https://ttpoll.com/p/200608

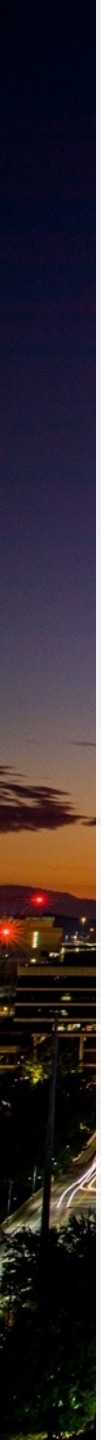
COSC 424/525: Deep Learning

Dr. Hector Santos-Villalobos

Dr. Santos

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Lecture 23: Transformers



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Class Announcements

Homework:

Bonus homework will be released by the end of the week.

Lectures:

- Special lecture on object segmentation on Canvas.
- 2-4 more special lectures
  - Auto-encoders
  - Generative Adversarial Networks
  - Diffusion techniques
  - Adapters and Mixtures of Experts

Exams:

- Final exam, May 9th, 10:30 am
  - Comprehensive (All lectures)
- I will generate video discussing Exam #2 key.

Course project:

- Create your own project title
- Loss plot with training and validation

https://ttpoll.com/p/200608

Transformer

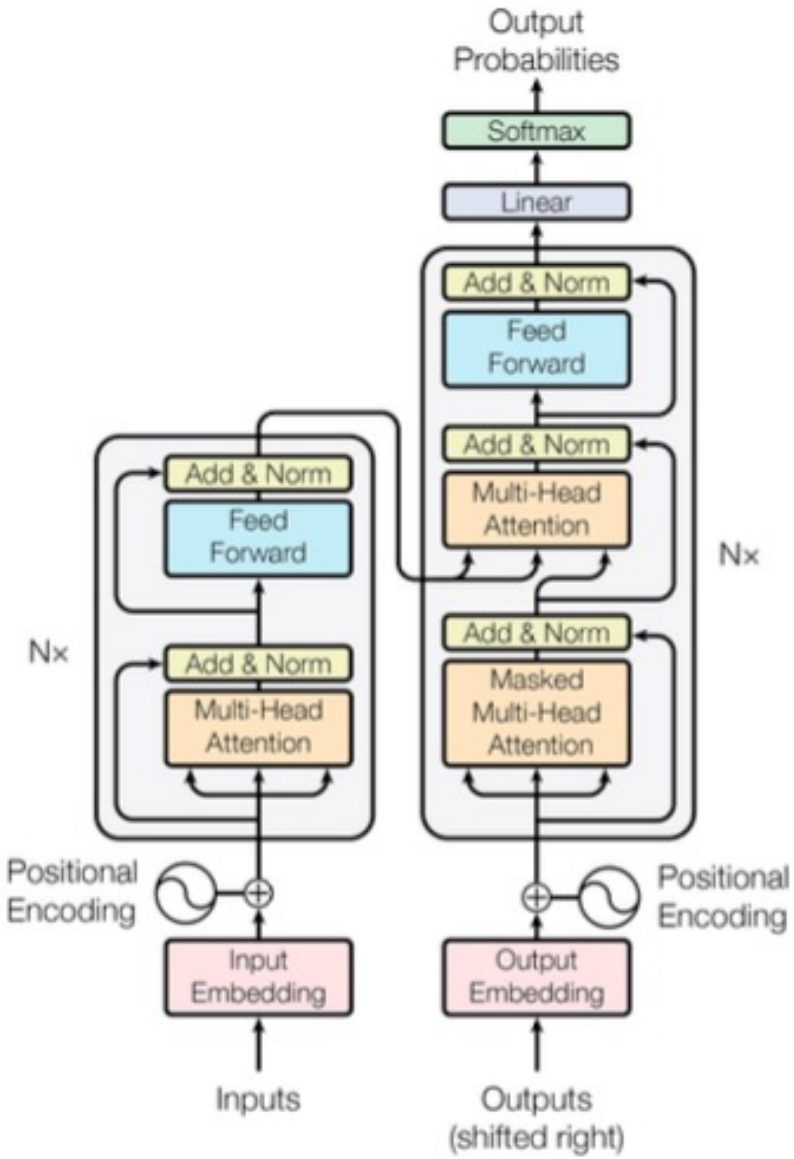
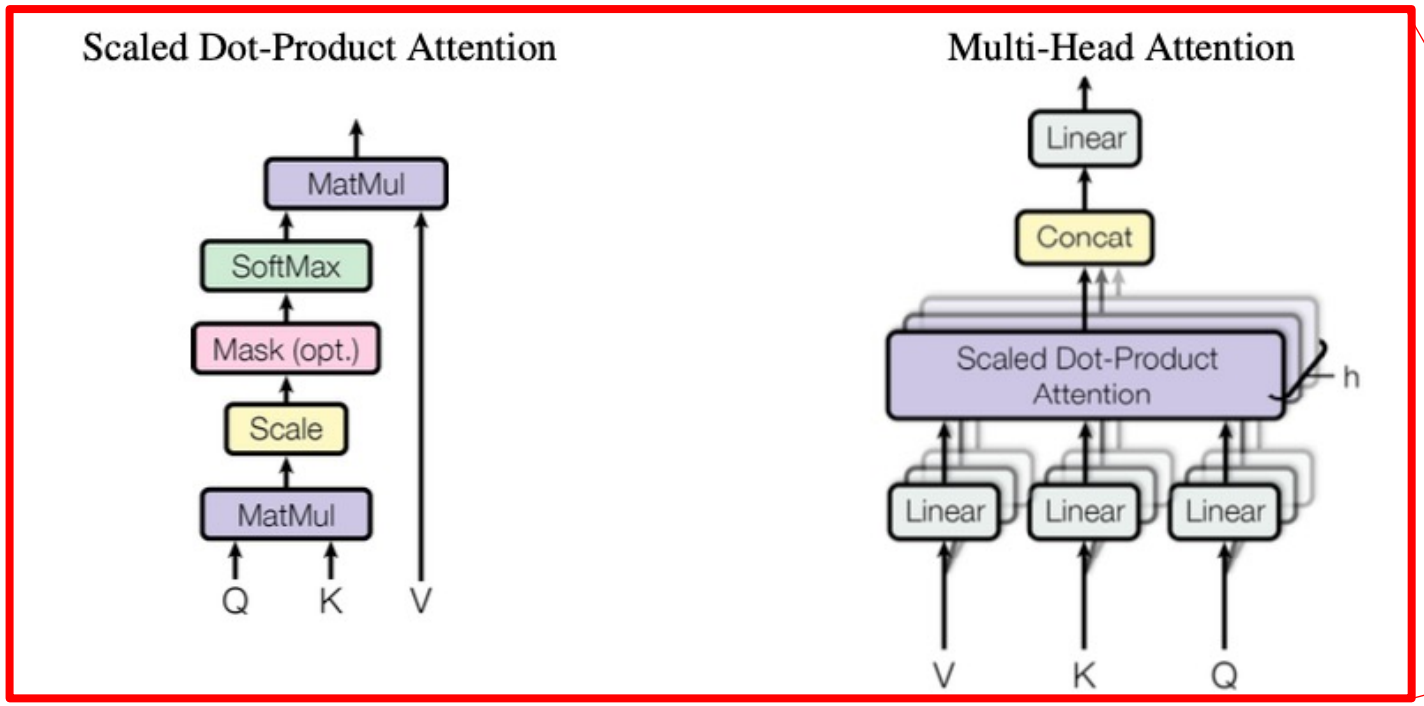


