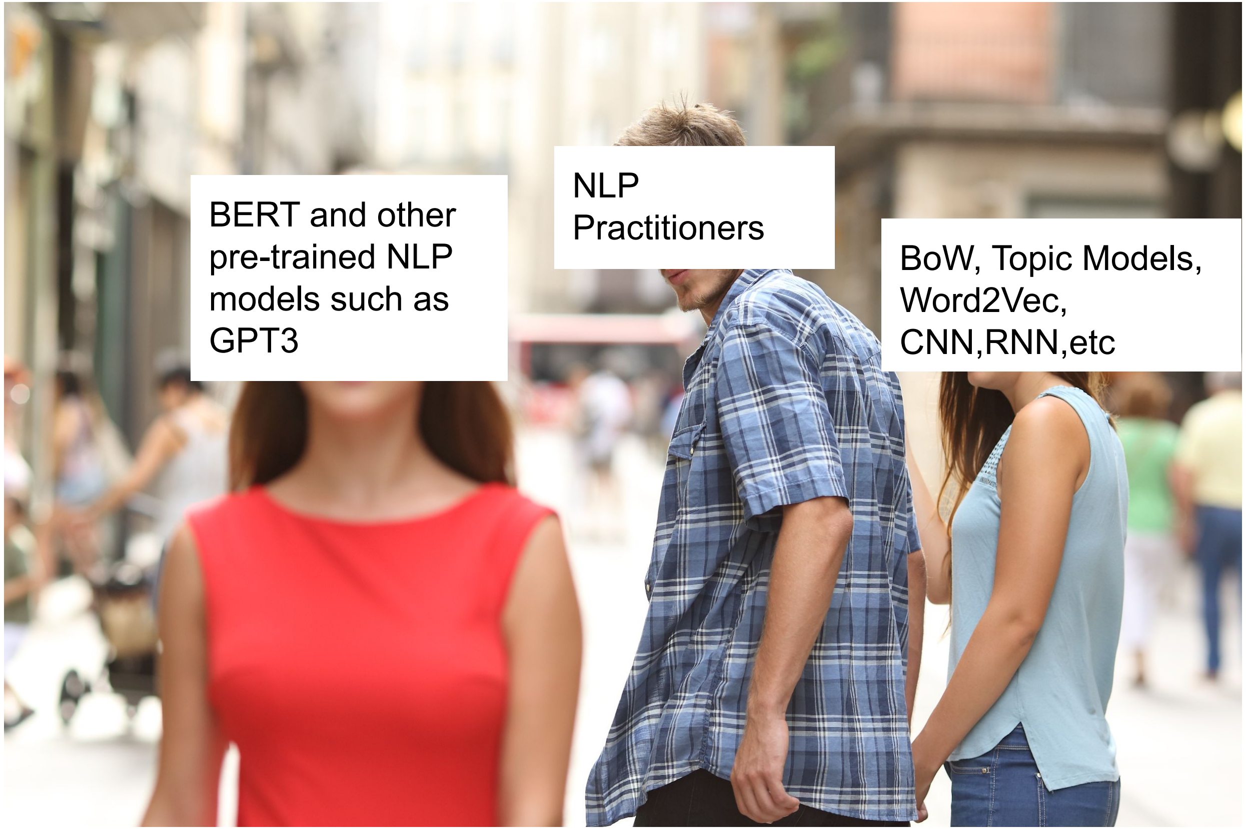
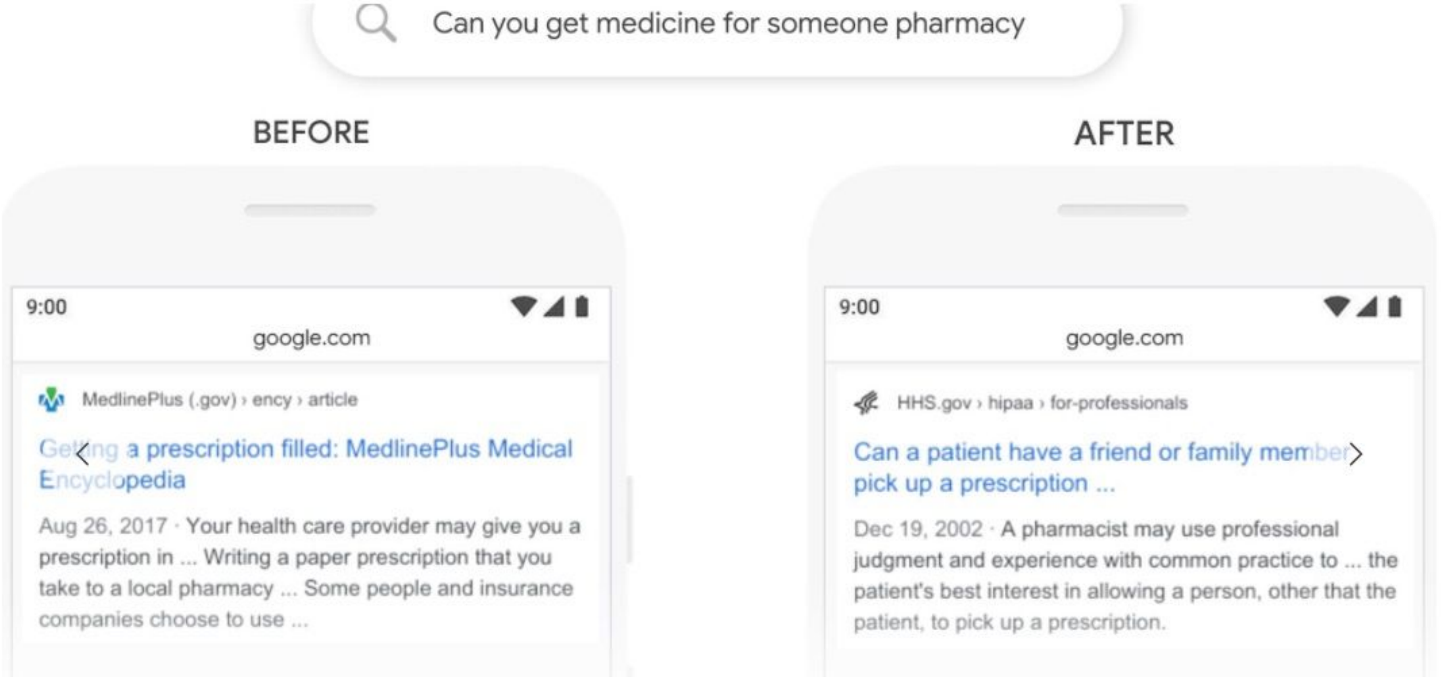
Frontiers in NLP II

Pre-trained NLP Model: BERT


BERT and other pre-trained NLP models such as GPT3
NLP Practitioners
BoW, Topic Models, Word2Vec, CNN,RNN,etc

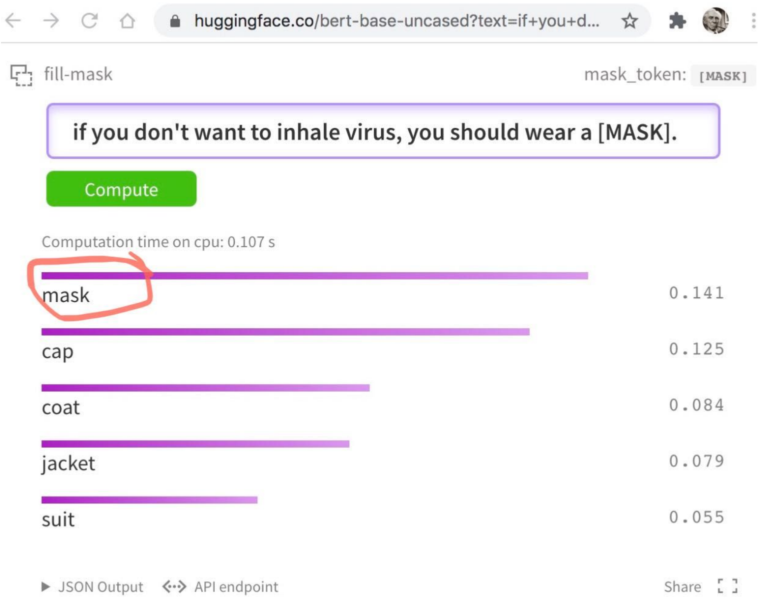
BERT in Google Search



With the latest advancements from our research team in the science of language understanding--made possible by machine learning--we're making a significant improvement to how we understand queries, representing the biggest leap forward in the past five years, and one of the biggest leaps forward in the history of Search.

https://blog.google/products/search/search-language-understanding-bert/

BERT vs Somebody



Try:
https://huggingface.co/bert-base-uncased?text=if+you+don%27t+want+to+inhale+virus+is%2C+you+should+wear+a+%5BMASK%5D

Extraction-based QA using BERT

```
question = "What can we learn in NUS?"
answer_text = "The National University of Singapore (NUS) is the national research university of Singapore. \
               Founded in 1905 as the Straits Settlements and Federated Malay States Government Medical School, NUS is the oldest higher education institution in Singapore. \
               It is consistently ranked within the top 20 universities in the world and is considered to be the best university in the Asia-Pacific. \
               NUS is a comprehensive research university, \
               offering a wide range of disciplines, including the sciences, medicine and dentistry, design and environment, law, arts and social sciences, engineering, business, computing and music \
               at both the undergraduate and postgraduate levels."
```

BERT

```
print('Answer: "' + answer + '"')
```

```
Answer: "sciences , medicine and dentistry , design and environment , law , arts and social sciences , engineering , business , computing and music"
```

```
question = "What does NUS mean?"
answer_text = "The National University of Singapore (NUS) is the national research university of Singapore. \
               Founded in 1905 as the Straits Settlements and Federated Malay States Government Medical School, NUS is the oldest higher education institution in Singapore. \
               It is consistently ranked within the top 20 universities in the world and is considered to be the best university in the Asia-Pacific. \
               NUS is a comprehensive research university, \
               offering a wide range of disciplines, including the sciences, medicine and dentistry, design and environment, law, arts and social sciences, engineering, business, computing and music \
               at both the undergraduate and postgraduate levels."
```

BERT

```
print('Answer: "' + answer + '"')
```

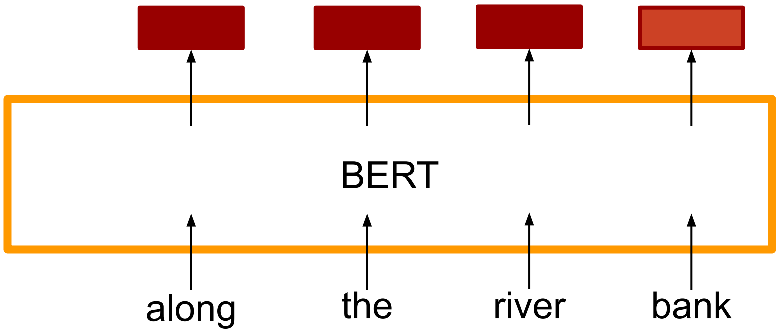
```
Answer: "national university of singapore"
```

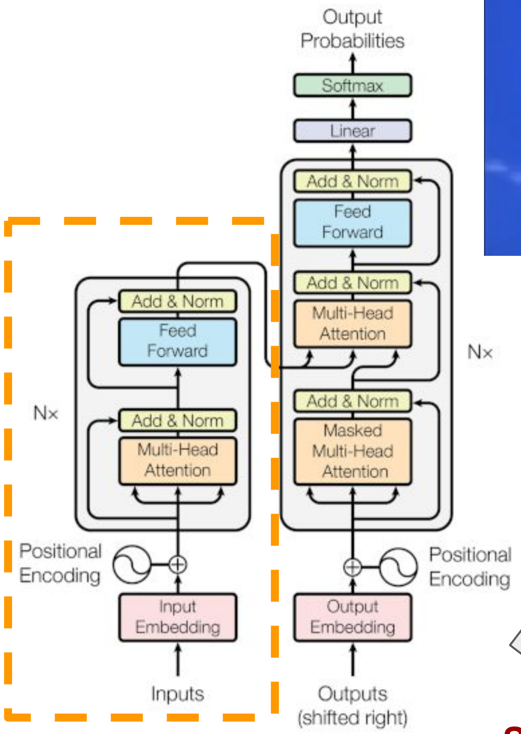
What is BERT

BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (**BERT**)
- BERT: Encoder of Transformer,

Transformer





Given a sequence of words, generate a sequence of vectors and then can be used for various NLP tasks

