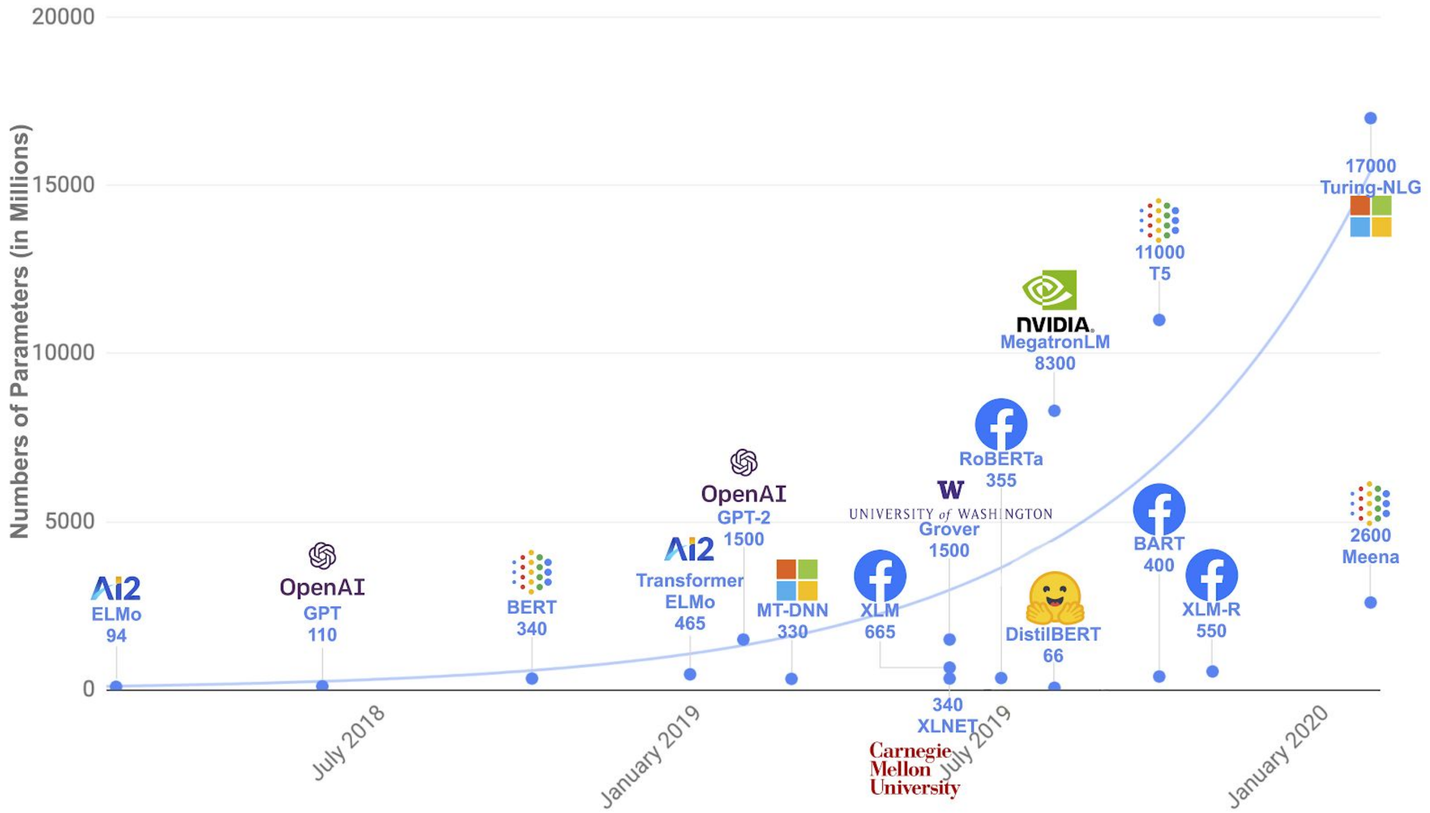


Figure adapted from this Tensorflow blog post, initially designed by the HuggingFace team.

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