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Transformer



$$A(V,K,Q) = Softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

V, K, Q

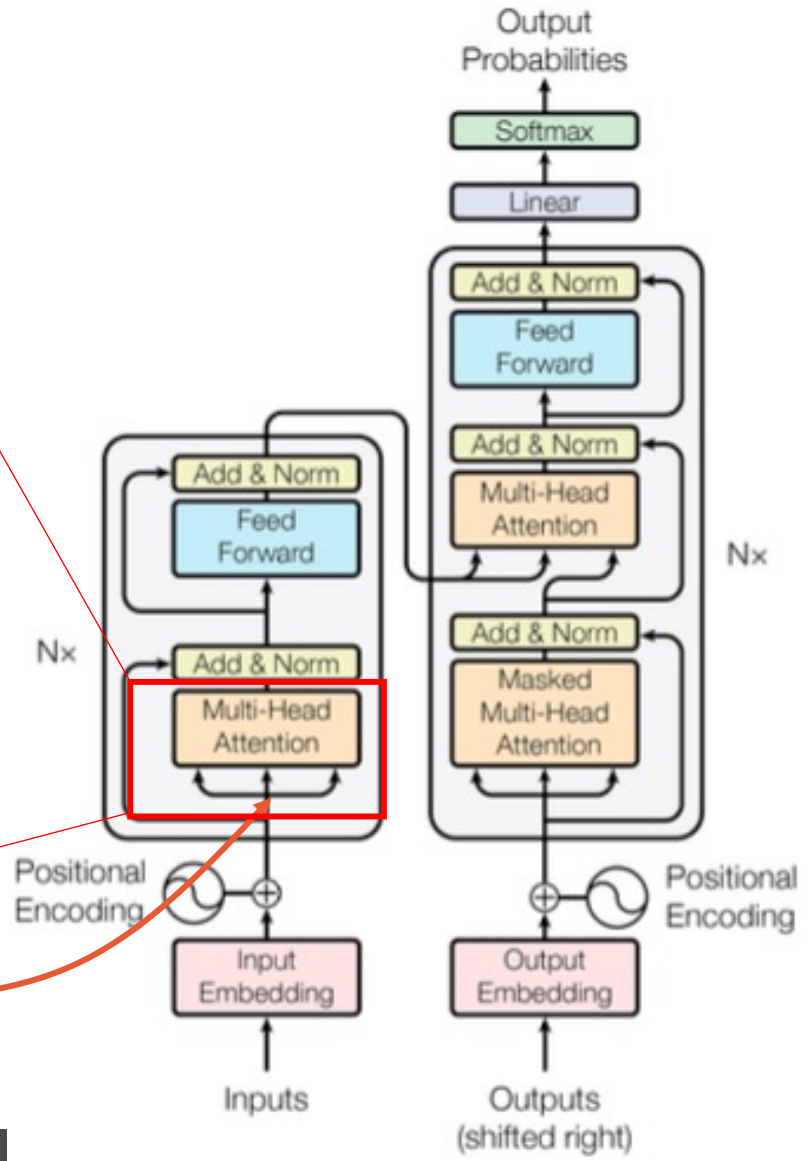


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Transformer

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

Multi-Layer Perceptron

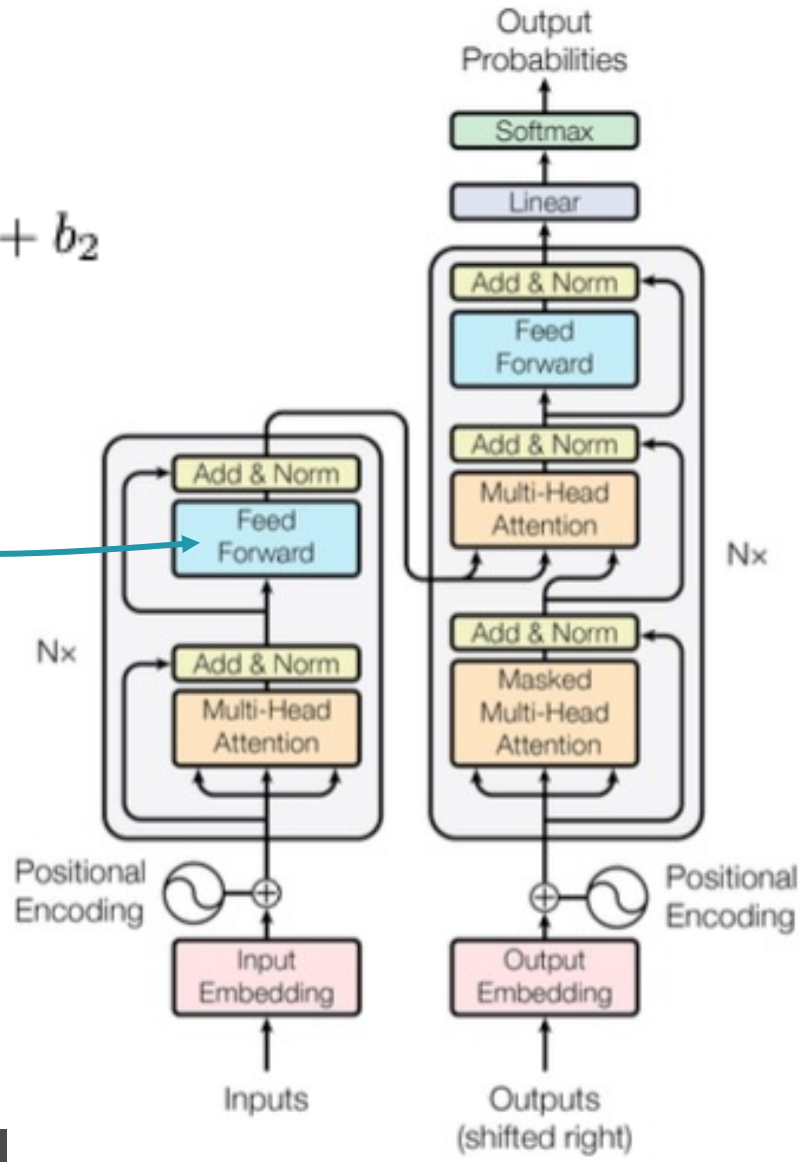


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Pop Quiz

What Self-Attention operation is similar to the gate operations in GRU and LSTM?

A. The linear transformation of the inputs into the V,K,Q components.

B. The softmax of the dot product between Q and K

C. The softmax of the dot product between Q and V

D. The softmax of the dot product between V and K

https://ttpoll.com/p/200608

Positional Encoders

https://ttpoll.com/p/200608

Transformer

No convolutions or recurrence.
Therefore, we don't know where the words are located.

