HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN
UNIVERSITY OF APPLIED SCIENCES

Deep NLP 3: Transformers and Attention

Oliver Guhr

Identify offensive language using Transformers

Quiz Time!

How is the pace of this course?

A) too slow
B) just right
C) too fast

What is unsupervised learning?

A) compressing sparse into dense vectors
B) learning based on example input-output pairs
C) a different name for transfer learning
D) an algorithm that learns patterns from untagged data

What does the Distributional Hypothesis say?

A) Words can be encoded in a vector space
B) Words are described by their context words
C) Words can be drawn on maps
D) Similarity between words can be calculated using the euclidean distance

Why do we need dense vector representations for texts?

A) to efficiently compute neural networks
B) to encode the relationships between words
C) to create word clouds
D) to pretrain neural networks

Transfer learning for NLP works by:

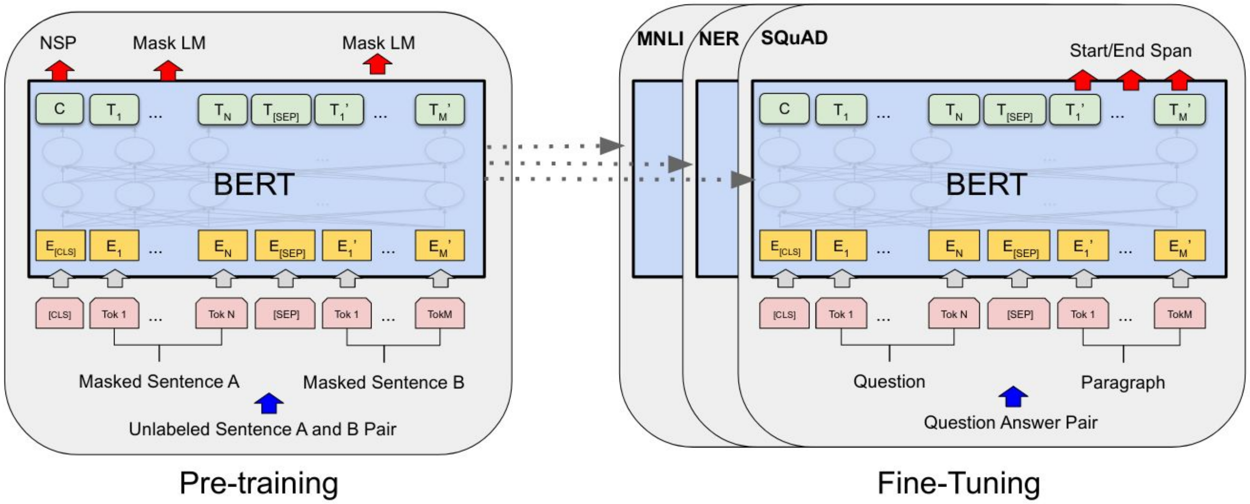
A) training a model with an unsupervised task and retraining it with labeled data

B) pretraining a model with a labeled data and retraining it with an unsupervised task

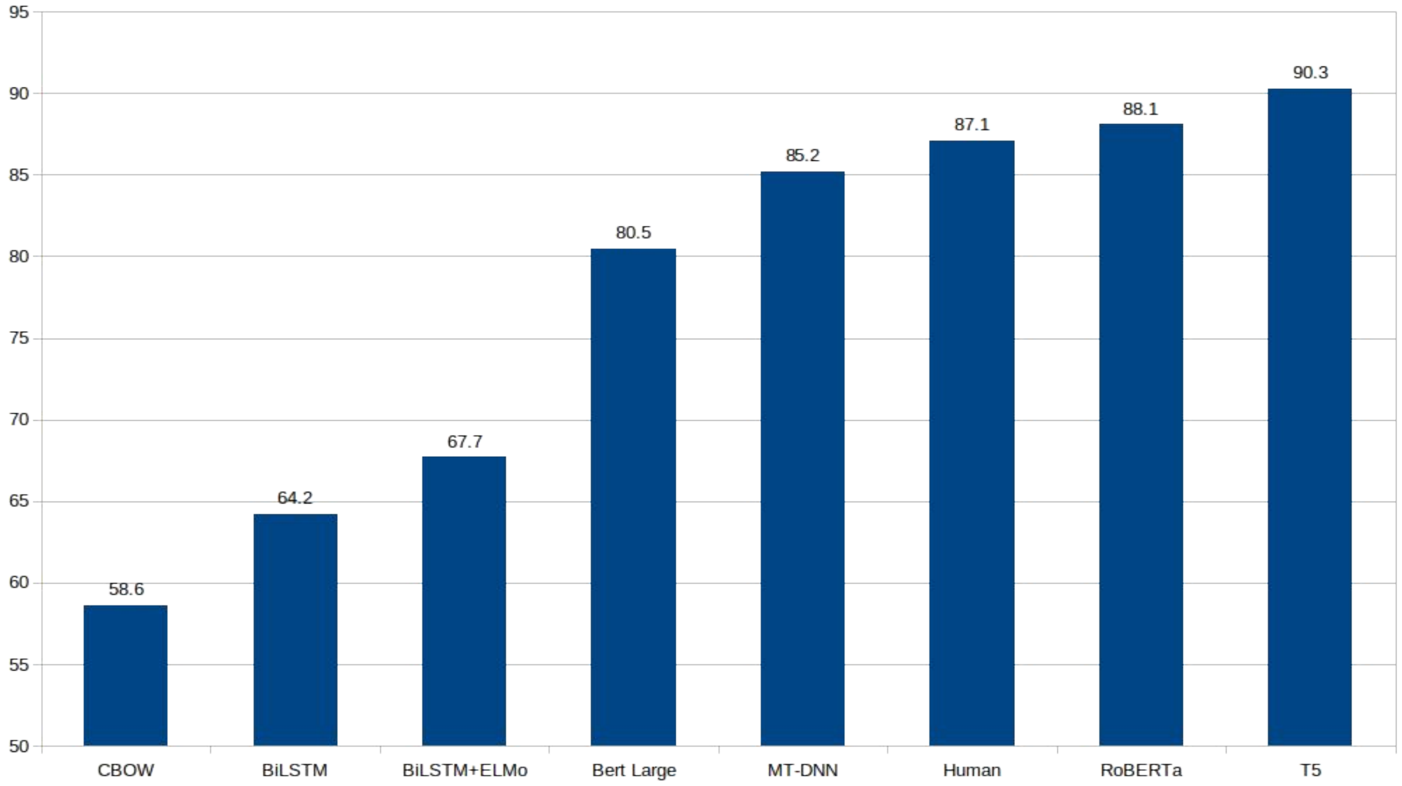
Recap

Bert

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- Paper by Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
- Published in 2018
- improved the state-of-the-art in most important benchmarks



GLUE Benchmark



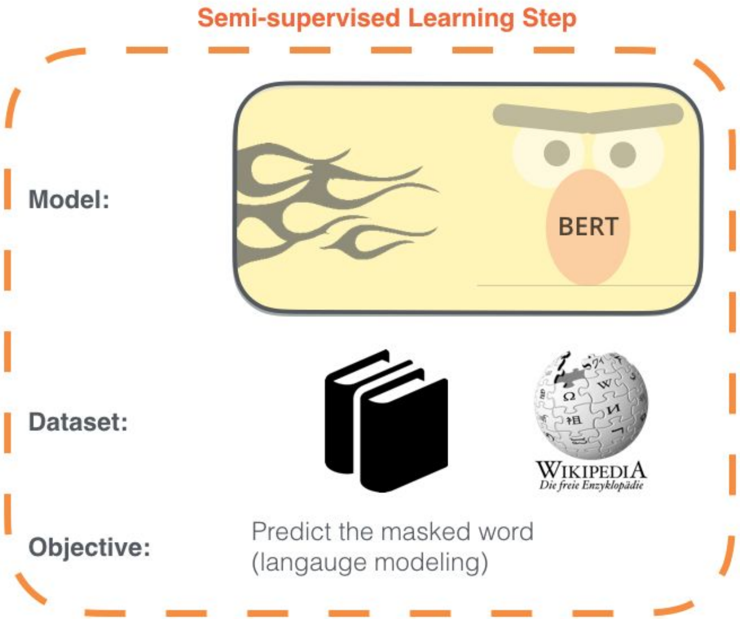
GLUE Leaderboard: https://gluebenchmark.com/leaderboard

How deep are these models?

Bert

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

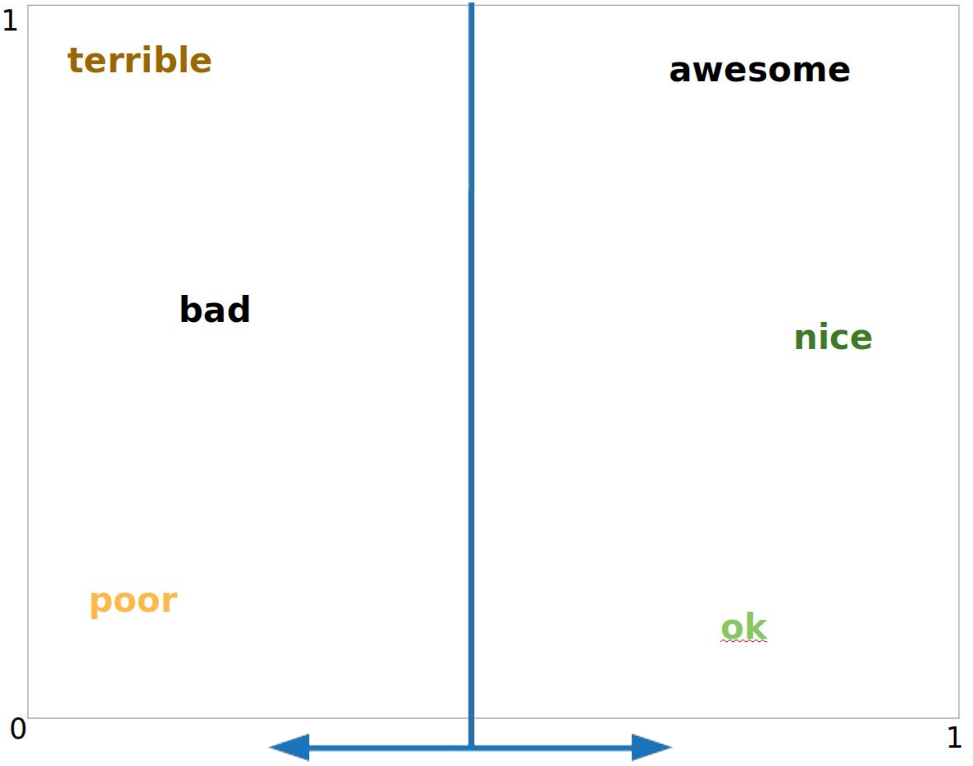


Distributional Hypothesis

Words that occur in the same contexts tend to have similar meanings.
Harris (1954)