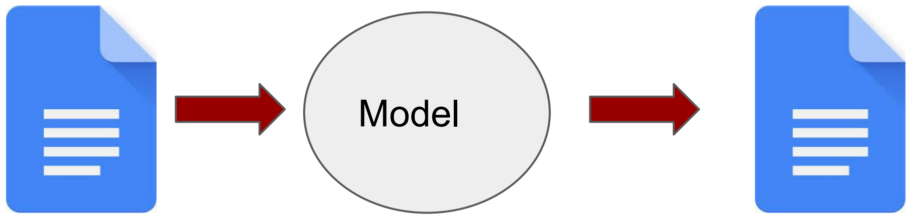
Solve Seq2Seq Task

Agenda

1. Seq2Seq
2. Transformers
3. BERT Model:
    a. How to pre-train BERT
    b. How to use BERT

Seq2Seq

Seq2Seq Task

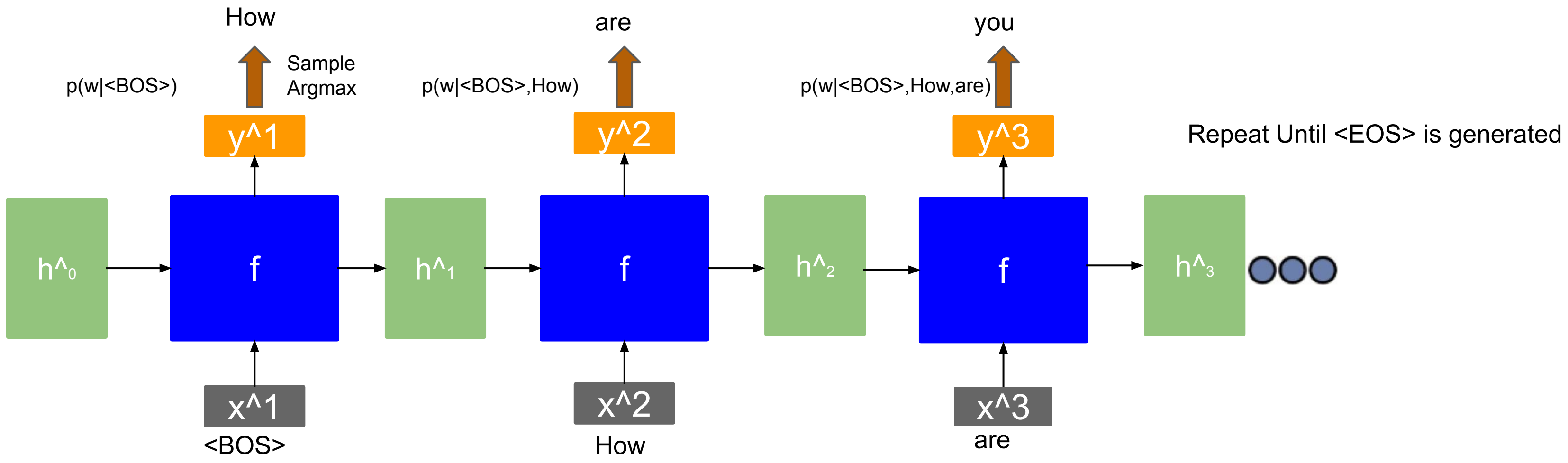


One Sequences of Words

General Sequences of Words

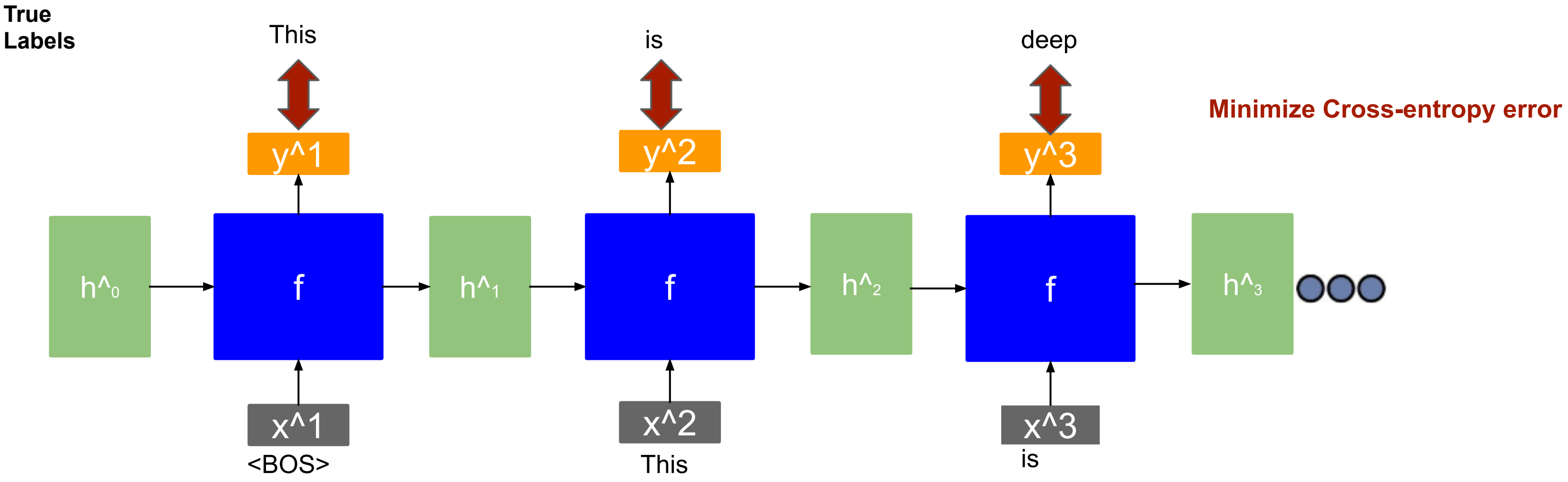
Sequence Generation: Inference

- Sentences are sequences of words/characters
- Generate a word/character each time by RNN

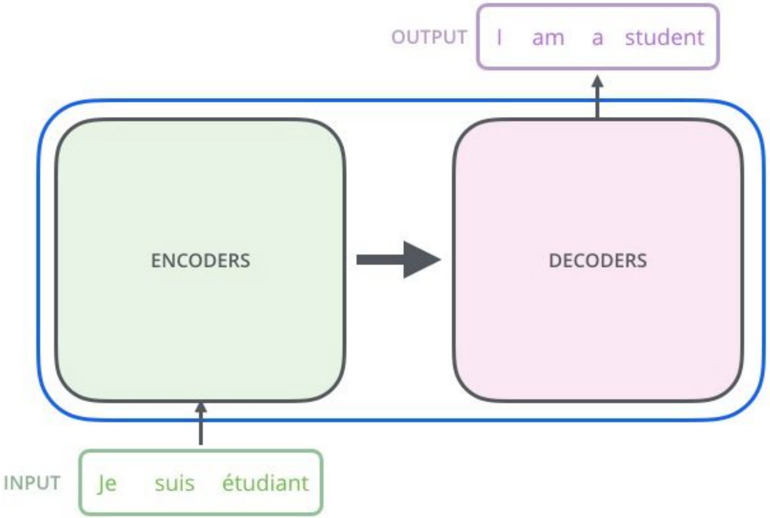


Sequence Generation: Training

- Training (Language Model):
    - Training data/corpus: This is deep learning

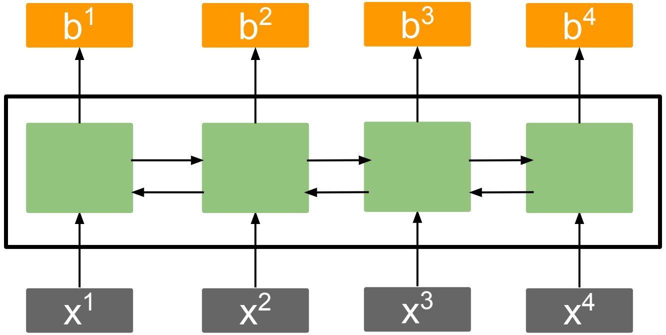


Seq2Seq



Source:
http://jalammar.github.io/illustrated-transformer/

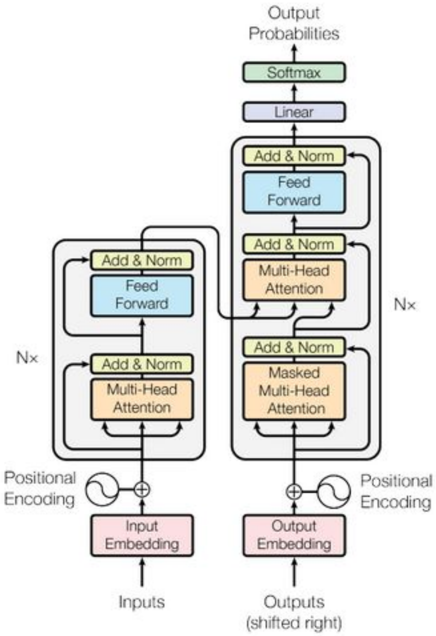
Encoder and Decoder:
Take a sequence as input, generate a sequence as output



RNN is slow, which can not be parallelized.

Transformer

What is Transformer



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

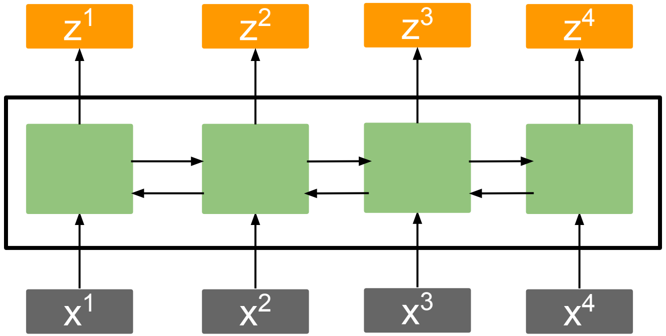
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

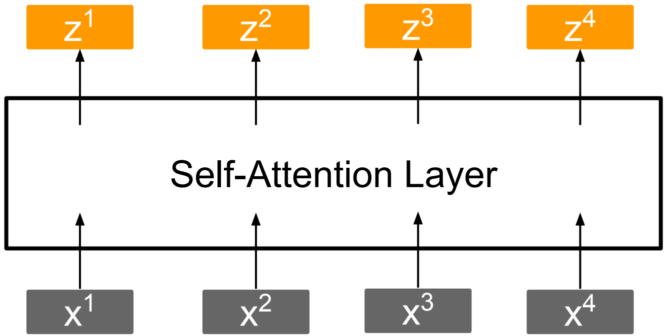
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Transformer is a sequence to sequence model (Encoder and Decoder), but it replace Recurrent Neural Networks with self-attentional modules.

Seq2Seq

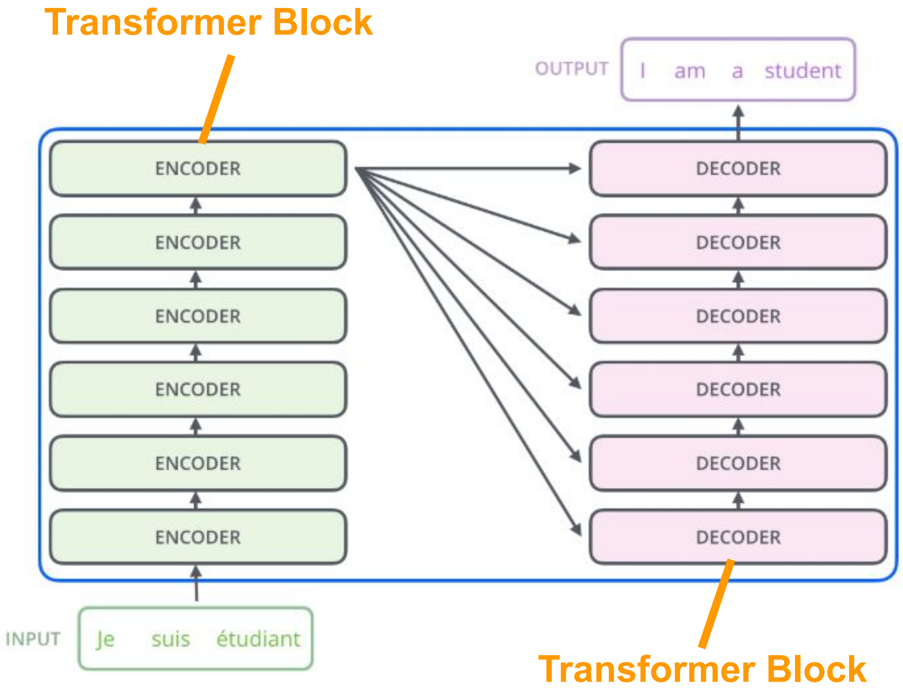




b^i can be computed **parallelly** based on **the whole input sequence.**

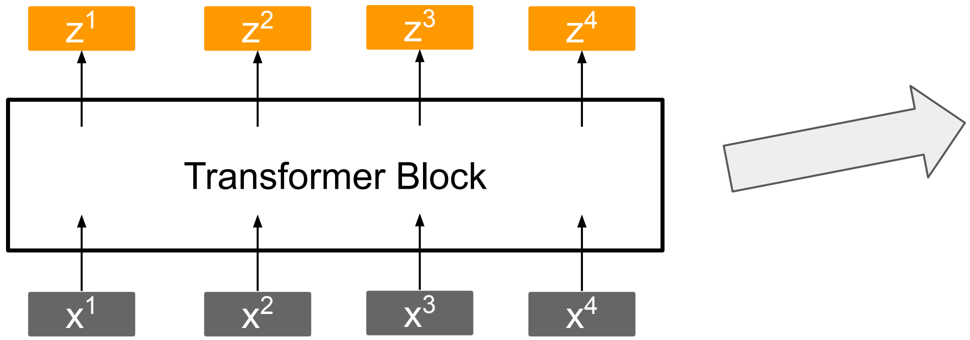
Transformer

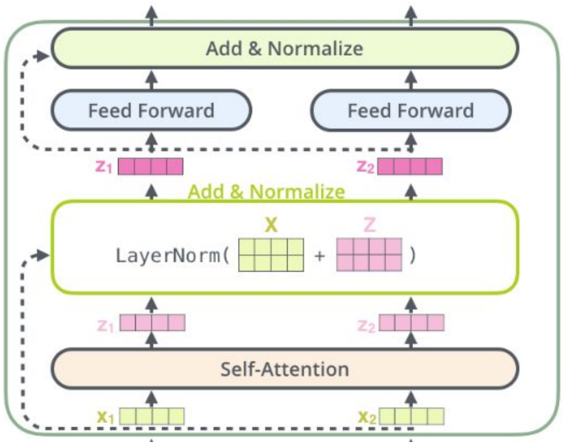
1. Transformer Block: Key component in Transformer (Like layer computation in neural network)
2. **Encoder**:
    a. Stack 6 **transformer blocks**
    b. Learn representations for the input sequence
3. Decoder:
    a. Stack another 6 transformer blocks.
    b. Generate output sequences conditioned on the learned representations from encoder.



Here, we only focus on ENCODER parts.