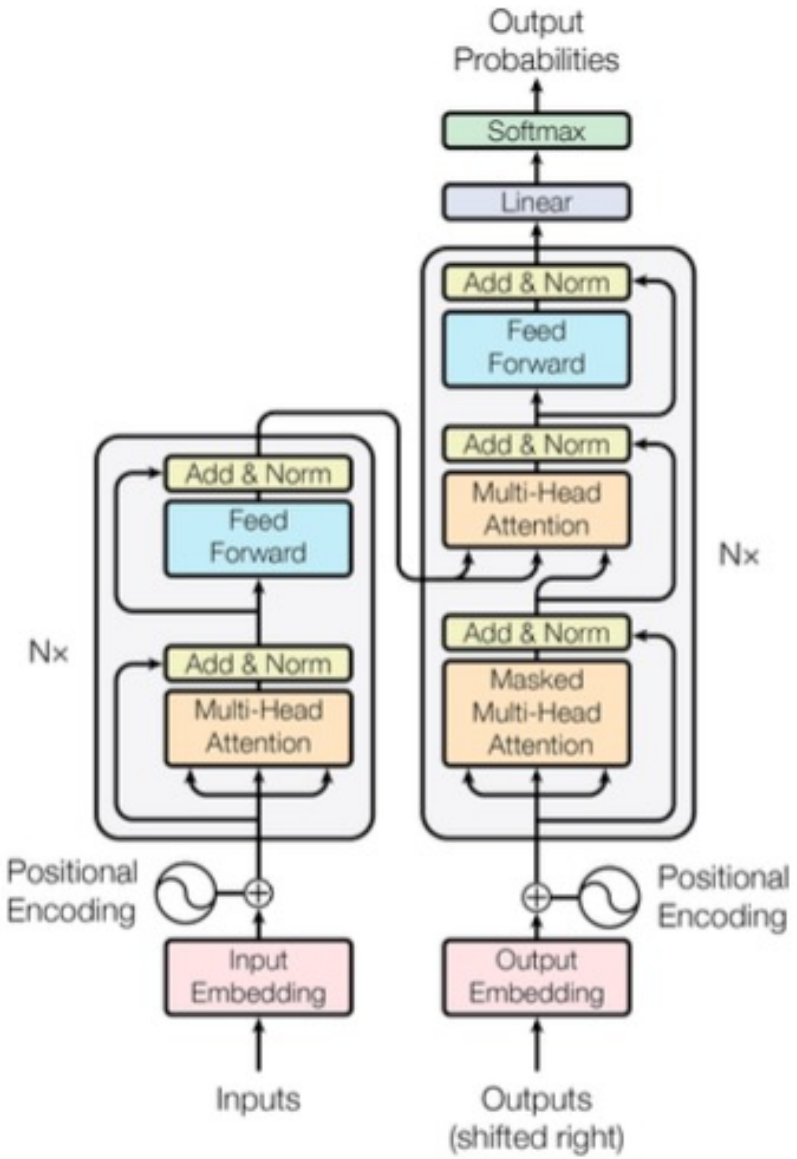
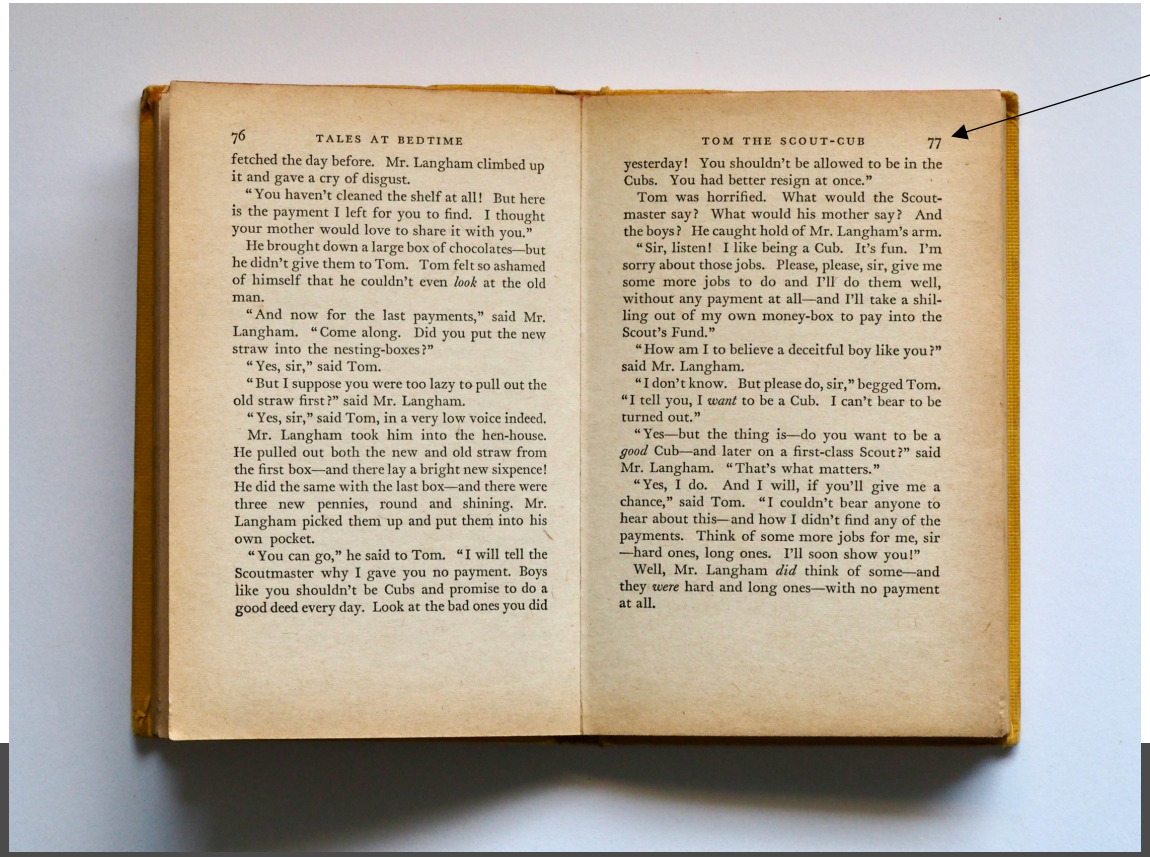


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Transformer

No convolutions or recurrence.
Therefore, we don't know where the words are located.



76 TALES AT BEDTIME

fetched the day before. Mr. Langham climbed up it and gave a cry of disgust.

"You haven't cleaned the shelf at all! But here is the payment I left for you to find. I thought your mother would love to share it with you."

He brought down a large box of chocolates—but he didn't give them to Tom. Tom felt so ashamed of himself that he couldn't even *look* at the old man.

"And now for the last payments," said Mr. Langham. "Come along. Did you put the new straw into the nesting-boxes?"

"Yes, sir," said Tom.

"But I suppose you were too lazy to pull out the old straw first?" said Mr. Langham.

"Yes, sir," said Tom, in a very low voice indeed.

Mr. Langham took him into the hen-house. He pulled out both the new and old straw from the first box—and there lay a bright new sixpence! He did the same with the last box—and there were three new pennies, round and shining. Mr. Langham picked them up and put them into his own pocket.

"You can go," he said to Tom. "I will tell the Scoutmaster why I gave you no payment. Boys like you shouldn't be Cubs and promise to do a good deed every day. Look at the bad ones you did

TOM THE SCOUT-CUB 77

yesterday! You shouldn't be allowed to be in the Cubs. You had better resign at once."

Tom was horrified. What would the Scoutmaster say? What would his mother say? And the boys? He caught hold of Mr. Langham's arm.

"Sir, listen! I like being a Cub. It's fun. I'm sorry about those jobs. Please, please, sir, give me some more jobs to do and I'll do them well, without any payment at all—and I'll take a shilling out of my own money-box to pay into the Scout's Fund."

"How am I to believe a deceitful boy like you?" said Mr. Langham.

"I don't know. But please do, sir," begged Tom. "I tell you, I *want* to be a Cub. I can't bear to be turned out."

"Yes—but the thing is—do you want to be a *good* Cub—and later on a first-class Scout?" said Mr. Langham. "That's what matters."

"Yes, I do. And I will, if you'll give me a chance," said Tom. "I couldn't bear anyone to hear about this—and how I didn't find any of the payments. Think of some more jobs for me, sir—hard ones, long ones. I'll soon show you!"

Well, Mr. Langham *did* think of some—and they *were* hard and long ones—with no payment at all.