A word is characterized by the company it keeps.
Firth (1957)

Word Vectors - Klassifikation



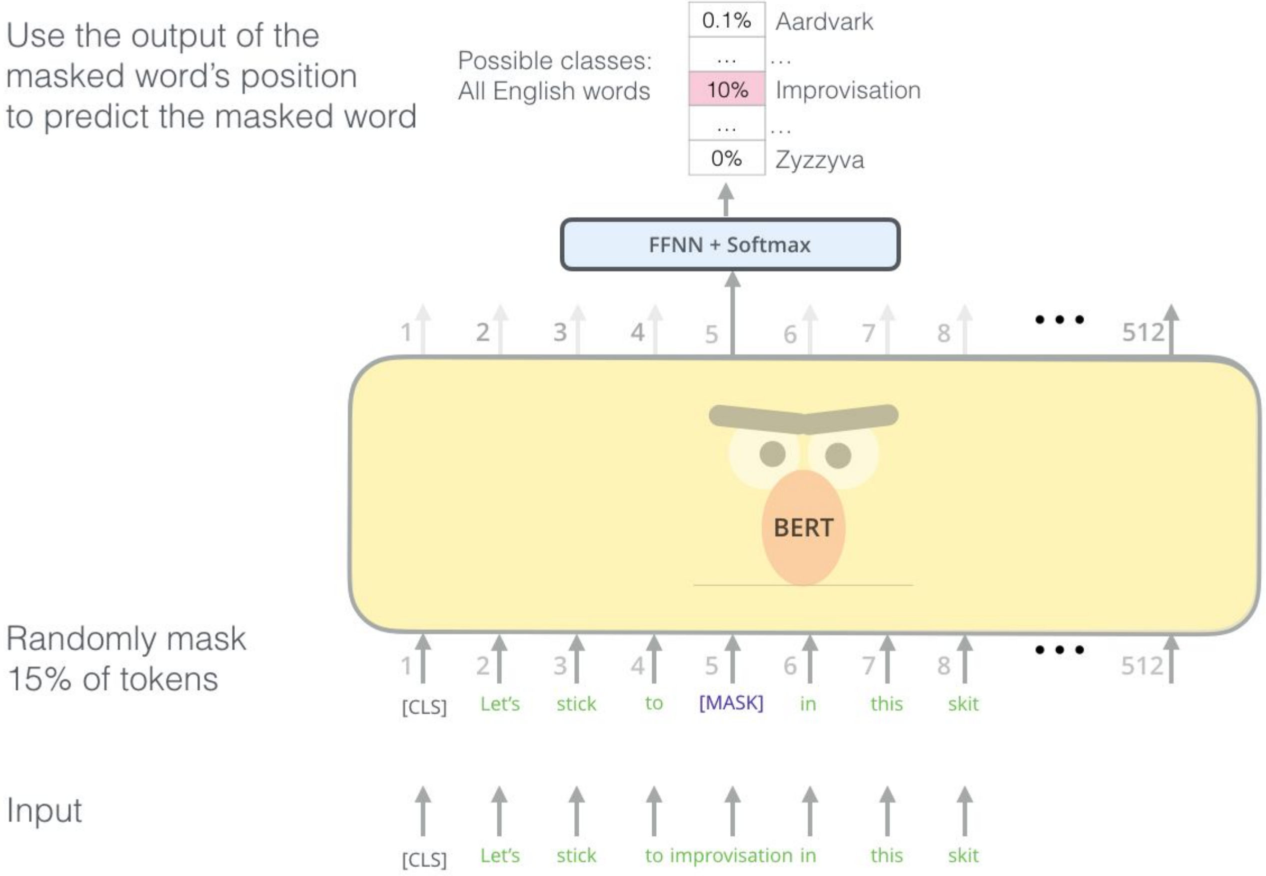
$v_{ok} = [0.75, 0.15]$

$v_{nice} = [0.85, 0.50]$

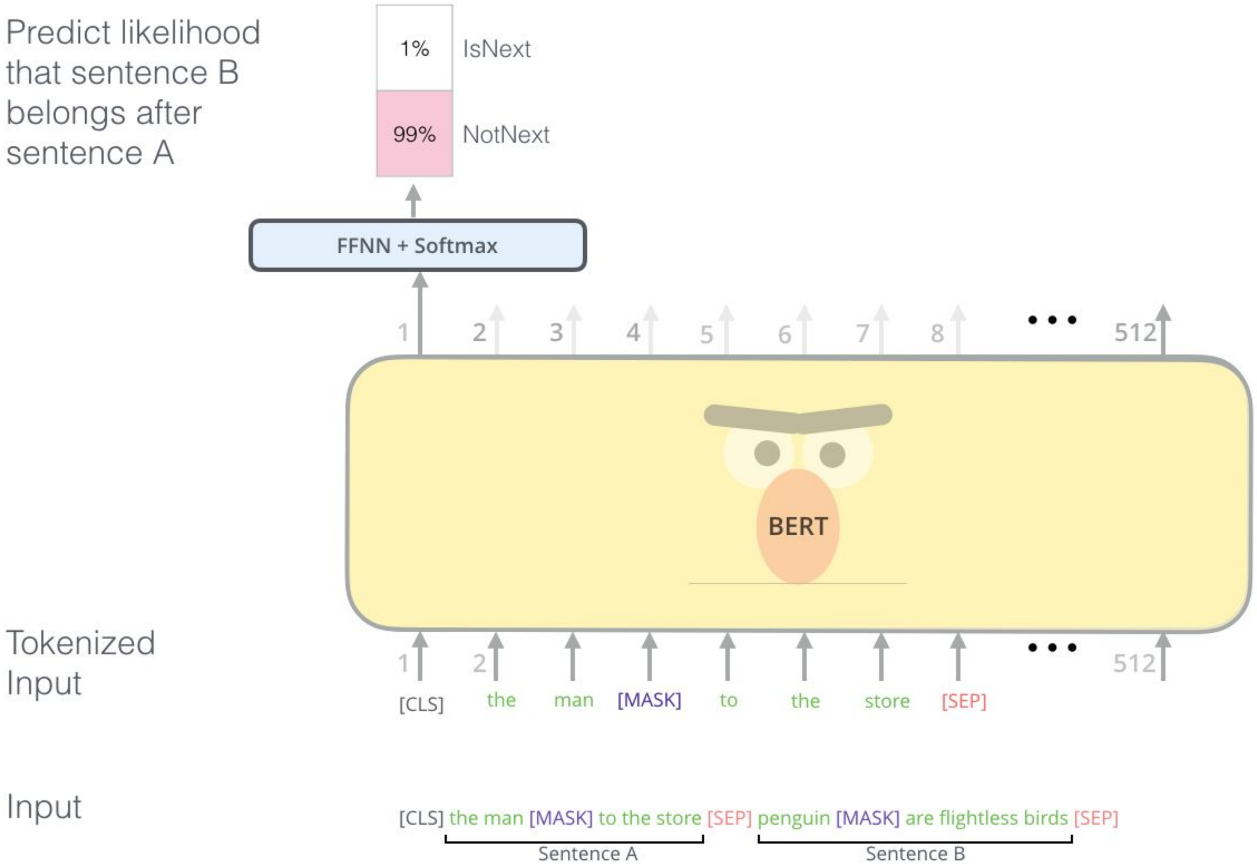
$v_{poor} = [0.15, 0.18]$

$v_{terrible} = [0.10, 0.91]$

Task One: Mask Words



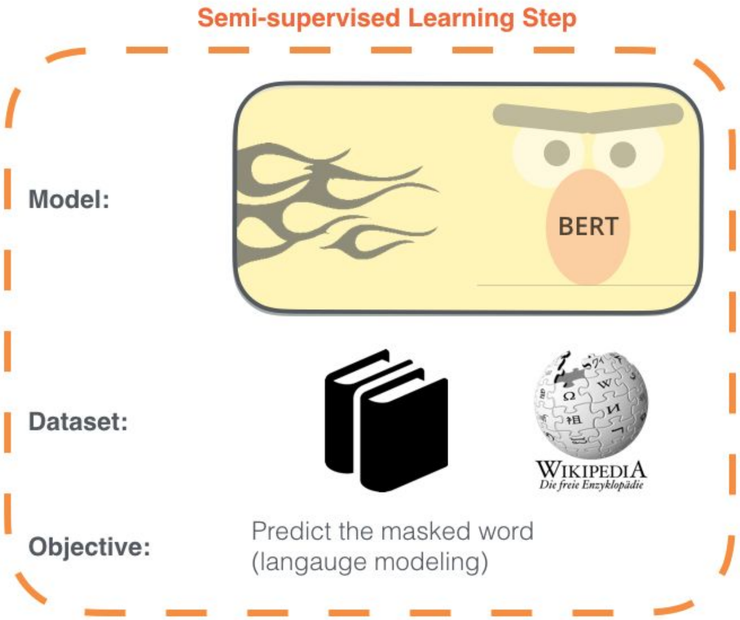
Task Two: Next Sentence Prediction



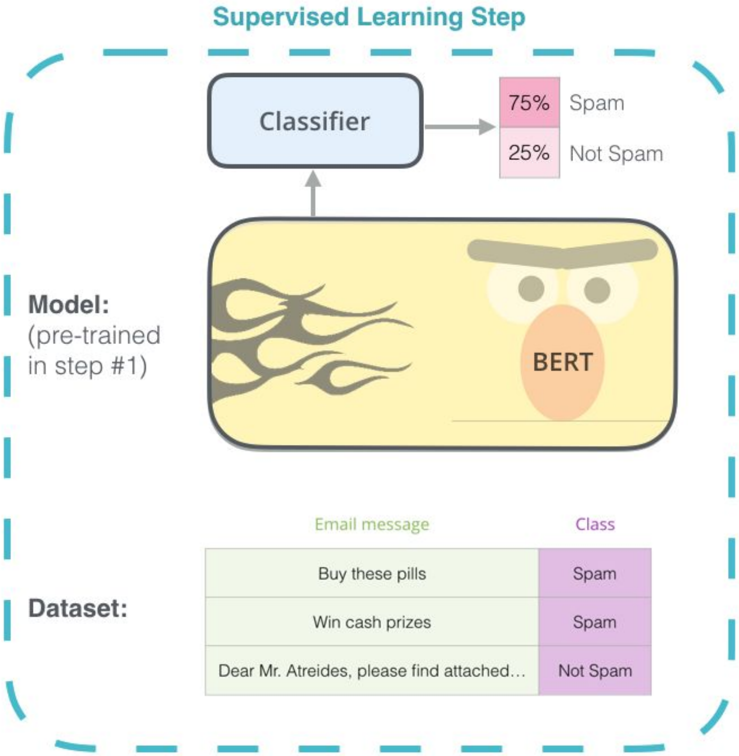
Bert

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

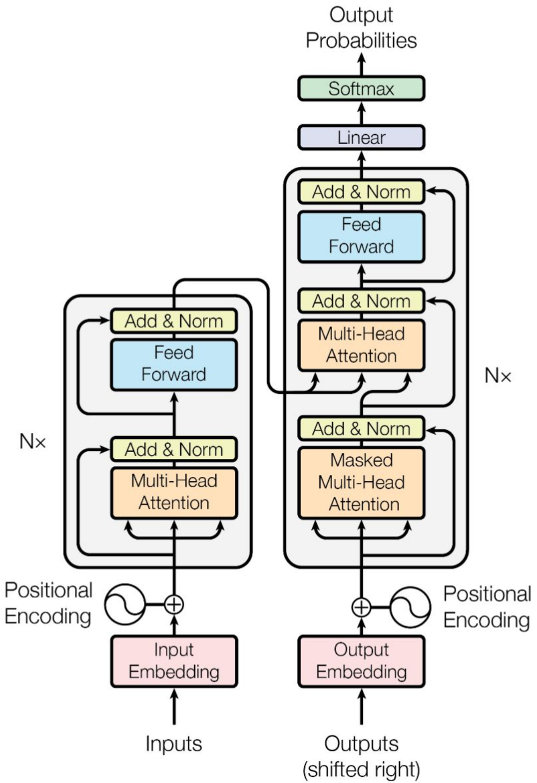


2 - Supervised training on a specific task with a labeled dataset.



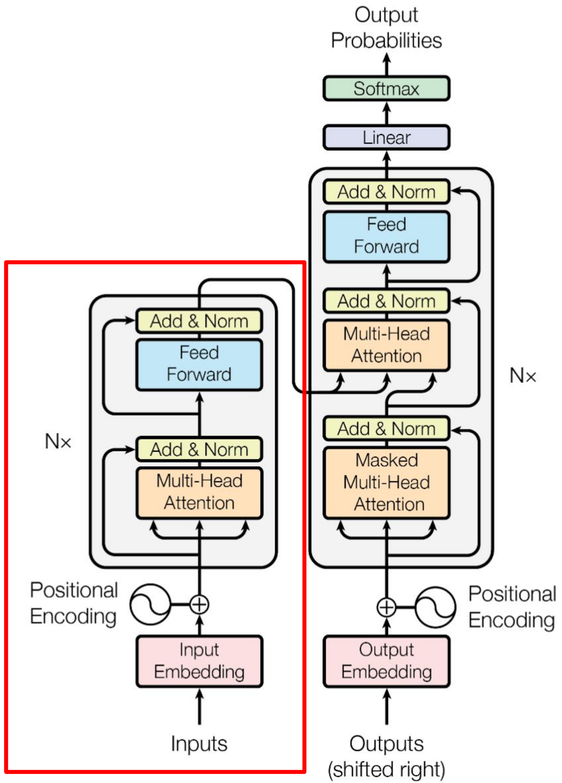
