CS 584 Natural Language Processing

Introduction to Transformer and BERT

Department of Computer Science
Chang Lu
clu13@stevens.edu

Transformer

❖ **Vaswani, Ashish, et al. "Attention is all you need." NeurIPS 2017.**

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Attention is **all you need**

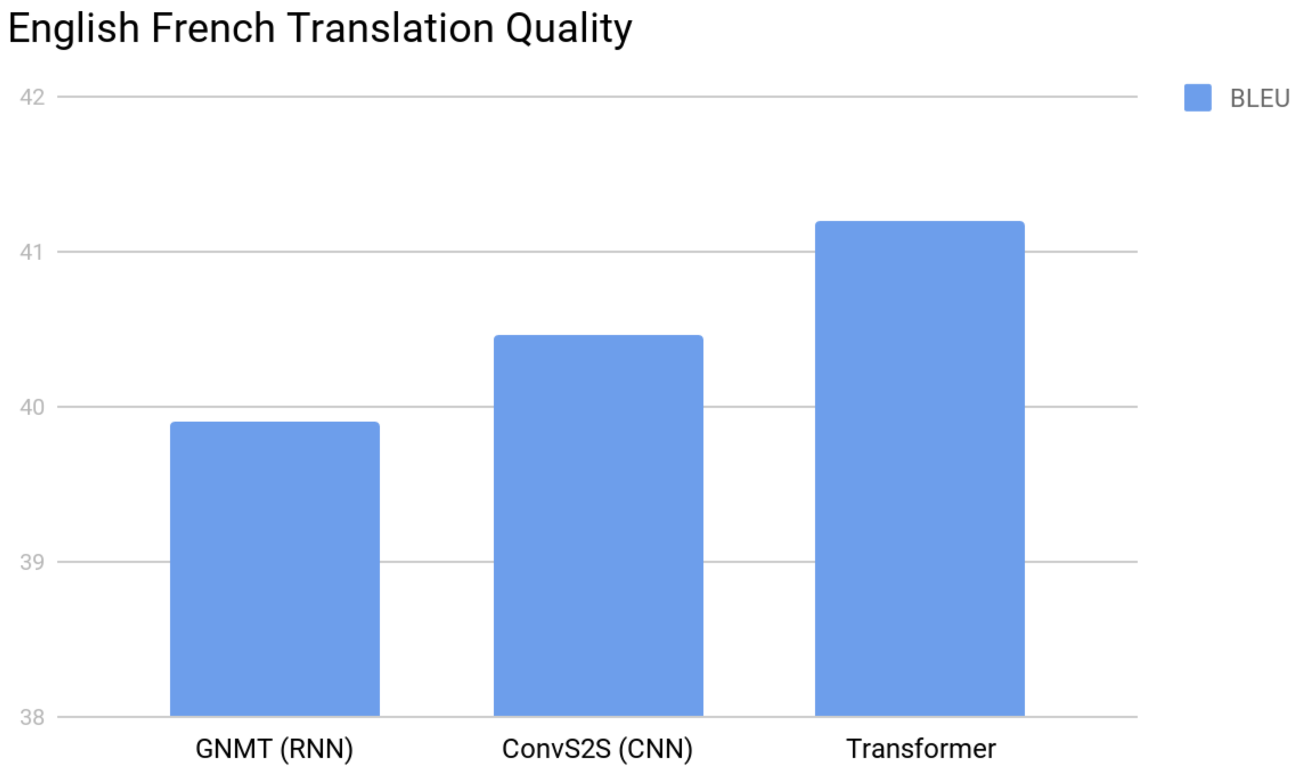
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

… the number of **attention** heads and the **attention** key and value dimensions, keeping the amount of computation constant, as described in Section 3.2.2. While single-head **attention** is 0.9 …

☆ Save 99 Cite Cited by 57295 Related articles All 46 versions Import into BibTeX

Transformer: Overview

- **BLEU score: EN-FR**



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

Transformer: Overview

- Seq2seq: Encoder + Decoder
- Encoder: Self-attention
- Decoder: Self-attention
  + Cross-attention

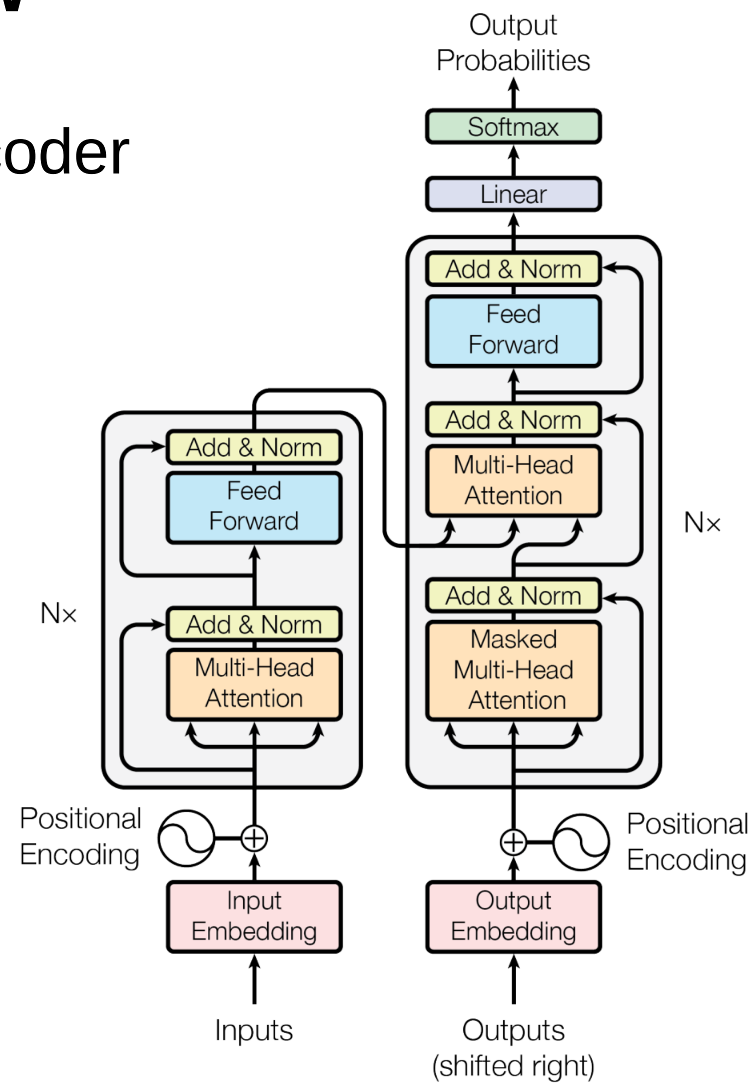
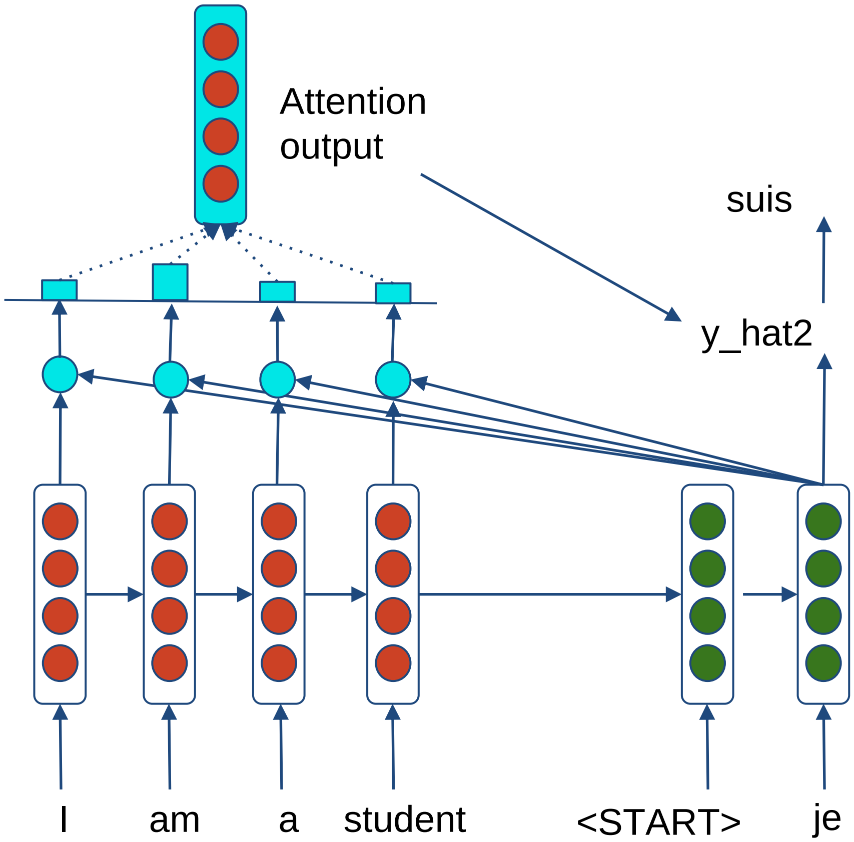


Figure 1: The Transformer - model architecture.

