Carnegie Mellon University

Intermediate Deep Learning for Engineers

Spring 2025, Deep Learning for Engineers
April 1, 2025, 7th Session

Amir Barati Farimani
Associate Professor of Mechanical Engineering and Bio-Engineering
Carnegie Mellon University

Transformers

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

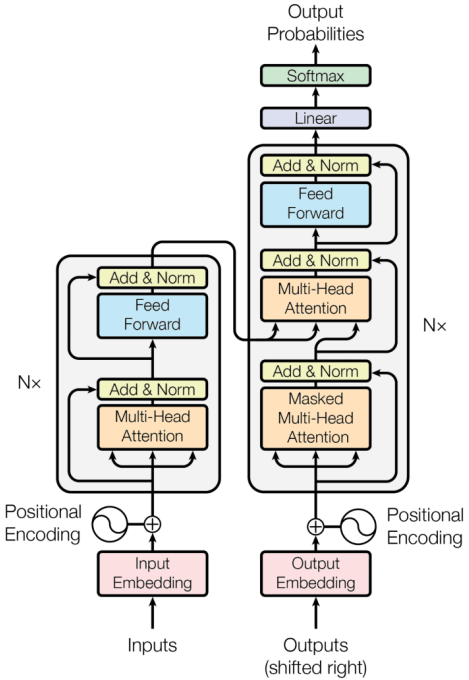
Jakob Uszkoreit* Google Research usz@google.com

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

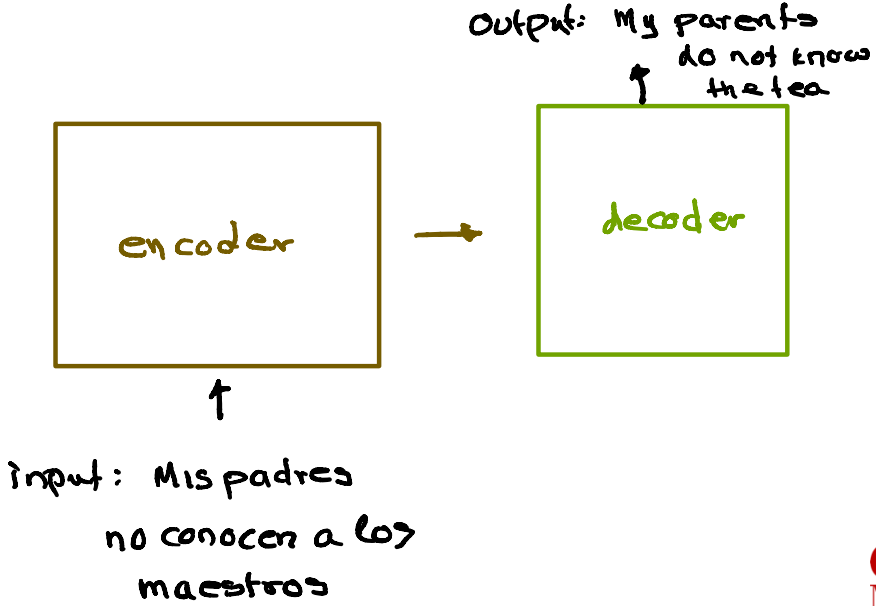


Carnegie Mellon University

Transformers

The Transformer is a model that uses attention to boost the speed with which seq2seq with attention models can be trained. The biggest benefit, however, comes from how The Transformer lends itself to **parallelization**. We will break it apart and look at how it functions.

In its heart it contains an encoding component, a decoding component, and connections between them.

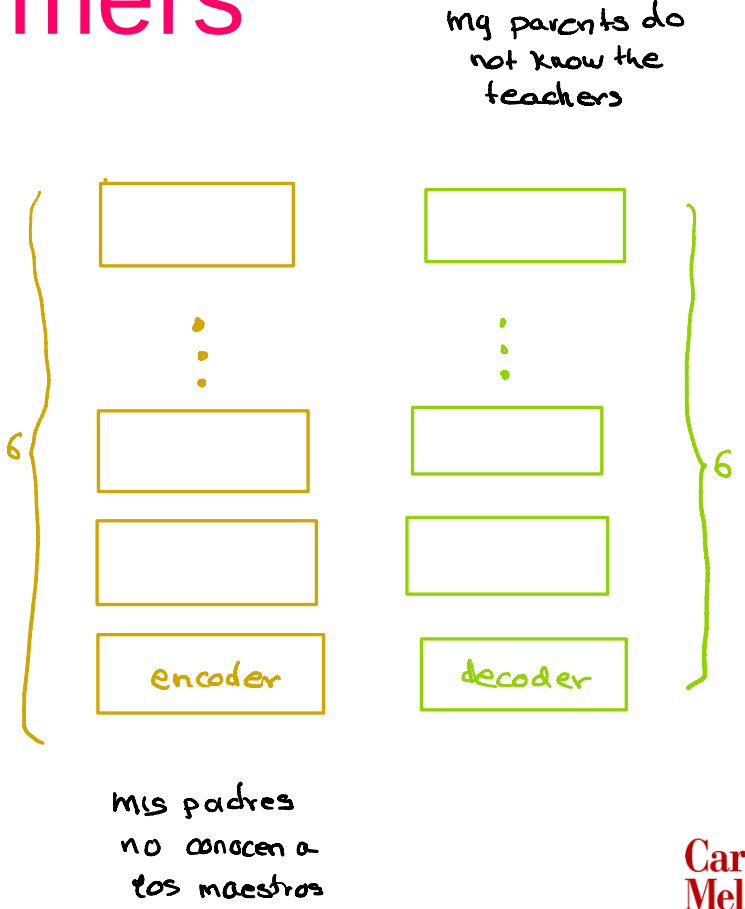


Transformers

The encoding is a stack of encoders.

The original paper stacks six of them on top of each other – there's nothing magical about the number six, one can definitely experiment with other arrangements).

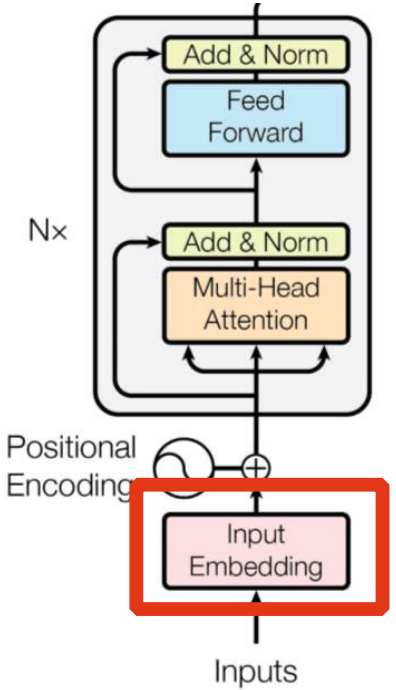
The decoding is a stack of decoders of the same number.



Transformers (Embedding)

Input embedding of words based on dictionary:

Original sentence (tokens)	YOUR	CAT	IS	A	LOVELY	CAT
Input IDs (position in the vocabulary)	105	6587	5475	3578	65	6587
Embedding (vector of size 512)	952.207	171.411	621.659	776.562	6422.693	171.411
	5450.840	3276.350	1304.051	5567.288	6315.080	3276.350
	1853.448	9192.819	0.565	58.942	9358.778	9192.819
	...	...	...	...	...	...
	1.658	3633.421	7679.805	2716.194	2141.081	3633.421
	2671.529	8390.473	4506.025	5119.949	735.147	8390.473



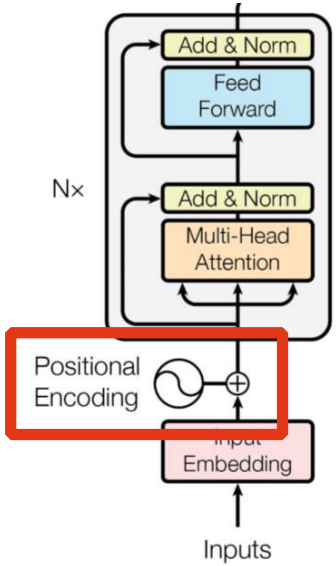
We define d_{model} = 512, which represents the size of the embedding vector of each word

Carnegie Mellon University

Transformers (Pos Encoding)

Positional Encoding

- We want each word to carry some information about its position in the sentence.
- We want the model to treat words that appear close to each other as “close” and words that are distant as “distant”.
- We want the positional encoding to represent a pattern that can be learned by the model.



Transformers (Embed+Pos)

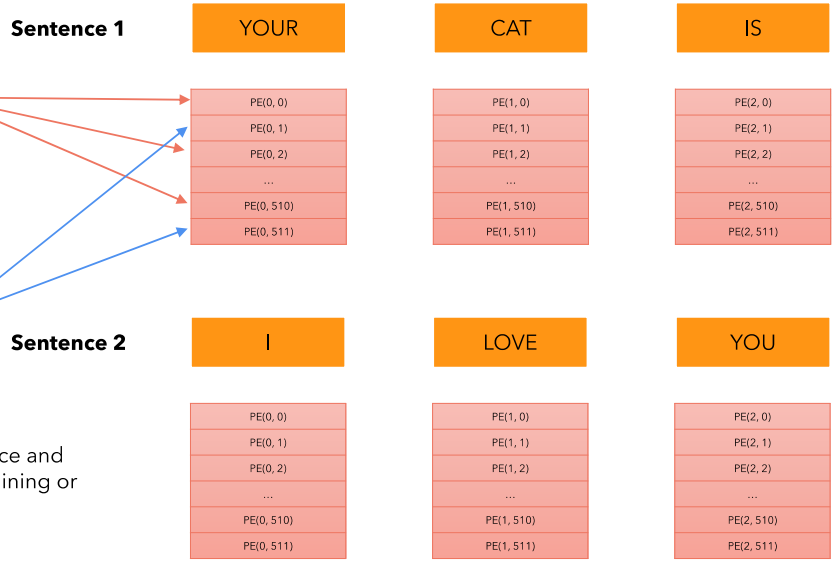
Adding the positional encoding to the embedding

Transformers (Positional Encoding)

What is positional encoding and how is it implemented

$$PE(pos, 2i) = \sin \frac{pos}{10000^{\frac{2i}{d_{model}}}}$$

$$PE(pos, 2i+1) = \cos \frac{pos}{10000^{\frac{2i}{d_{model}}}}$$



We only need to compute the positional encodings once and then reuse them for every sentence, no matter if it is training or inference.

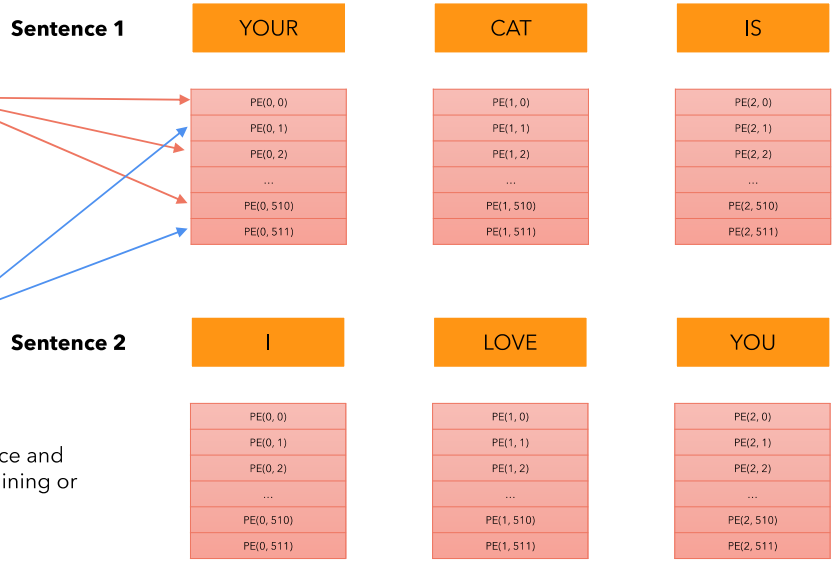
Carnegie Mellon University

Transformers (Positional Encoding)

What is positional encoding and how is it implemented

$$PE(pos, 2i) = \sin \frac{pos}{10000^{\frac{2i}{d_{model}}}}$$

$$PE(pos, 2i+1) = \cos \frac{pos}{10000^{\frac{2i}{d_{model}}}}$$



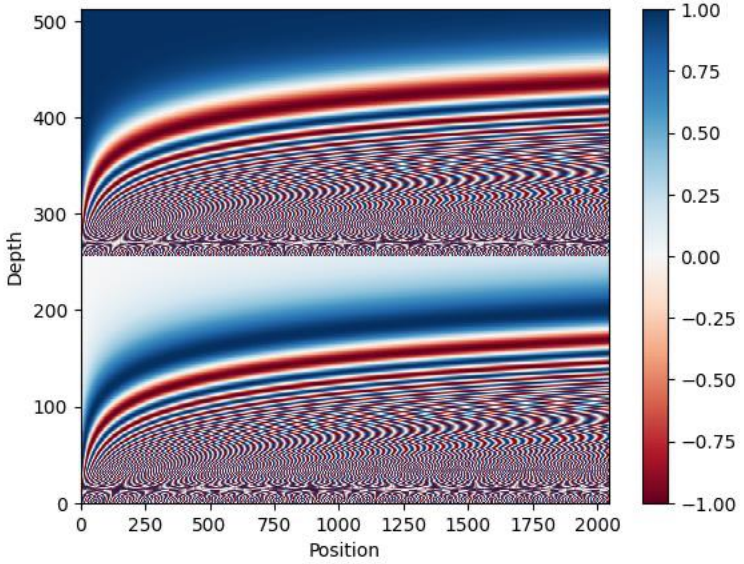
We only need to compute the positional encodings once and then reuse them for every sentence, no matter if it is training or inference.

Carnegie Mellon University

Transformers (Positional Encoding)

What is positional encoding and how is it implemented

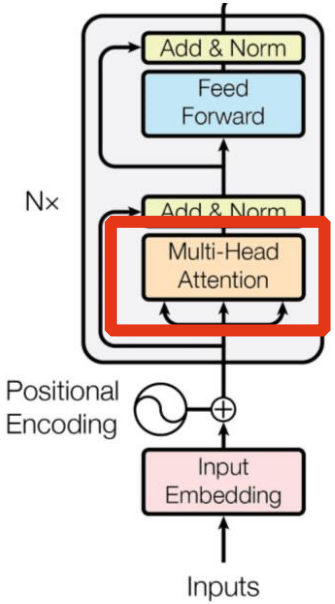
Trigonometric functions like cos and sin naturally represent a pattern that the model can recognize as continuous, so relative positions are easier to see for the model. By watching the plot of these functions, we can also see a regular pattern, so we can hypothesize that the model will see it too.