Transformer Block in Encoder





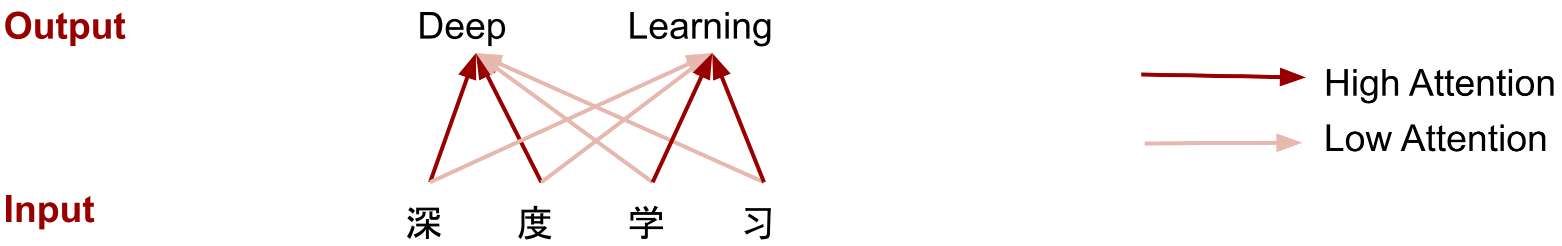
1. Input: A sequence of vectors
2. Output: A sequence of vectors
3. Key Components:
   a. Self-attention Layer
   b. Positional Embeddings
   c. Residual and Normalization Layer
   d. Fully-connected Layer

The target is to map all input sequences into an abstract continuous representation that holds the learned information for that entire sequence.

Attention Mechanism

Attention in deep learning:

1. Attention vectors: a vector of importance weights (how strongly the output variable is correlated with other elements)



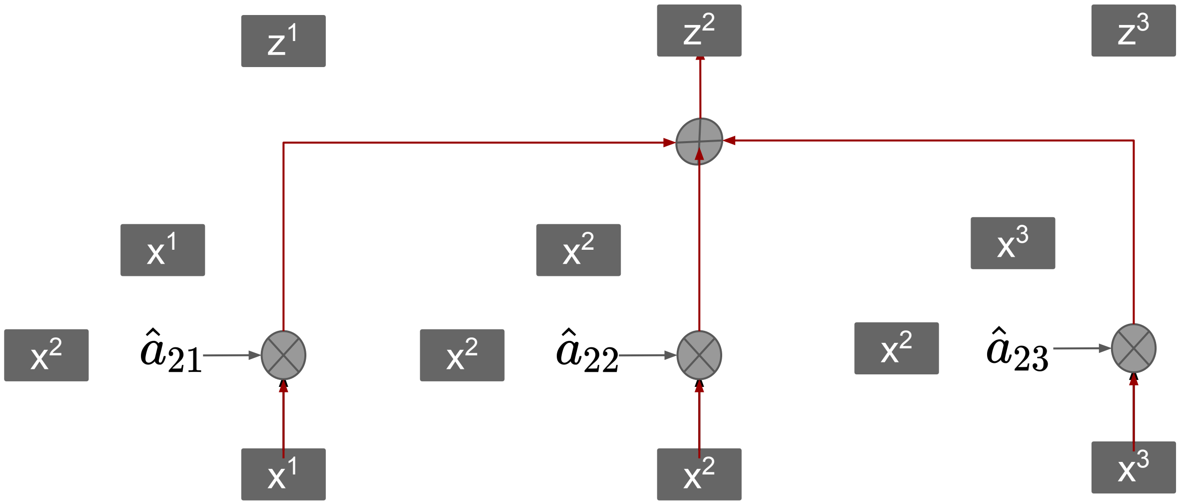
2. The target is approximated by the sum of their input values weighted by the attention scores.

$$Vec_{deep}=0.5*Vec_{\text{深}} + 0.5*Vec_{\text{度}} + 0*Vec_{\text{学}} + 0*Vec_{\text{习}}$$

Basic Self-Attention

Self-attention (**intra**-attention):

1. A sequence-to-sequence operation taking a sequence of vectors in and generate a sequence of vectors out
2. Relating different positions of the input sequence in order to compute the representation.

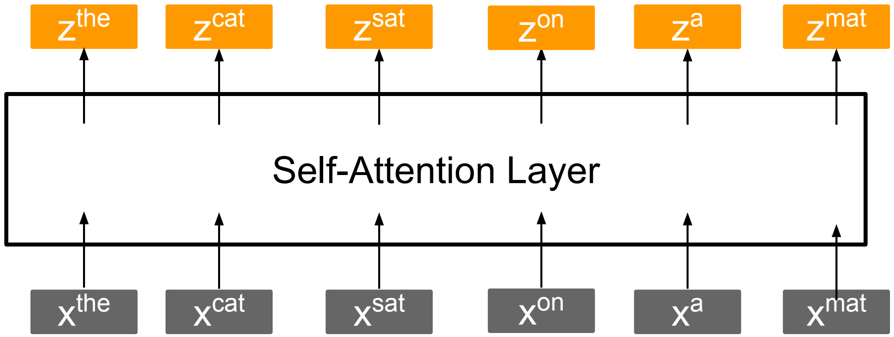


$$z^i = \sum_j \hat{a}_{ij} x^j$$

$$a_{ij} = (x^i)^T (x^j)$$

$$\hat{a}_{ij} = \frac{e^{a_{ij}}}{\sum_j e^{a_{ij}}}$$

Why Self-Attention Works



Toy Example:

1. “the” is not relevant to the interpretation of the other words.
2. To interpret what “sat” means in this sentence, it is very helpful to know “who” was sitting? Therefore, we hope “cat” and “sat” can have a high attention value.

Self-Attention:

1. The dot product express how related two vectors in the input sequence are, with “related” defined by the learning task
2. The output vectors are weighted sums over the whole input sequence, with the weights determined by these dot products.

To fully understand language, it is not sufficient to understand the individual words that make up a sentence, the model must capture how the words relate to each other in the context of the sentence.

Basic Self-Attention

1. There are no model parameters. It is totally determined by the embedding layer.

Embedding Layer:
map word index to a vector

2. Self attention is permutation equivariant. It ignores the order information.

Self-Attention Layer

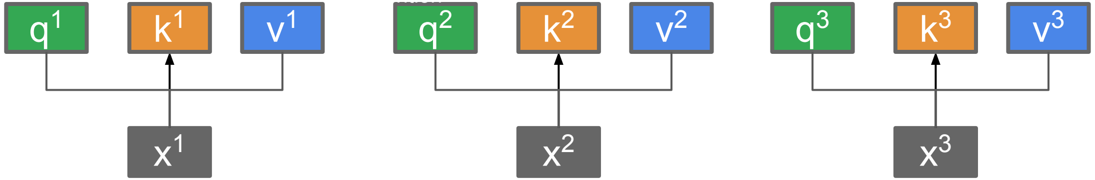
Step 1: Generate **query**, **key**, and **value** vector for the **input** vector at each time step.

q Query (to match others): $q^i = W^q x^i$

k Key (to be matched): $k^i = W^k x^i$

v Value (representation): $v^i = W^v x^i$

Model parameters are introduced here.



Word embeddings

Self-Attention Layer

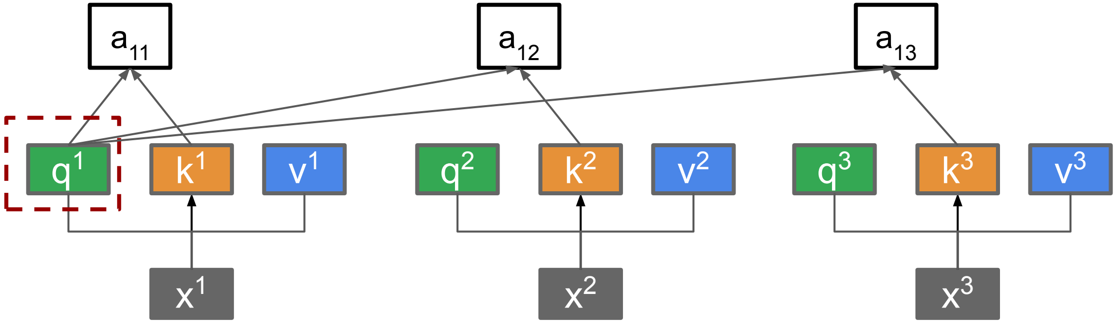
Step 2: Compute attention scores using query vectors and key vectors

To encode the i-th word in the sequence, we need to compute the attention scores between this i-th word and all the words in the sequence.

1. Pick the query vector from the i-th word: q^i
2. Attention score computation between q^i and all key vectors

$$a_{i,j} = \frac{q^i \cdot k^j}{\sqrt{d_k}}$$

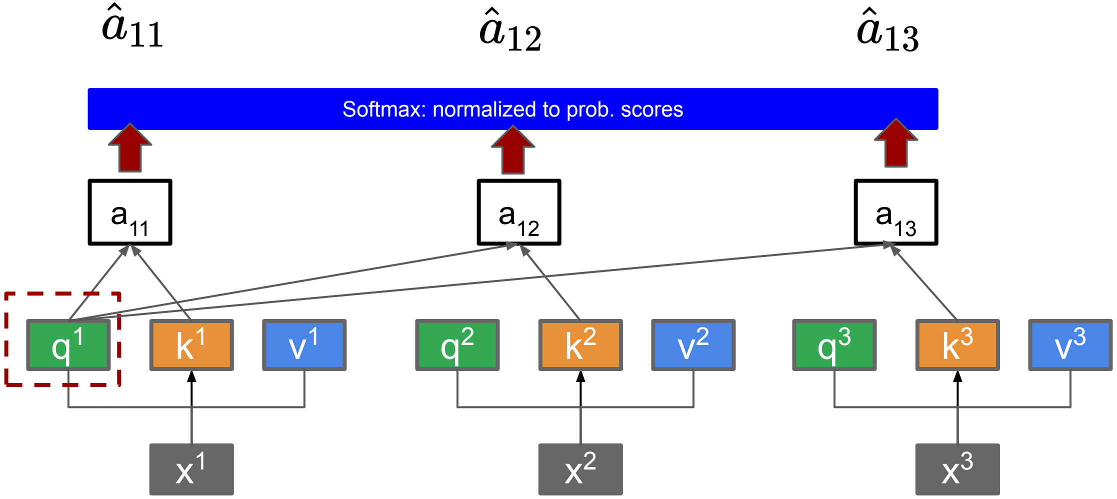
Dim of key vectors



Word vectors

Self-Attention Layer

Step 3: Fed unscaled attention scores into softmax layers $\hat{a}_{1i} = \frac{e^{a_{1i}}}{\sum_j e^{a_{1j}}}$



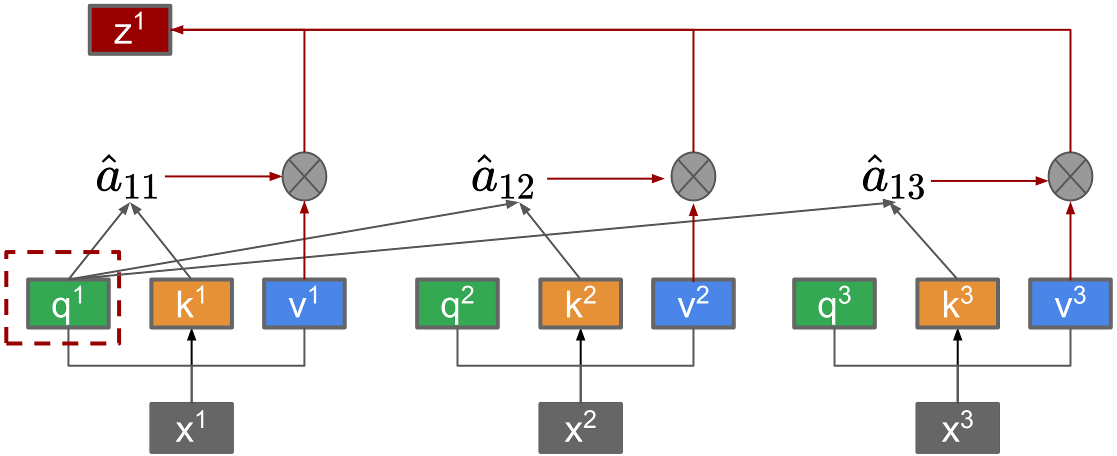
Word vectors

Self-Attention Layer

Step 4: Take the sum of all the value vectors weighted by the attention scores.