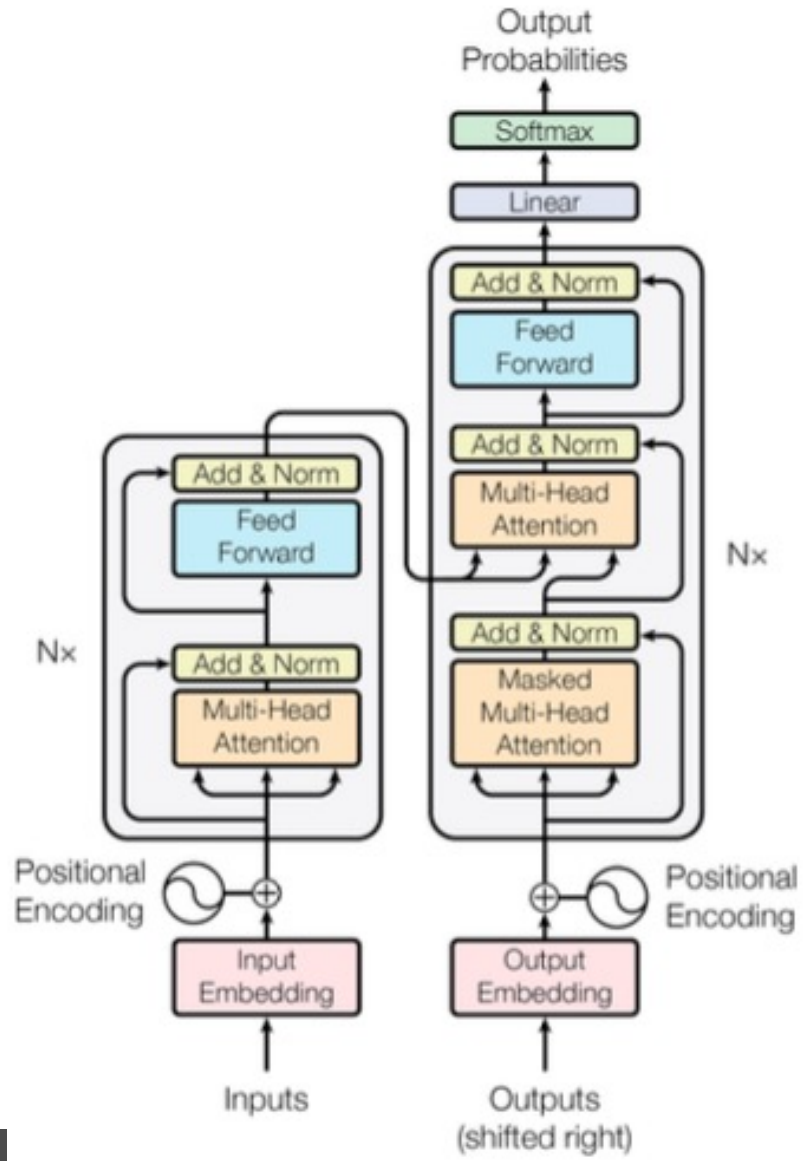


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

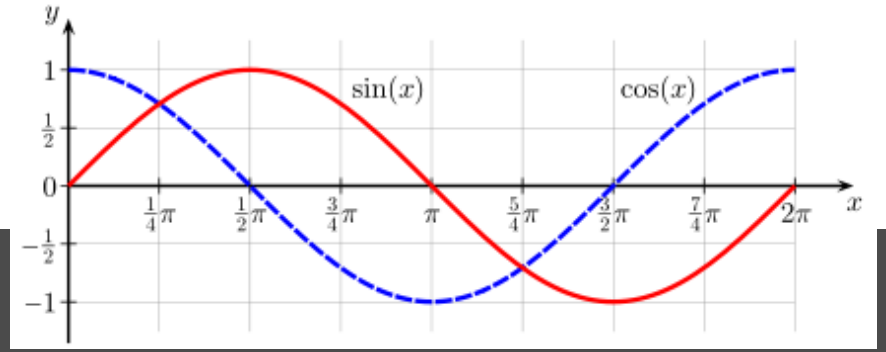
Transformer

No convolutions or recurrence.
Therefore, we don't know where the words are located.

Positional encodings

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

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



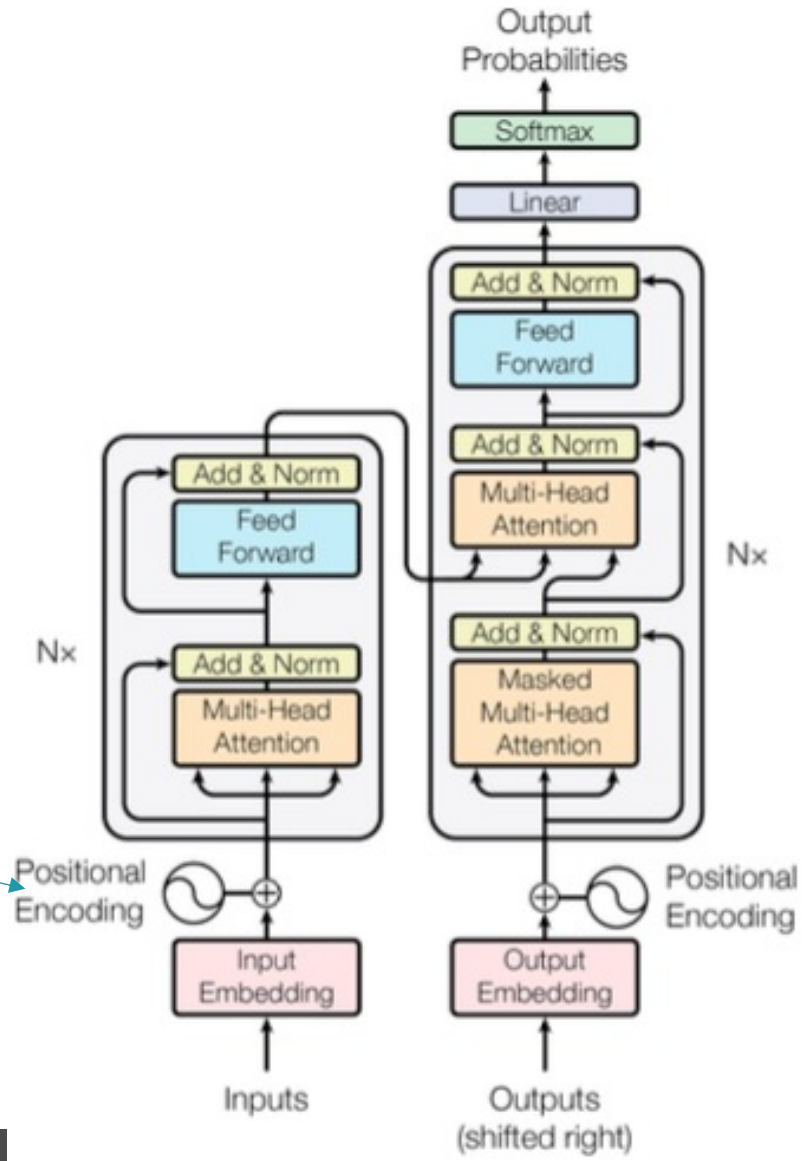
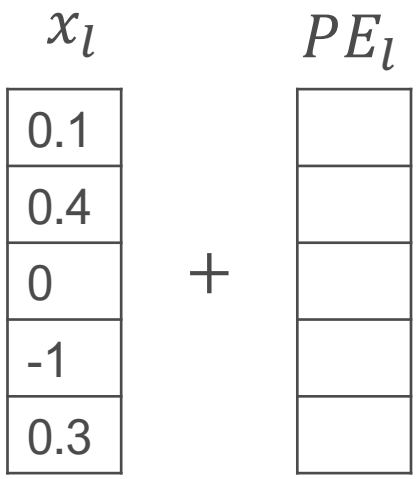


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Positional Encoding

We need deeper networks.



Position of word

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

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

Positional encoding dimension

Input encoding size

Which one of the two PE functions to sample from.

Positional Encoding

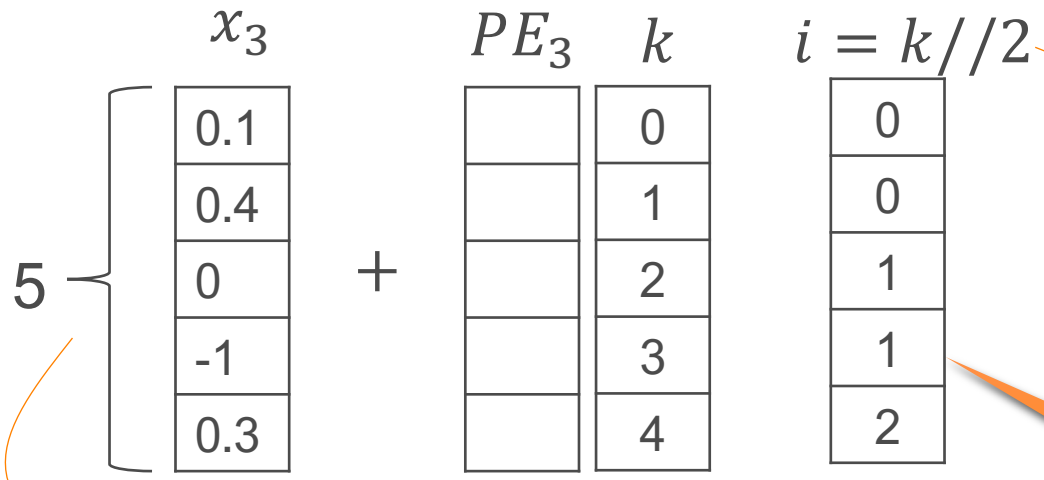
We need deeper networks.