How do Transformers work?

Attention is all you need



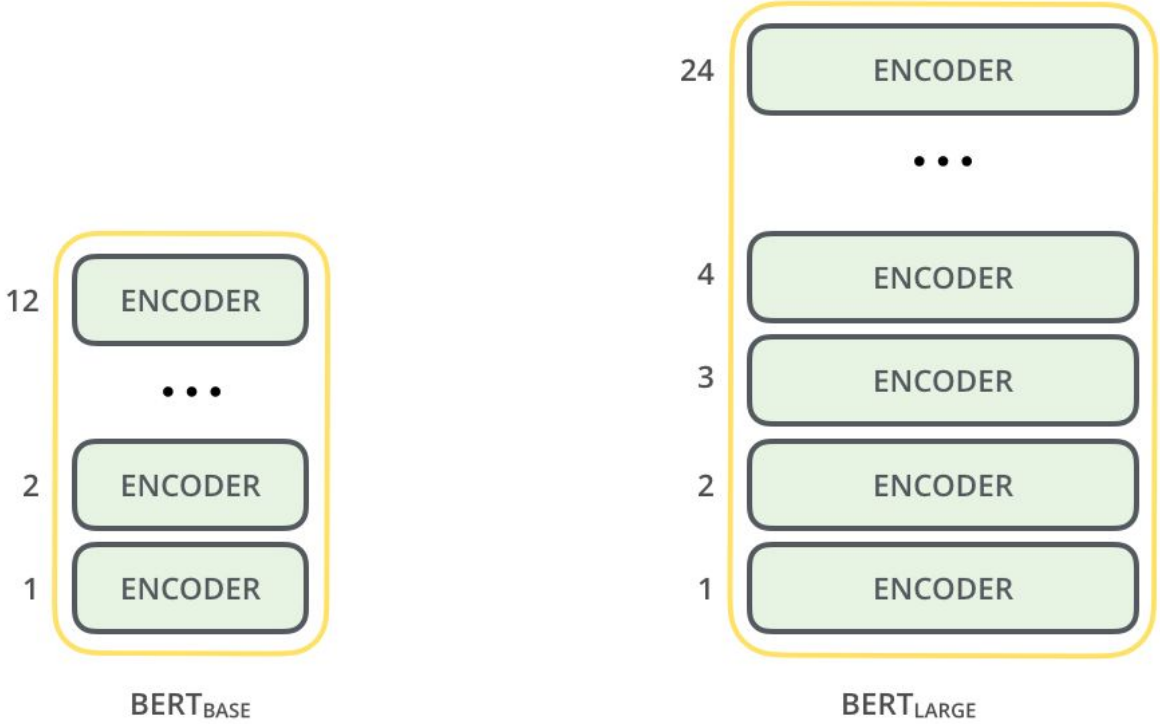
Attention Is All You Need, Vaswani et al. https://arxiv.org/abs/1706.03762

Attention is all you need

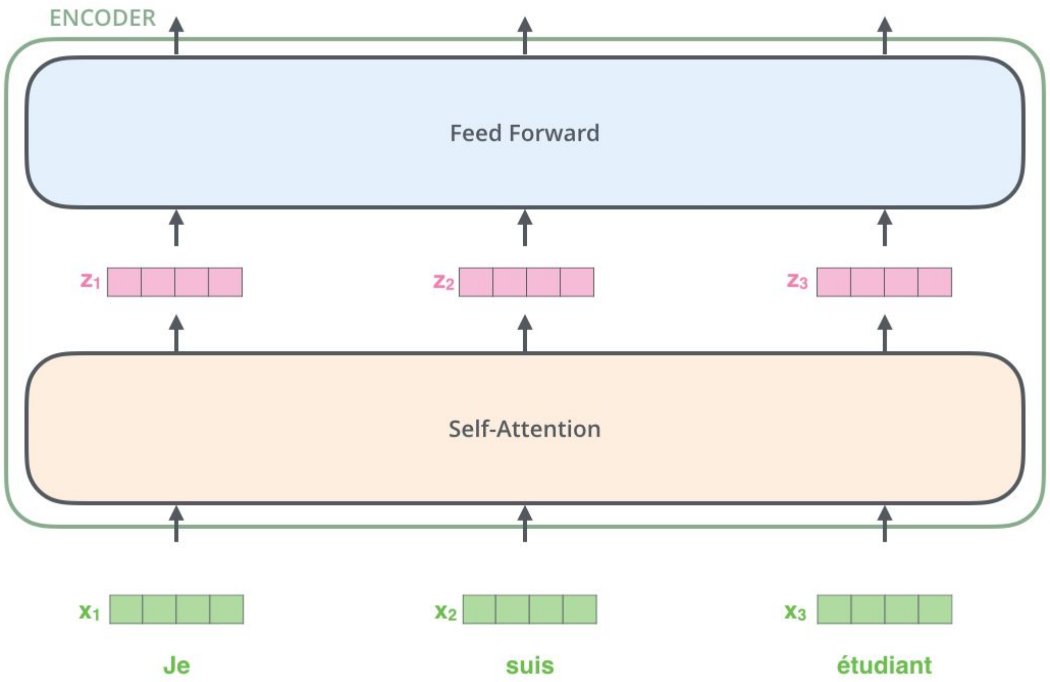


Attention Is All You Need, Vaswani et al. https://arxiv.org/abs/1706.03762

How encoders work.

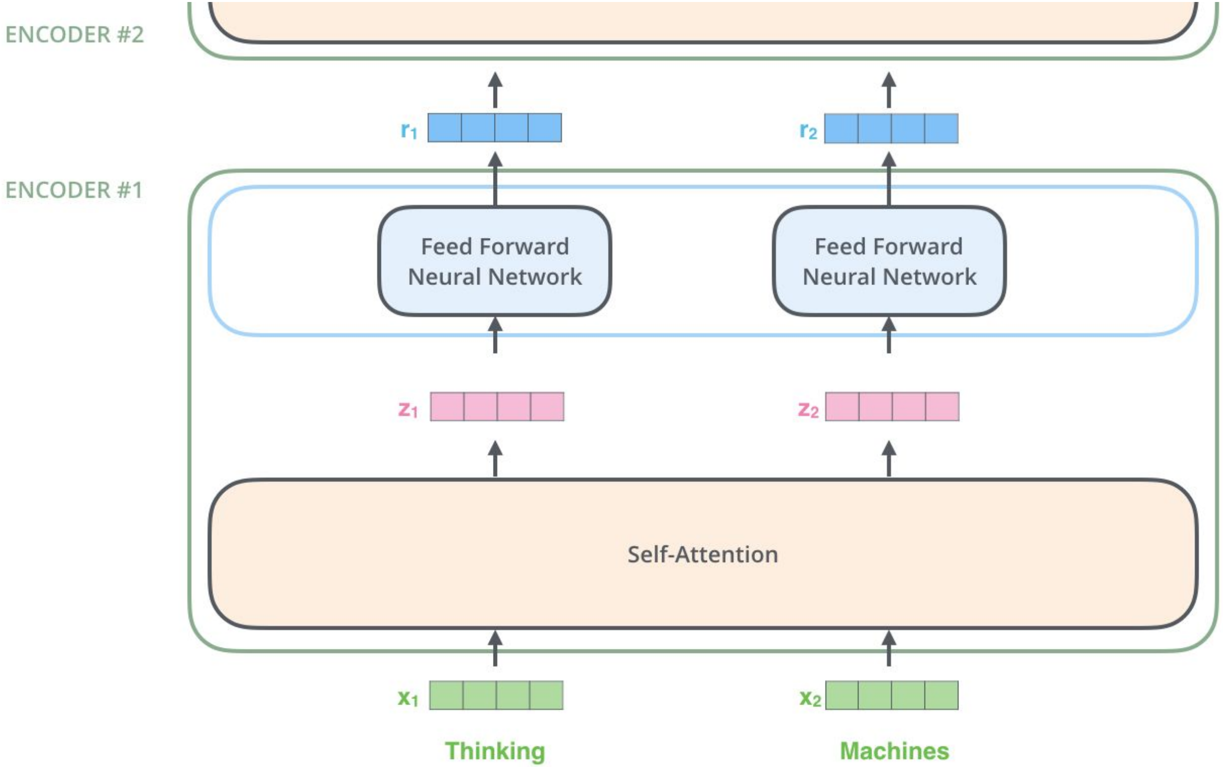


Transformer Encoder



The Illustrated BERT, Jay Alammar: http://http://jalammar.github.io/illustrated-transformer/

Transformer Encoder



The Illustrated BERT, Jay Alammar: http://http://jalammar.github.io/illustrated-transformer/

What is self attention?