Carnegie Mellon University

Transformers (Multi-head Attention)

First we need to learn Self attention



Self Attention

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

Self-Attention allows the model to relate words to each other.
In this simple case we consider the sequence length seq = 6 and $d_{model} = d_k = 512$. The matrices Q, K and V are just the input sentence.

$$\text{softmax}\left(\frac{Q \text{ (6, 512)} \times K^T \text{ (512, 6)}}{\sqrt{512}}\right) =$$

	YOUR	CAT	IS	A	LOVELY	CAT	Σ
YOUR	0.268	0.119	0.134	0.148	0.179	0.152	1
CAT	0.124	0.278	0.201	0.128	0.154	0.115	1
IS	0.147	0.132	0.262	0.097	0.218	0.145	1
A	0.210	0.128	0.206	0.212	0.119	0.125	1
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174	1
CAT	0.195	0.114	0.203	0.103	0.157	0.229	1

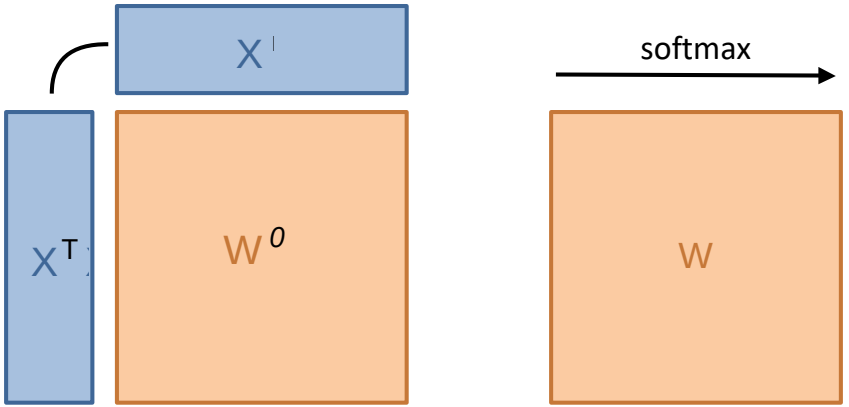
(6, 6)

* all values are random.

Carnegie Mellon University

Self Attention

VECTORIZED



$W^0 = X^T X$

$W = \text{softmax}(W^0)$

$$w_{ij}^0 = x_i^T x_j$$

$$w_{ij} = \frac{\exp w_{ij}^0}{\sum_j \exp w_{ij}^0}$$

Self Attention

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

	YOUR	CAT	IS	A	LOVELY	CAT
YOUR	0.268	0.119	0.134	0.148	0.179	0.152
CAT	0.124	0.278	0.201	0.128	0.154	0.115
IS	0.147	0.132	0.262	0.097	0.218	0.145
A	0.210	0.128	0.206	0.212	0.119	0.125
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174
CAT	0.195	0.114	0.203	0.103	0.157	0.229

(6, 6)

X

V

(6, 512)

=

Attention

(6, 512)

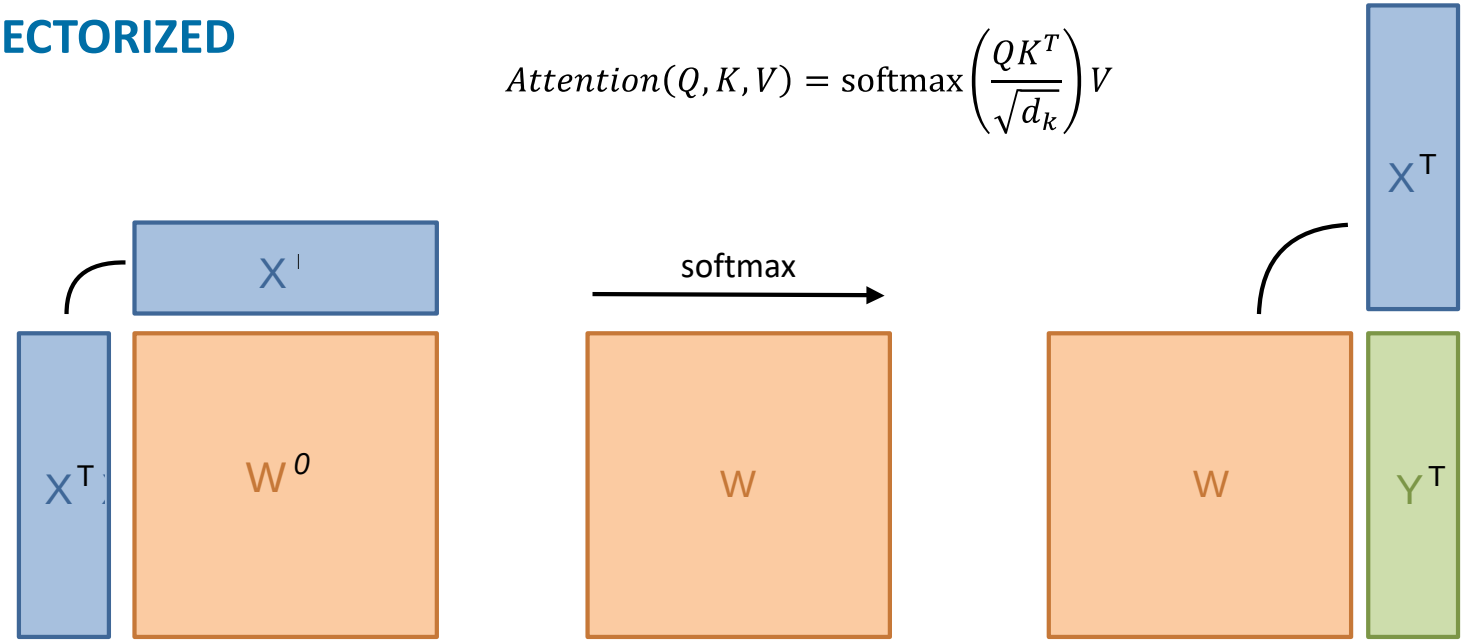
Each row in this matrix captures not only the meaning (given by the embedding) or the position in the sentence (represented by the positional encodings) but also each word's interaction with other words.

Carnegie Mellon University

Self Attention

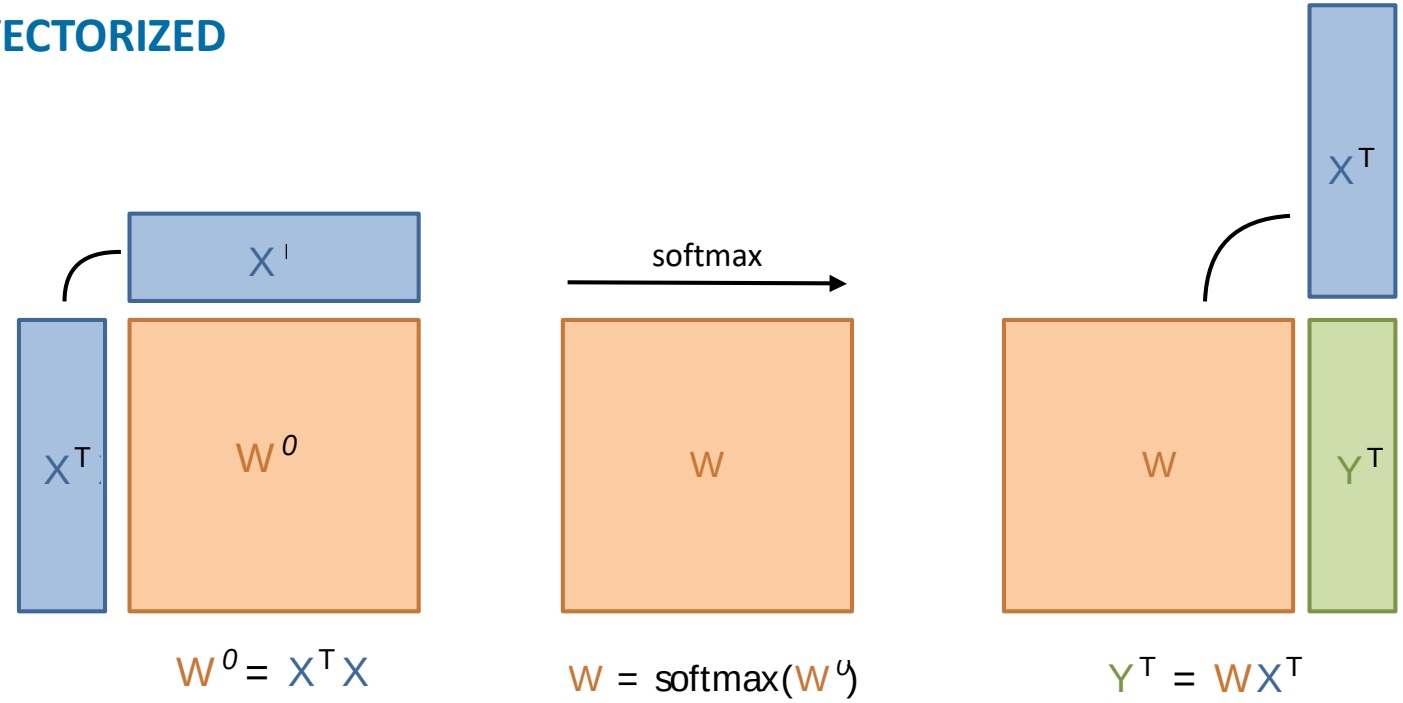
VECTORIZED

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



Self Attention

VECTORIZED



Self Attention

- Self-Attention is permutation invariant.

- Self-Attention requires no parameters. Up to now the interaction between words has been driven by their embedding and the positional encodings. This will change later.

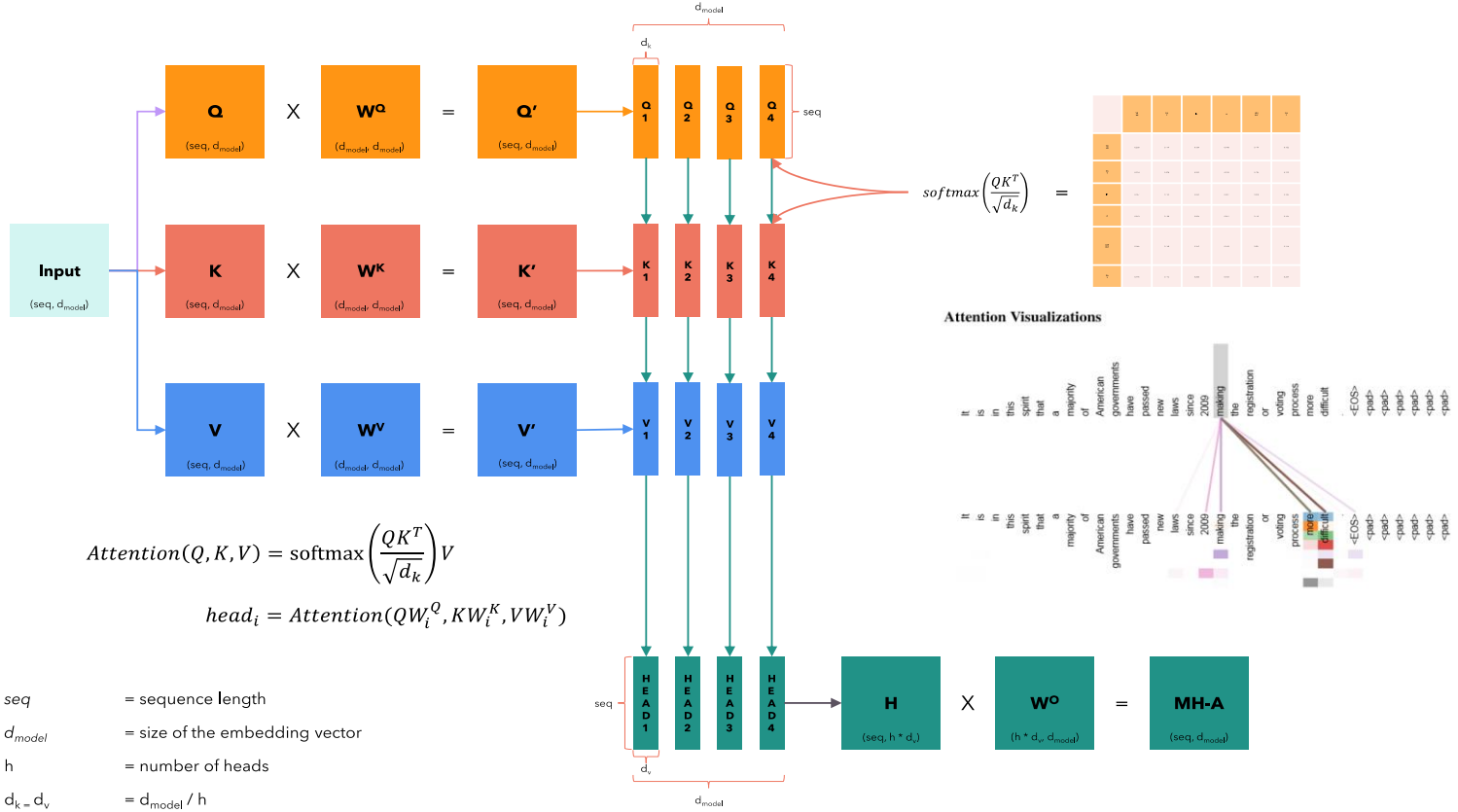
- We expect values along the diagonal to be the highest.

- If we don't want some positions to interact, we can always set their values to $-\infty$ before applying the softmax in this matrix and the model will not learn those interactions. We will use this in the decoder.

	YOUR	CAT	IS	A	LOVELY	CAT
YOUR	0.268	0.119	0.134	0.148	0.179	0.152
CAT	0.124	0.278	0.201	0.128	0.154	0.115
IS	0.147	0.132	0.262	0.097	0.218	0.145
A	0.210	0.128	0.206	0.212	0.119	0.125
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174
CAT	0.195	0.114	0.203	0.103	0.157	0.229

Carnegie Mellon University

Multi-head Attention



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

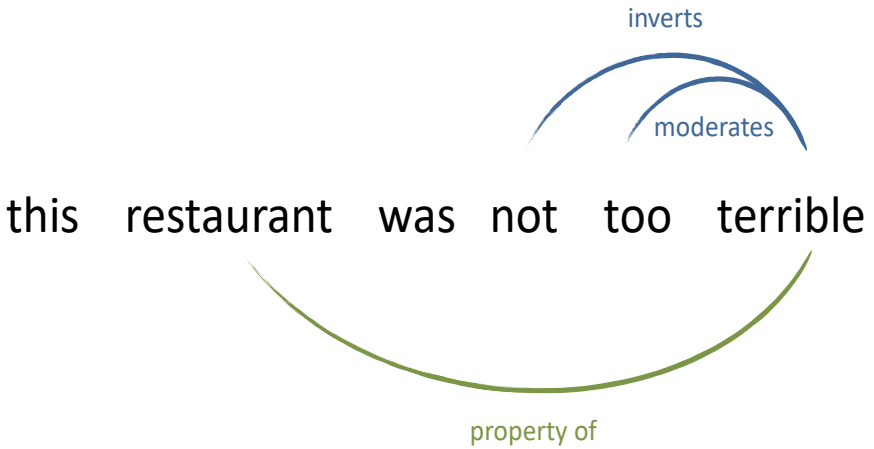
$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

seq	= sequence length
d_{model}	= size of the embedding vector
h	= number of heads
$d_k = d_v$	= d_{model} / h

Carnegie Mellon University

Attention Learns functional relation via Multihead

In many sentences, there are different relations to model. Here, the word meaning of the word “terrible” is inverted by “not” and moderated by “too”. Its relation to the word restaurant is completely different: it describes a property of the restaurant.
The idea behind multi-head self-attention is that multiple relations are best captured by different self-attention operations.