Encoded vector for the first element

$$z^1 = \sum_i \hat{a}_{1i} v^i$$



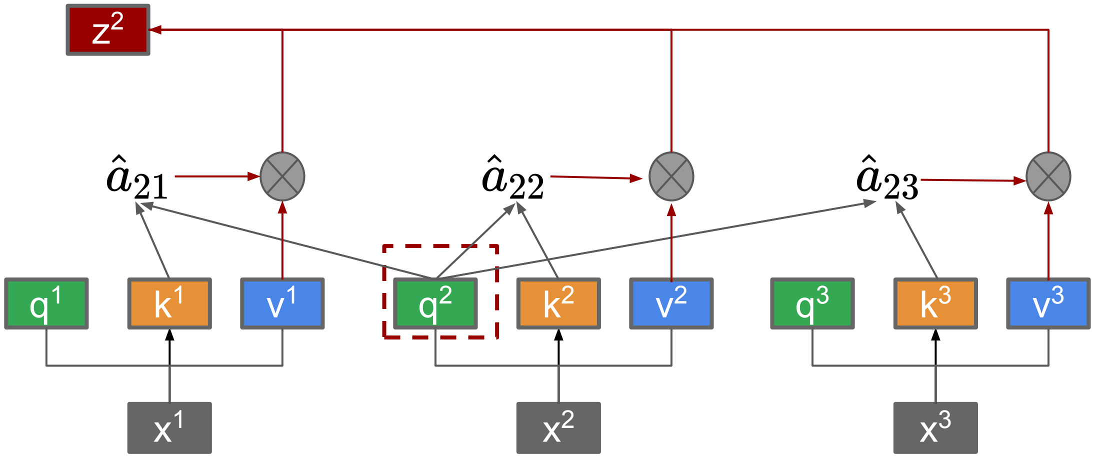
Word vectors

Self-Attention Layer

Step 5: All elements in input sequence x^i will be encoded into new vectors b^i

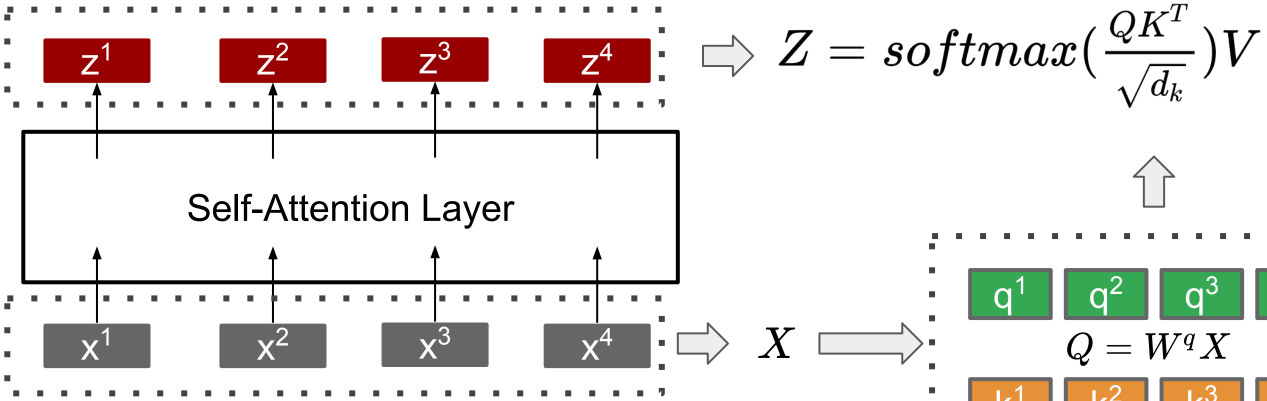
Encoded vector for the second element

$$z^2 = \sum_i \hat{a}_{2i} v^i$$

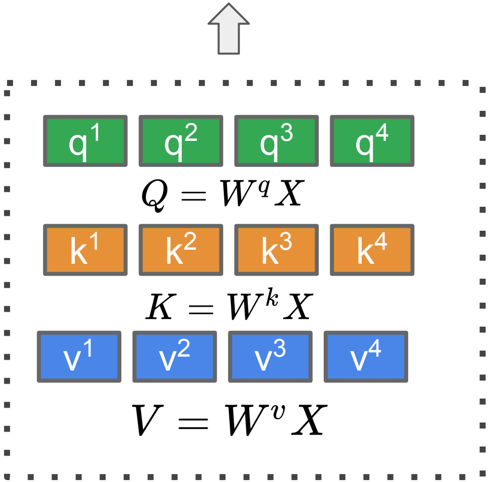


Word vectors

Matrix Formulation



b^i can be computed **parallelly** based on **the whole input sequence.**

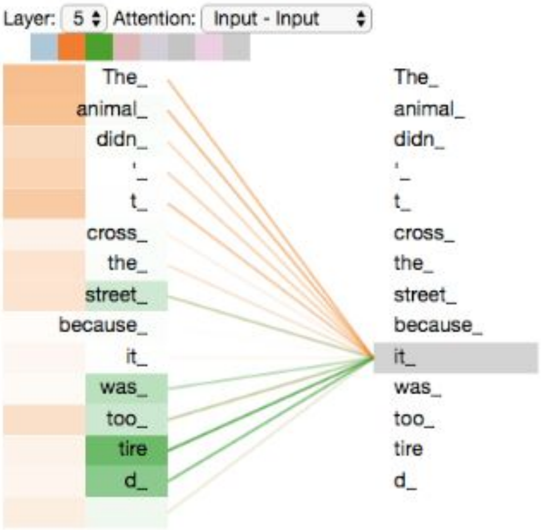


Matrix Multiplication

Multi-head Self-Attention

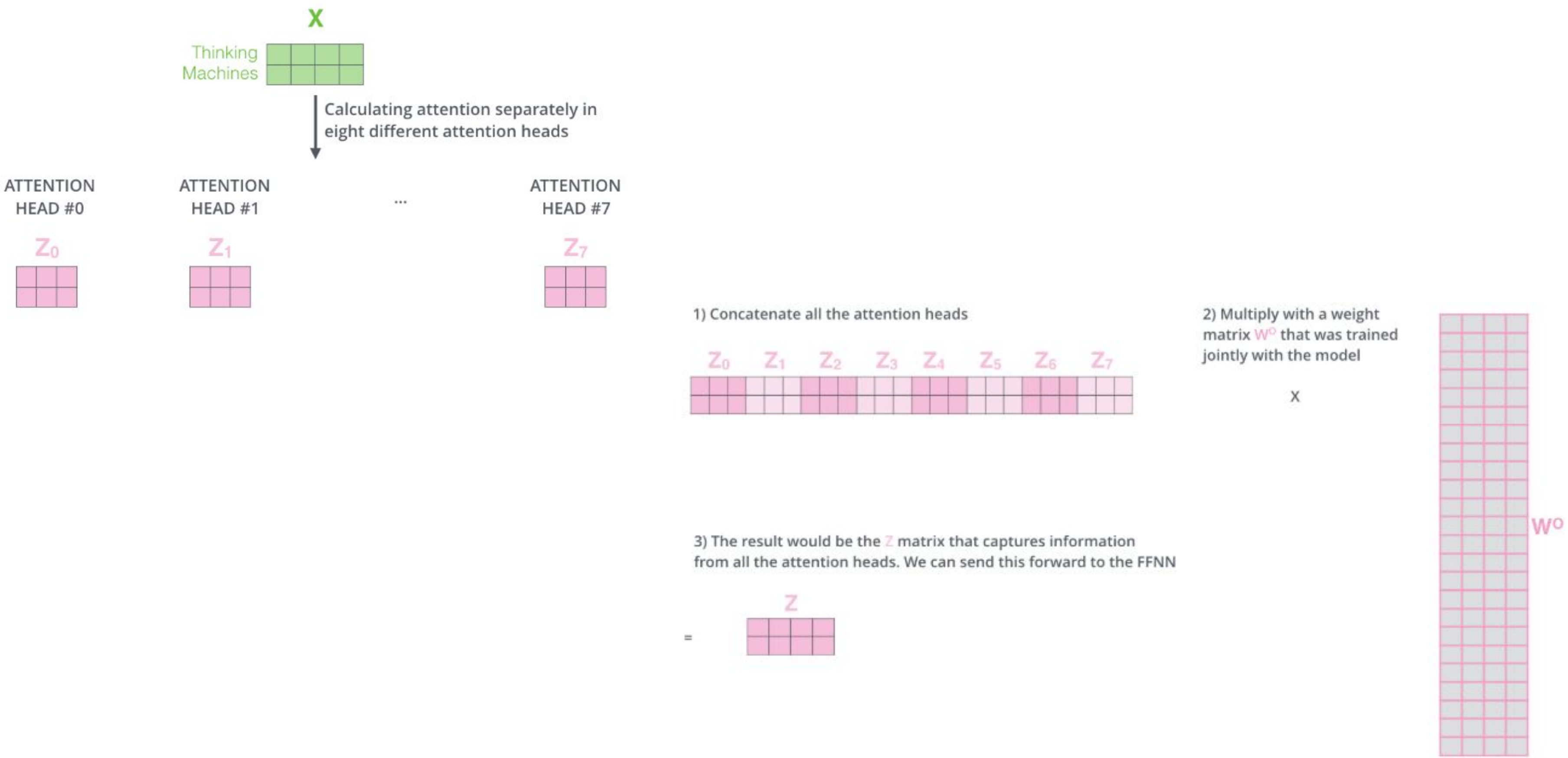
1. Model parameters: W^k, W^q, W^v specific one kind of attention
2. Multi-head means separate W^k, W^q, W^v matrices
   a. Expands the model's ability to focus on different positions
   b. Gives the attention layer multiple "representation subspaces"

High attention from one head



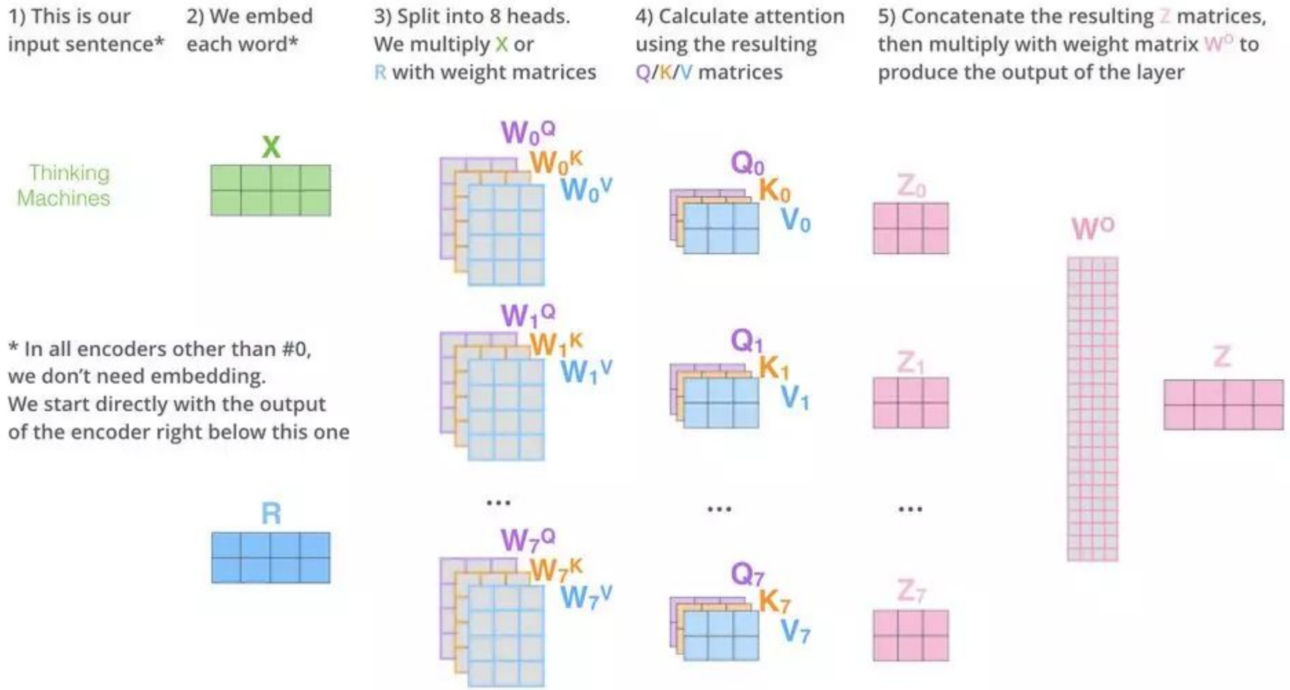
High attention from another head

Multi-head Self-Attention



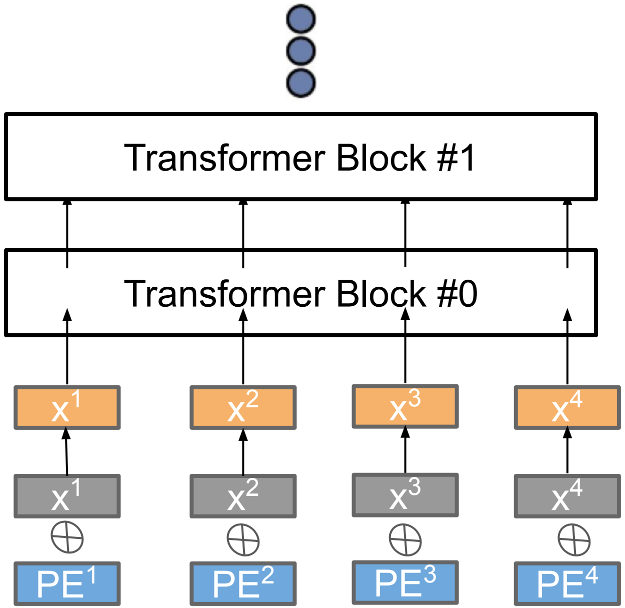
Self-Attention Layer

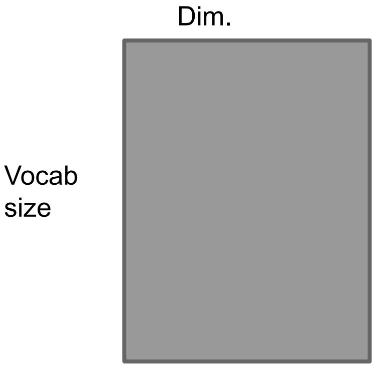
That's pretty much all there is to multi-headed self-attention. It's quite a handful of matrices, I realize. Let me try to put them all in one visual so we can look at them in one place

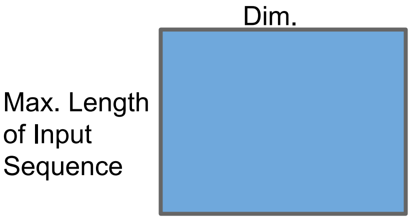


Positional Embeddings

1. No position information in self-attention
2. Positional Embeddings: each position has a unique positional vector PE(pos)
   a. Add this vector to each input embeddings
   b. Expands the model’s ability to focus on different positions.