Recall: RNN + Attention

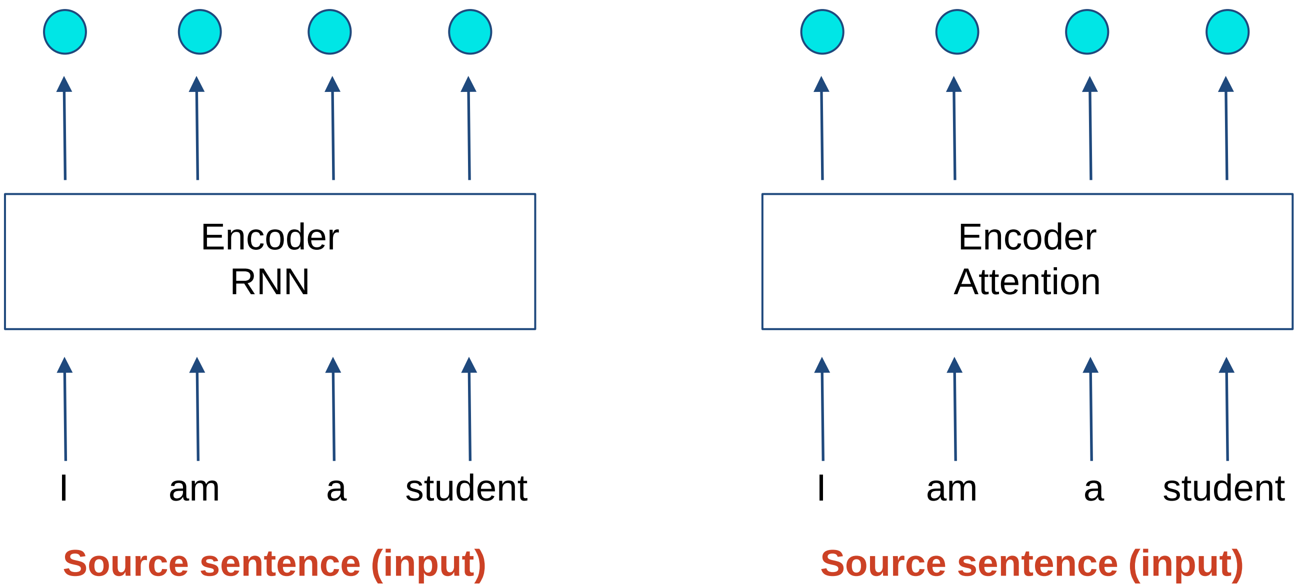
- Encoder: RNN
- Decoder: RNN + Attention
- Problem:
  - Gradient vanishing / exploding, i.e., forget on long inputs
  - RNN is slow



Idea of Transformer

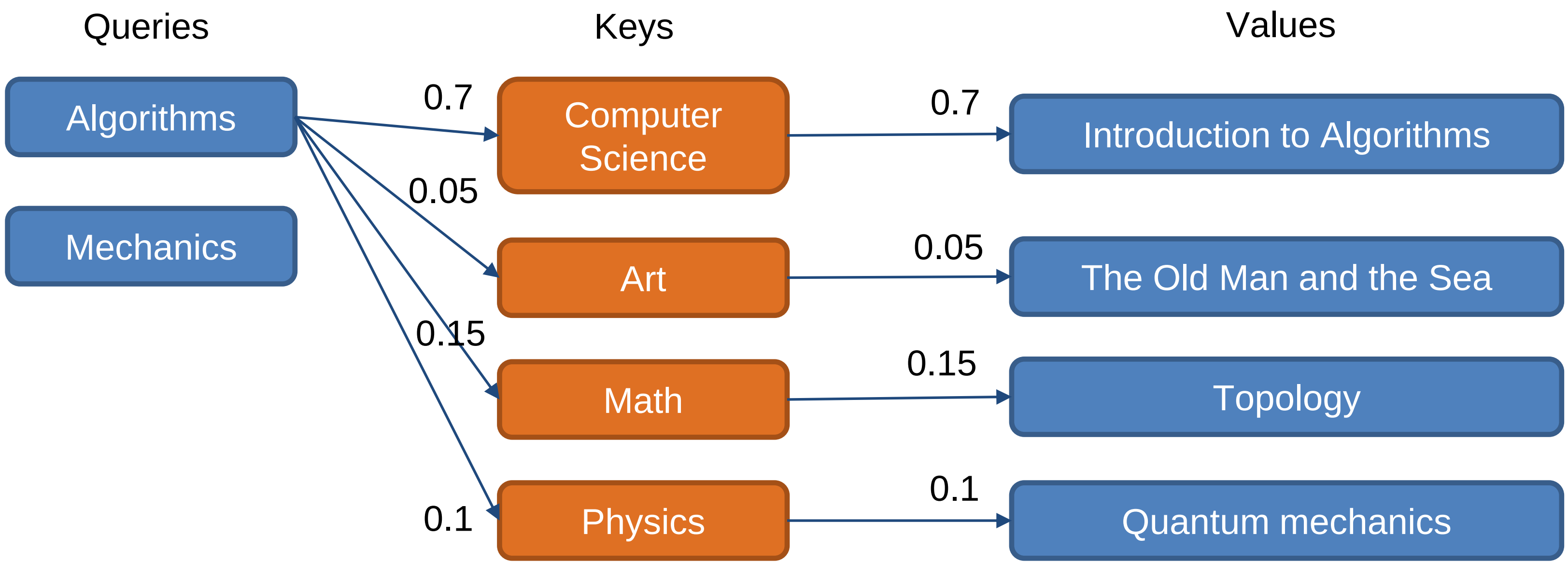
❖ **Use Attention to replace RNN**

- Every token can see all other/previous tokens



Self-attention

- **Scaled dot-product attention: query, key, value**
- **Example: book search**



Self-attention

❖ **Scaled dot-product attention: query, key, value**

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

- $Q, K, V: R^{n \times T \times d}$, (batch_size x seq_len x embedding_size)
- d_k: scaling factor, embedding size

❖ **Explanation:**

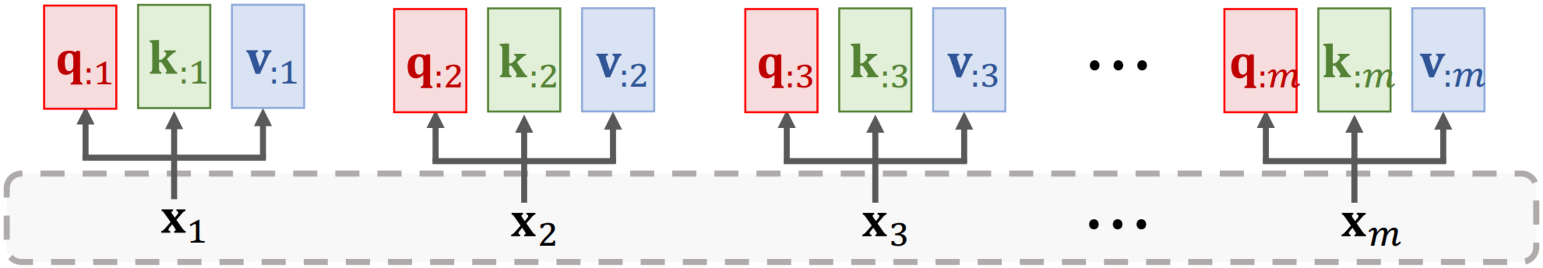
- $QK^\top: R^{n \times T \times T}$, dot product between query and key
- Softmax: $R^{n \times T \times T}$, probability distribution $\boldsymbol{\alpha}$
- Weighted average of values