Carnegie Mellon University

Attention as a Soft Dictionary

- **Attention** is a *soft* dictionary
- key, query and value are vectors
- every key matches the query *to some extent* as determined by their dot-product
- a *mixture* of all values is returned with softmax-normalized dot products as mixture weights
- ***Self*-attention** Attention with keys, queries and values from the same set.

```
d = {'a' : 1, 'b' : 2, 'c' : 3}

d['b'] = 3
```

<- query

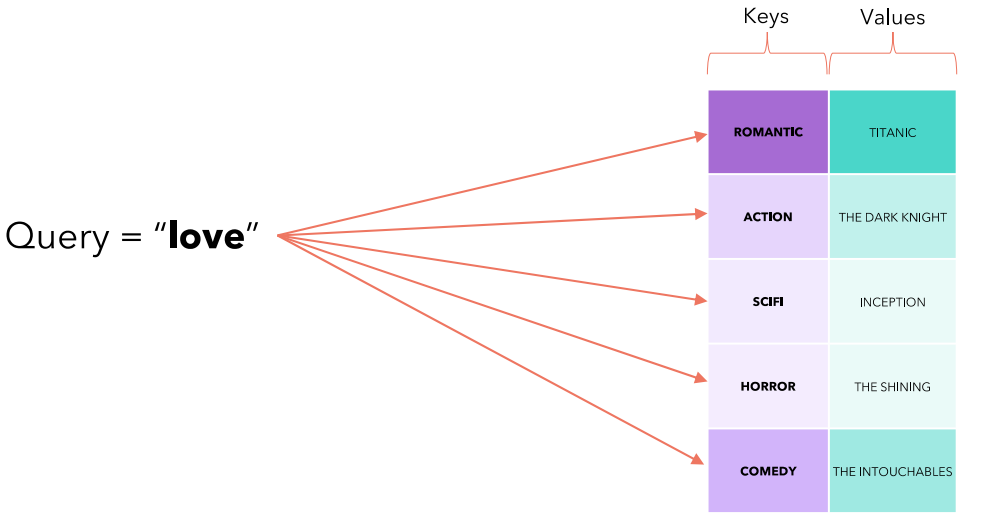
<- key

<- value

key	value
a	1
b	2
c	3

Carnegie Mellon University

K, Q, V interpretation



* this could be a Python dictionary or a database table.

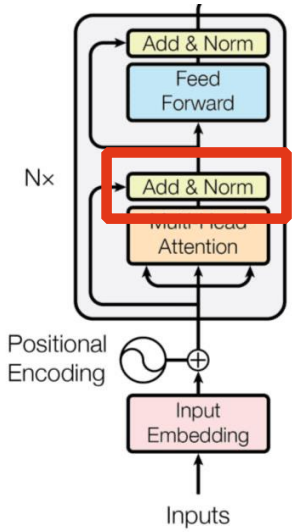
Carnegie Mellon University

Add&Norm

Batch of 3 items

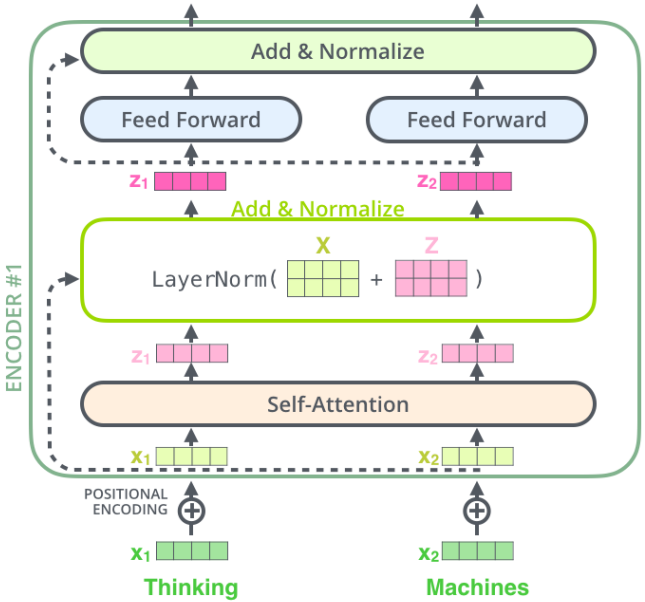
ITEM 1	ITEM 2	ITEM 3
50.147	1242.223	9.370
3314.825	688.123	4606.674
...	...	...
...	...	...
8463.361	434.944	944.705
8.021	149.442	21189.444
μ_1	μ_2	μ_3
σ_1^2	σ_2^2	σ_3^2

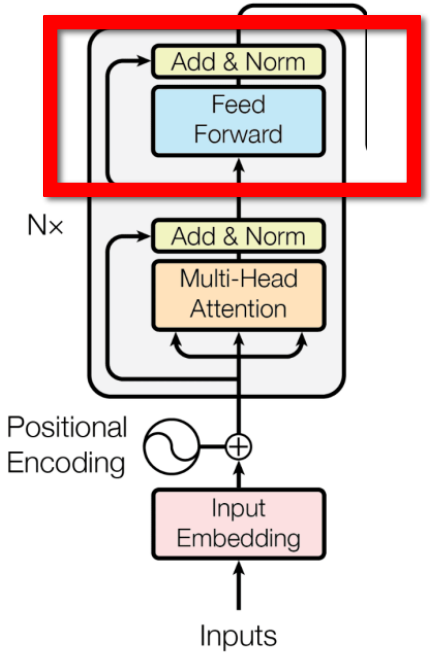
$$\hat{x}_j = \frac{x_j - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$



Carnegie Mellon University

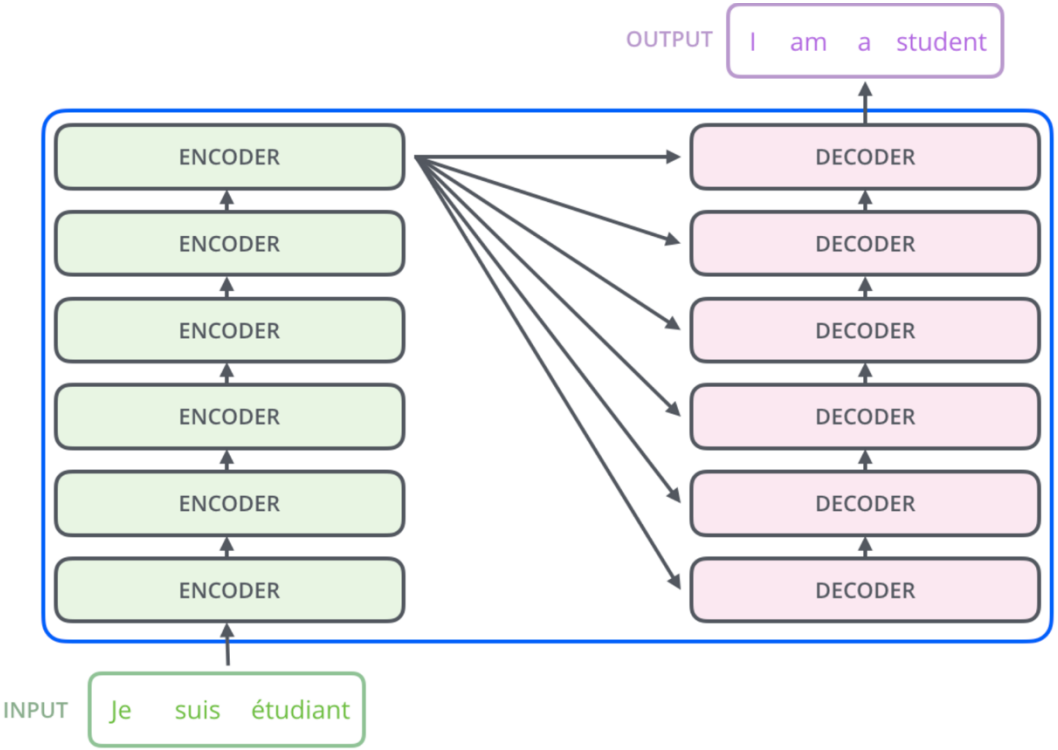
Feed Forward NN





Carnegie Mellon University

Encoders on top of each other



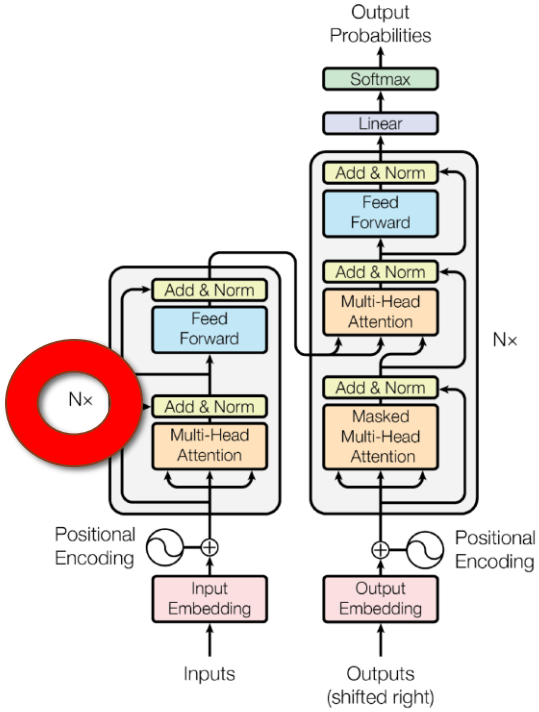


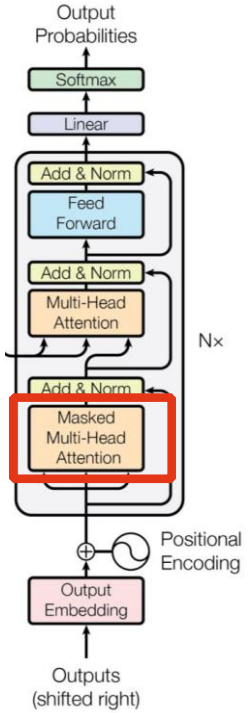
Figure 1: The Transformer - model architecture.

Carnegie Mellon University

Decoder

What is Masked Multi-Head Attention?

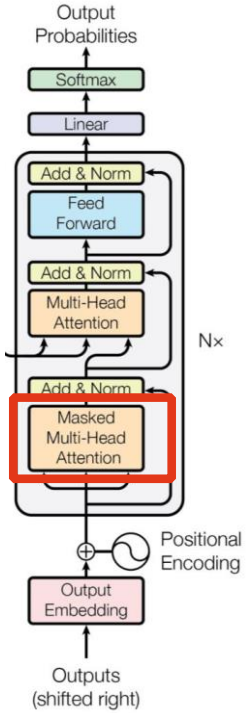
Our goal is to make the model causal: it means the output at a certain position can only depend on the words on the previous positions. The model must not be able to see future words.



Decoder

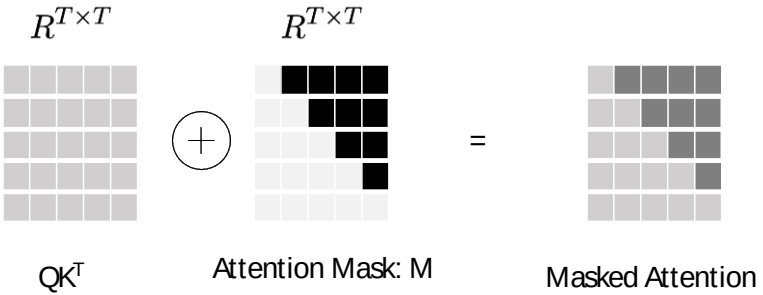
What is Masked Multi-Head Attention?

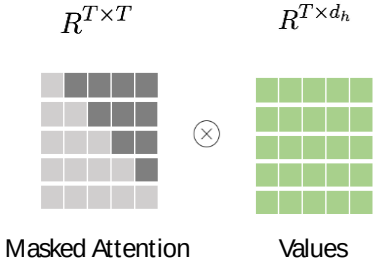
	YOUR	CAT	IS	A	LOVELY	CAT
YOUR	0.268	[illegible]	~~0.134~~	~~0.148~~	~~0.179~~	~~0.152~~
CAT	0.124	0.278	[illegible]	~~0.128~~	~~0.154~~	~~0.115~~
IS	0.147	0.132	0.262	~~0.097~~	~~0.218~~	~~0.145~~
A	0.210	0.128	0.206	0.212	~~0.119~~	~~0.125~~
LOVELY	0.146	0.158	0.152	0.143	0.227	~~0.174~~
CAT	0.195	0.114	0.203	0.103	0.157	0.229

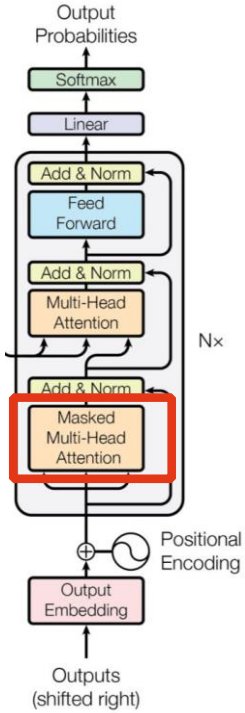


Decoder

What is Masked Multi-Head Attention?

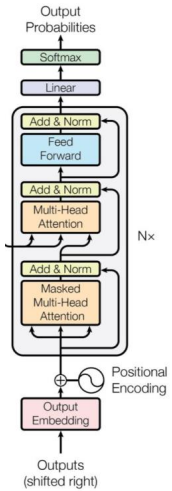


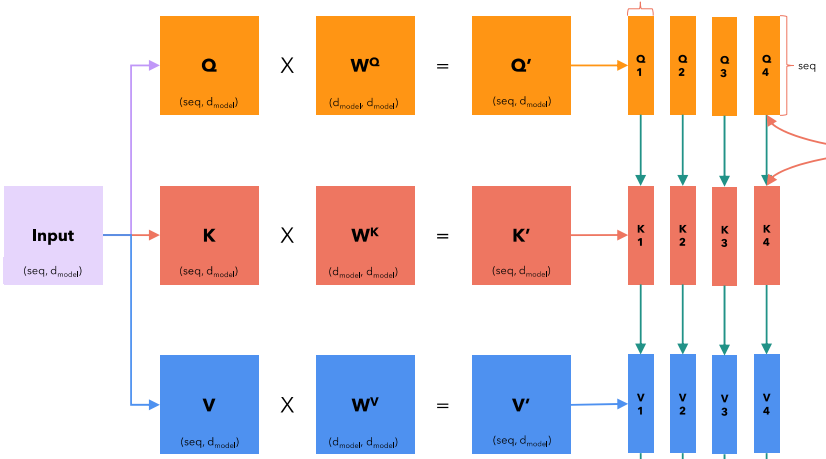


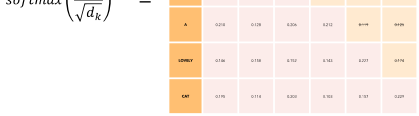


Carnegie Mellon University

Decoder



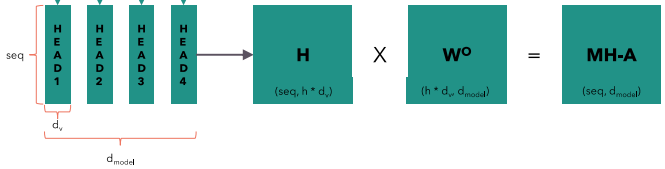




All the values above the diagonal are replace with **-∞** *before* applying the softmax, which will replace them with zero.