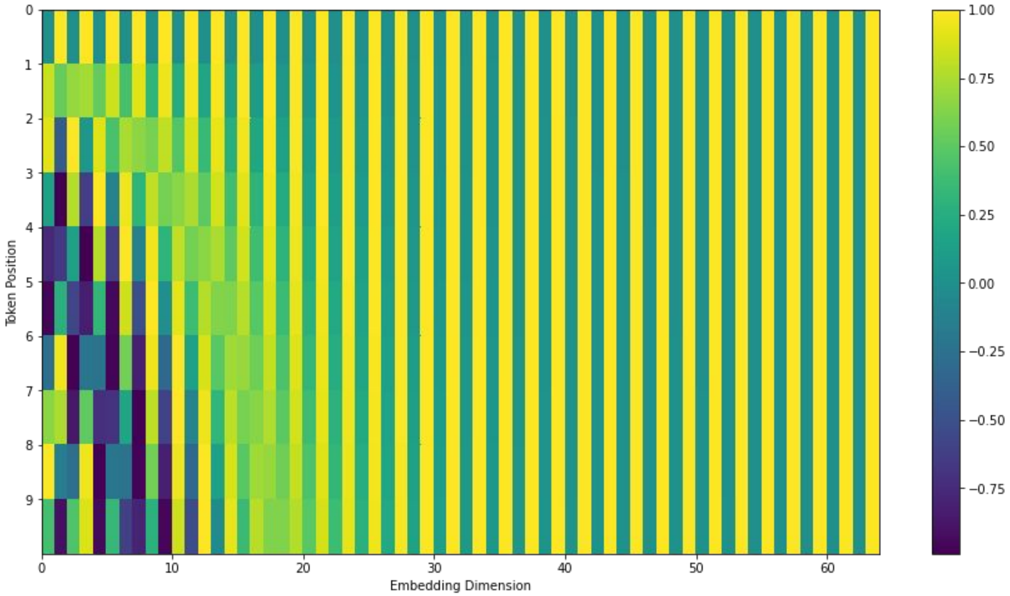
Input Embedding Matrix (Learned from the data)

Positional Embedding Matrix (Pre-defined and Fixed)

Positional Embeddings

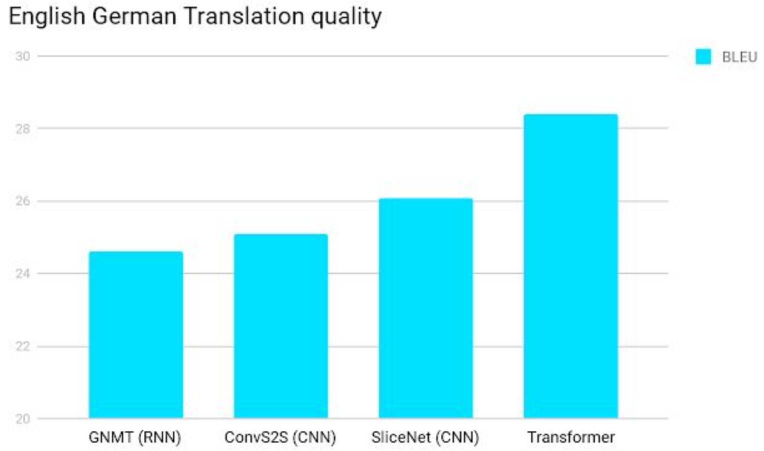
The equation in the original paper:

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

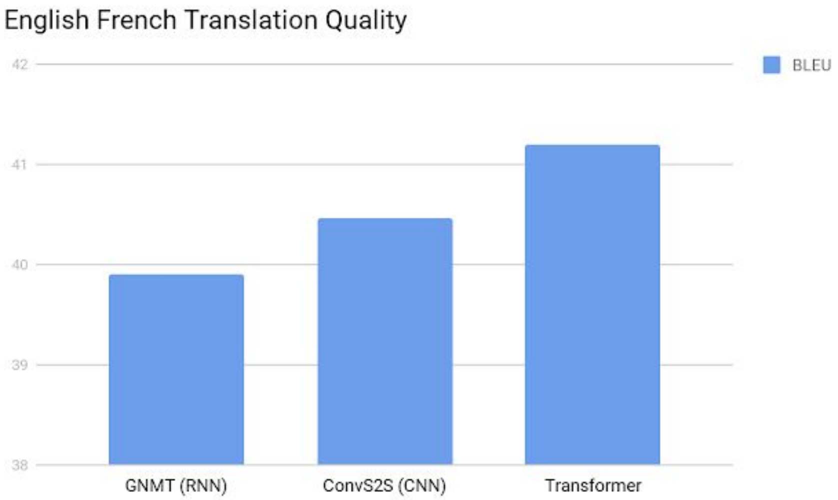


More details:
https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Better Feature Extractor for Text

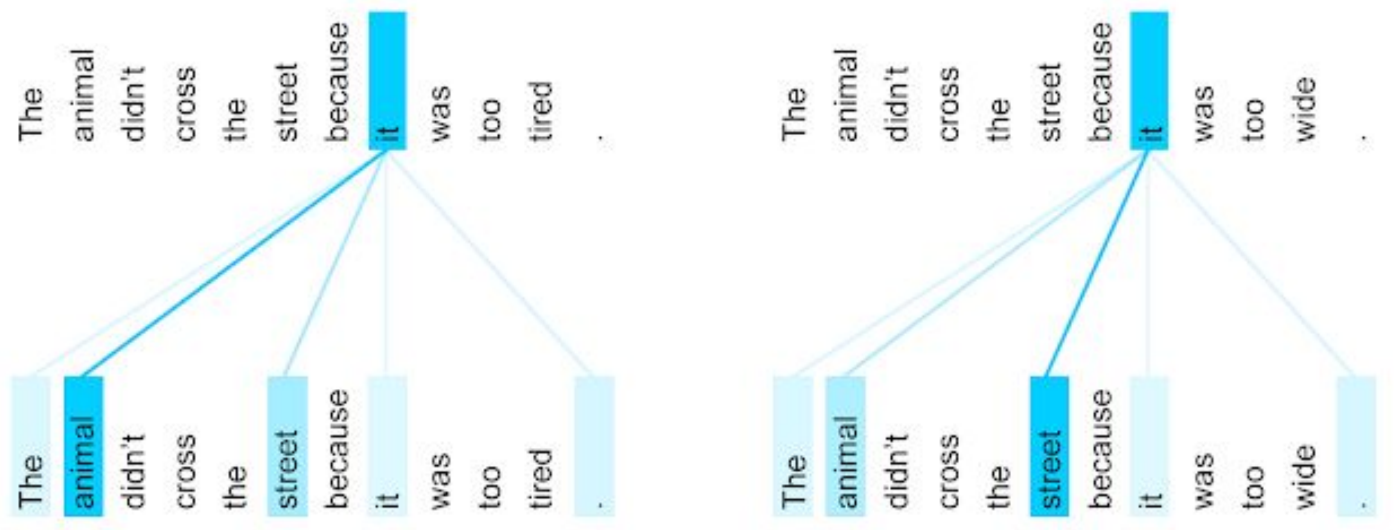


BLEU scores (higher is better) of single models on the standard WMT newstest2014 English to German translation benchmark.



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

Attention Visualization



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).

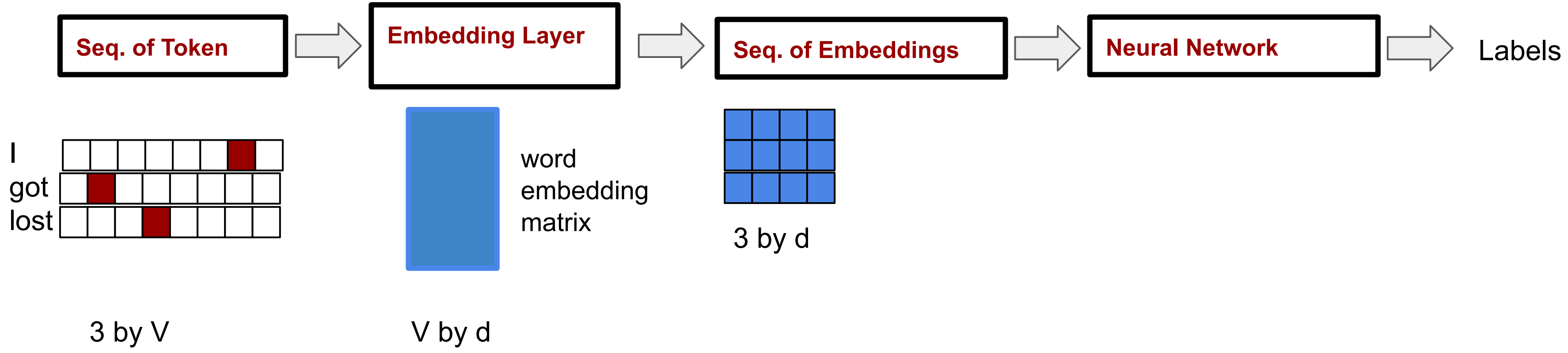
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

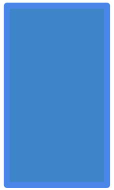
Resources for Transformer

1. https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html
2. http://jalammar.github.io/illustrated-transformer/
3. https://nlp.seas.harvard.edu/2018/04/03/attention.html
4. https://github.com/jessevig/bertviz#attention-head-view
5. https://arxiv.org/abs/1706.03762

BERT

Neural Networks for NLP





Each word has a fixed vector

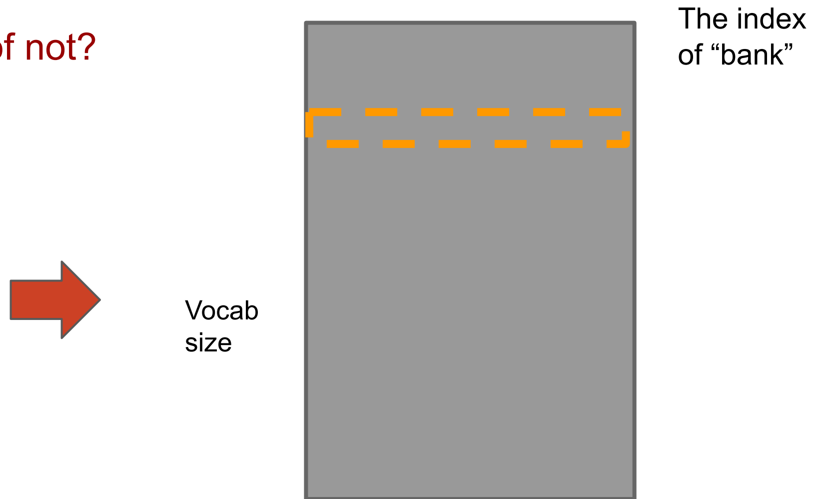
Can not address multi-sense problem!

Multiple Senses of Words

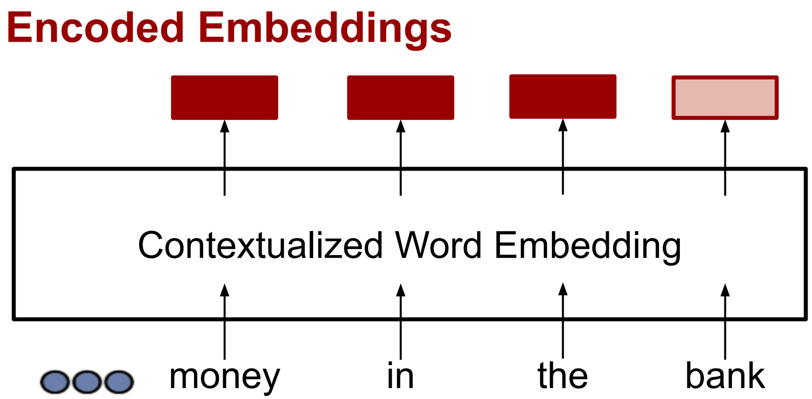
- It is safest to deposit your money in the **bank**.
- All the animals lined up along the river **bank**.
- Today, blood **banks** collect blood.

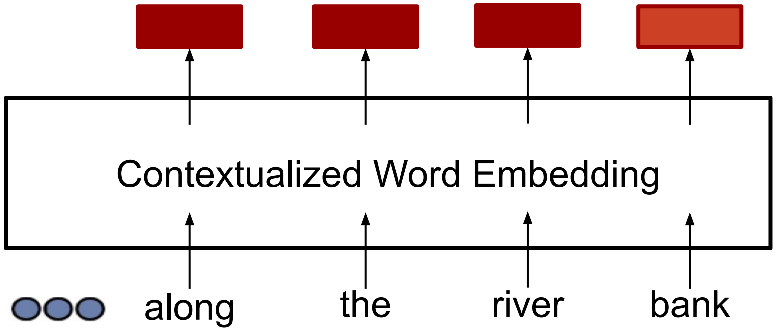
The third sense of not?