Self-attention

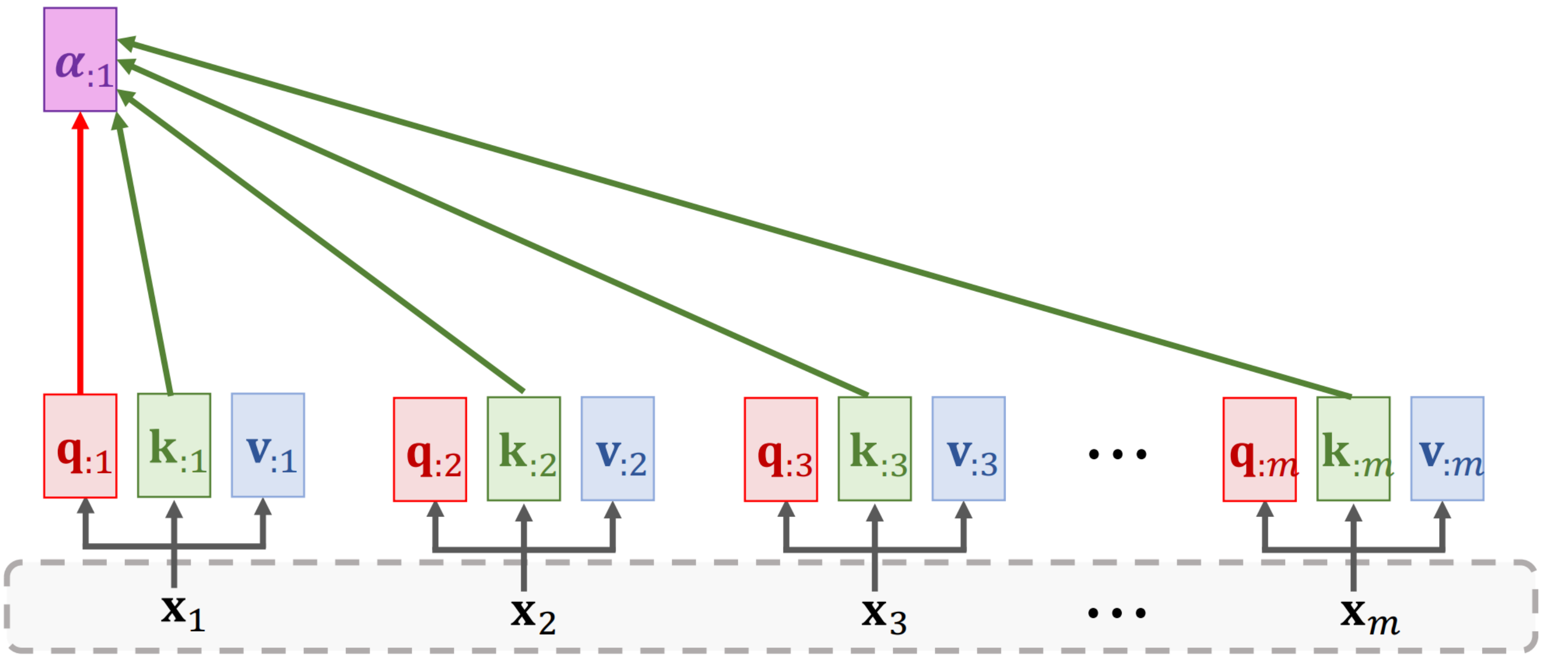
Inputs:

$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\cdots$ $\mathbf{x}_m$

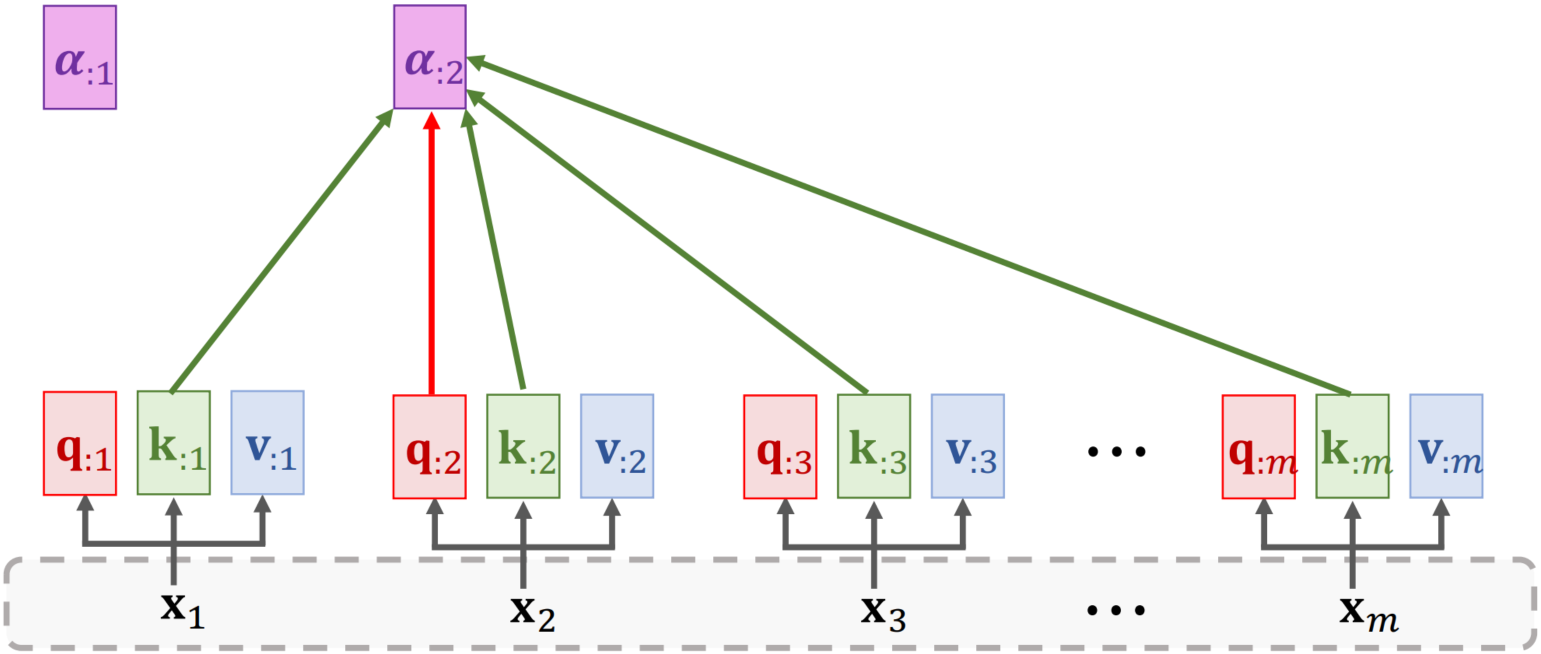
Self-attention



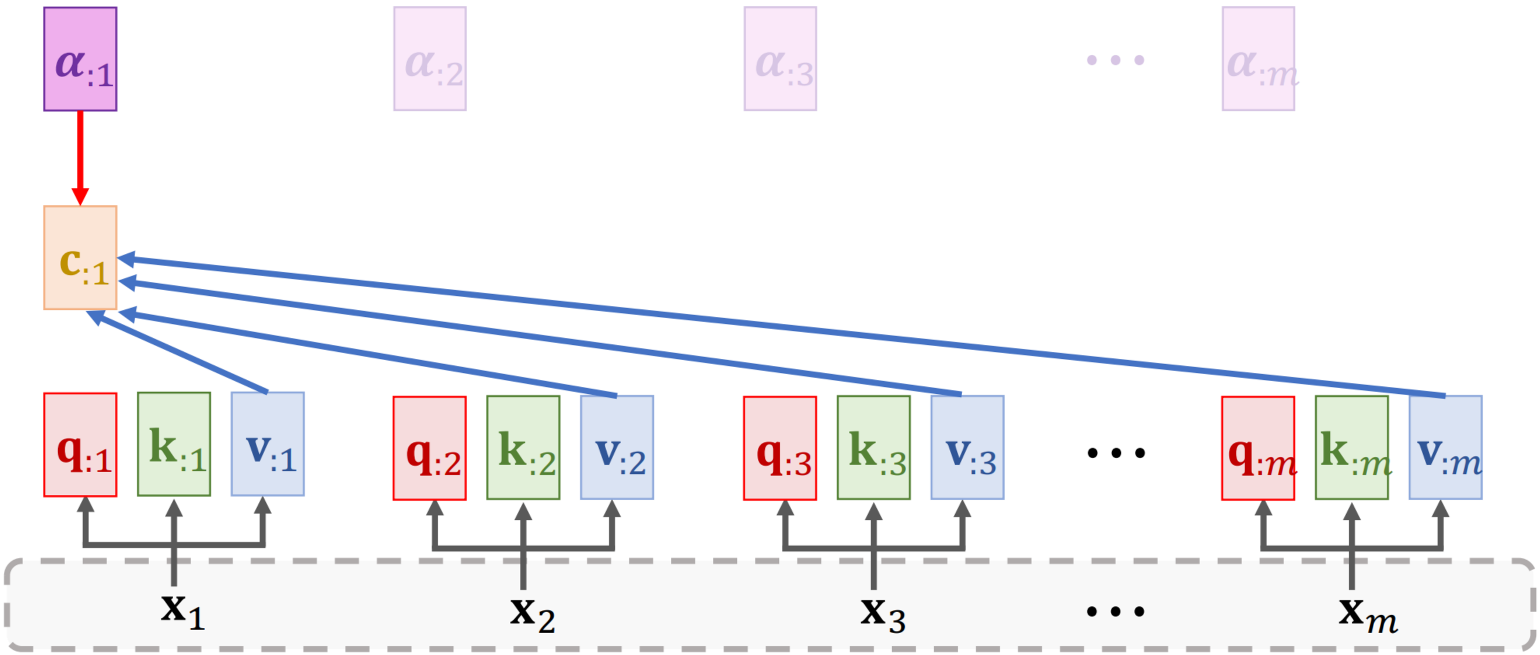
Self-attention



Self-attention

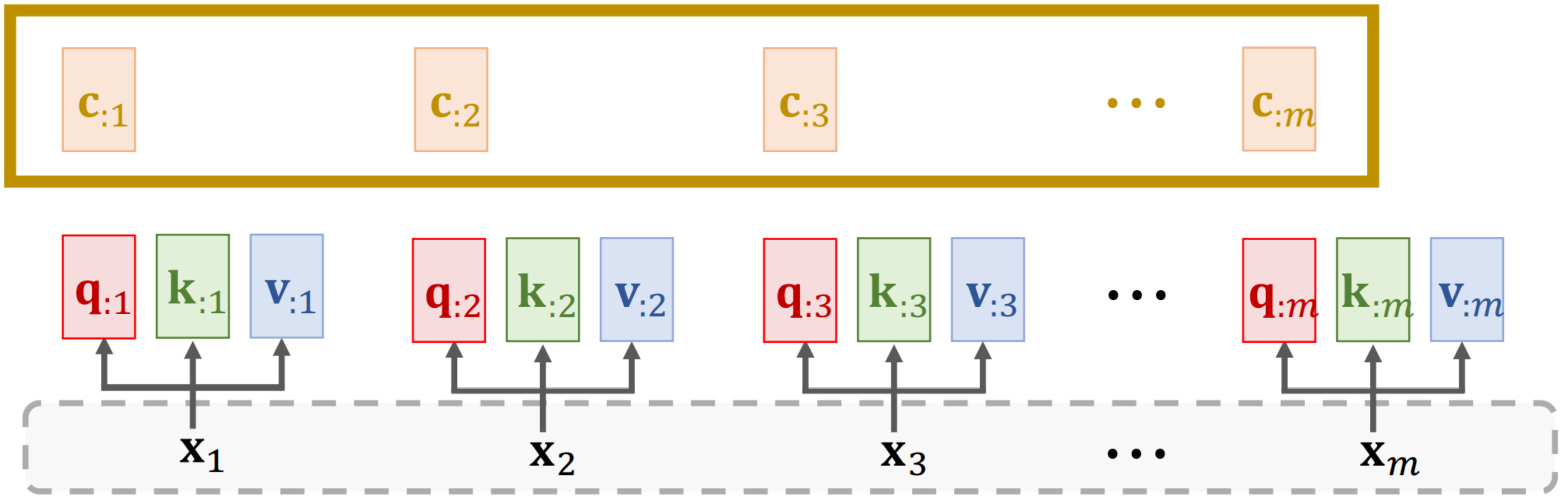


Self-attention



Self-attention

Output of self-attention layer:



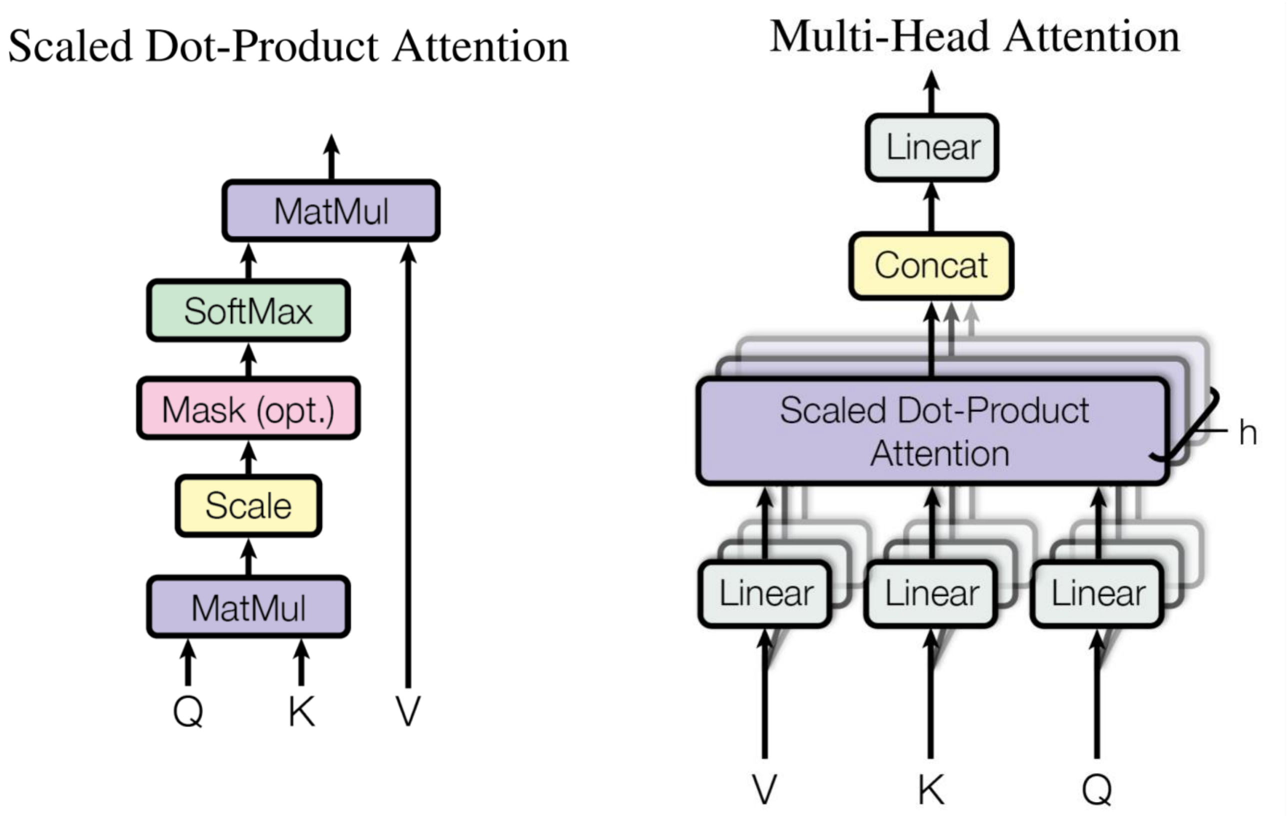
Multi-head attention

- Project Q, K, V for h times -> self-attention -> concat
- Allow model focus on different subspace