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

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$



$$MultiHead(Q,K,V) = Concat(head_1 \ldots head_h)W^O$$

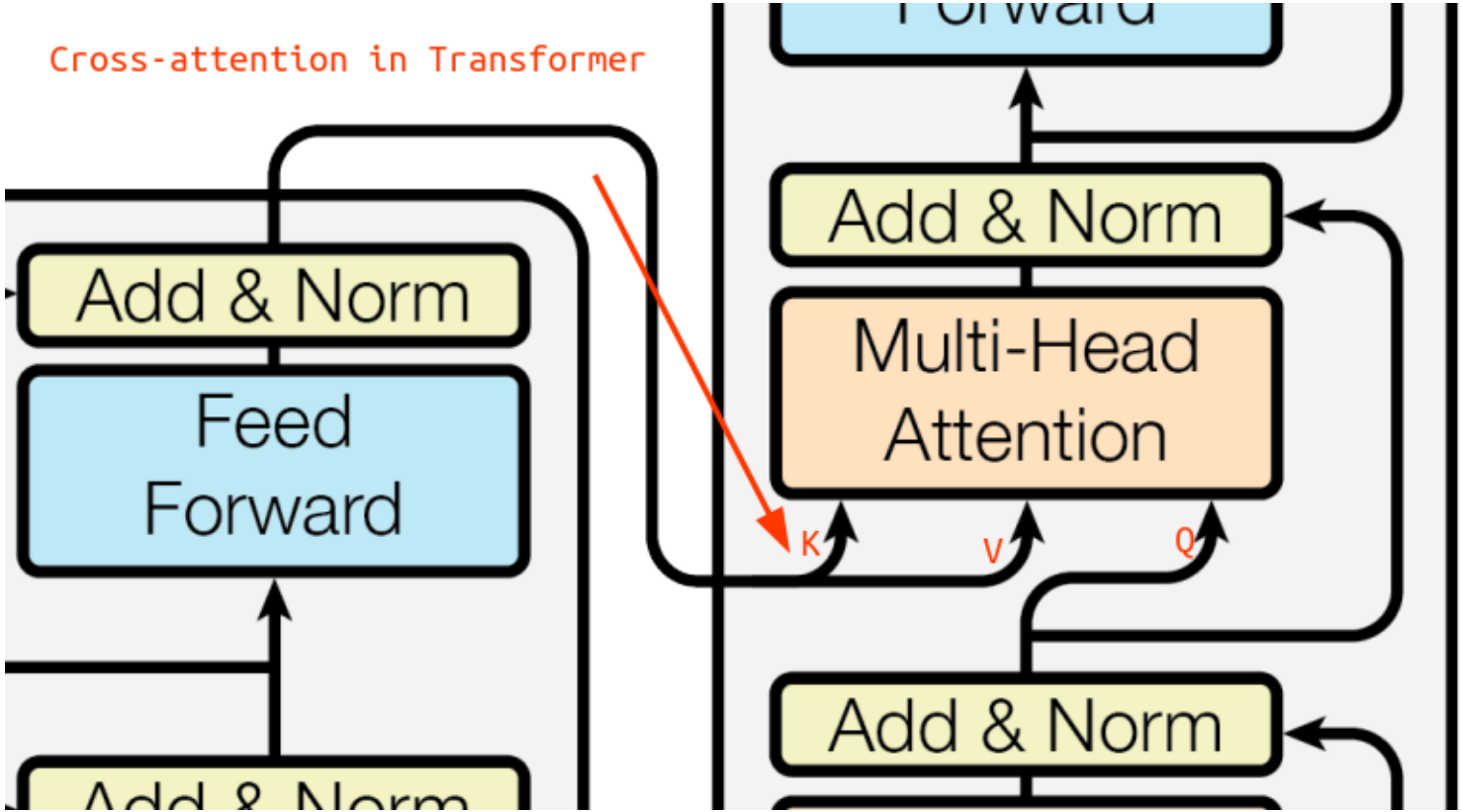
seq = sequence length

d_{model} = size of the embedding vector

h = number of heads

Carnegie Mellon University

Cross Attention



Training

I love you very much

Ti amo molto

Training

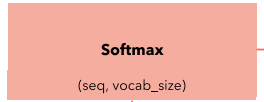
Training

Time Step = 1

It all happens in one time step!

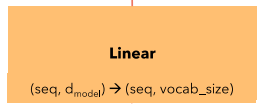
Ti amo molto <EOS>

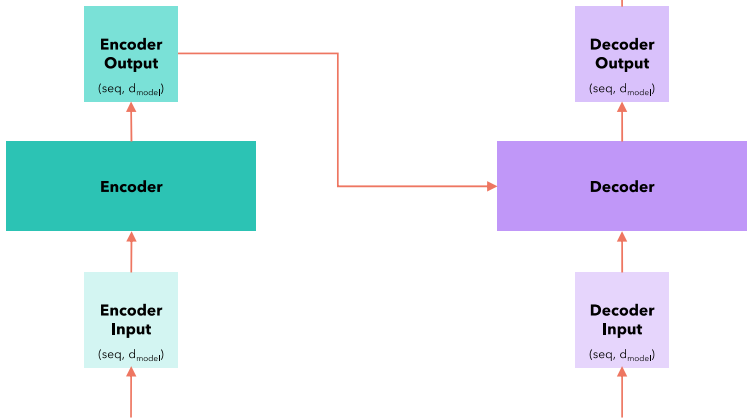
* This is called the "label" or the "target"



Cross Entropy Loss

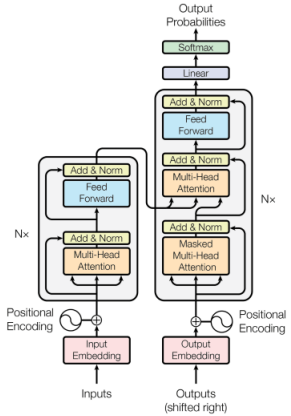
The encoder outputs, for each word a vector that not only captures its meaning (the embedding) or the position, but also its interaction with other words by means of the multi-head attention.





<SOS> I love you very much <EOS>

<SOS> Ti amo molto

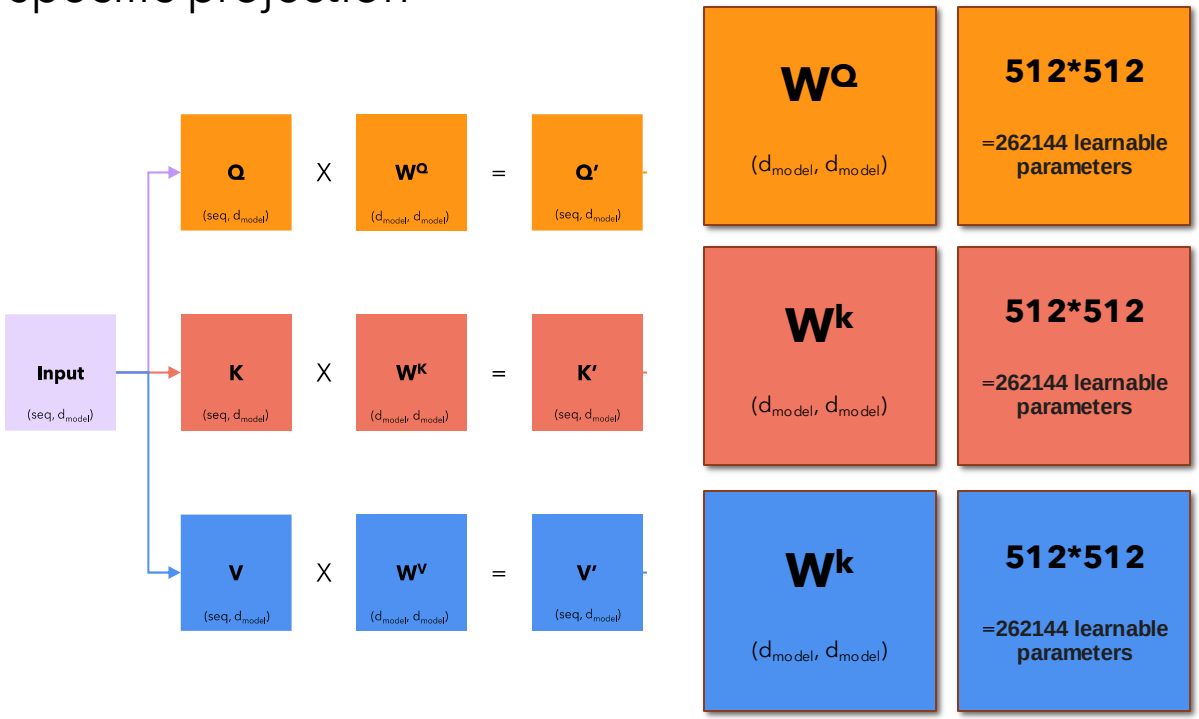


We prepend the <SOS> token at the beginning. That's why the paper says that the decoder input is shifted right.

Carnegie Mellon University

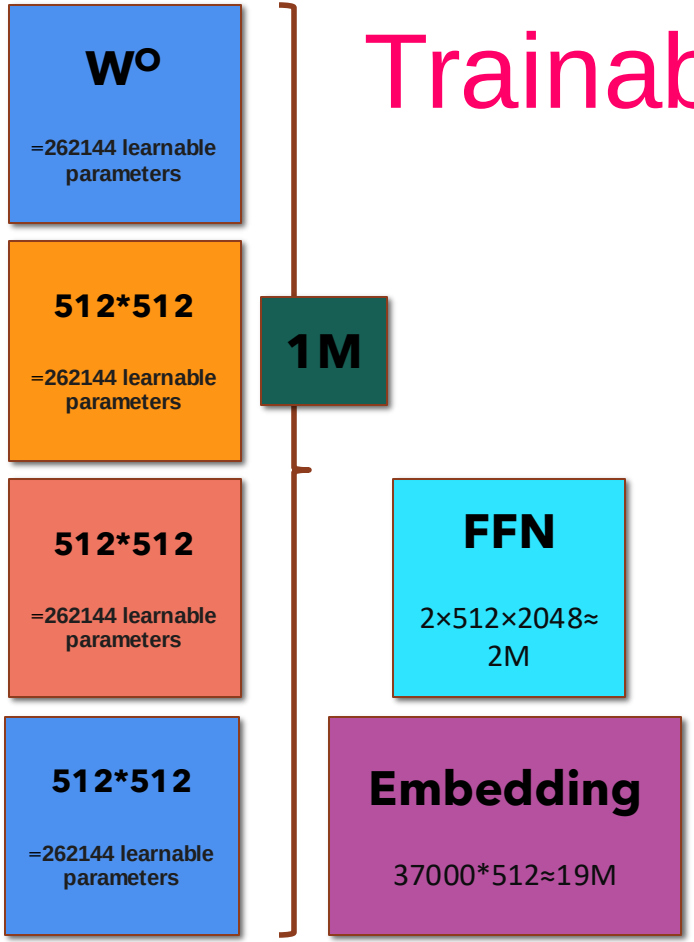
Trainable Parameters

In the multi-head setup, the keys, values, and queries are a result of a head-specific projection



Carnegie Mellon University

Trainable Parameters



Number of parameters in each multi-head attention layer:

$$N_{att} = N(W^O) + (N(W_i^Q) + N(W_i^K) + N(W_i^V)) \times h$$

$$= h \times d_v \times d_{model} + (d_{model} \times d_k + d_{model} \times d_k + d_{model} \times d_v) \times h$$

where $N(.)$ is the size (number of elements) of a matrix.

Number of parameters in each Feed Forward Network:

$$N_{FFN} = 2 \times d_{model} \times d_{ff} + d_{model} + d_{ff}$$

Therefore, the total number of parameters is:

$$2N \times N_{FFN} + (N + 2N)N_{att} + N_{voc} \times n_{model}$$

where N_{voc} (the size of the vocabulary) is 37000 according to section 5.1.

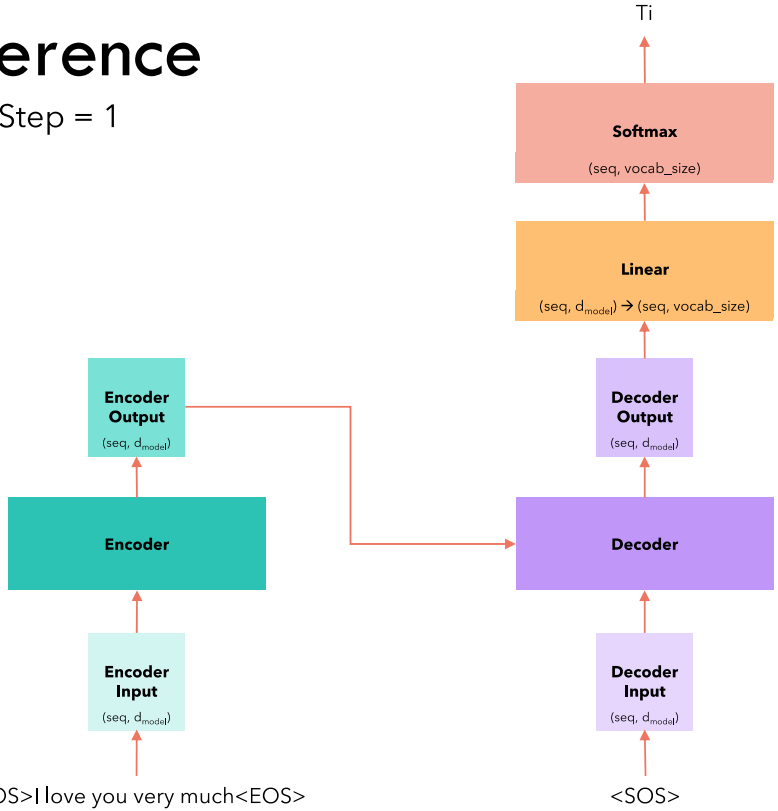
12 (2M+1M+1M)+19M=64M

Carnegie Mellon University

Inference

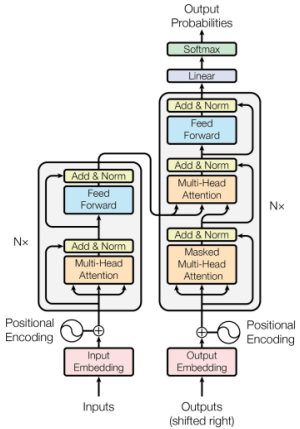
Inference

Time Step = 1



We select a token from the vocabulary corresponding to the position of the token with the maximum value.