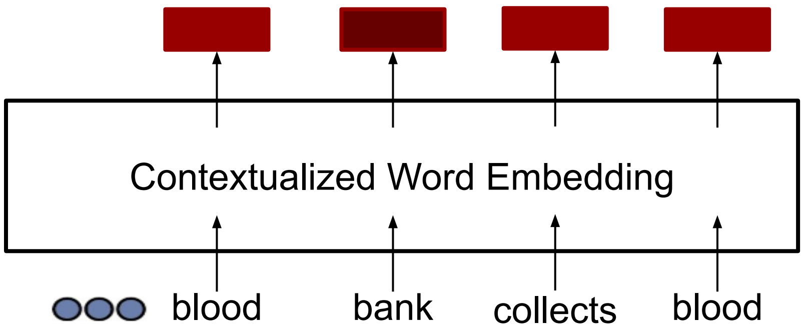
Word2Vec, Fasttext,Glove and other word embedding models



Contextualized Word Embeddings



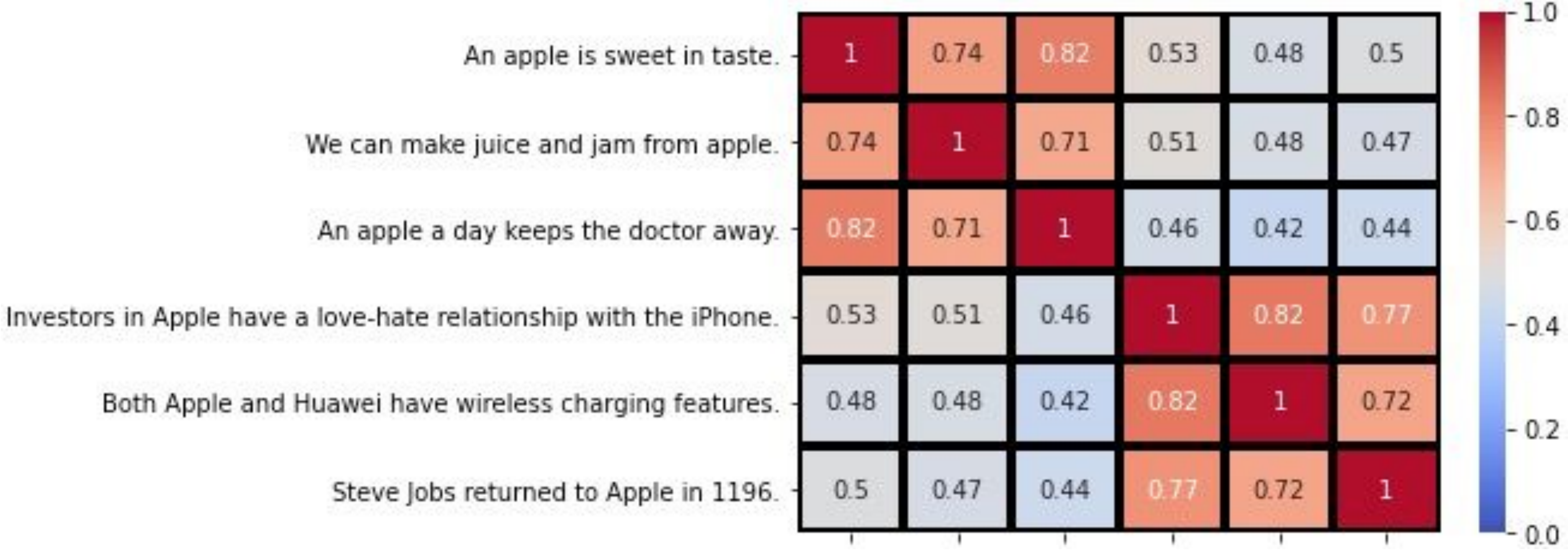




Different Contexts,
Different Encoded Embeddings for bank.

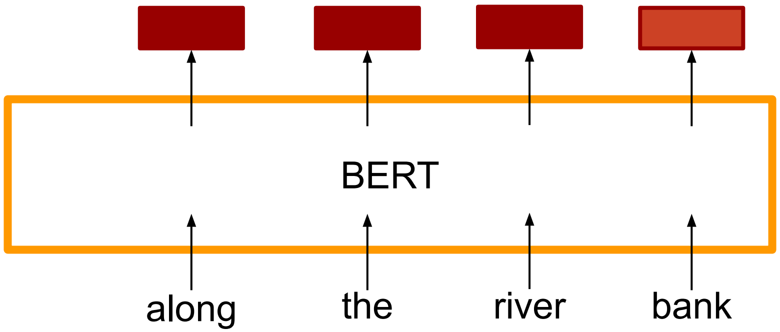
Embeddings generated from BERT

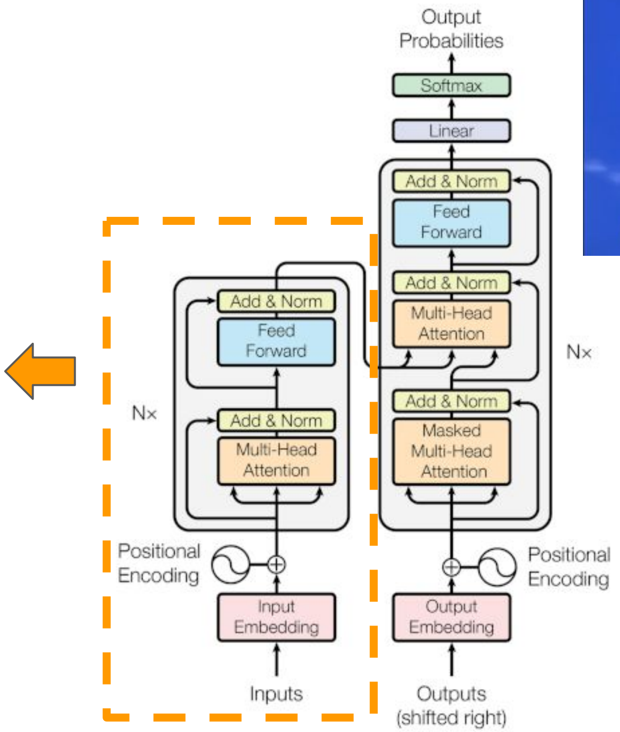
Cos-similarities among vectors of “apple” in different context



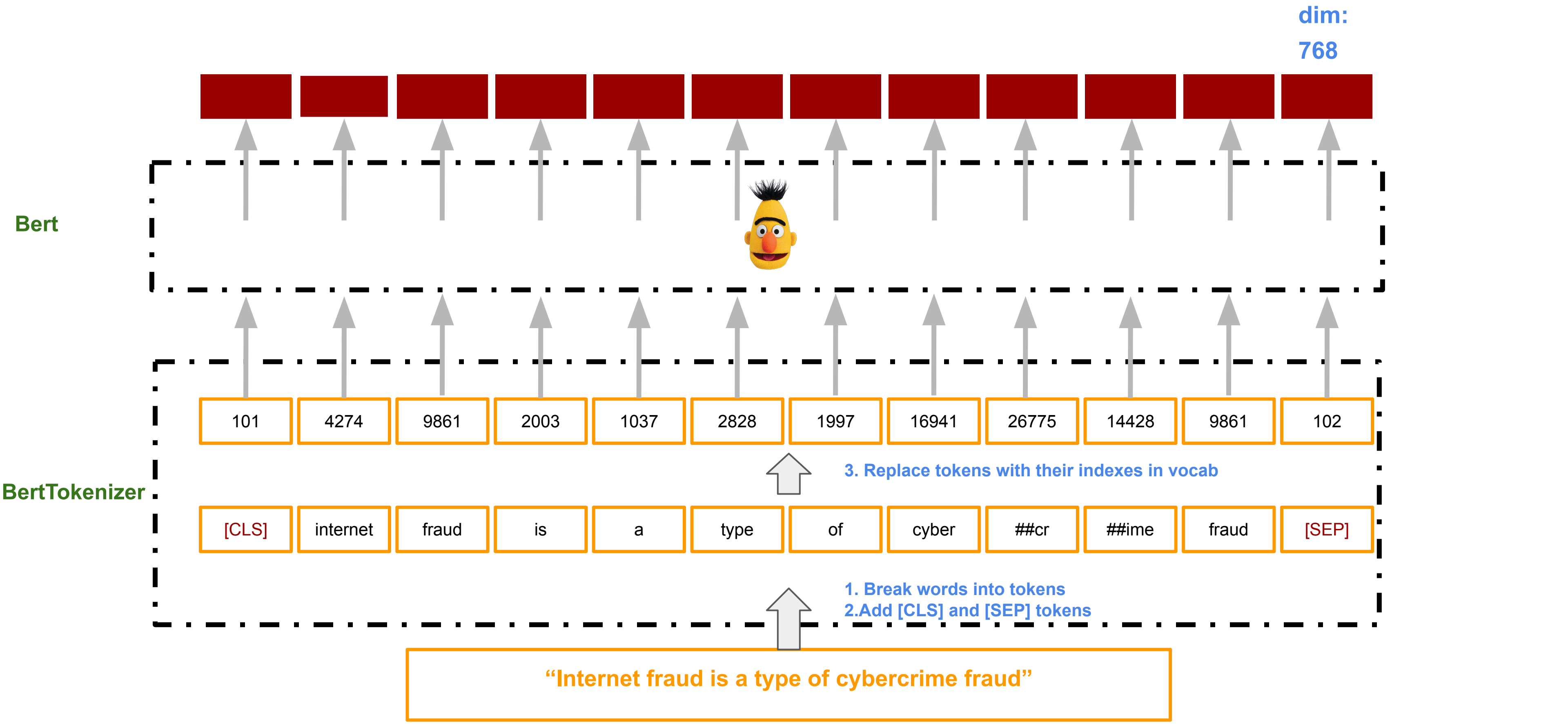
BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (**BERT**)
- BERT: Encoder of Transformer,

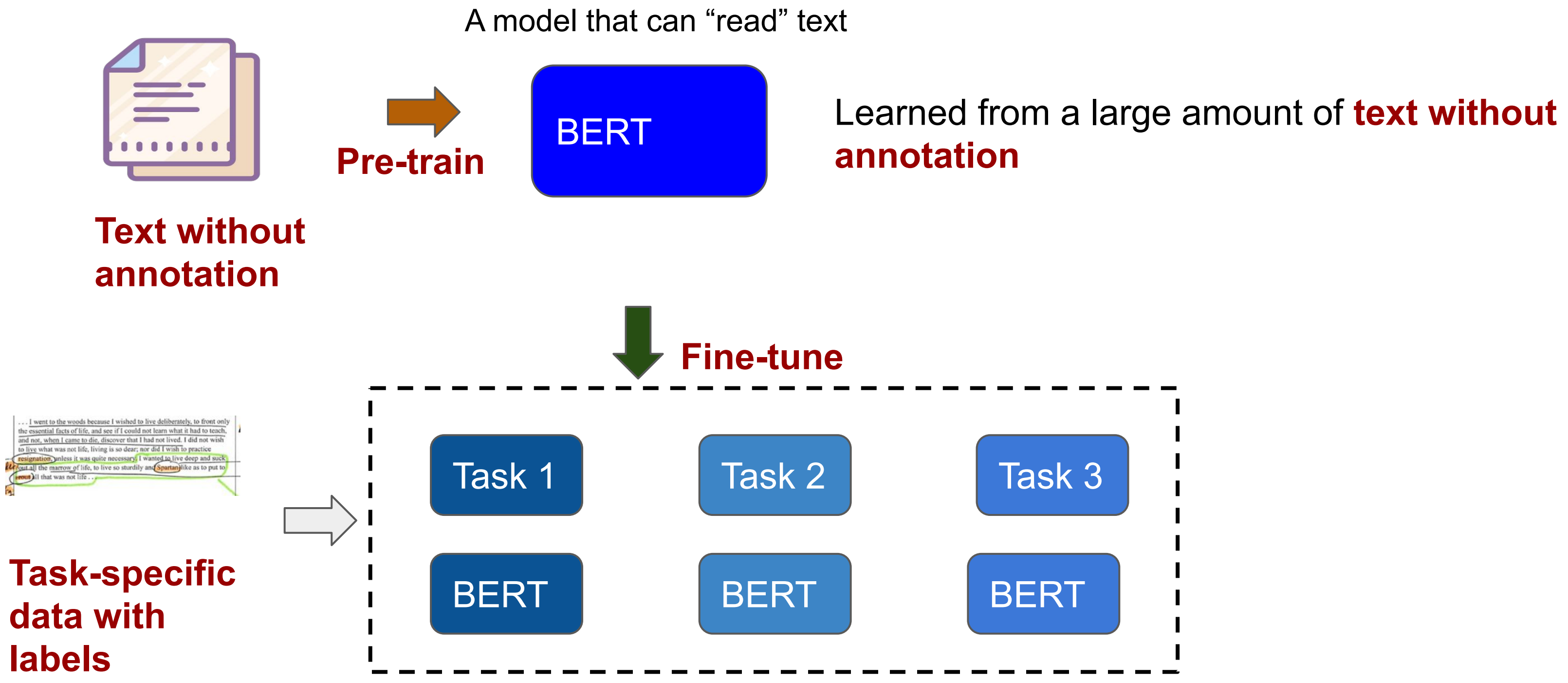




How does BERT compute

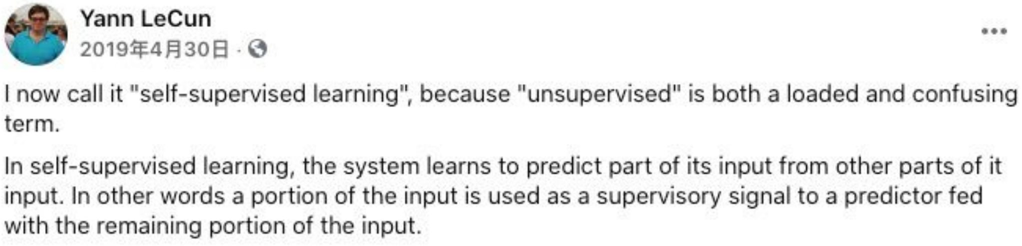


How to use BERT



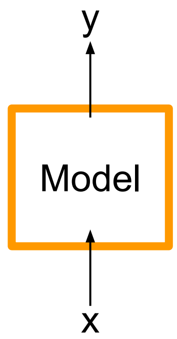
How to Pre-Train

The answer is **self-supervised learning**.

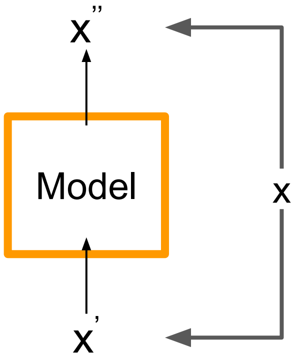

Yann LeCun
2019年4月30日 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.