Scaled dot product attention

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

Query Key Value

Scaled dot product attention

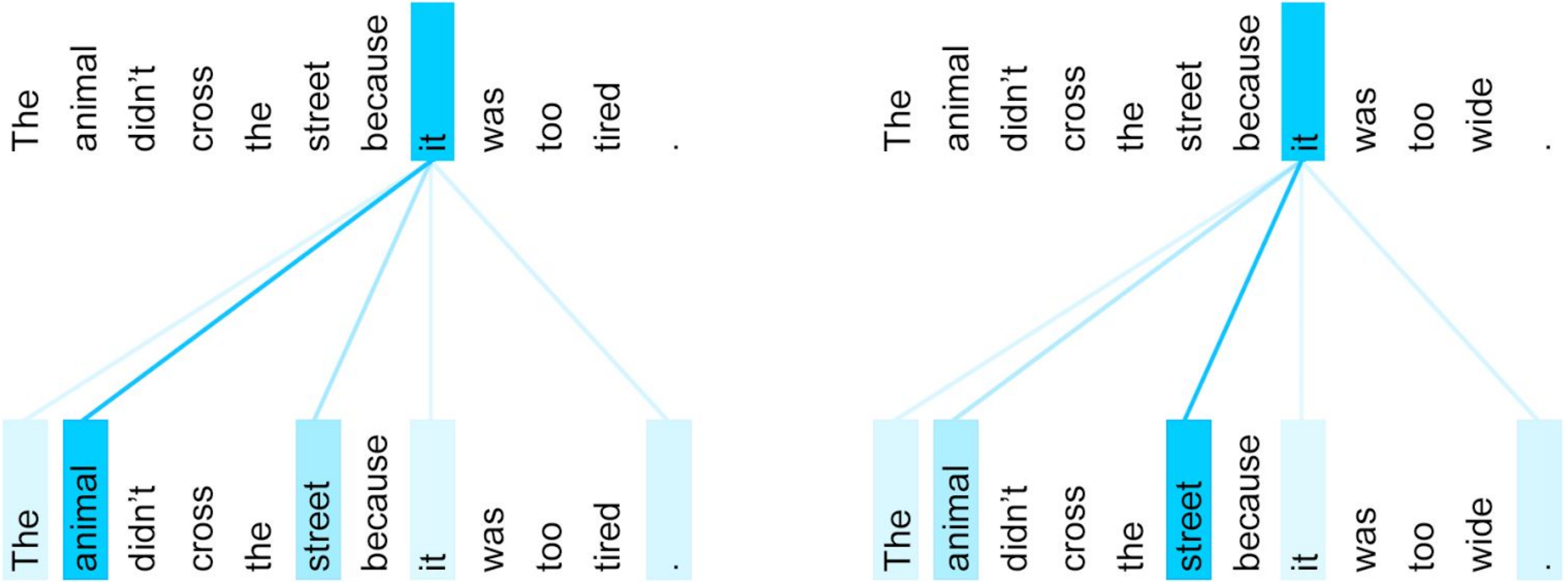
$$\text{Attention}(\underline{Q}, \underline{K}, \underline{V}) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Query **Key** **Value**

Take the current word or token, find the most similar key and return the corresponding value.

What does Attention do?

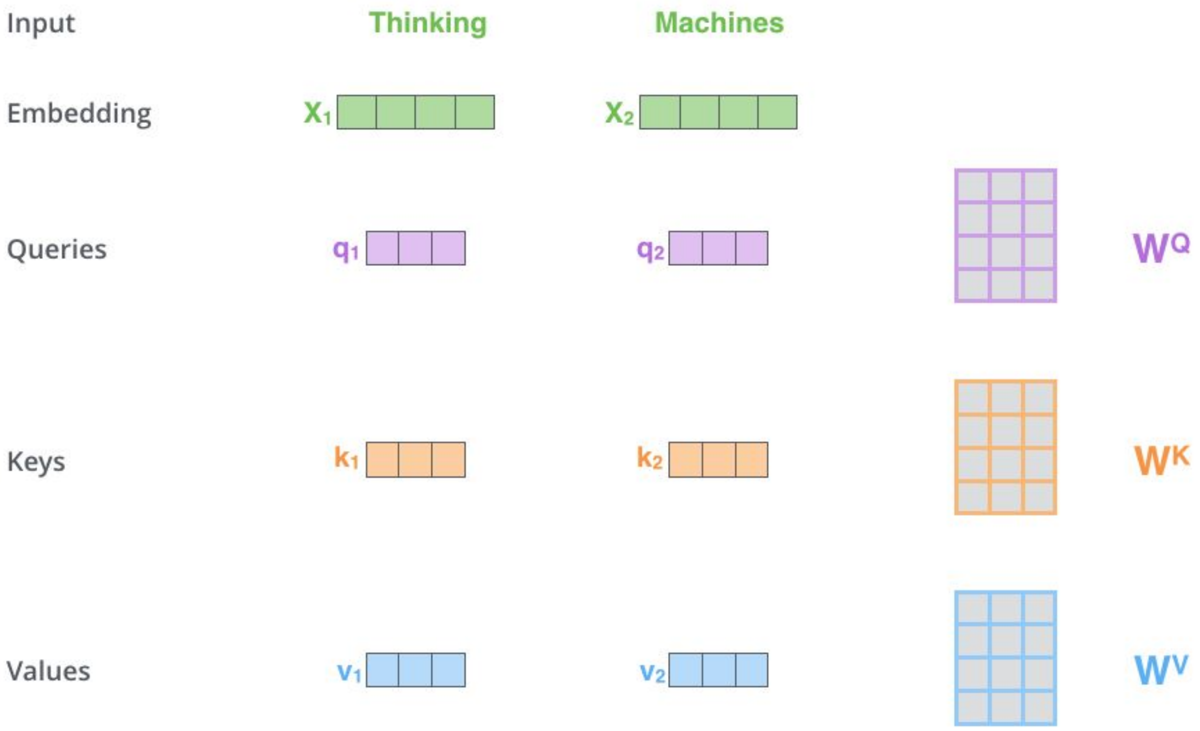
What does Attention do?



The encoder self-attention distribution for the word "it" from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

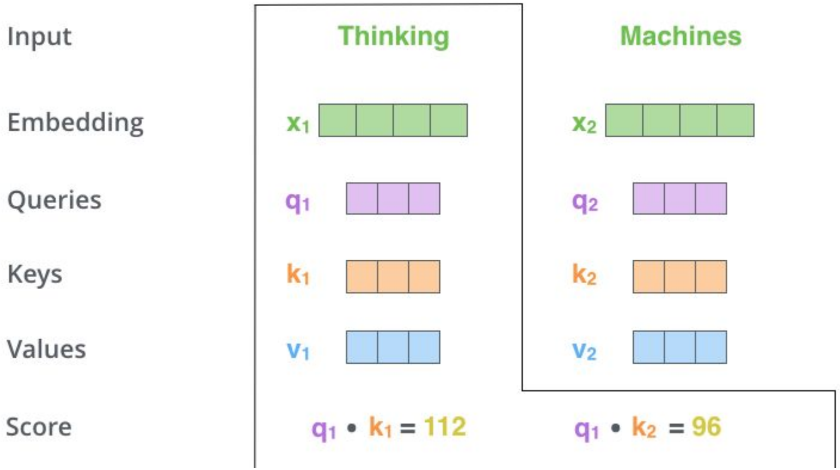
Source: https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Attention

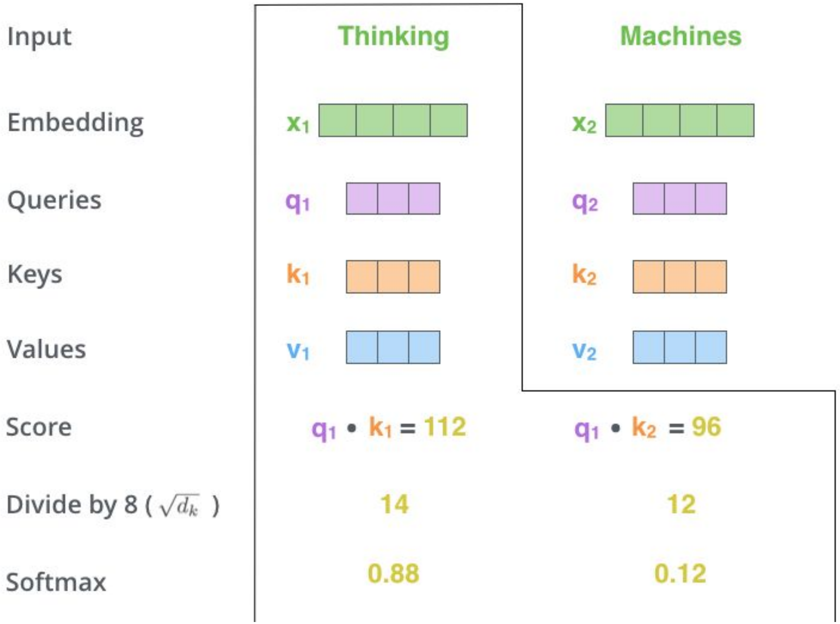


Input
Thinking
Machines
Embedding
x_1
x_2
Queries
q_1
q_2
W^Q
Keys
k_1
k_2
W^K
Values
v_1
v_2
W^V