The output of the last layer is commonly known as **logits**



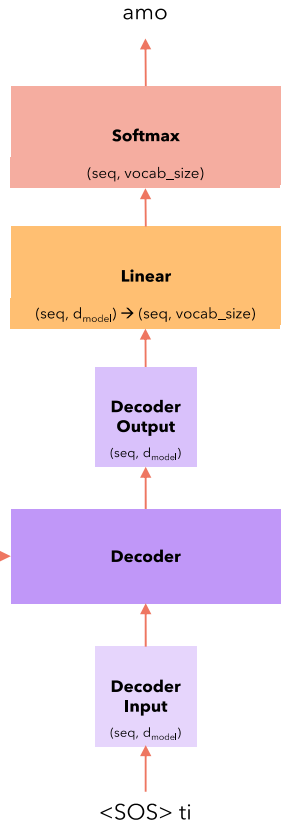
* Both sequences will have same length thanks to padding

Carnegie Mellon University

Inference

Inference

Time Step = 2

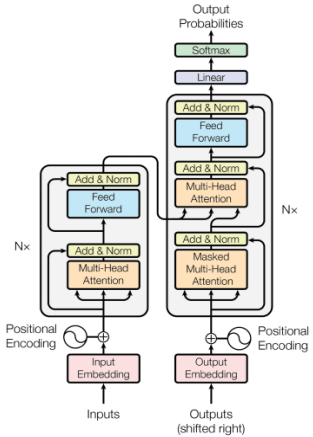


Use the encoder output from the first time step

<SOS>I love you very much<EOS>

<SOS> ti

Since decoder input now contains **two** tokens, we select the softmax corresponding to the second token.



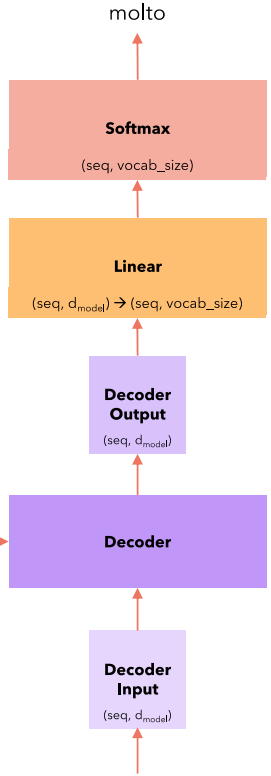
Append the previously output word to the decoder input

Carnegie Mellon University

Inference

Inference

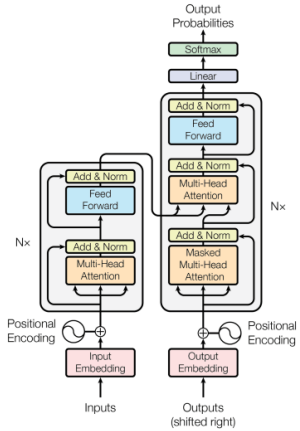
Time Step = 3



Use the encoder output from the first time step

<SOS>I love you very much<EOS>

Since decoder input now contains **three** tokens, we select the softmax corresponding to the third token.



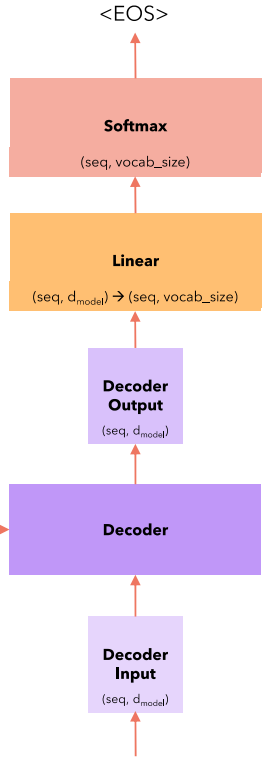
Append the previously output word to the decoder input

Carnegie Mellon University

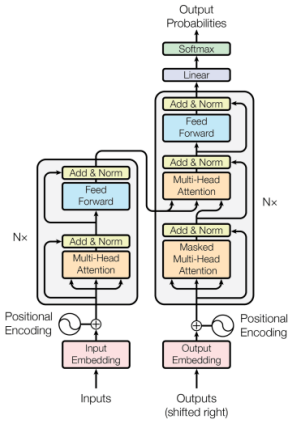
Inference

Inference

Time Step = 4



Since decoder input now contains **four** tokens, we select the softmax corresponding to the fourth token.



Use the encoder output from the first time step

<SOS>I love you very much<EOS>

<SOS> ti amo molto

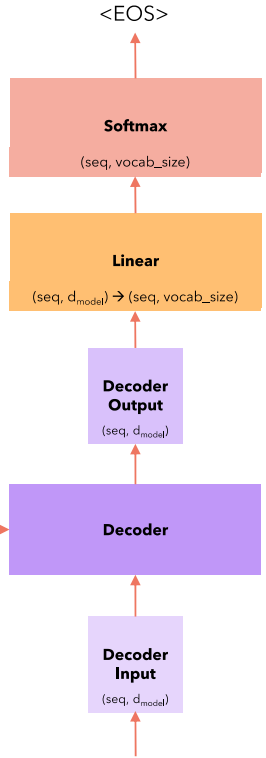
Append the previously output word to the decoder input

Carnegie Mellon University

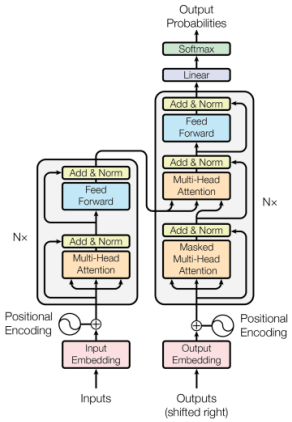
Inference

Inference

Time Step = 4



Since decoder input now contains **four** tokens, we select the softmax corresponding to the fourth token.



Use the encoder output from the first time step

<SOS>I love you very much<EOS>

<SOS> ti amo molto

Append the previously output word to the decoder input

Carnegie Mellon University

SOTA on BLEU

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Carnegie Mellon University

How attention learns the context?

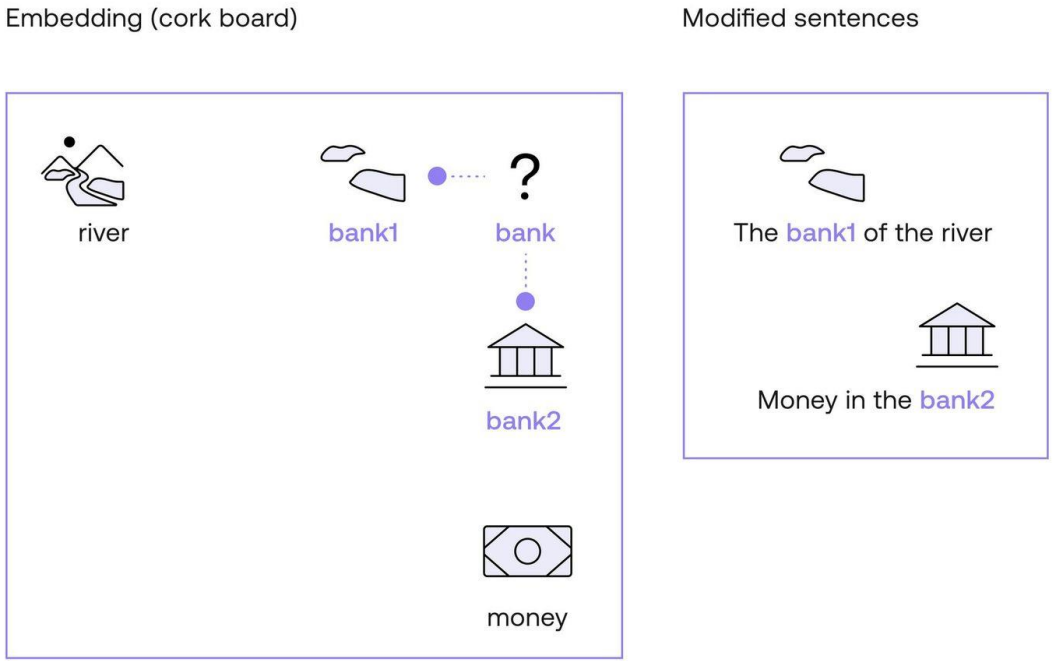
Attention:
Telling context in words

The bank of the river

Money in the bank

How attention learns the context?

- Modified sentence 1:
The bank1 of the river.
-
Modified sentence 2:
Money in the bank2.

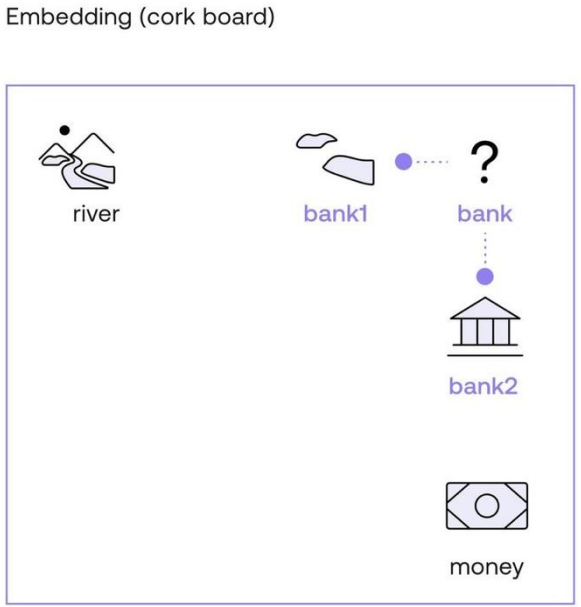


How attention learns the context?

•What do you mean by “moving a word closer to another one”?

•Why did you ignore the other words in the sentence? How did you know the words “river” and “money” were the ones dictating the context, instead of the words “the”, “in”, or “of”?

• As humans, we know which words provide context, but a computer wouldn’t have a clue.

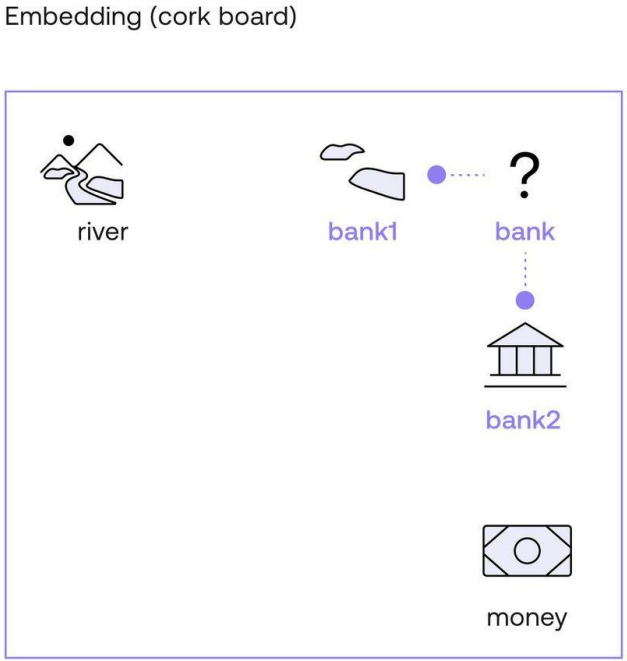


Carnegie Mellon University

How attention learns the context?

what we mean by “moving a word closer to another one”. The way I like to imagine this, is to average two words. For example, let’s say that I want to move the word “bank” 10% closer to the word river. I now think of the word “0.9*Bank + 0.1*River”. That is, “bank1” is 90% “bank”, and 10% “river”. Also, let’s say that “bank2” is 80% “bank” and 20% “money”. So let’s say these are the modified words:

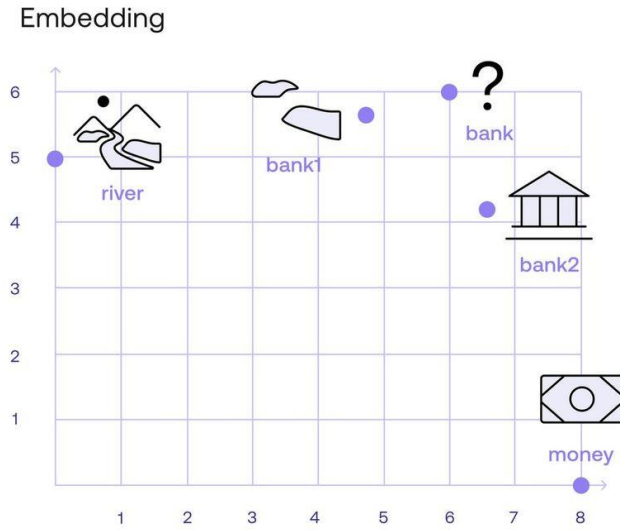
- Bank1 = 0.9*Bank + 0.1*River
- Bank2 = 0.8*Bank + 0.2*Money



How attention learns the context?

we learned that a word embedding consists of assigning a vector (list) of numbers to each word. The Cohere embedding associates each word with a vector of length 4096 (that is, a list of 4096 numbers per word). For simplicity, let's imagine an embedding that associates a vector of two numbers to each word, and that the following are the numbers:

- River: [0,5]
- Money: [8,0]
- Bank: [6,6]

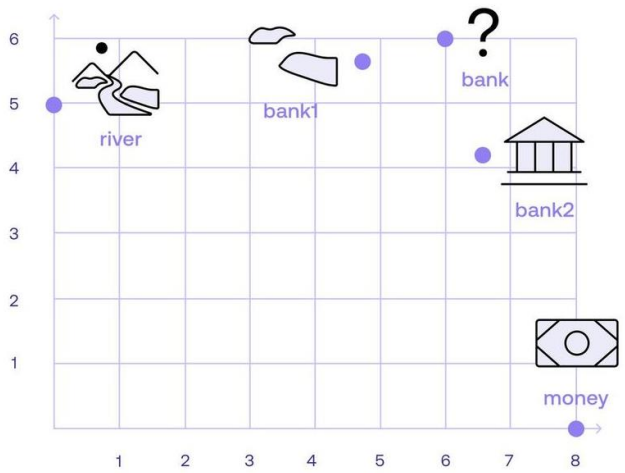


Carnegie Mellon University

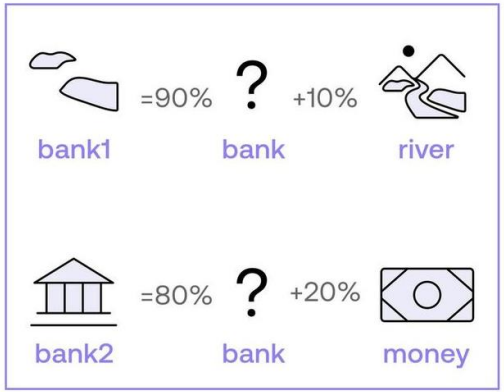
How attention learns the context?

•Bank1: 0.9*Bank + 0.1*River = 0.9*[6, 6] + 0.1*[0, 5]
= [5.4, 5.4] + [0, 0.5]= [5.4, 5.9]
•Bank2: 0.8*Bank + 0.2*Money = 0.8*[6,6] + 0.2*[8,0]
= [4.8, 4.8] + [1.6, 0]= [6.4, 4.8]

Embedding



Equations



Carnegie Mellon University

How attention learns the context?