Supervised



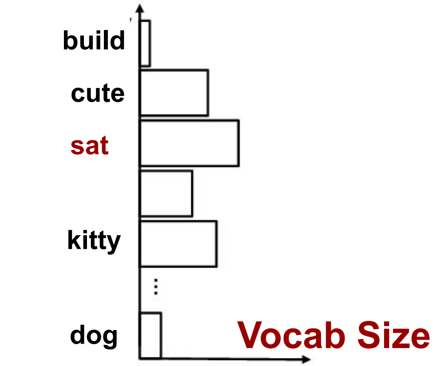
Generated by Rules

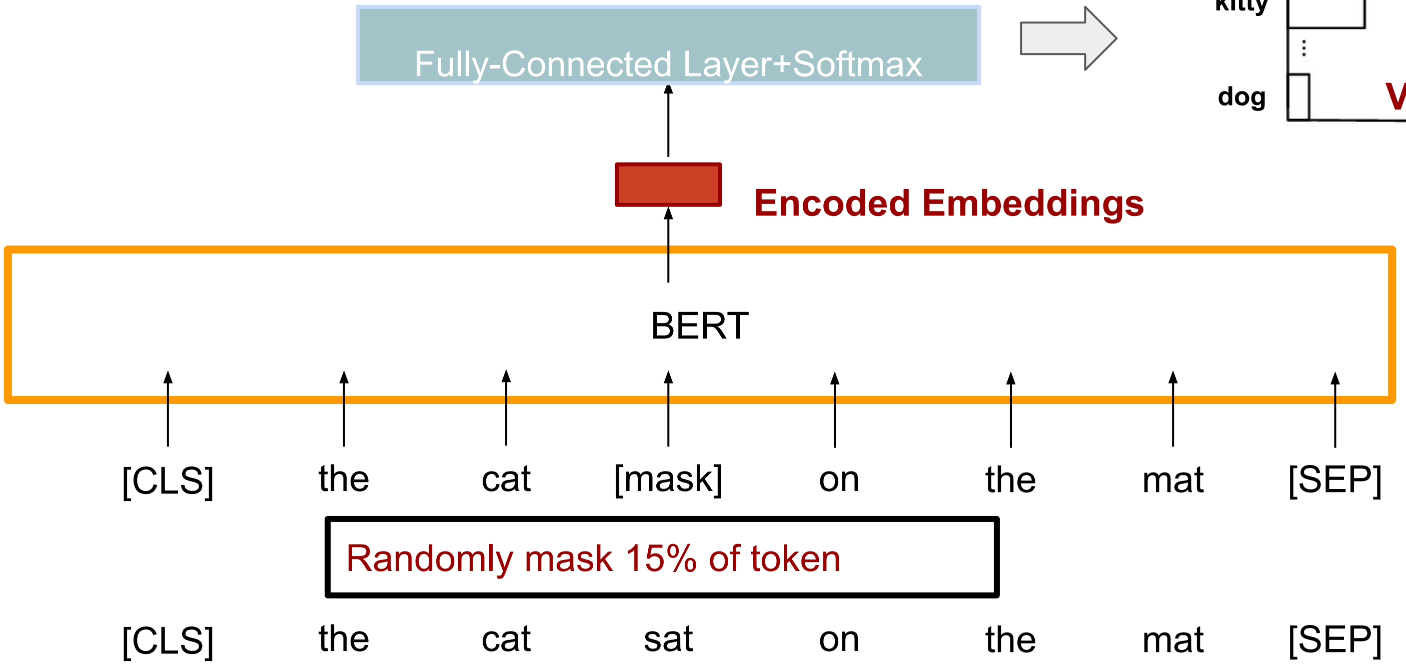
Self-Supervised

Automatically generate some kind of supervisory tasks

Pretraing Task I: MLM

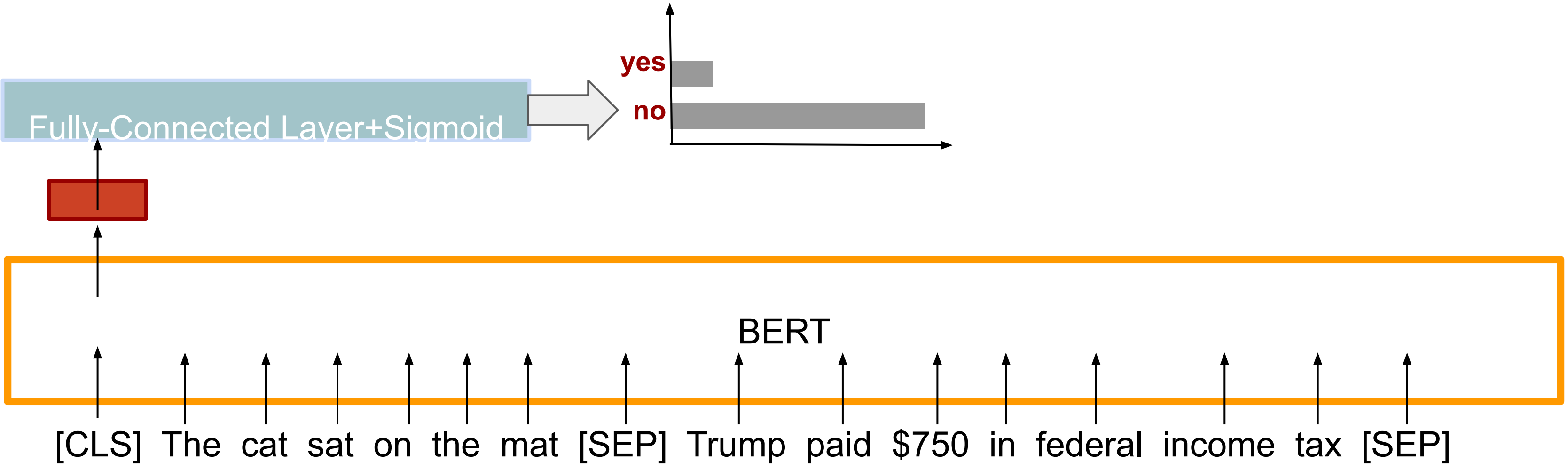
- **Masked Language Model**
  - Use the encoded embeddings of the masked word's to predict the masked word.





Pretraing Task II: NSP

- **Next sentence prediction**
    - Given two sentences A and B, is B likely to be the sentence followed by A?
    - Make bert good at handling relationships between multiple sentences



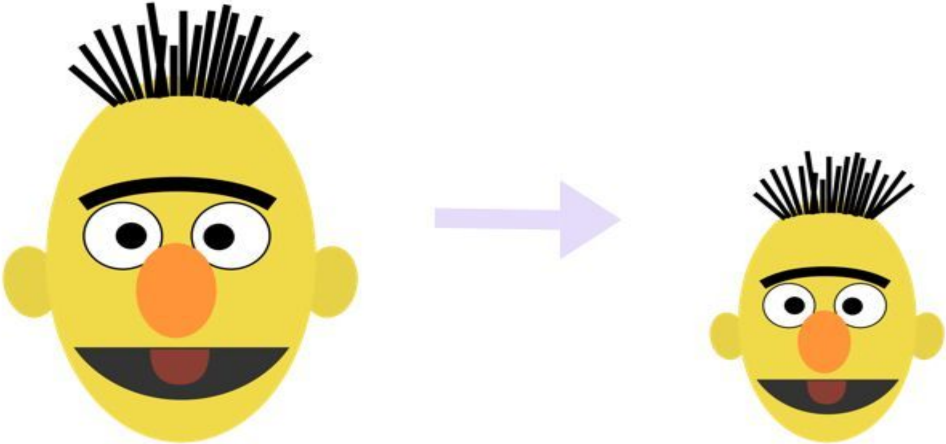
Huge Model Size

12 attention heads
110 million model parameters

Training of $\text{BERT}_{\text{BASE}}$ was performed on 4 Cloud TPUs in Pod configuration (16 TPU chips total).[13] Training of $\text{BERT}_{\text{LARGE}}$ was performed on 16 Cloud TPUs (64 TPU chips total). Each pre-training took 4 days to complete.

16 attention heads
345 million model parameters

Smaller Model



Published as a conference paper at ICLR 2020

ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS

Zhenzhong Lan[1] Mingda Chen[2*] Sebastian Goodman[1] Kevin Gimpel[2]
Piyush Sharma[1] Radu Soricut[1]

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

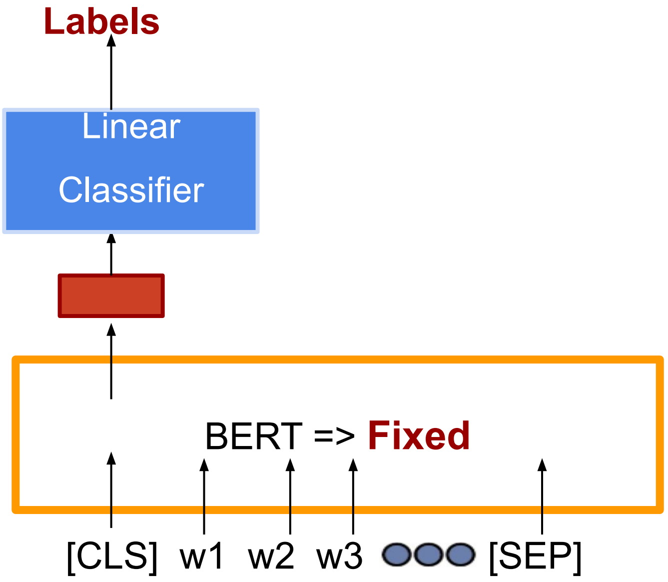
Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
{victor,lysandre,julien,thomas}@huggingface.co

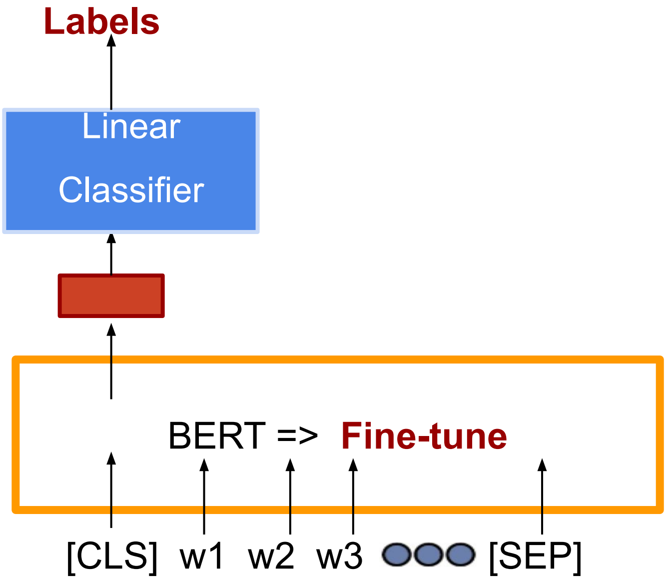
Abstract

Good summary:
http://mitchgordon.me/machine/learning/2019/11/18/all-the-ways-to-compress-BERT.html

BERT Usage I

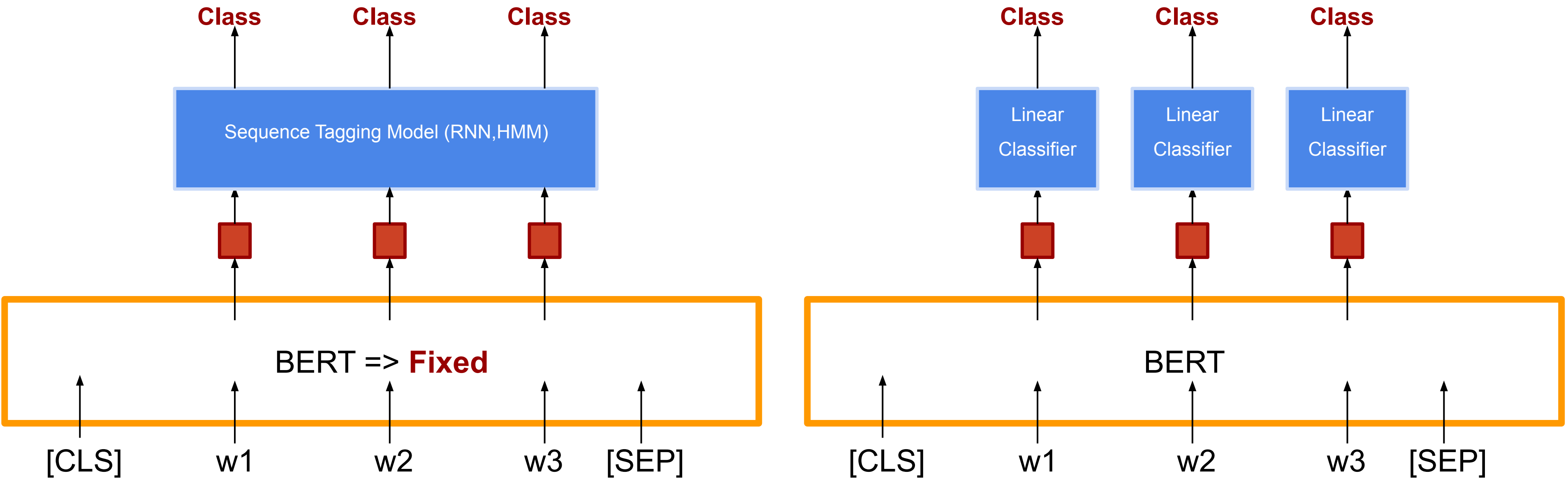
- **Input: Single Sentence** **Output: Class**
  - Sentiment Analysis
  - Document Classification