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

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

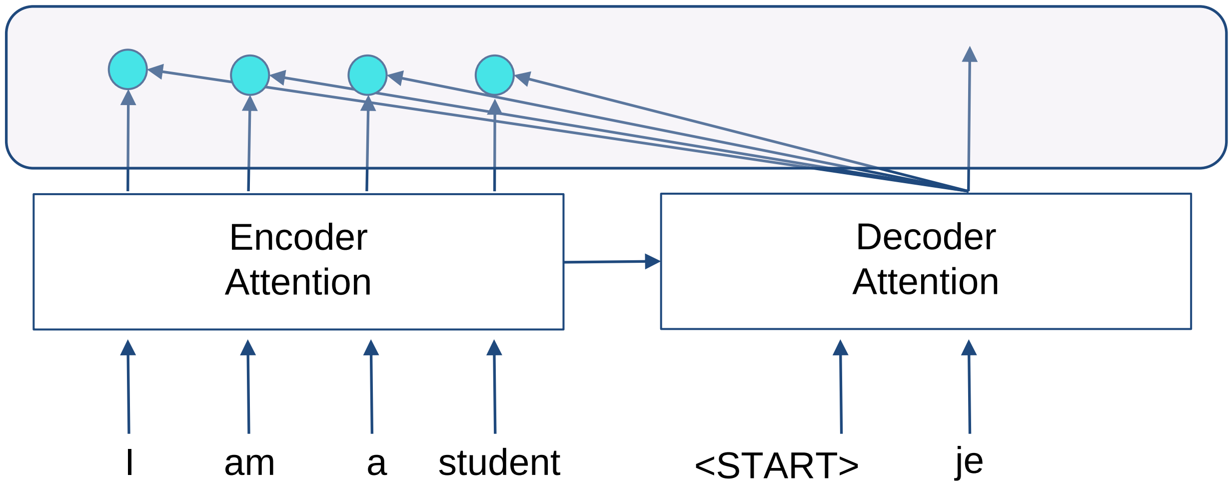
Multi-head attention



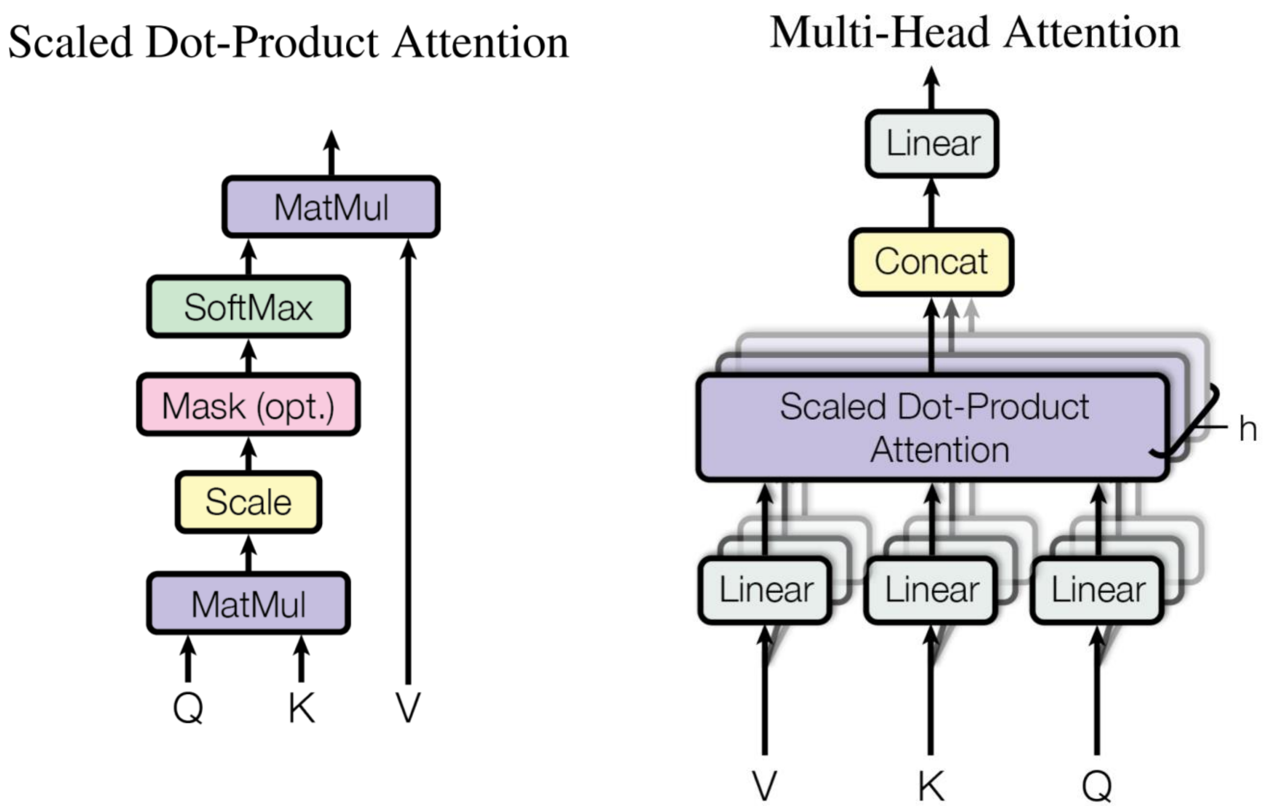
- Q, K, V of self-attention: embedding of input tokens or outputs from previous attention layers

Cross-attention

- ❖ **RNN part is replaced in both Encoder and Decoder by self-attention**
- ❖ **What is next?**
  - Decoder attention with encoder contexts



Cross-attention



- Q: Output of decoder self-attention
- K, V of cross attention: Encoder output
- Query which source token is important to the current target token

Feed-forward

- **Dense layers after self-attention/cross-attention**

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Transformer: Overview

❖ Seq2seq: Encoder + Decoder

❖ Encoder: Self-attention

❖ Decoder: Self-attention + Cross-attention

- Stack N times for each (attention + feed-forward)