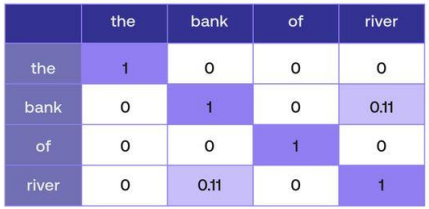
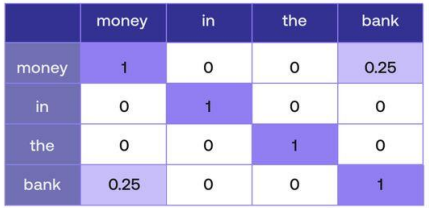
The bank of the river

	the	bank	of	river
the	1	0	0	0
bank	0	1	0	0.11
of	0	0	1	0
river	0	0.11	0	1

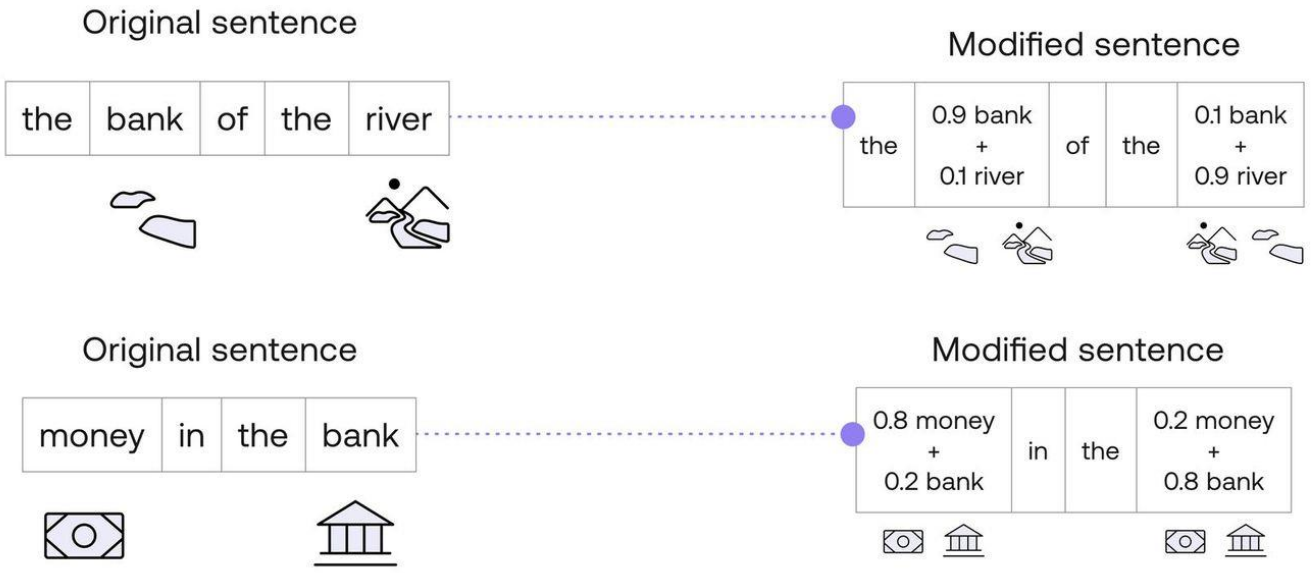
Money in the bank

	money	in	the	bank
money	1	0	0	0.25
in	0	1	0	0
the	0	0	1	0
bank	0.25	0	0	1

How attention learns the context?

Similarities	New words	Attention step	Normalization step
The bank of the river			
	the1	the	the
	bank1	bank +0.11 river	+0.9 bank +0.1 river
	of1	of	of
	river1	river +0.11 bank	+0.9 river +0.1 bank
Money in the bank			
	money2	money +0.25 bank	+0.8 money +0.2 bank
	in2	in	in
	the2	the	the
	bank2	bank +0.25 money	+0.8 bank +0.2 money

How attention learns the context?



Masked Language Models and Pre-training

Word Embedding

1-of-N Encoding

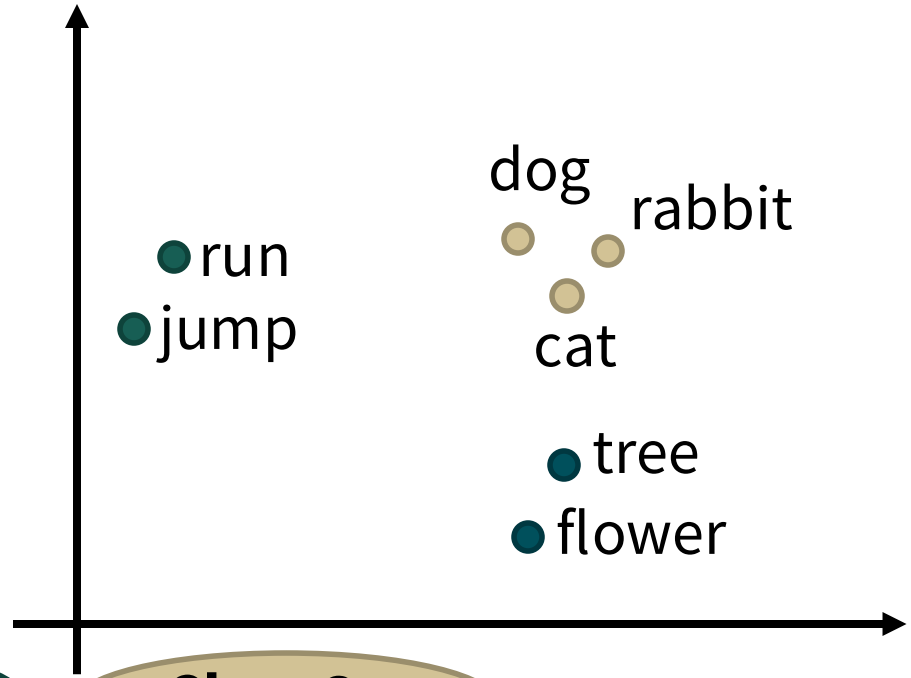
apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

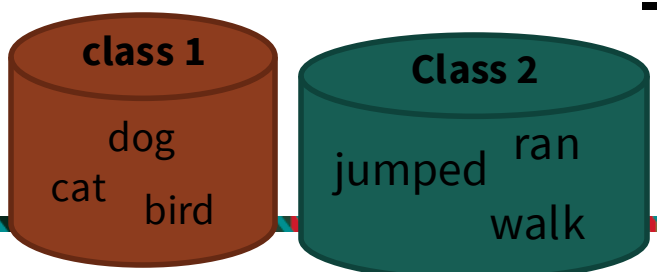
cat = [0 0 1 0 0]

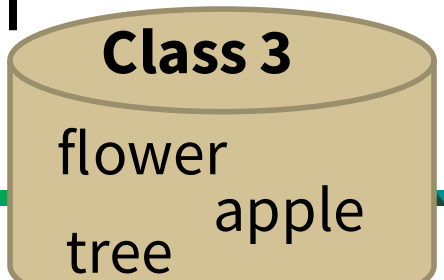
dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]



Word Class





Carnegie Mellon University

A word can have multiple senses

Have you paid that money to the bank yet ?

It is safest to deposit your money in the bank .

The victim was found lying dead on the river bank .

They stood on the river bank to fish.

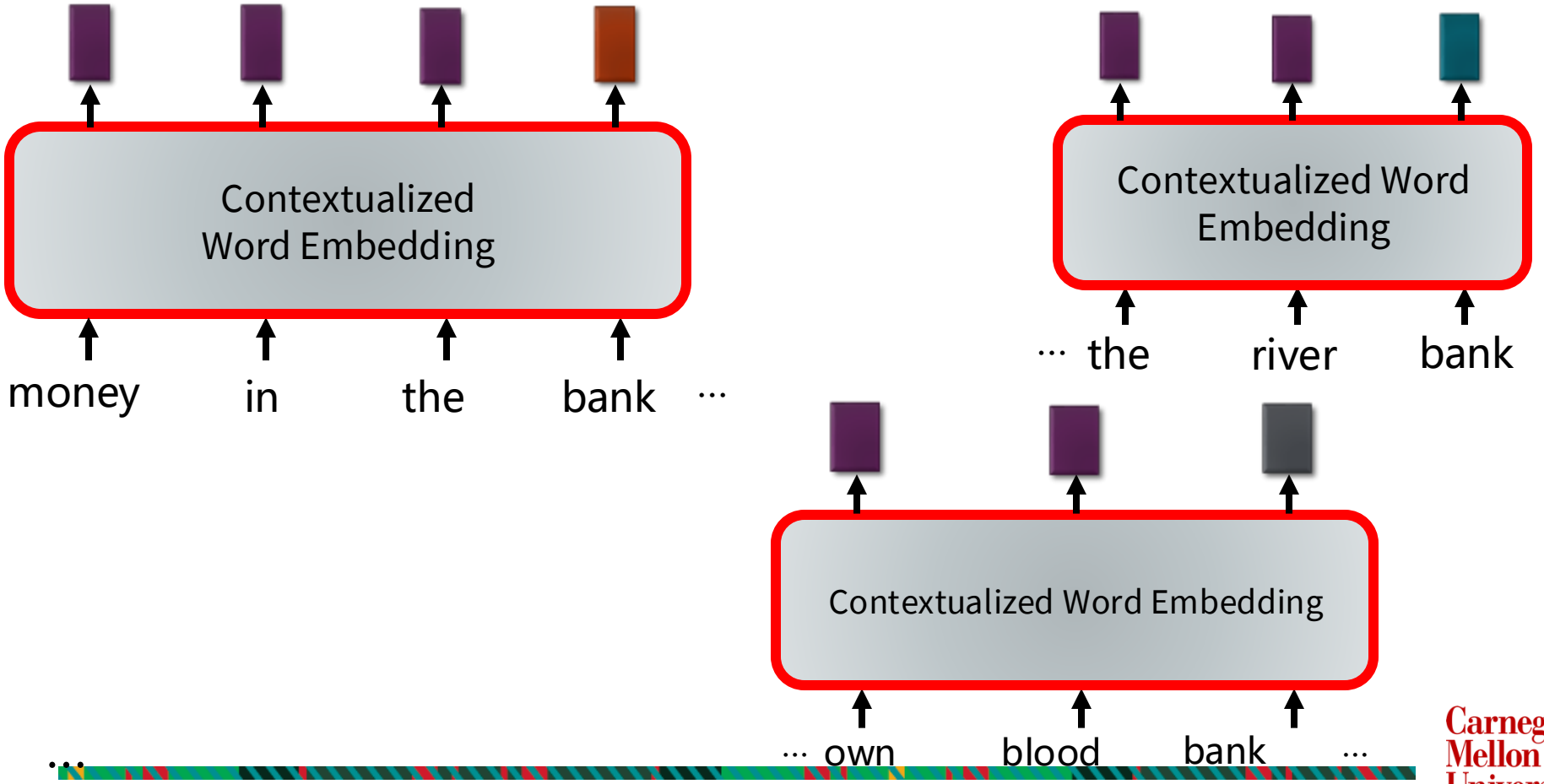
The hospital has its own blood bank.

The third sense or not?

https://arxiv.org/abs/1902.06006

Carnegie Mellon University

Contextualized Word Embedding

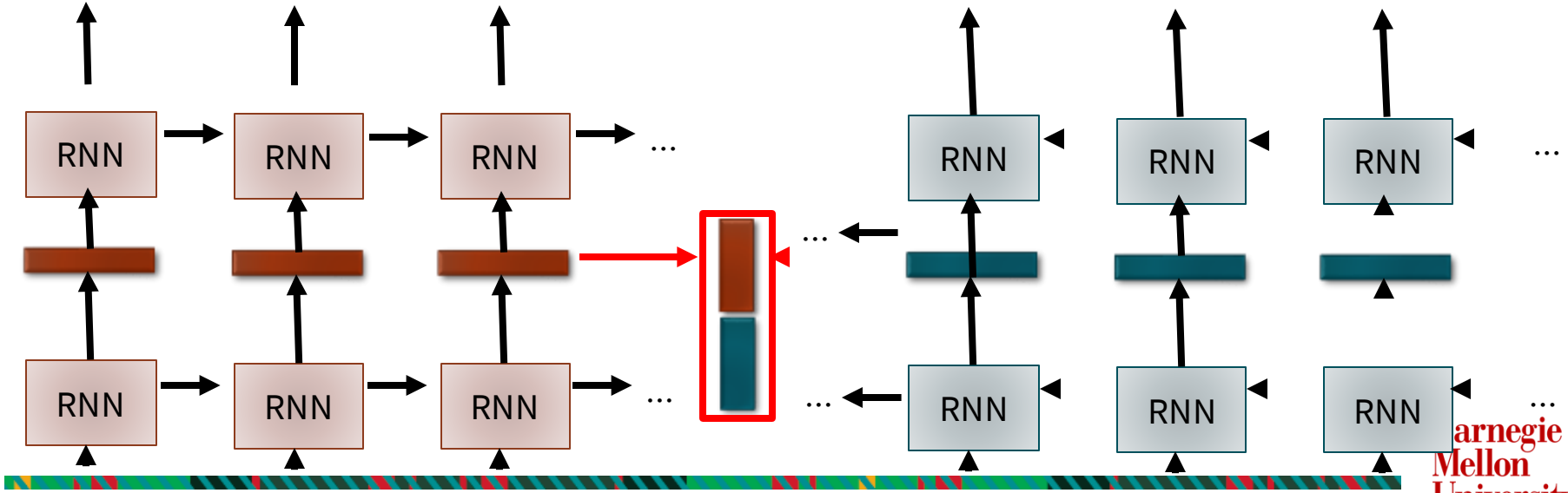


Embeddings from Language Model (ELMO)

Embeddings from Language Model (ELMO)

https://arxiv.org/abs/1802.05365

RNN-BASED LANGUAGE MODELS (TRAINED FROM LOTS OF SENTENCES)



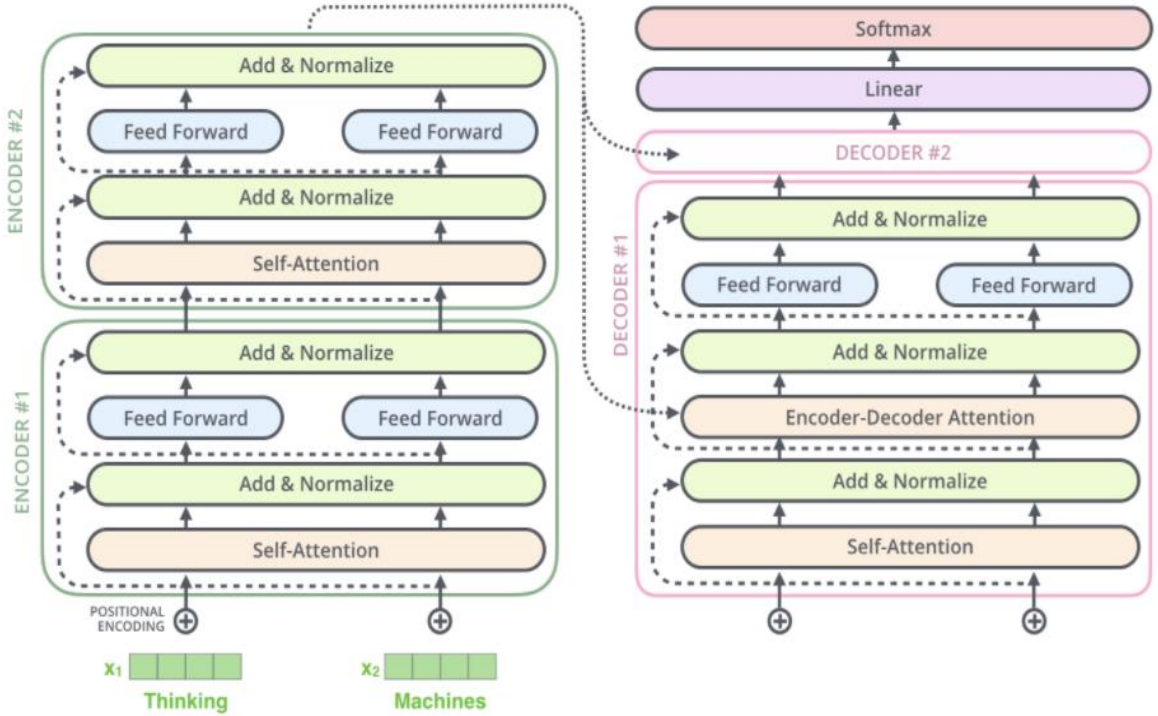
Why BERT?