Attention

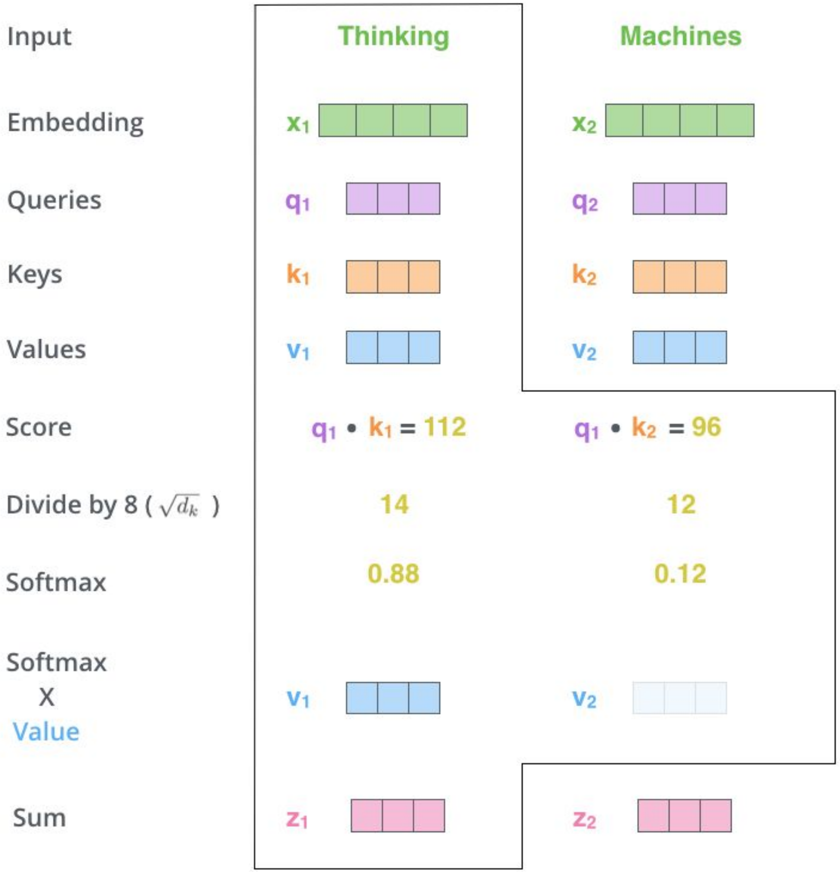

Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96

Attention



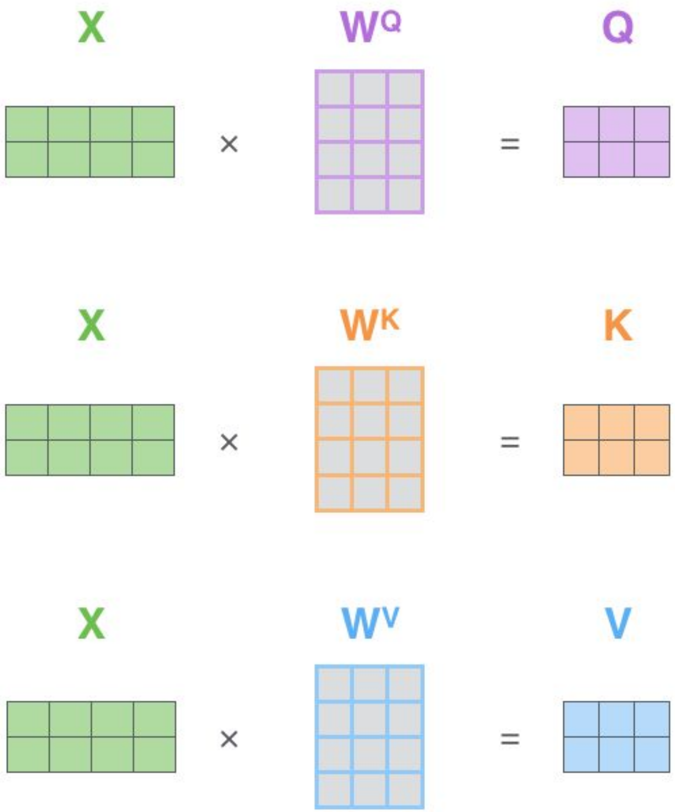
Input
Thinking
Machines
Embedding
x_1
x_2
Queries
q_1
q_2
Keys
k_1
k_2
Values
v_1
v_2
Score
$q_1 \bullet k_1 = 112$
$q_1 \bullet k_2 = 96$
Divide by 8 ($\sqrt{d_k}$)
14
12
Softmax
0.88
0.12

Attention

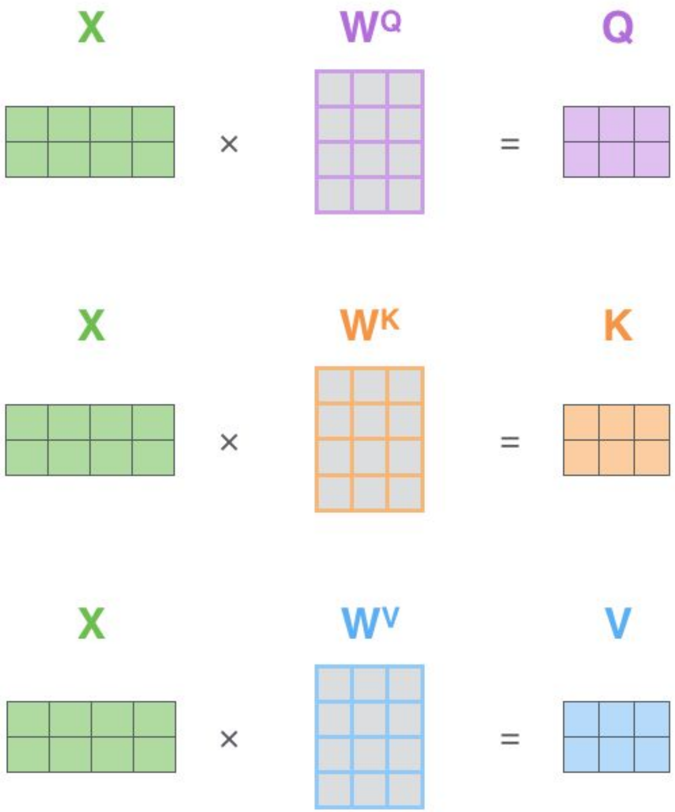


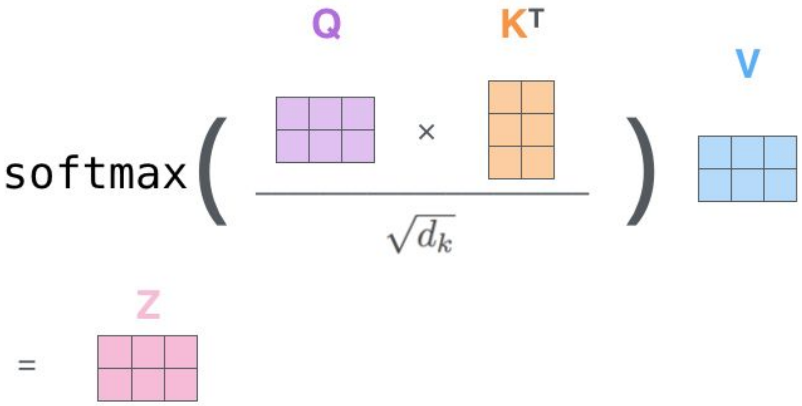
Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96
Divide by 8 ($\sqrt{d_k}$)
14
12
Softmax
0.88
0.12
Softmax
X
Value
v1
v2
Sum
z1
z2