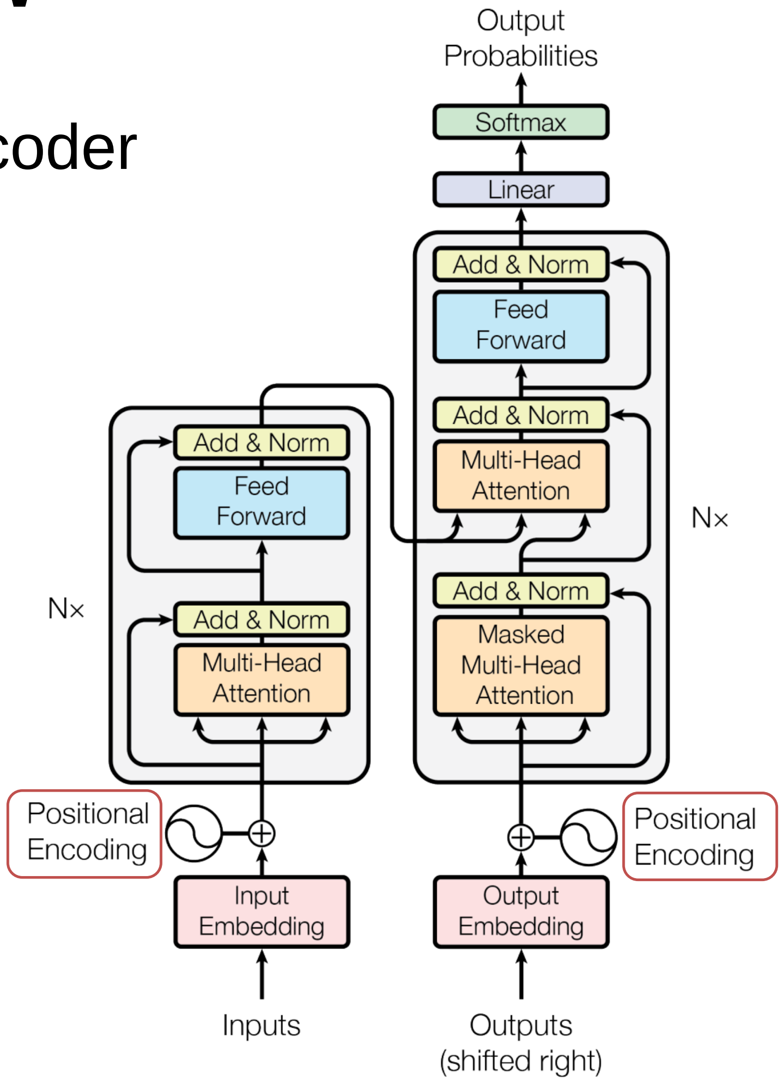


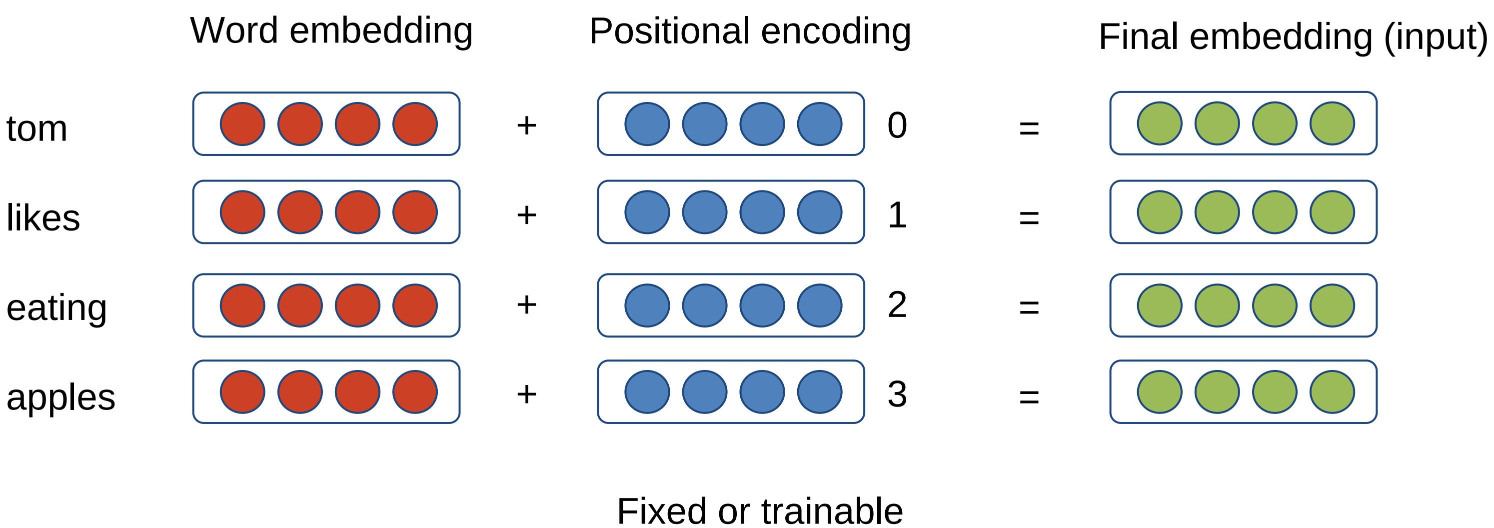
Figure 1: The Transformer - model architecture.

Positional Encoding

- Self-attention does not consider ordering in a sequence

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

- “tom likes eating apples” = “apples like eating tom”
- Add position information in word embedding



BERT

- Bidirectional Encoder Representations from Transformers
- Kenton, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL 2019.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin   Ming-Wei Chang   Kenton Lee   Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Bert: Pre-training of deep bidirectional transformers for language understanding
J Devlin, MW Chang, K Lee, K Toutanova - arXiv preprint arXiv …, 2018 - arxiv.org
We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language …
Save   Cite   Cited by 52952   Related articles

BERT

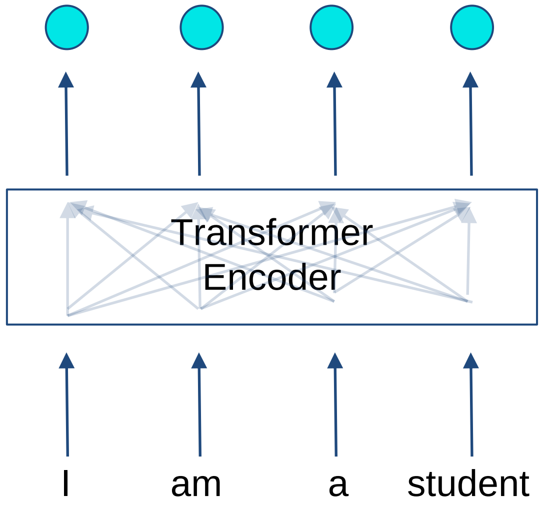
- ❖ Goals:
  - Provide a universal pretrained language model
  - Easily fine-tune on downstream tasks
- ❖ Contribution: Lead the scheme of “Pre-training + fine-tuning” in NLP

BERT: Structure

❖ Transformer Encoder

❖ Why bidirectional?

- Every token can see previous and following tokens

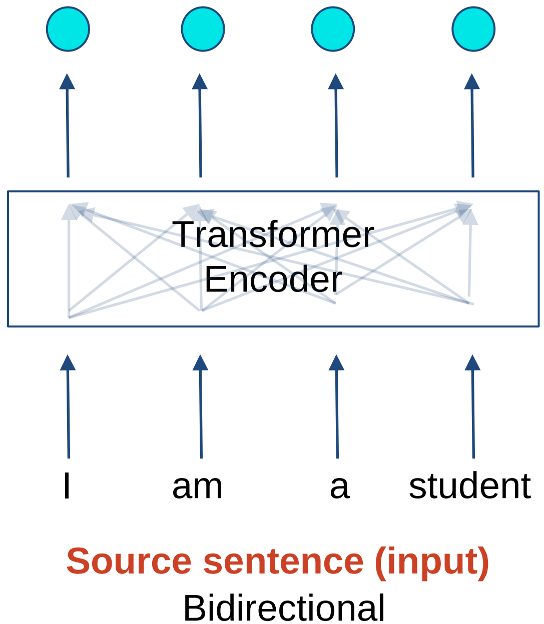


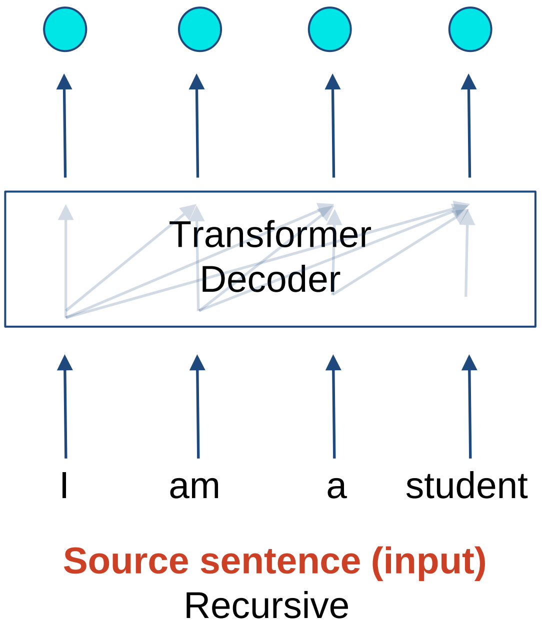
Source sentence (input)