BERT Usage II

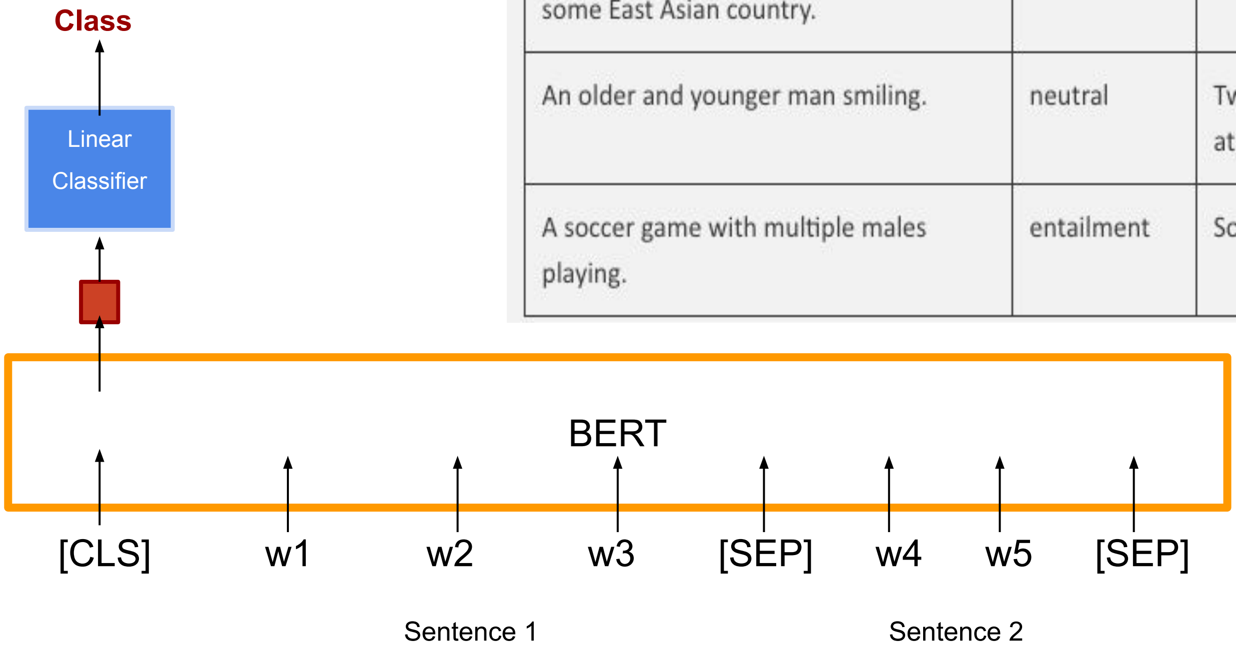
- **Input: Single Sentence Output: Class per each token**
  - NER, POS Tagging



BERT Usage III

- **Input: Two Sentences Output: Class**
  - Natural Language Inference

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.



BERT Usage IV

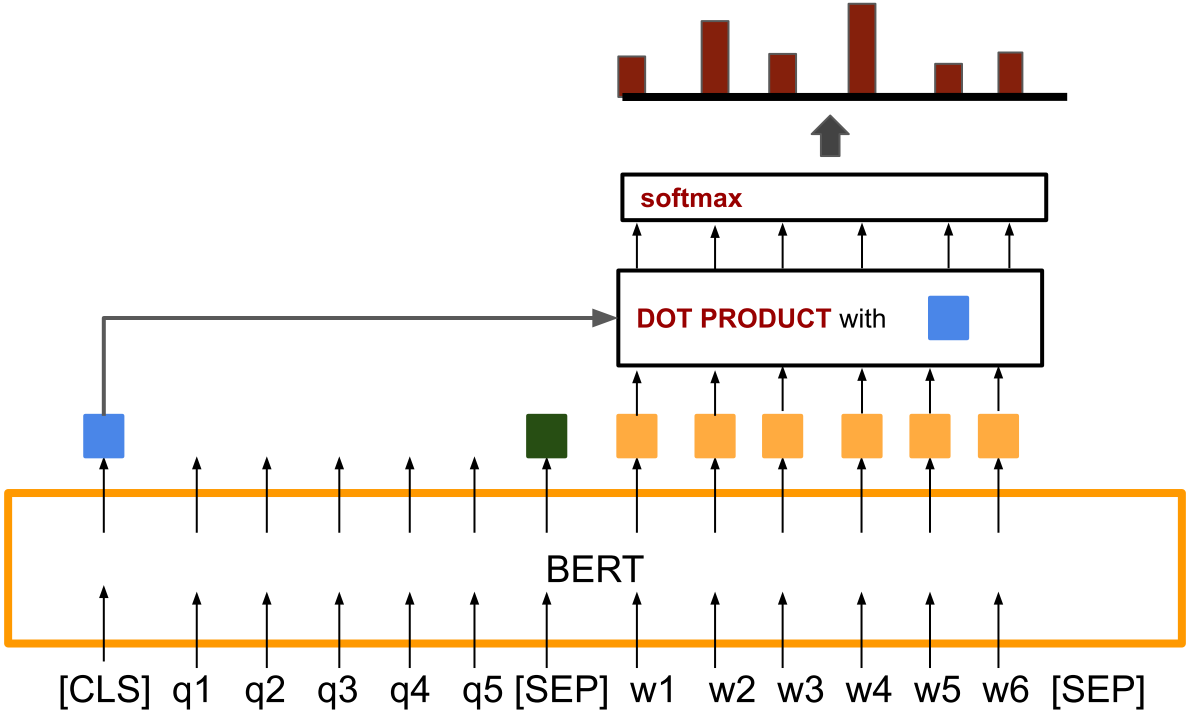
- **Extraction-based Question Answering (SQuAD):**
  - **Input: two "sentences" (Question and Reference Text)**
  - **Output: start and end positions in Reference (Answer)**

Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

BERT Usage IV

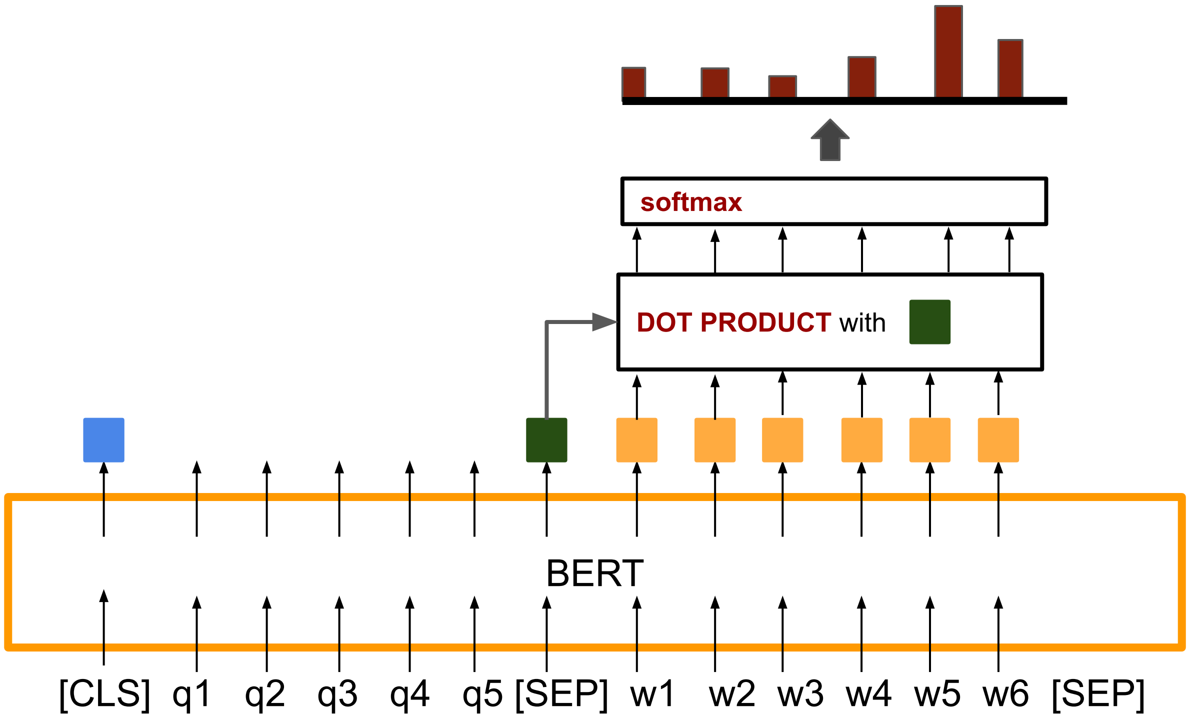
- **Extraction-based Question Answering (SQuAD):**
  - **Question {q1,q2,q3,q4,q5} Reference Text{w1,w2,w3,w4,w5,w6}**



The starting position for answer in reference is 4

BERT Usage IV

- **Extraction-based Question Answering (SQuAD):**
  - **Question {q1,q2,q3,q4,q5} Reference Text{w1,w2,w3,w4,w5,w6}**



The starting position for answer in reference is 4

The ending position for answer in reference is 5

The answer is w4w5

Superior Performance of BERT

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 20, 2019	BERT + DAE + AoA (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	**87.147**	**89.474**
2 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) *Layer 6 AI*	86.730	89.286
3 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	86.673	89.147
4 May 21, 2019	XLNet (single model) *XLNet Team*	86.346	89.133
5 Apr 13, 2019	SemBERT(ensemble) *Shanghai Jiao Tong University*	86.166	88.886

1 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	**90.724**	**93.011**
2 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
2 Apr 05, 2020	Retro-Reader (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.578	92.978
3 Jul 31, 2020	ATRLP+PV (ensemble) *Hithink RoyalFlush*	90.442	92.877
3 May 04, 2020	ELECTRA+ALBERT+EntitySpanFocus (ensemble) *SRCB_DML*	90.442	92.839
4 Jun 21, 2020	ELECTRA+ALBERT+EntitySpanFocus (ensemble) *SRCB_DML*	90.420	92.799

Superior Performance of BERT

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

System	Dev F1	Test F1
ELMo+BiLSTM+CRF	95.7	92.2
CVT+Multi (Clark et al., 2018)	-	92.6
$\text{BERT}_{\text{BASE}}$	96.4	92.4
$\text{BERT}_{\text{LARGE}}$	**96.6**	**92.8**

Table 3: CoNLL-2003 Named Entity Recognition results. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. Test results were scored against the hidden labels by the SWAG authors. †Human performance is measure with 100 samples, as reported in the SWAG paper.

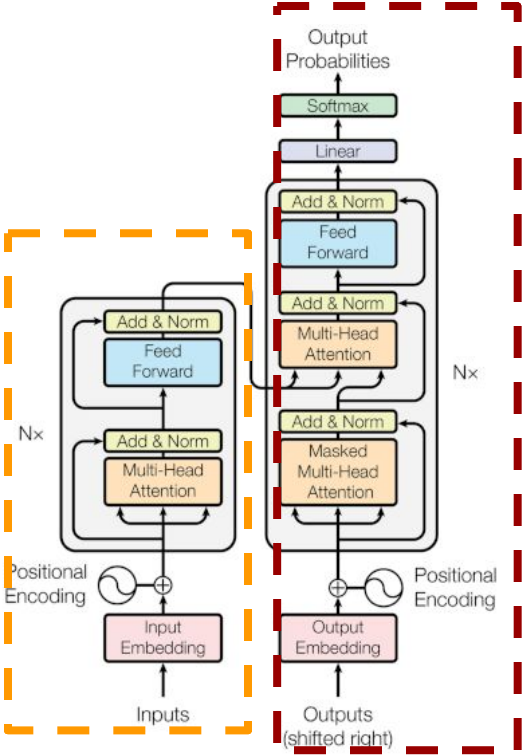
System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

BERT, GPT and Transformers

BERT

Encoder
Maksed Language Model



Transformer

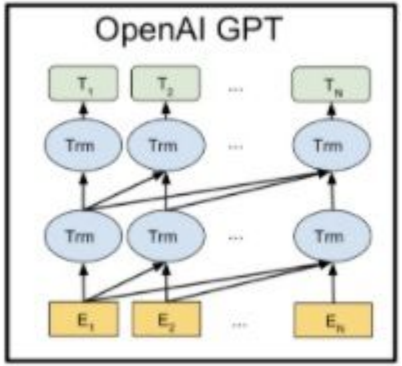
GPT2
GPT3

OpenAI

GPT-3, an autoregressive language model with 175 billion parameters

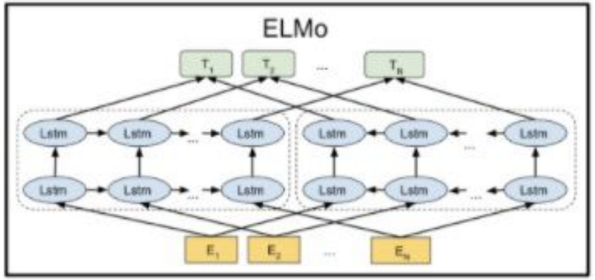
Decoder
Language Model

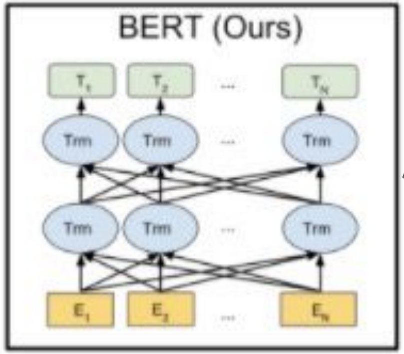
BERT vs Other Pre-trained Models



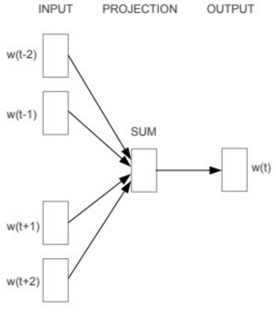
From
single-directional to
Bi-directional

From RNNs to
transformers





From FCNs to
transformers



CBOW

https://lilianweng.github.io/lil-log/2019/01/31/generalized-language-models.html

Where Next

What can we learn from BERT?

- Fully utilize the large-scale unannotated NLP data
- Embrace transformers (self-attention) instead of RNN and CNN
- Two stages in the NLP model development:
  - Large-scale pretraining
  - **Specific-task fine-tuning**

In the Future

- In the next few years, BERT will be used in almost all NLP applications (GPT2/3 may be more suitable for generative NLP tasks)
    - Build the specific NLP applications on pre-trained NLP models.

- Can we find a better feature extraction model than transformers?

- Can we find a better pre-training task than MLM and NSP?