Matrix Calculation

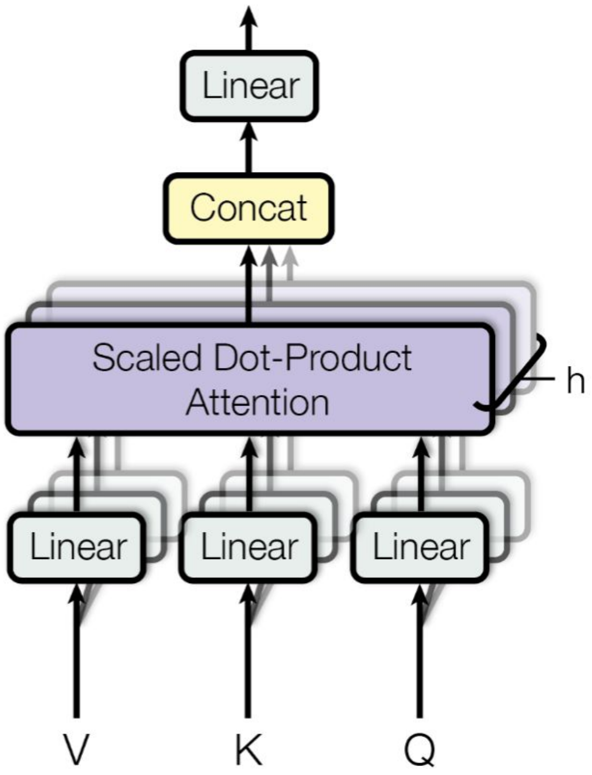


Matrix Calculation



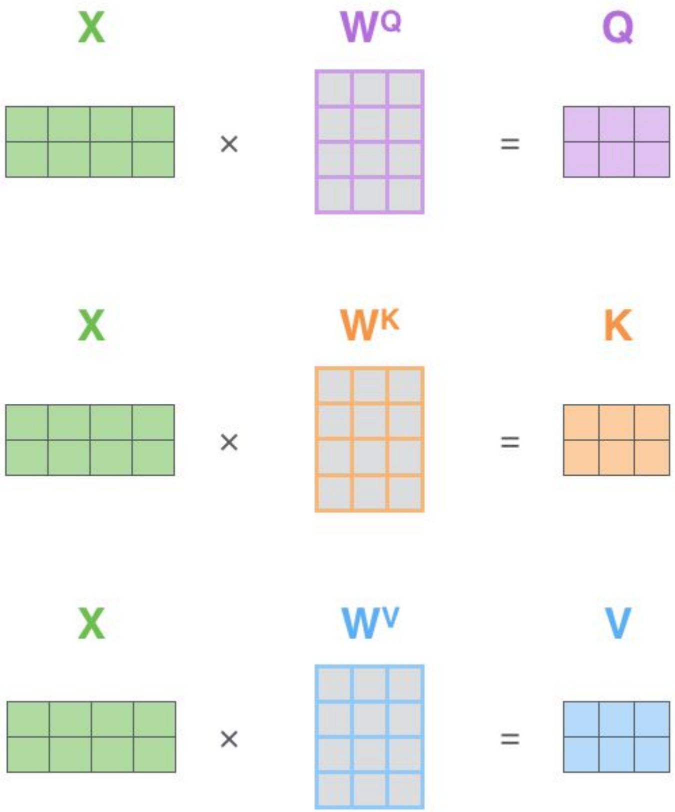


Multi Head Attention

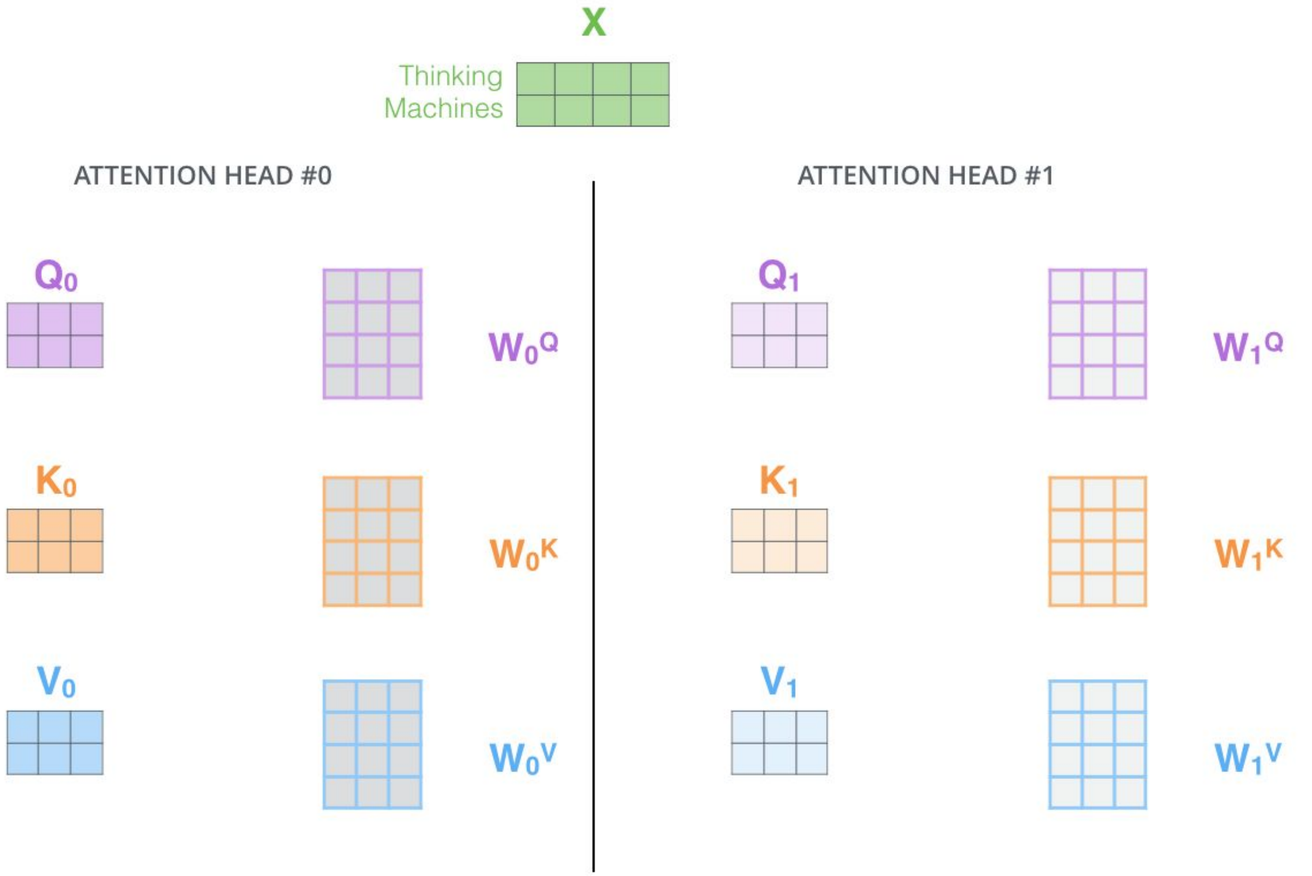


Attention Is All You Need, Vaswani et al. https://arxiv.org/abs/1706.03762

Multi Head Attention

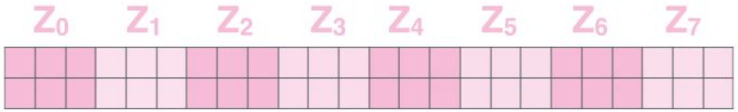


Multi Head Attention



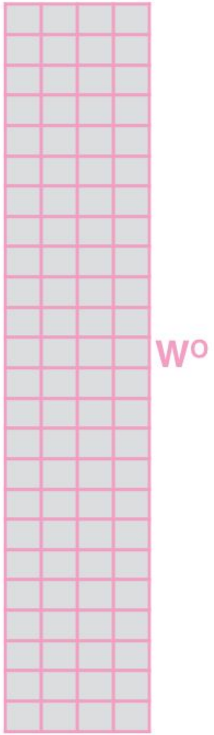
Multi Head Attention

1) Concatenate all the attention heads



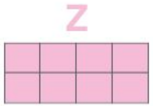
2) Multiply with a weight matrix W^O that was trained jointly with the model