BERT: Structure

❖ Bidirectional (Encoder)
  - Every token can see previous and following tokens

❖ Recursive (Decoder), such as GPT
  - Every token can only see previous tokens





BERT

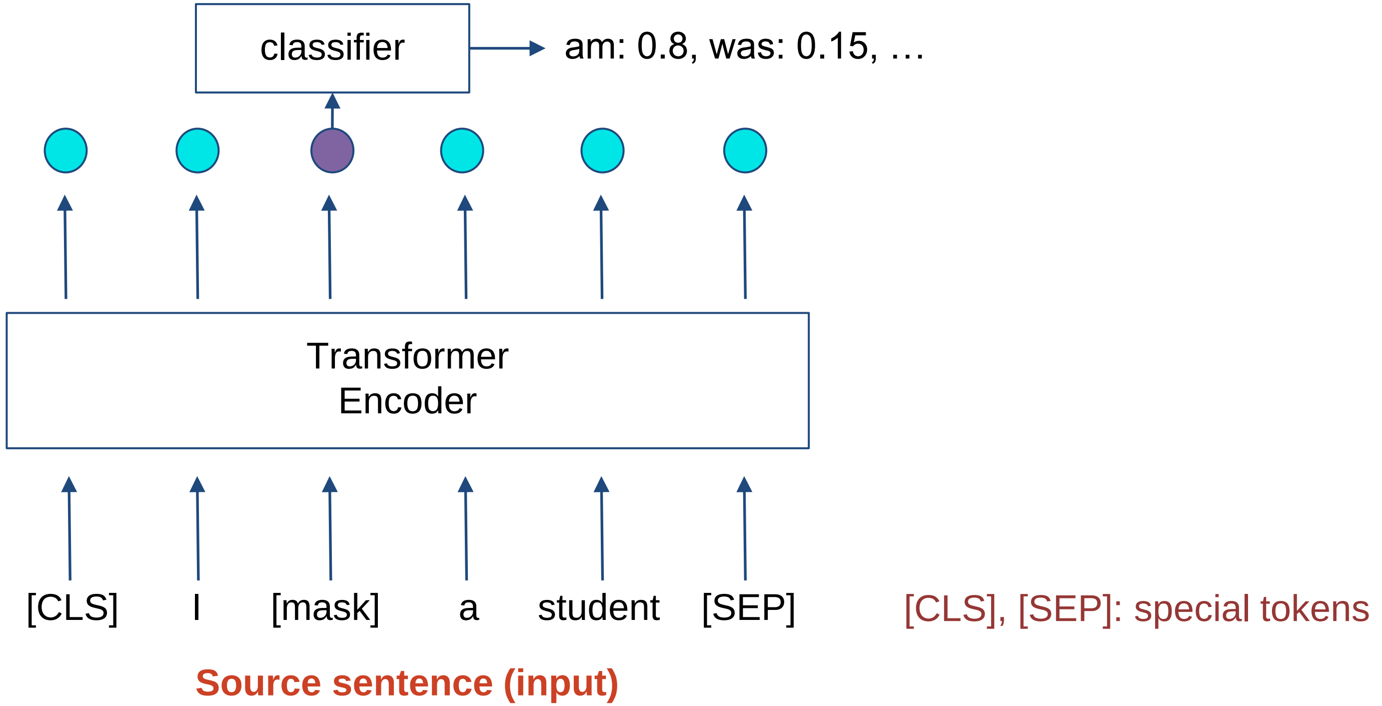
- ❖ Pretraining Goals:
  - Let the model understand natural languages
- ❖ Tasks:
  - Masked token prediction
  - Next sentence prediction

Masked Token Prediction

❖ Randomly mask some tokens in the dataset

- Mask rate: 15%

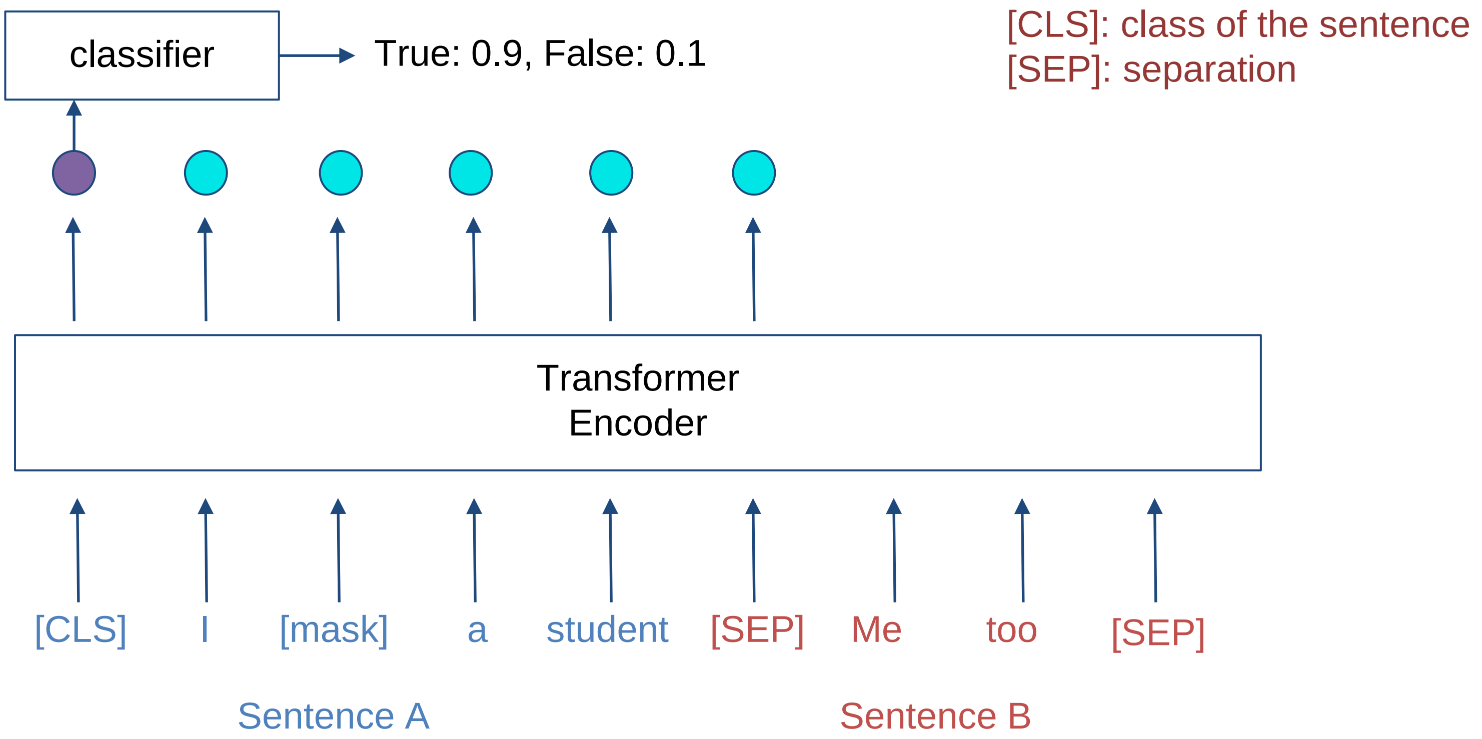
❖ Let the model predict the masked tokens in a sentence



Next Sentence Prediction

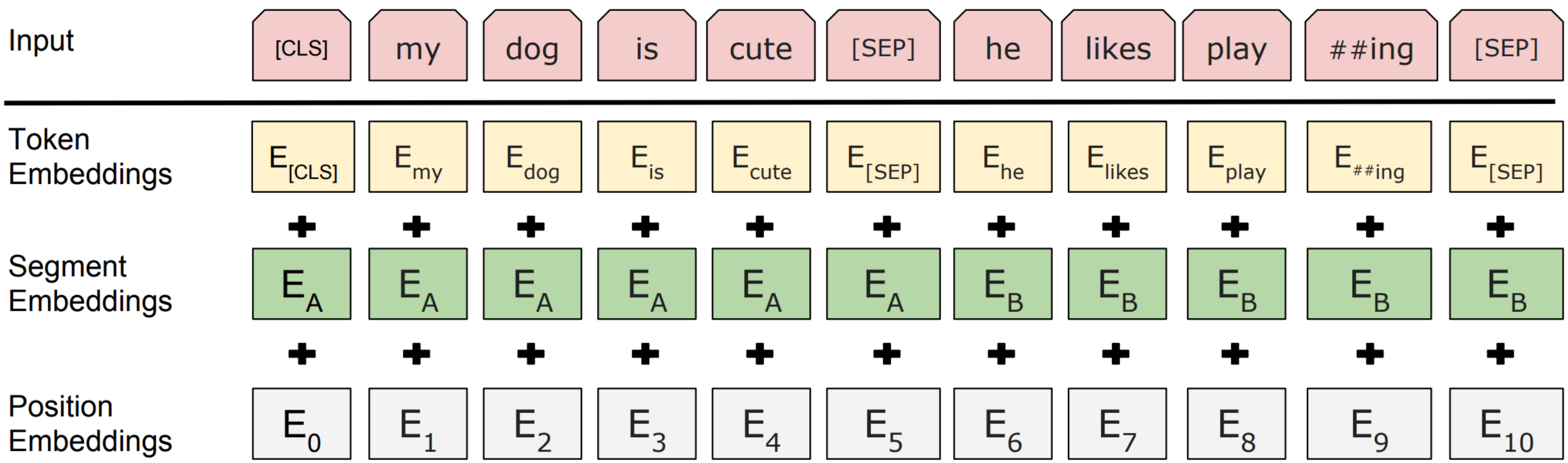
❖ Given two sentences A and B, let the model predict whether B is the next sentence of A