- Those contextual LMs have a problem: They only use left context or right context
- But language understanding is bidirectional
- The reason for unidirectionality: Words can “see themselves” in a bidirectional encoder
- BERT came to the rescue!

- BERT generates a language model by training in both directions which gives words more context
- BERT provided a way to more accurately pre-train models with less data
- It involved a pre-training and fine-tuning stages
- It is based on the Transformer architecture (encoders)

Transformer's Recap

The vanilla **Transformer** model has an Encoder and Decoder, and was used in a seq2seq manner.

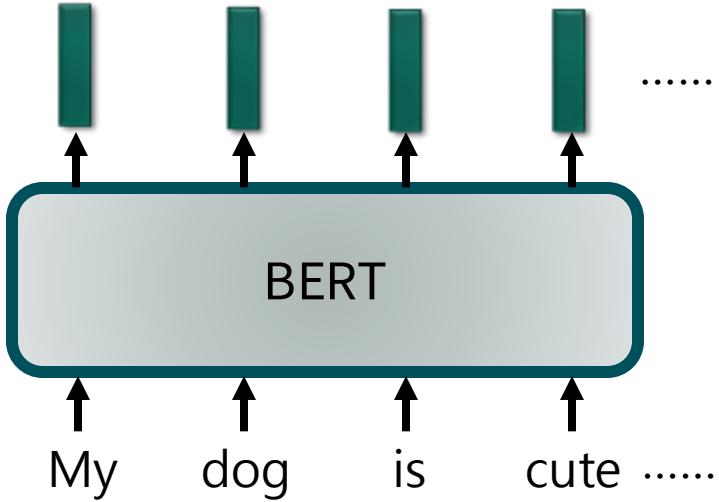


https://jalammar.github.io/illustrated-transformer/

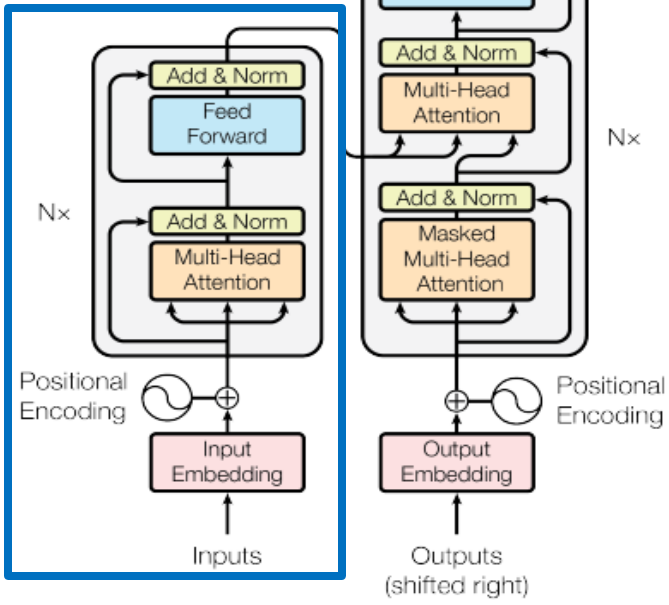
Bidirectional Encoder Representations from Transformers (BERT)

BERT = Encoder of Transformer

Learned from a large amount of text without annotation



Encoder



Bidirectional Encoder Representations from Transformers (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.

Semi-supervised Learning Step

Model: BERT

Dataset: WIKIPEDIA Die freie Enzyklopädie

Objective: Predict the masked word (langauge modeling)

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

Supervised Learning Step

Model: (pre-trained in step #1) BERT

Classifier → 75% Spam, 25% Not Spam

Dataset:

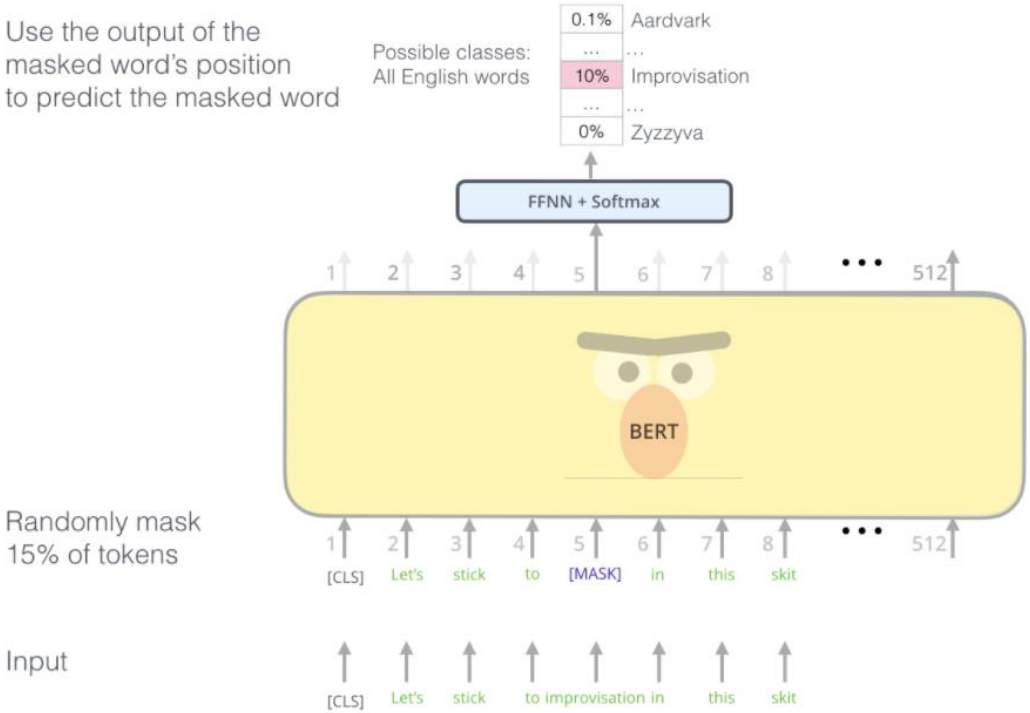
Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

Carnegie Mellon University

BERT

- **Model**: several Transformer Encoders. Input sentence or sentence pairs, [CLS] token, subword embeddings
- **Objective**: MLM and next-sentence prediction
- **Data**: BooksCorpus and Wikipedia

MLM objective



BERT's clever language modeling task masks 15% of words in the input and asks the model to predict the missing word.

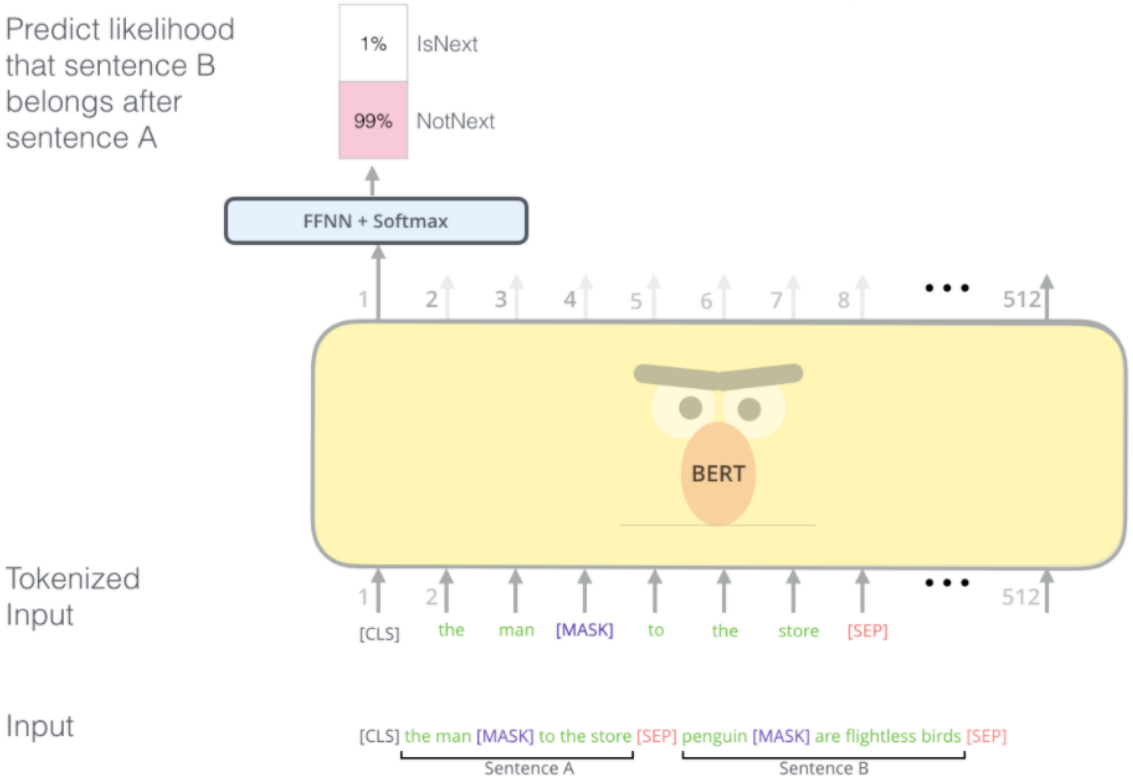
https://jalammar.github.io/illustrated-transformer/

BERT

BERT

- **Model**: several Transformer Encoders. Input sentence or sentence pairs, [CLS] token, subword embeddings
- **Objective**: MLM and next-sentence prediction
- **Data**: BooksCorpus and Wikipedia