$pos = 3$

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

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

$k =$



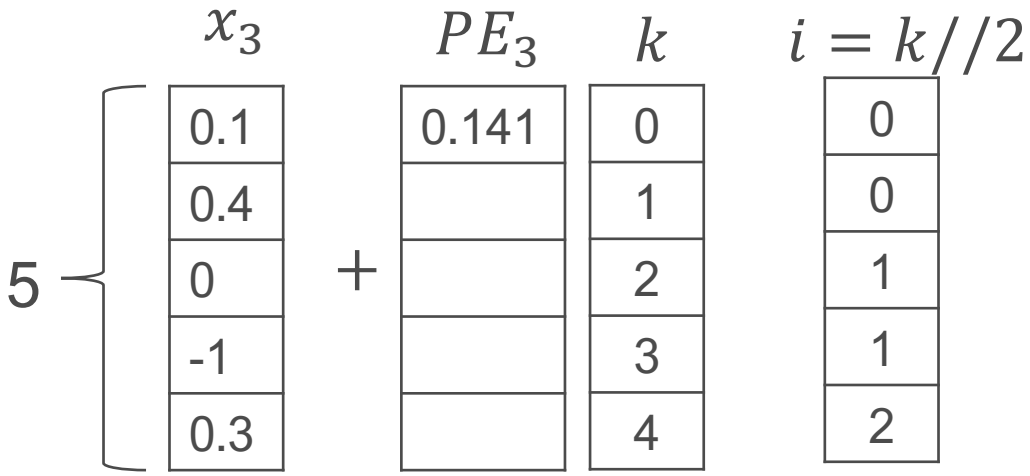
Waves are slower as index increases

https://ttpoll.com/p/200608

Positional Encoding

We need deeper networks.

$$pos = 3$$



$$PE_{(3,0)} = \sin(3) = 0.141$$

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

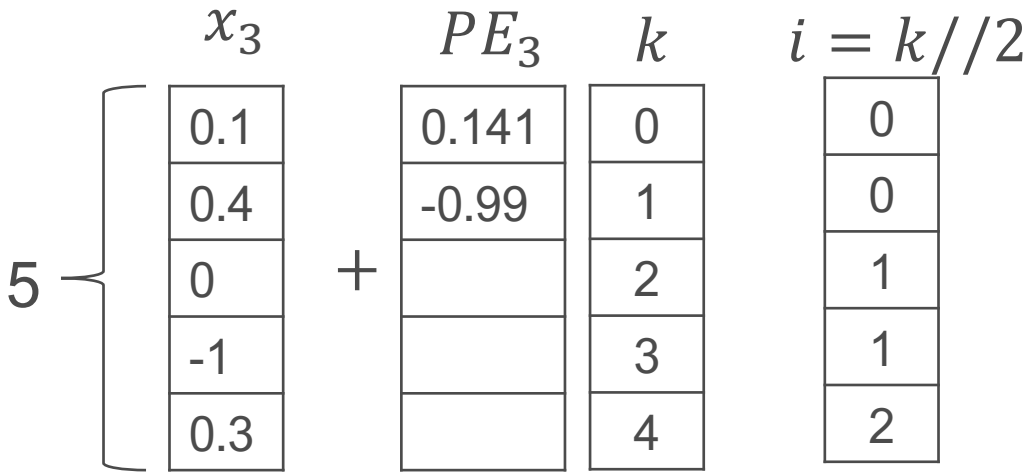
$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

https://ttpoll.com/p/200608

Positional Encoding

We need deeper networks.

$$pos = 3$$



$$PE_{(3,0)} = \sin(3) = 0.141$$

$$PE_{(3,1)} = \cos(3) = -0.99$$

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

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

Positional Encoding

We need deeper networks.

$$pos = 3$$

	x_3		PE_3	k	$i = k//2$
5	0.1	+	0.141	0	0
	0.4		-0.99	1	0
	0		0.075	2	1
	-1			3	1
	0.3			4	2

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

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

$$PE_{(3,0)} = \sin(3) = 0.141$$

$$PE_{(3,1)} = \cos(3) = -0.99$$

$$PE_{(3,2)} = \sin\left(\frac{3}{10000^{\frac{2*1}{5}}}\right) = 0.075$$

https://ttpoll.com/p/200608

Positional Encoding

We need deeper networks.

$$pos = 3$$

	x_3		PE_3	k	$i = k//2$
5	0.1	+	0.141	0	0
	0.4		-0.99	1	0
	0		0.075	2	1
	-1		0.997	3	1
	0.3		0.01	4	2

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

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

$$PE_{(3,0)} = \sin(3) = 0.141$$

$$PE_{(3,1)} = \cos(3) = -0.99$$

$$PE_{(3,2)} = \sin\left(\frac{3}{10000^{\frac{2*1}{5}}}\right) = 0.075$$

$$PE_{(3,3)} = \cos\left(\frac{3}{10000^{\frac{2*1}{5}}}\right) = 0.997$$

$$PE_{(3,4)} = \sin\left(\frac{3}{10000^{\frac{2*2}{5}}}\right) = 0.01$$

https://ttpoll.com/p/200608

Positional Encoding

x_3		PE_3		
0.1	+	0.141	=	0.241
0.4		-0.99		-0.59
0		0.075		0.075
-1		0.997		-0.003
0.3		0.01		0.31

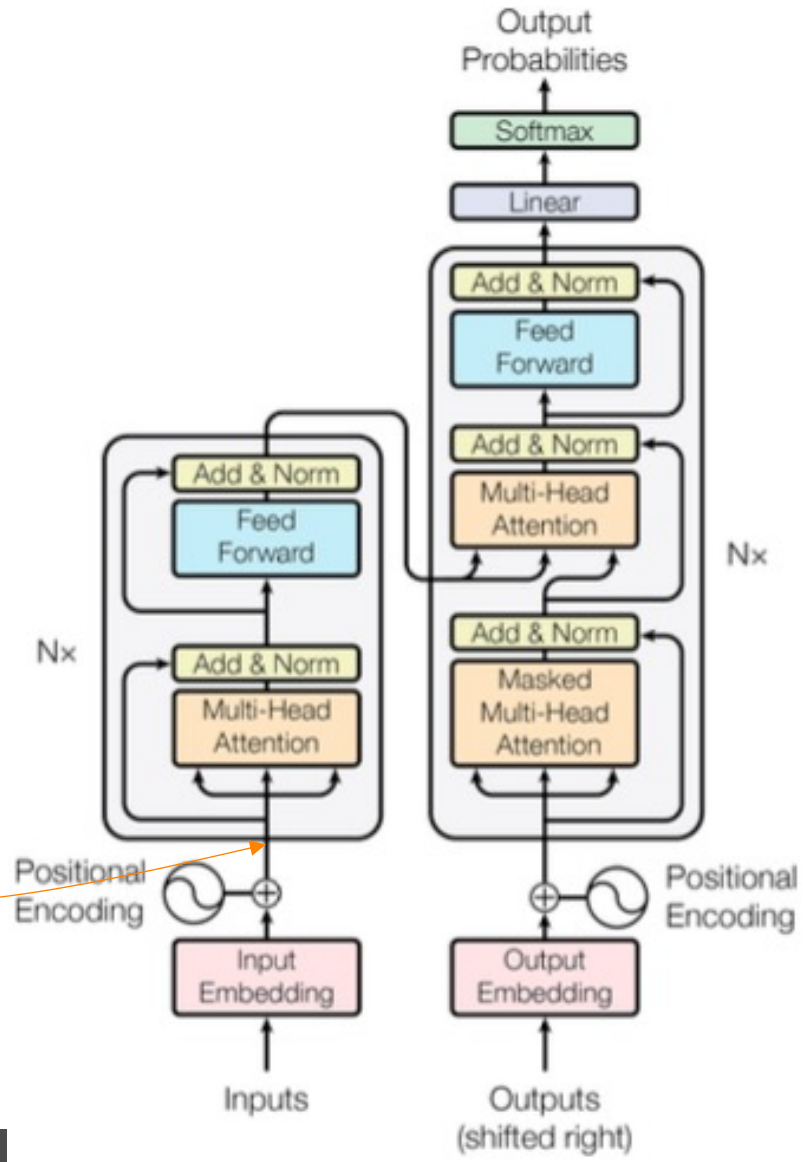
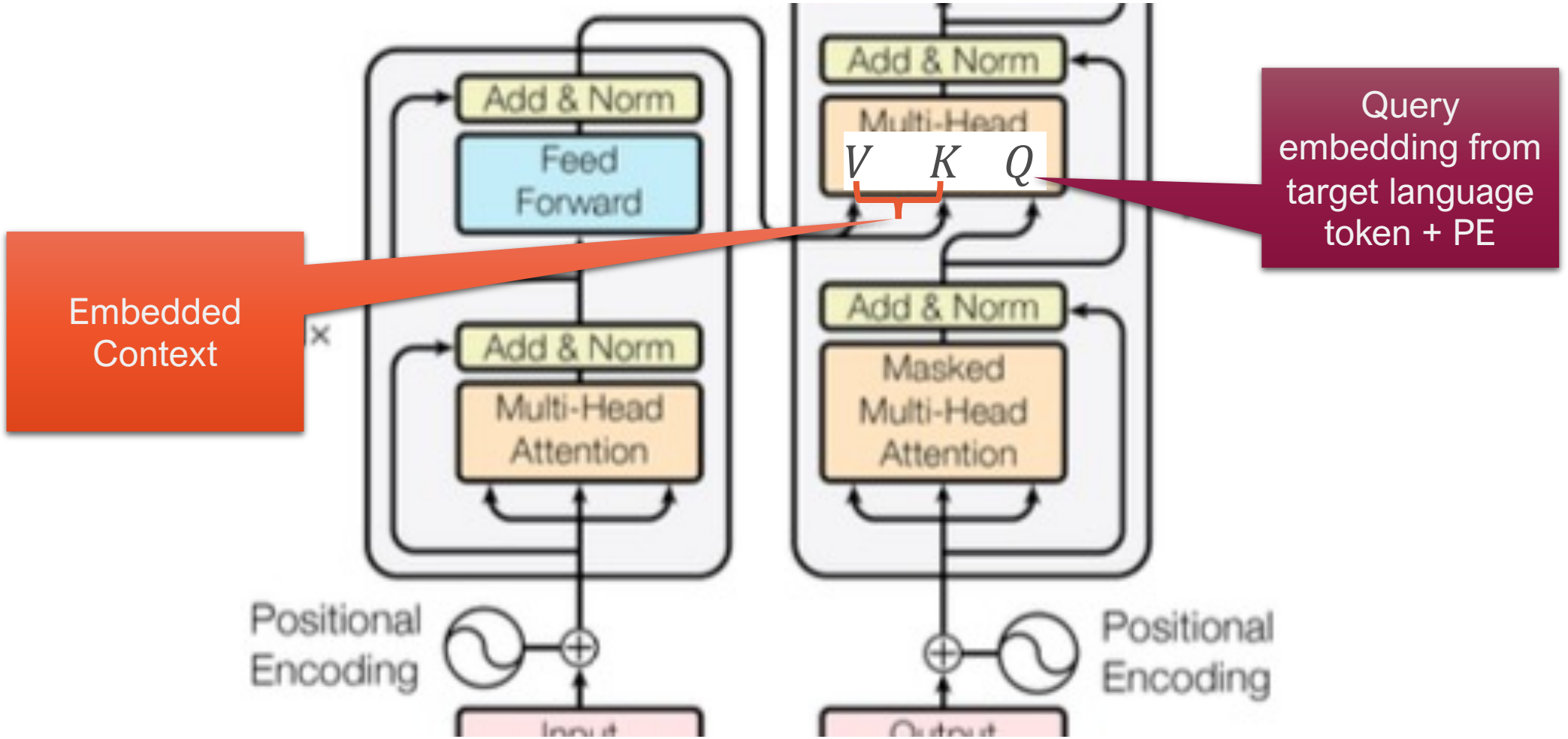