X

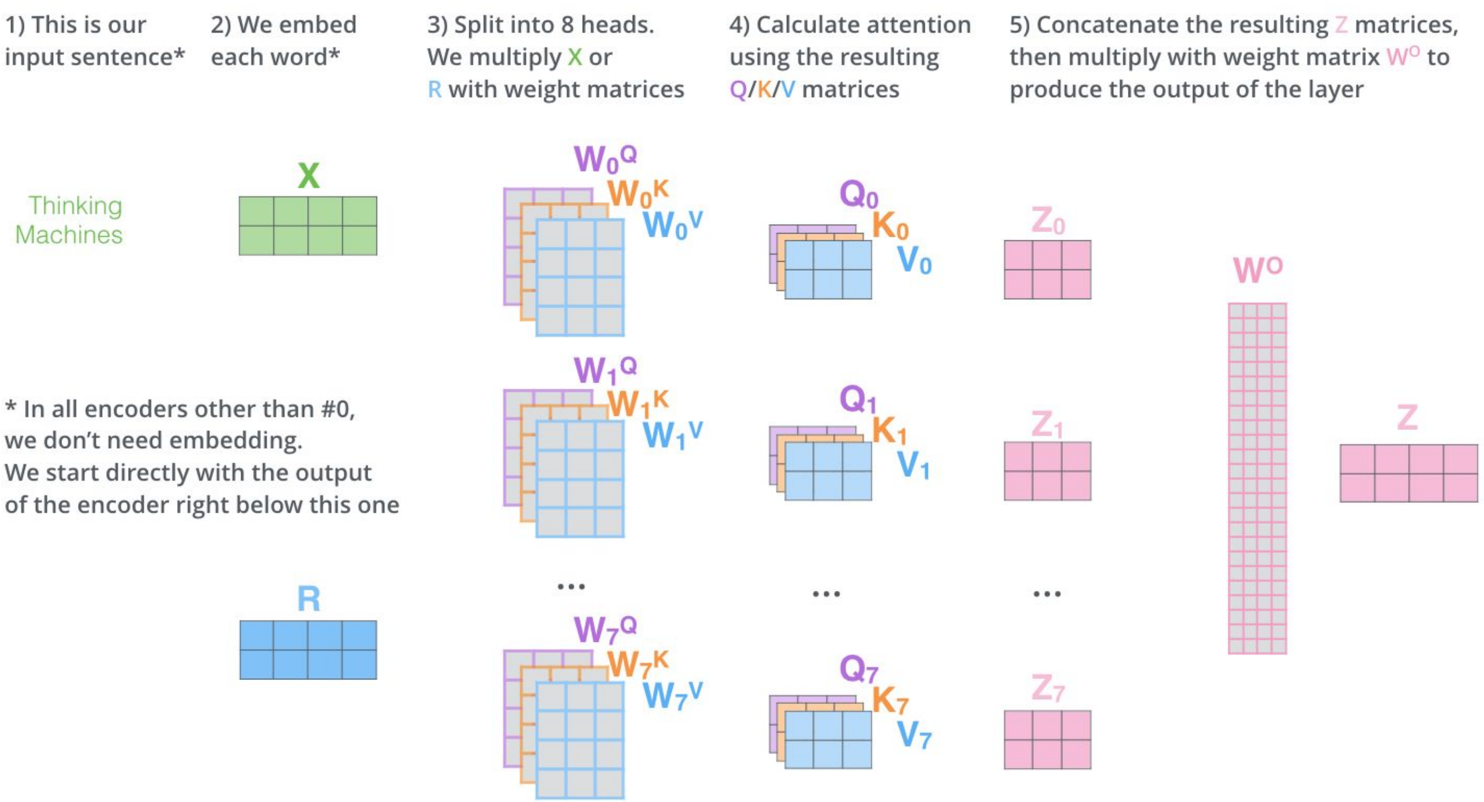


3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

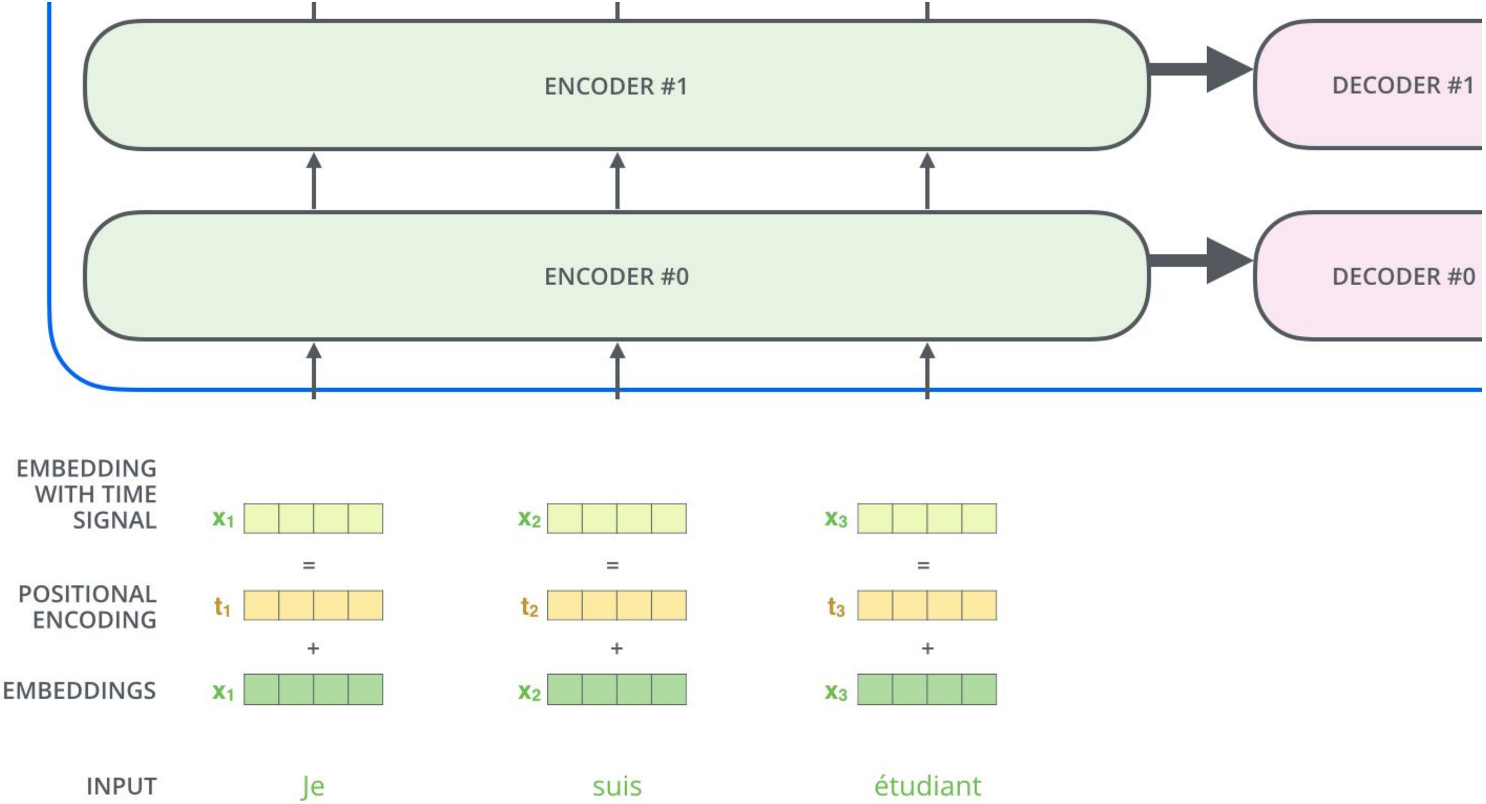
=



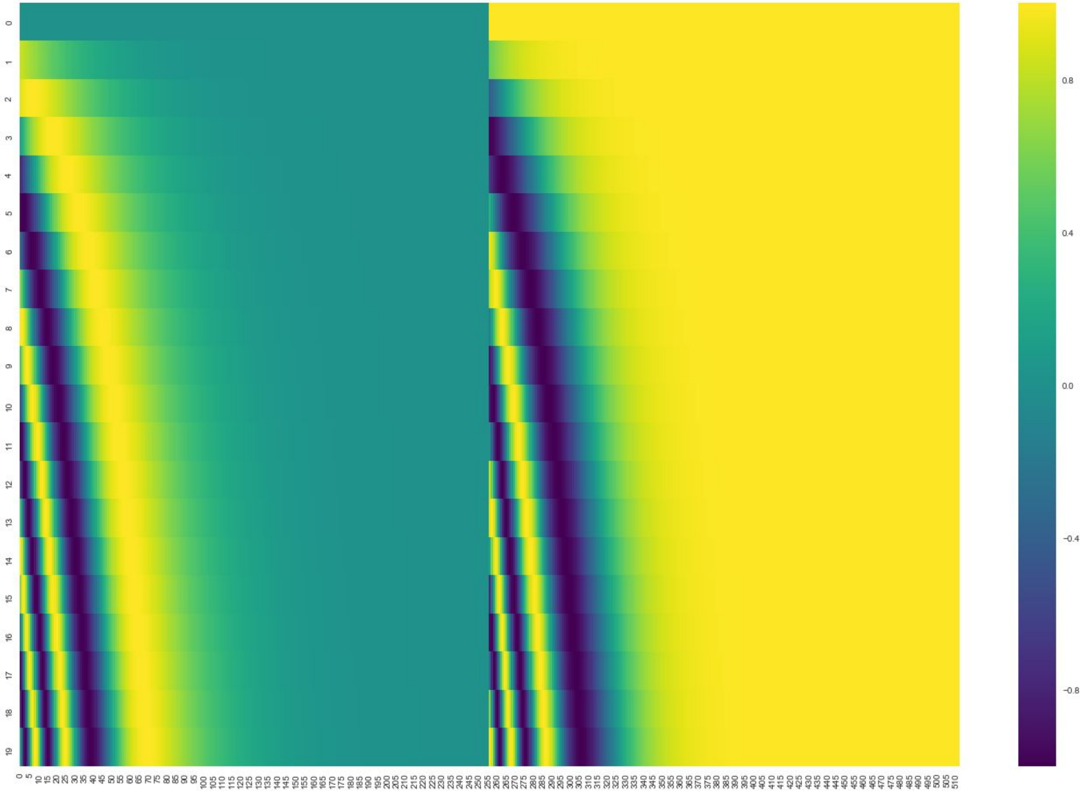
Multi Head Attention



Positional Encoding



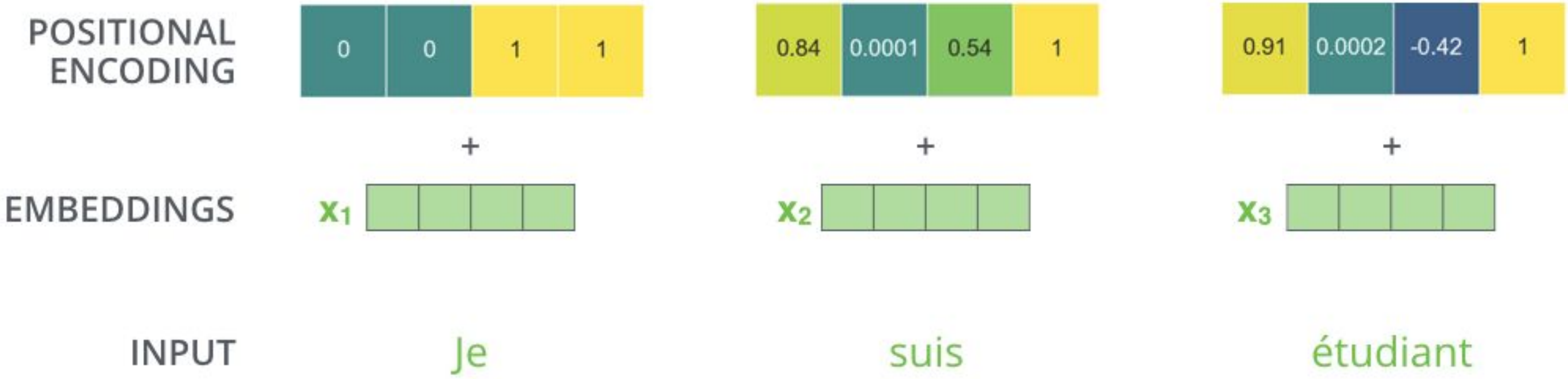
Positional Encoding



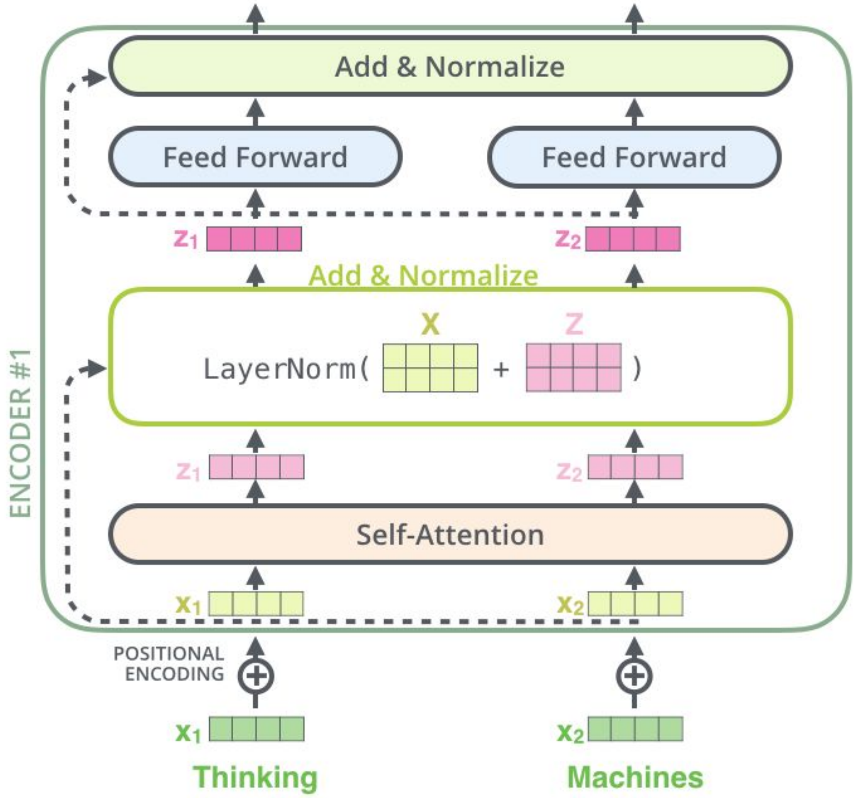
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Positional Encoding

For embedding with a dimensionality of 4 the encodings look like this:



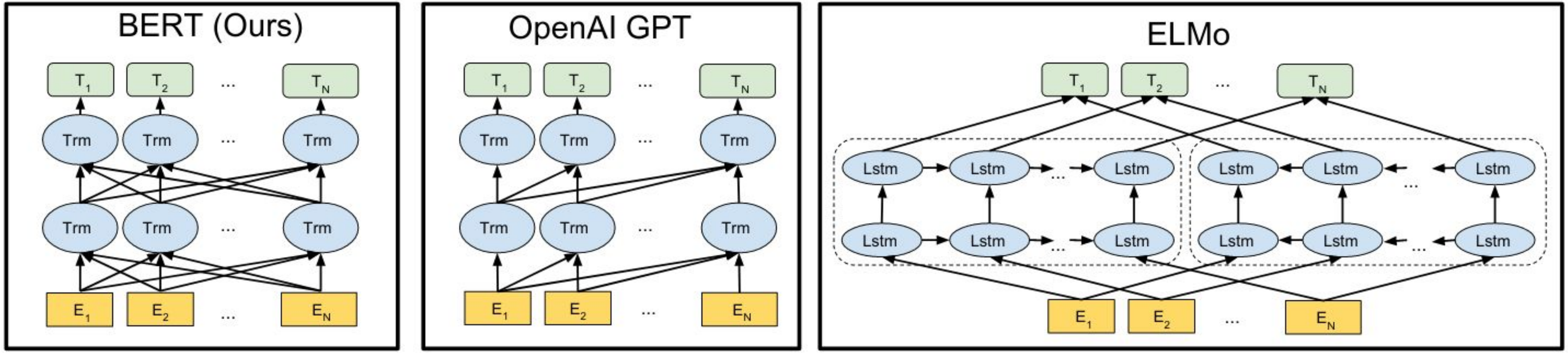
Add and Normalize



Layer Normalization Lei Ba et al. https://arxiv.org/abs/1607.06450

Transformers vs LSTMs

- Can we build something similar using LSTMs?
  - Yes, its called ELMo

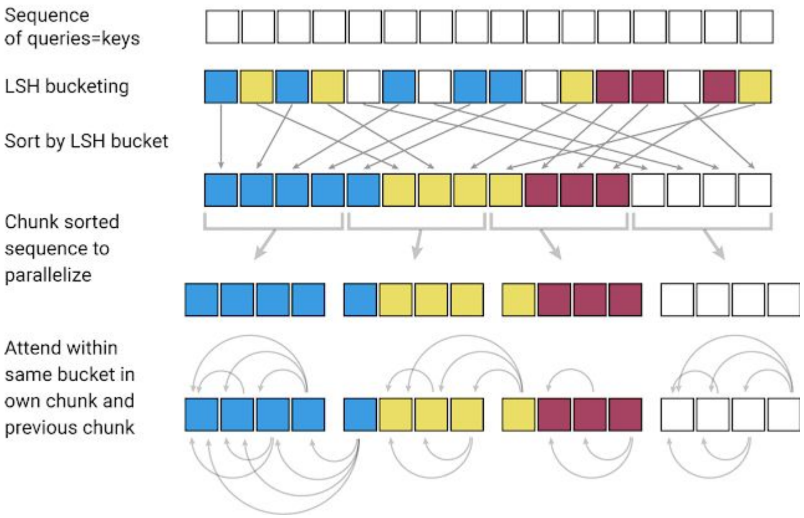


Source Bert Paper: https://arxiv.org/pdf/1810.04805.pdf

Future...

Reformer: The Efficient Transformer

- Improved efficiency of the attention algorithm
- **context windows of 1 million words** on a 16GB GPU (Transformer 512 Token)
- Main Contribution
  - locality-sensitive-hashing (LSH)
  - reversible residual layers
- **Similar ideas:**
  - Longformer, Linformer, [\w*]former



- More Information
  - [Paper by Kitaev, Kaiser and Levskya]
  - [Google AI Blog Post]
  - [Video Introduction]
  - [Background Info]

RealFormer: Transformer Likes Residual Attention

- **Resnets idea** but for Transformers: Residual connections for attention values
- Improves overall results but not by much
- [Paper by Ruining He, Anirudh Ravula, Bhargav Kanagal, Joshua Ainslie]

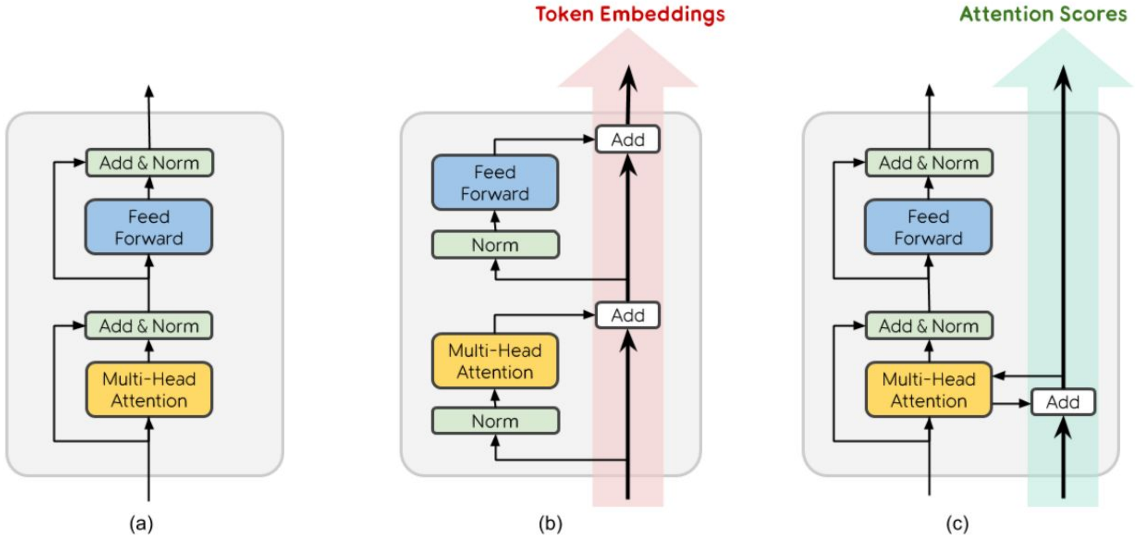
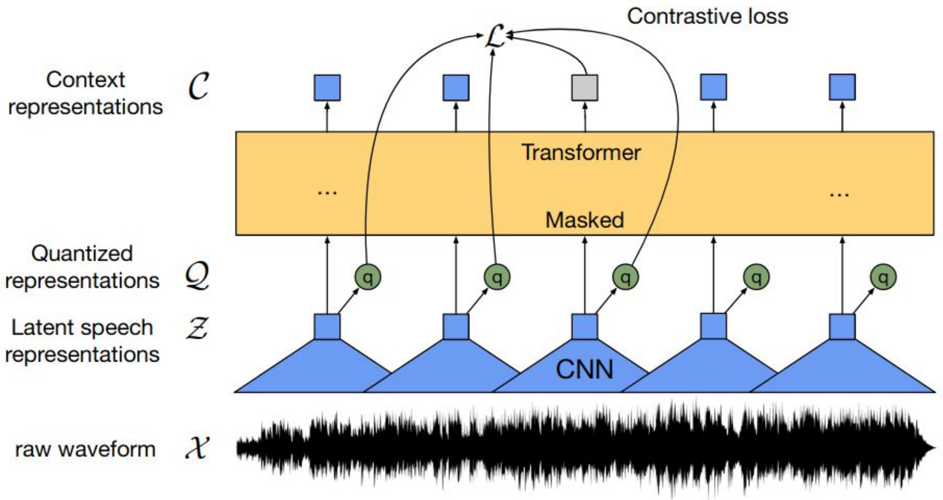


Figure 1: Comparison of different Transformer layers: (a) The prevalent Post-LN layer used by (*e.g.*) BERT; (b) Pre-LN layer used by (*e.g.*) GPT-2 that creates a "direct" path to propagate token embeddings; (c) Our RealFormer layer that creates a "direct" path to propagate attention scores (by adding a simple skip edge on top of (a)).

Automatic Speech Recognition

- wav2vec 2.0: A Framework for Self-SupervisedLearning of Speech Representations
- Key Ideas:
  - **CNN and Transformer based end to end model for speech recognition**
  - uses a novel **pretraining schema to learn for unlabeled audio data**
- outperforms the previous state of the art while using 100 times less labeled data
- can achieve good accuracy with very little data
- By Alexei Baevski, Henry Zhou, Abdelrahman Mohamed and Michael Auli

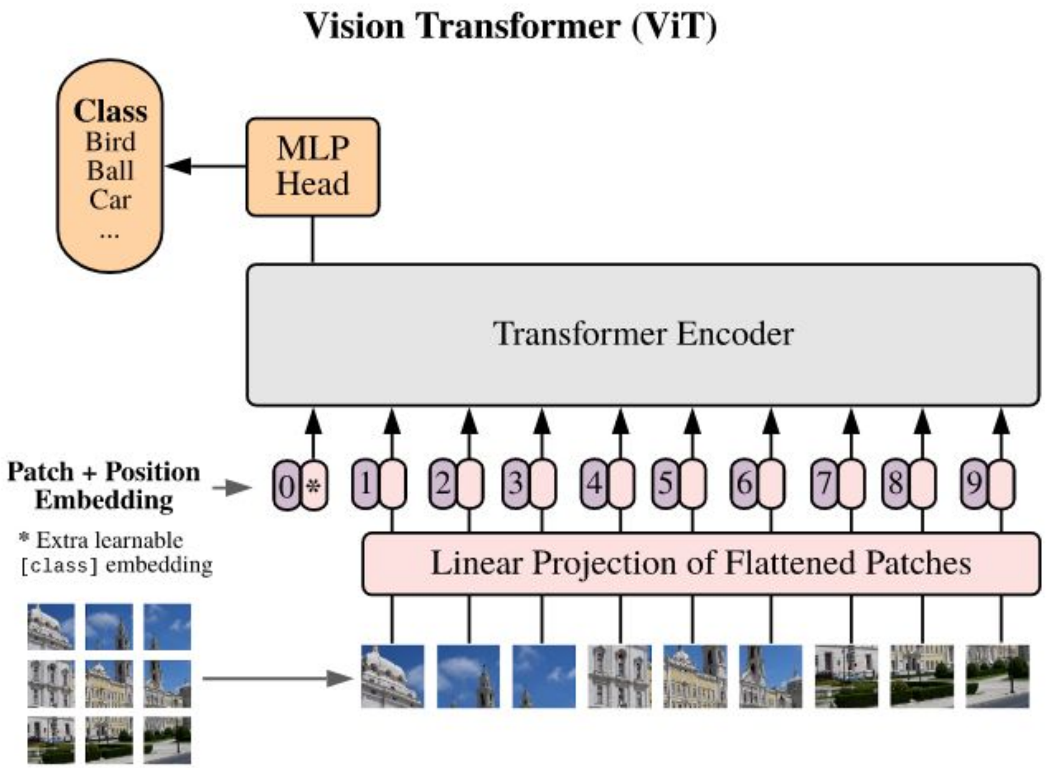


An Image is Worth 16x16 Words

- Imagenet and CIFAR with transformers
  - 88.55% on ImageNet,
  - 90.72% on ImageNet-ReaL,
  - 94.55% on CIFAR-100

- Paper by [Dosovitskiy et al.]

- Other approaches to vision tasks
  - [Taming Transformers for High-Resolution Image Synthesis]



Sources

Transformer

- Paper
  - Attention is all you need. Vaswani et al.
  - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Devlin et al.
  - Reformer: The Efficient Transformer Kitev et al.

- Good Read
  - Jay Alammars The Illustrated Transformer
  - Jay Alammars The Illustrated BERT

- Conference Talk:
  - Attention is all you need attentional neural network models by Łukasz Kaiser