Next sentence objective

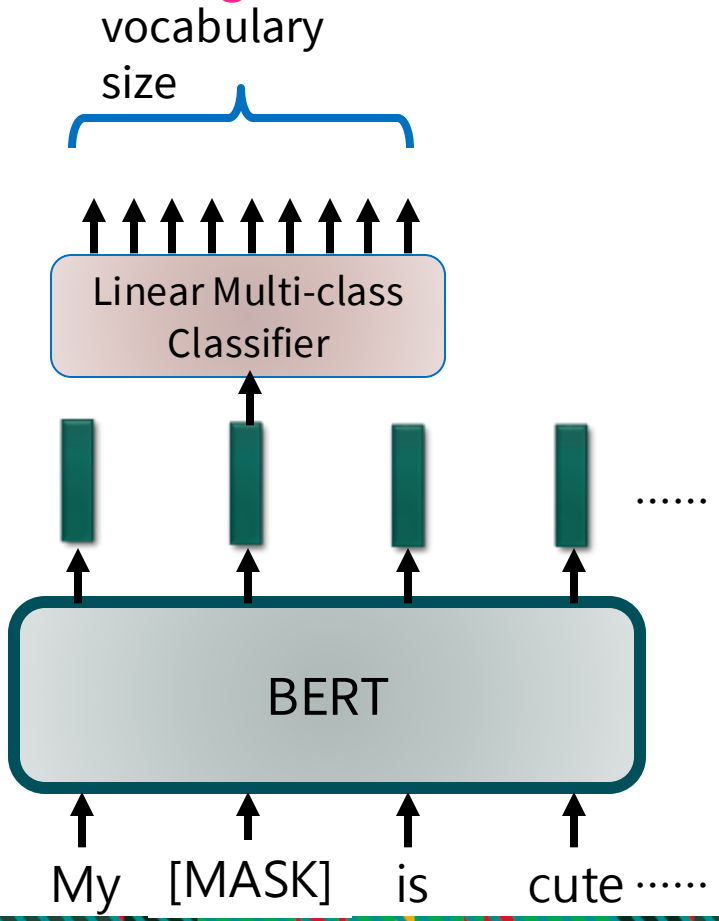


https://jalammar.github.io/illustrated-transformer/

Training of BERT

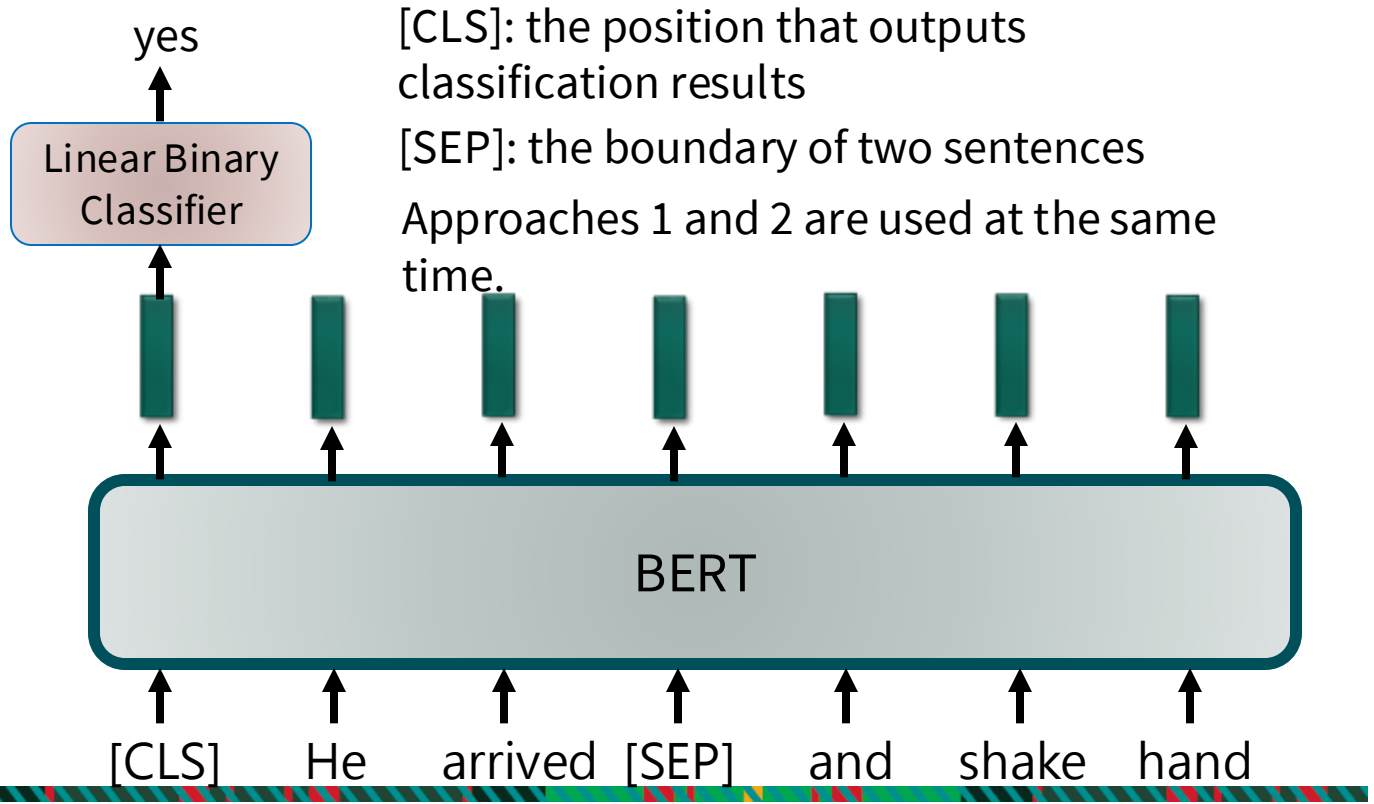
Approach 1: Masked LM

Predicting the masked word



Training of BERT

Approach 2: Next Sentence Prediction



How to use BERT – Case 1

class

Linear Classifier → Trained from Scratch

Input: single sentence, output: class

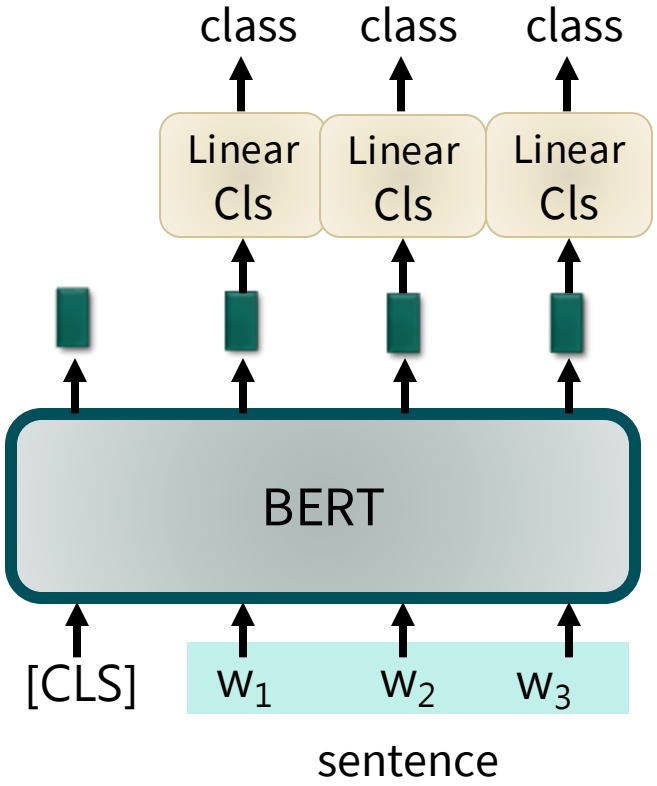
BERT → Fine-tune

[CLS] w_1 w_2 w_3

sentence

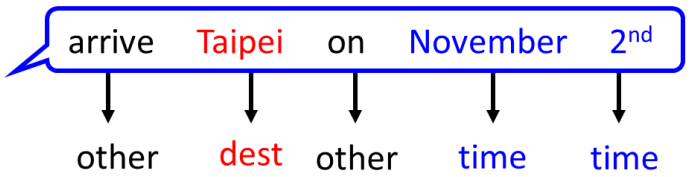
Example:
Sentiment analysis (our HW),
Document Classification

How to use BERT – Case 2



Input: single sentence,
output: class of each word

Example: Slot filling



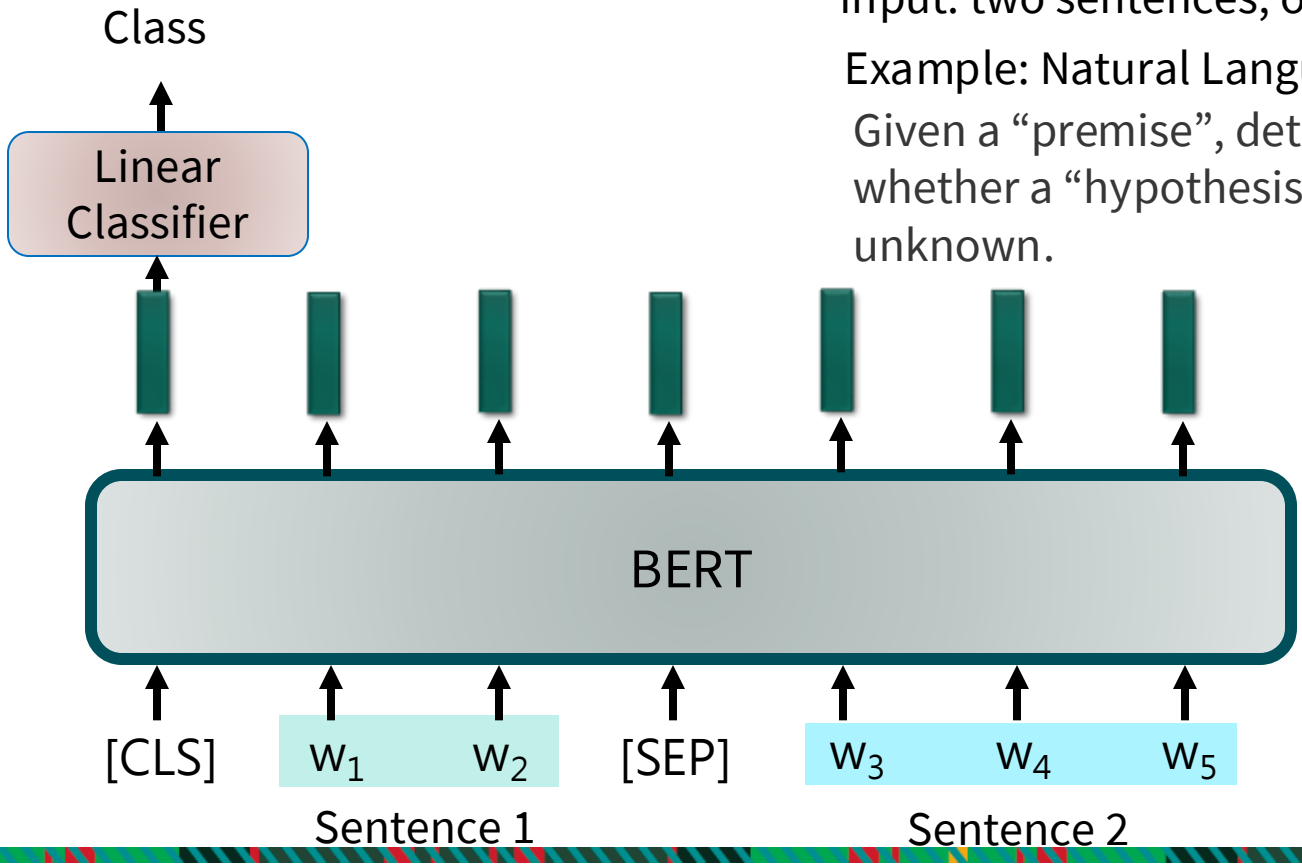
Carnegie Mellon University

How to use BERT – Case 3

Input: two sentences, output: class

Example: Natural Language Inference

Given a "premise", determining whether a "hypothesis" is T/F/ unknown.

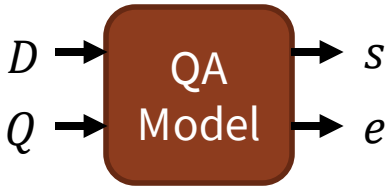


How to use BERT – Case 4

Extraction-based Question Answering (QA) (e.g. SQuAD)

Document: $D = \{d_1, d_2, \cdots, d_N\}$

Query: $Q = \{q_1, q_2, \cdots, q_N\}$



output: two integers (s, e)

Answer: $A = \{q_s, \cdots, q_e\}$

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. 17 in forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

77 79

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

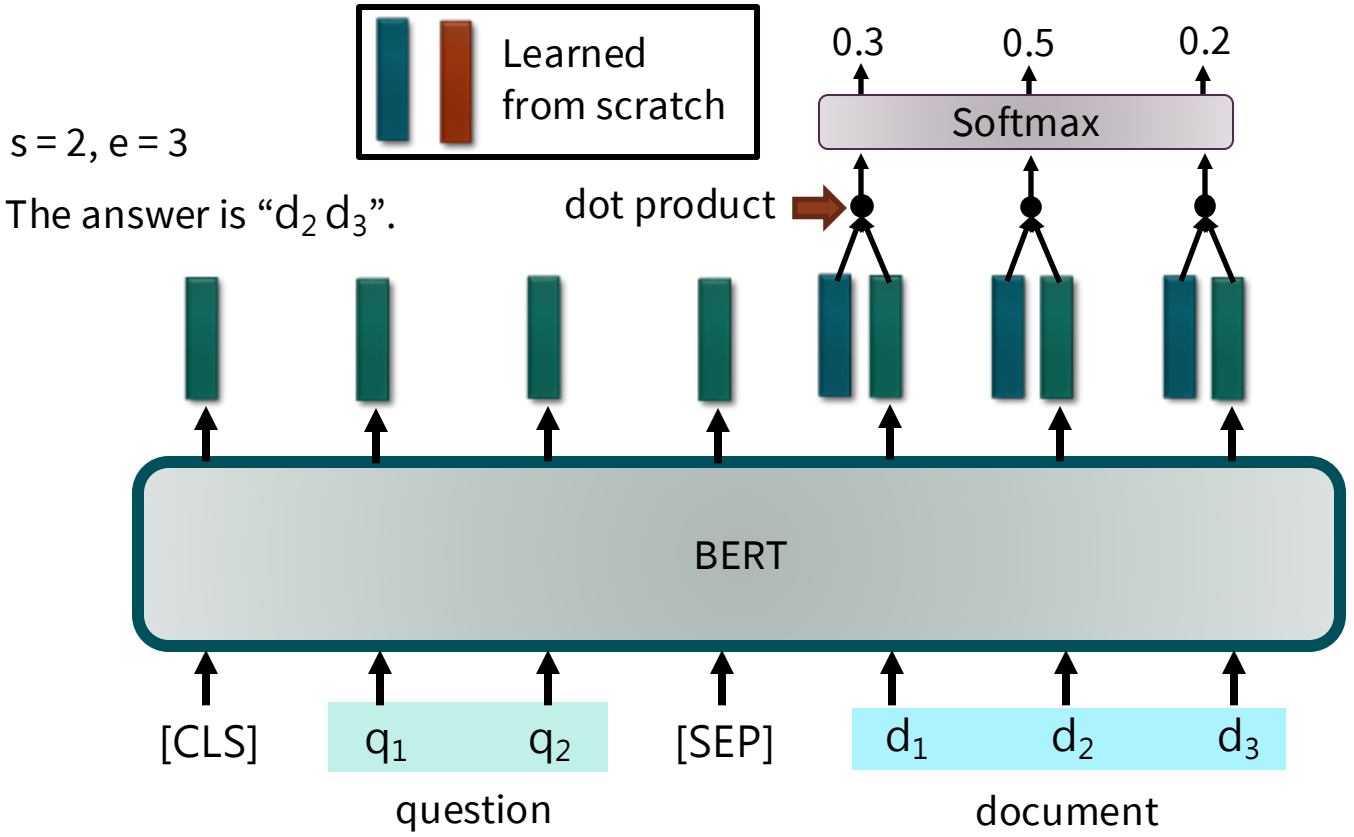
$s = 17, e = 17$

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

$s = 77, e = 79$

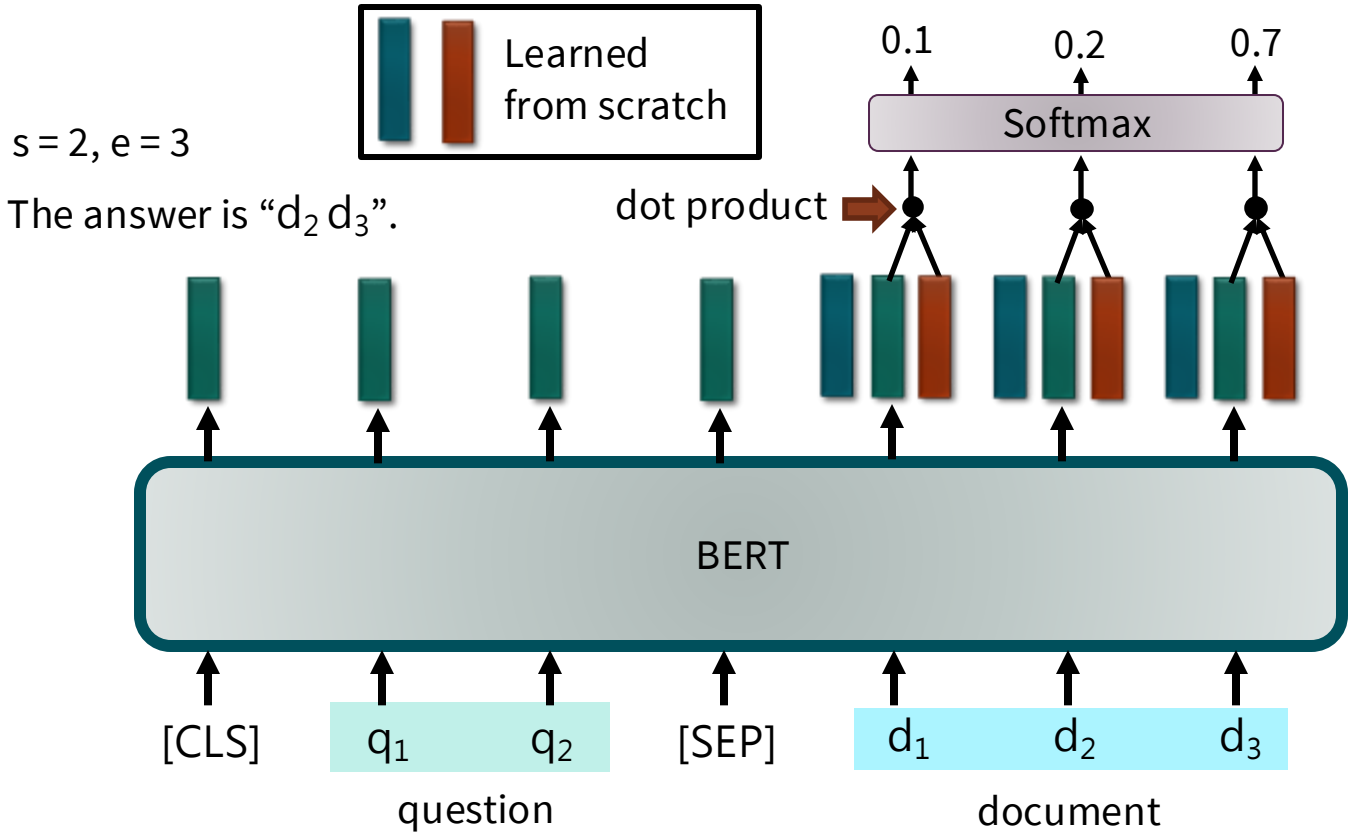
Carnegie Mellon University

How to use BERT – Case 4



Carnegie Mellon University

How to use BERT – Case 4



Carnegie Mellon University