- True/False = 50% / 50%



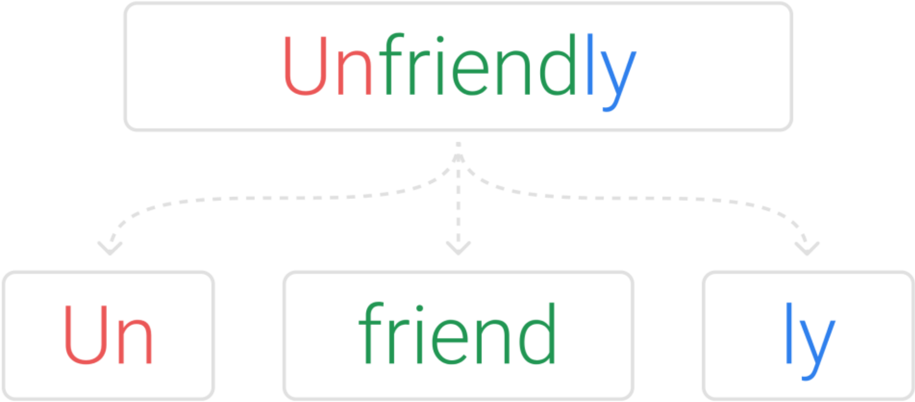
Positional Encoding

- ❖ Word embedding
- ❖ Segment embedding
- ❖ Position Embedding



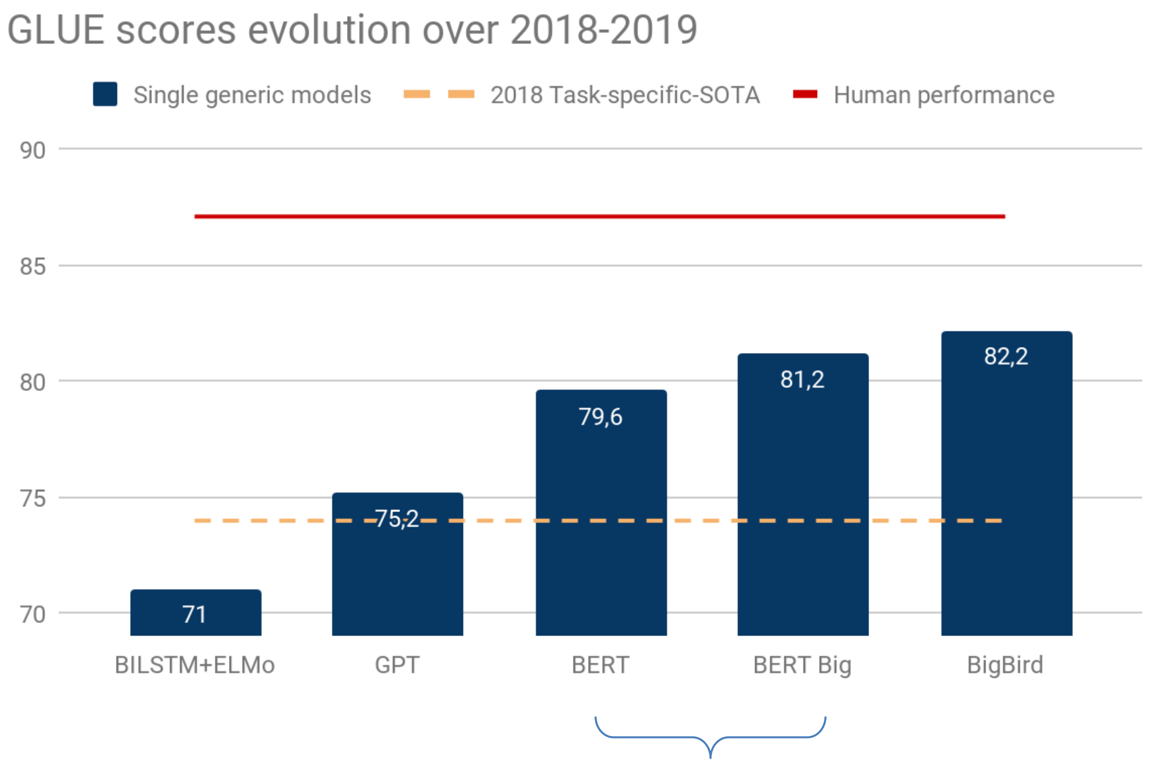
Tokenization

- ❖ Wordpiece: frequent subword

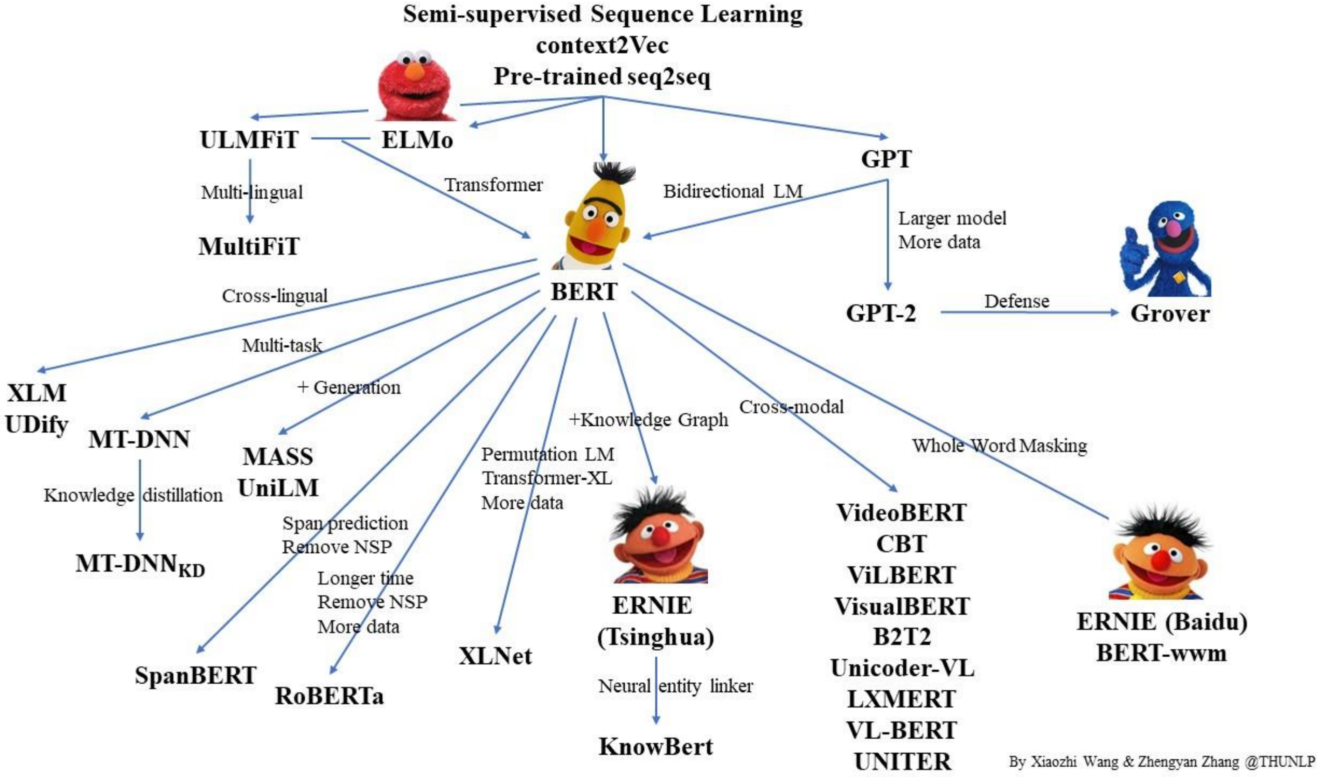


BERT Performance using GLUE

- GLUE: The General Language Understanding Evaluation benchmark on various tasks



BERT Family



Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
GPT
Multi-lingual
Transformer
Bidirectional LM
MultiFiT
Larger model
More data
BERT
Cross-lingual
GPT-2
Defense
Grover
Multi-task
XLM
UDify
+ Generation
MT-DNN
MASS
UniLM
+Knowledge Graph
Cross-modal
Whole Word Masking
Knowledge distillation
Permutation LM
Transformer-XL
More data
Span prediction
Remove NSP
MT-DNN$_{KD}$
Longer time
Remove NSP
More data
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
ERNIE
(Tsinghua)
ERNIE (Baidu)
BERT-wwm
SpanBERT
RoBERTa
XLNet
Neural entity linker
KnowBert
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

Q & A