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Transformer



https://ttpoll.com/p/200608

Transformer

Sets values of m_j or m_i to $-\infty$ for $j, i > l$

$$A(q_l, \boldsymbol{K}, \boldsymbol{V}) = \sum_{i=1}^{T} \frac{\exp\left(m_i q_l k_i^T\right)}{\sum_{j=1}^{T} \exp\left(m_j q_l k_j^T\right)} \times v_i$$

The mask prevents the decoder from seeing future tokens during training.

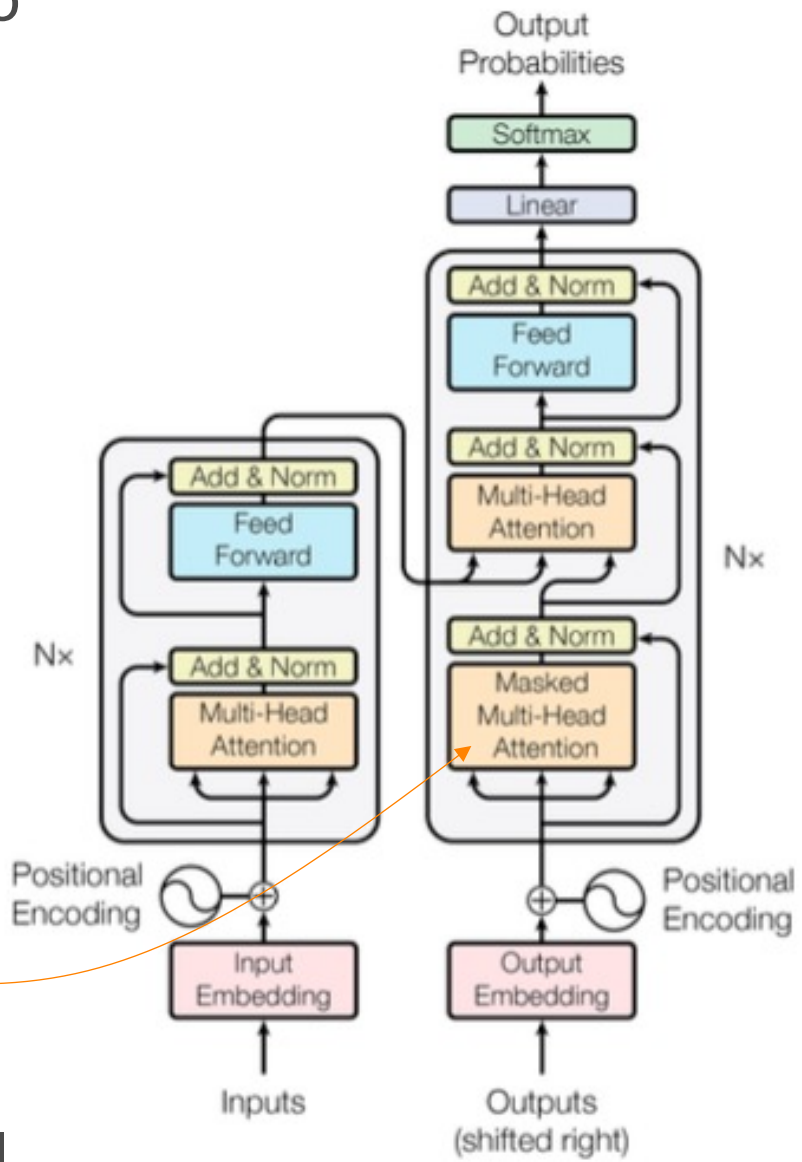


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Transformer

Softmax as large as the vocabulary.

Predict one word at a time.

Pick the word with the highest probability.

Pass the predicted word to decoder input.

Continue until <EOS>

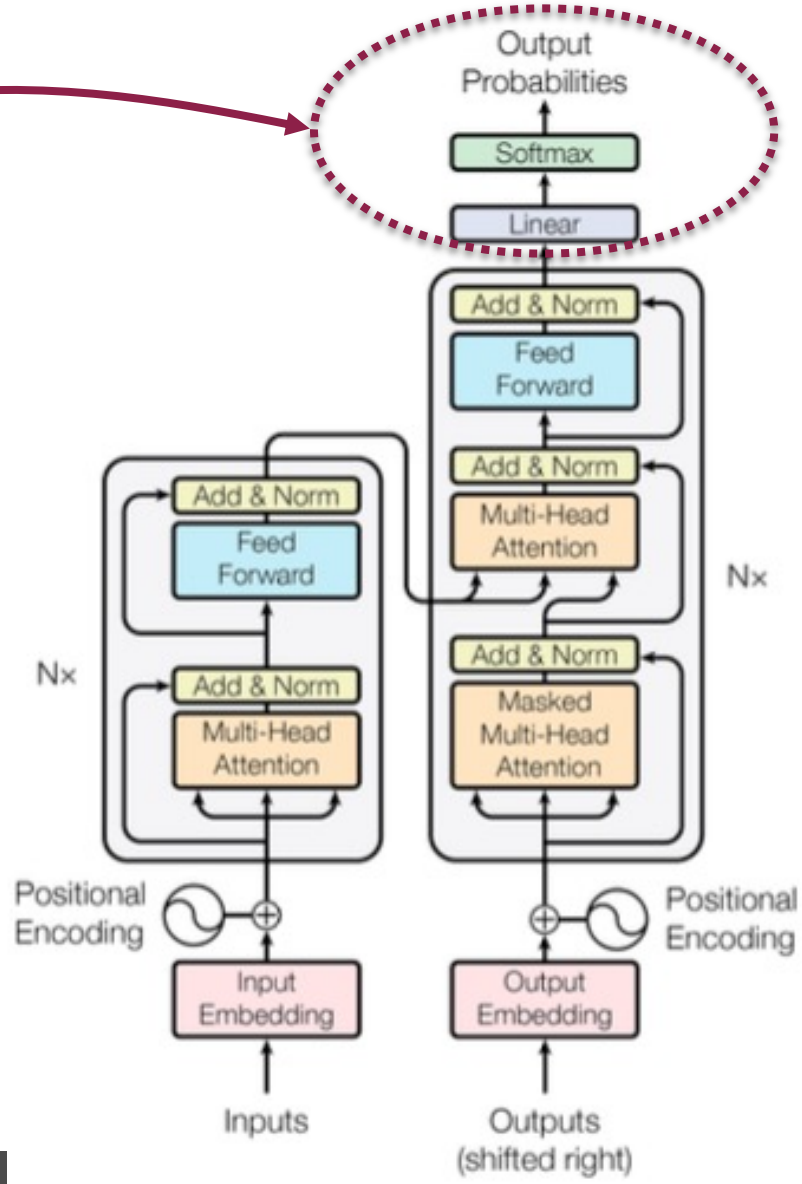
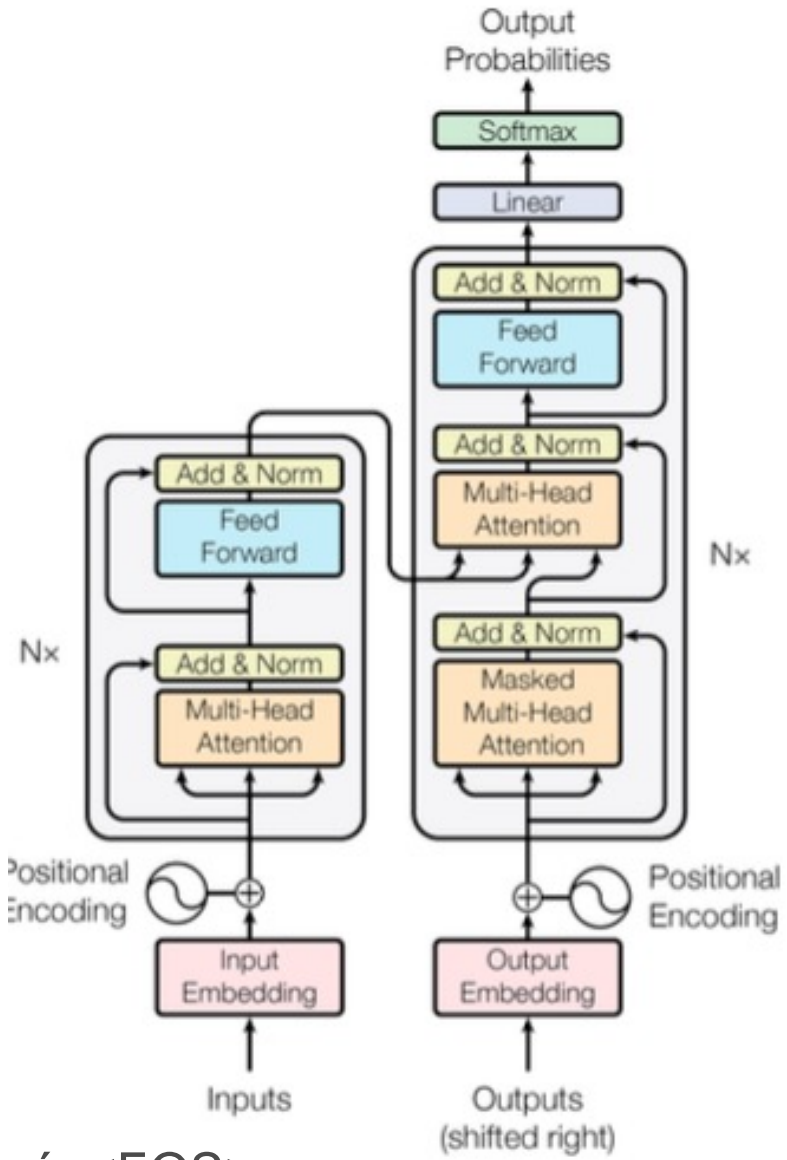


Figure 1: The Transformer - model architecture.

https://ttpoll.com/p/200608

Transformer

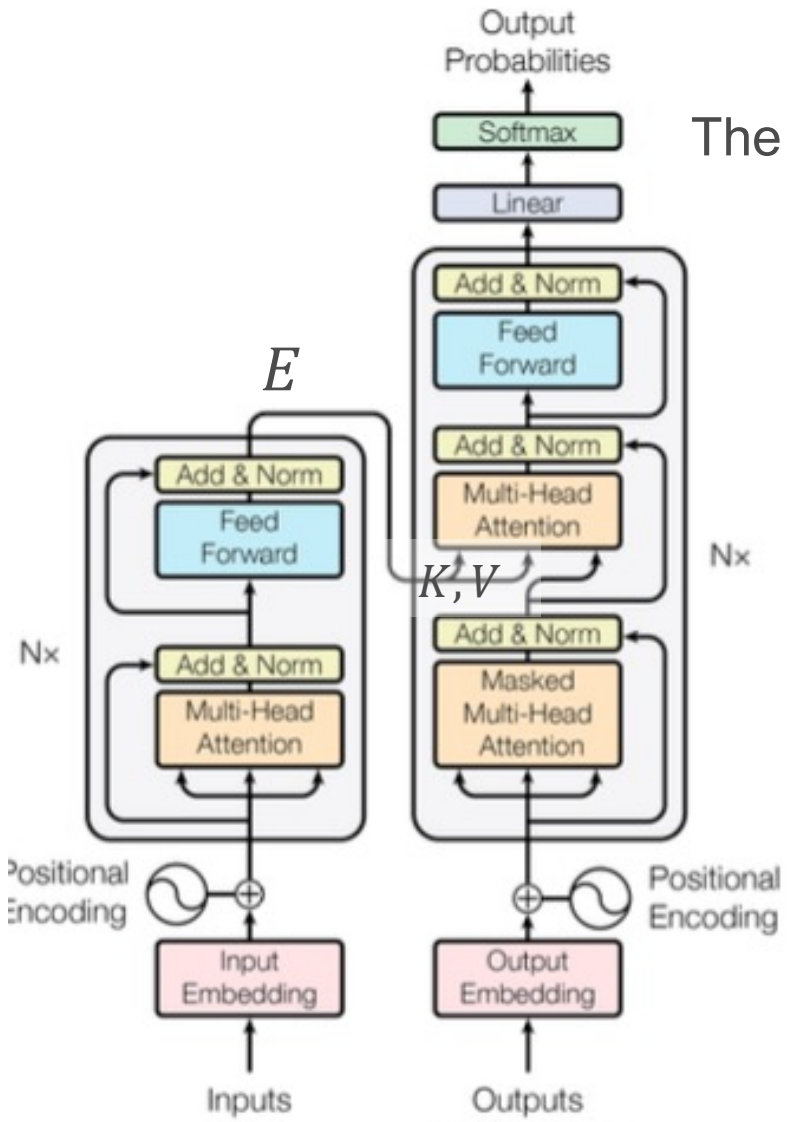


<SOS> La zorra huyó. <EOS>

<SOS> The fox ran away. <EOS>

https://ttpoll.com/p/200608

Transformer



The

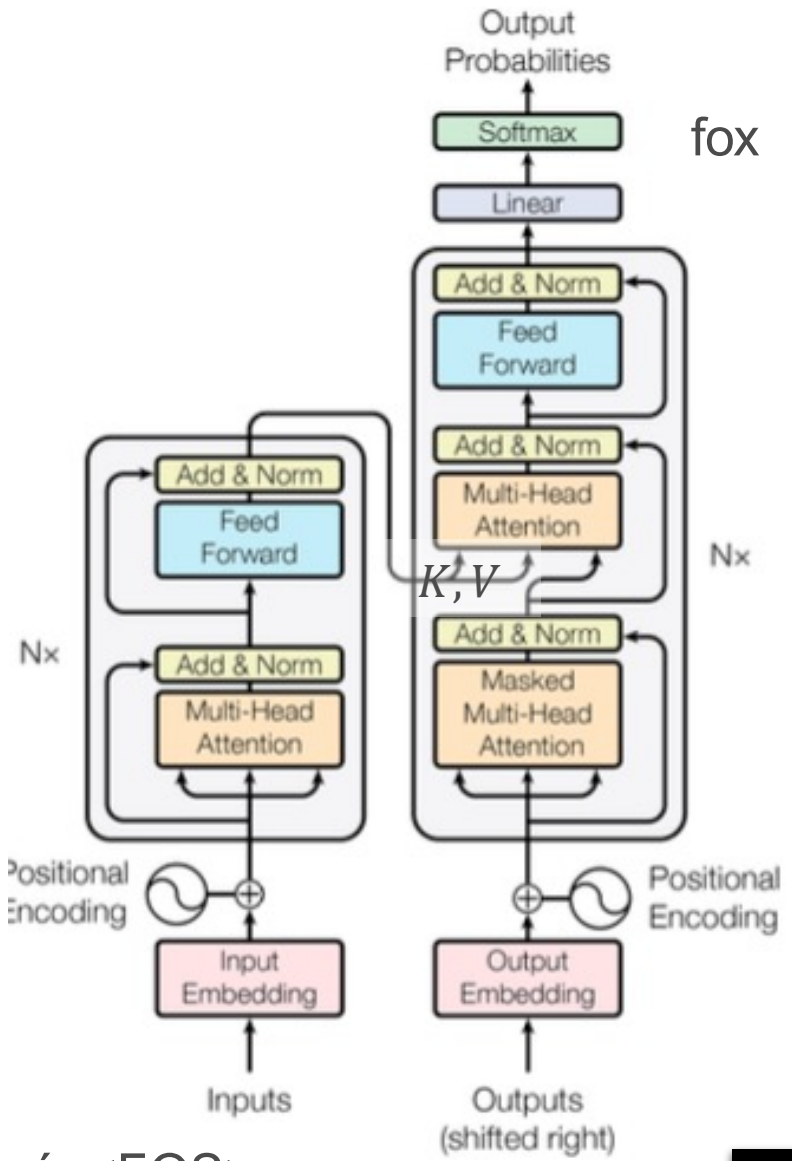
Output: The

<SOS> La zorra huyó. <EOS>

<SOS> [redacted]

https://ttpoll.com/p/200608

Transformer



fox

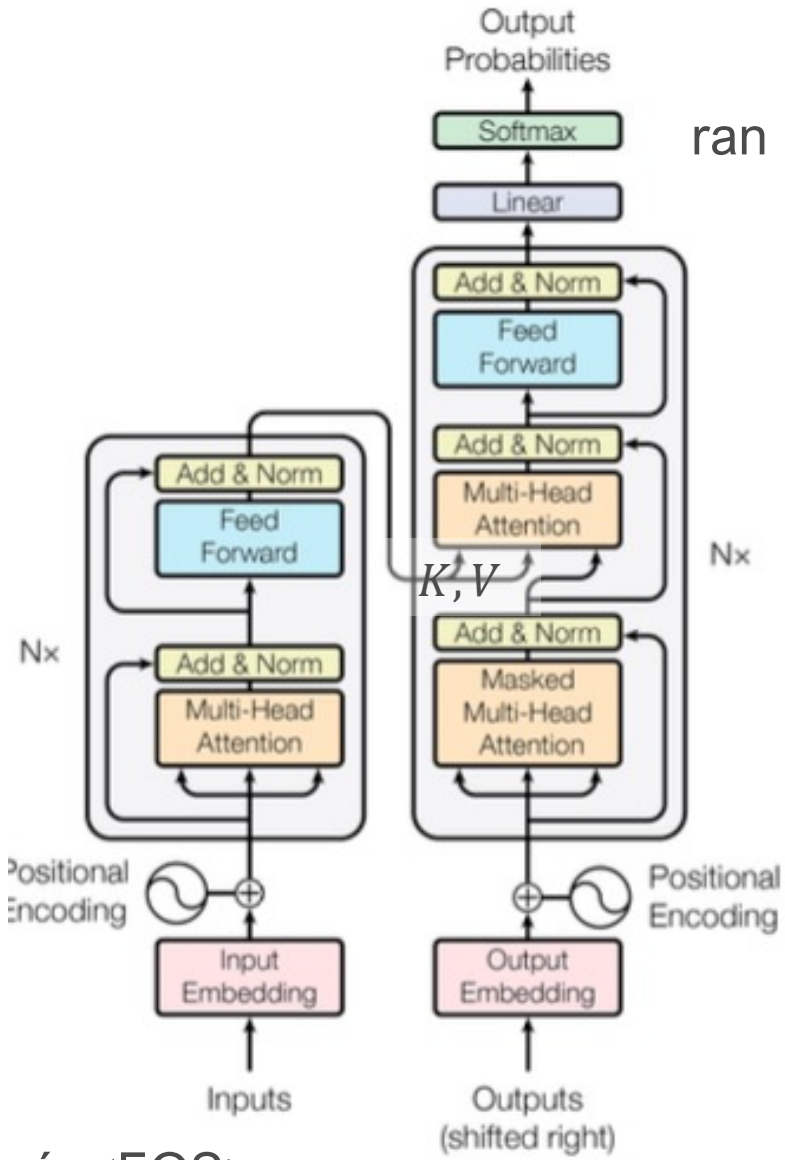
Output: The fox

<SOS> La zorra huyó. <EOS>

<SOS> The [redacted]

https://ttpoll.com/p/200608

Transformer



ran

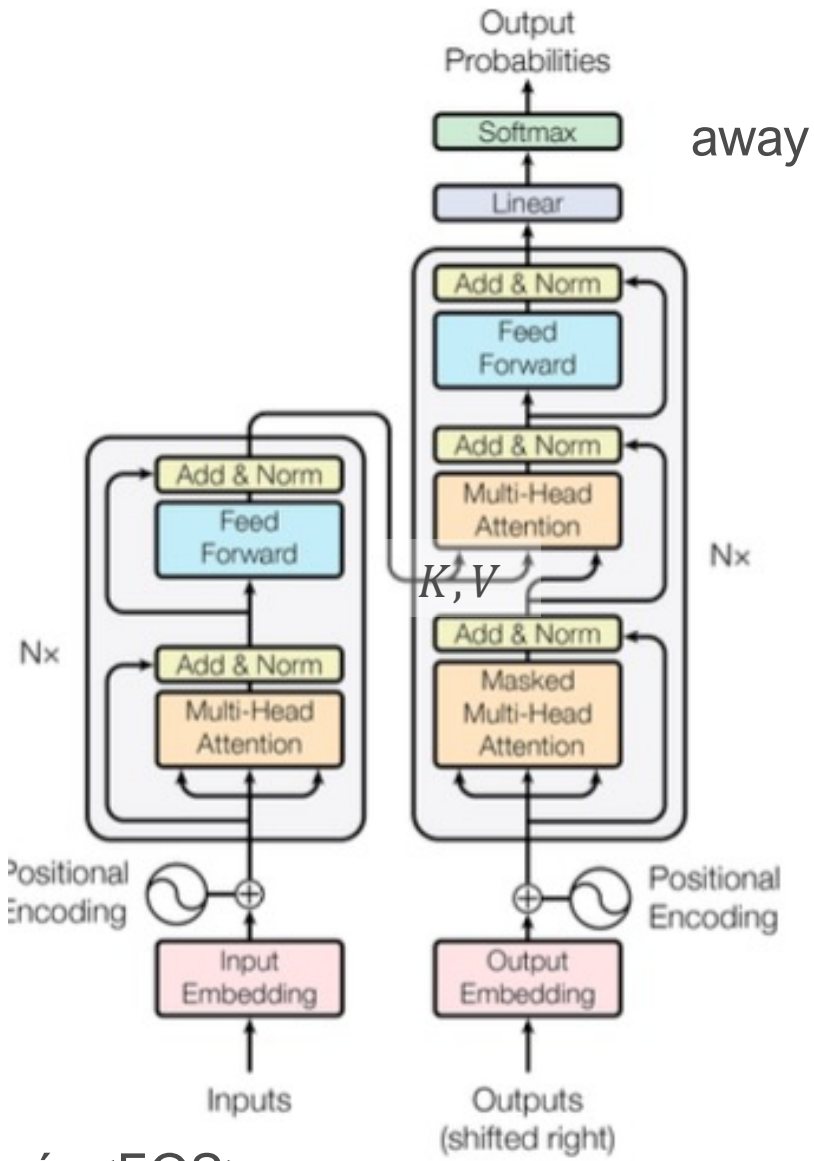
Output: The fox ran

<SOS> La zorra huyó. <EOS>

<SOS> The fox [redacted]

https://ttpoll.com/p/200608

Transformer



away

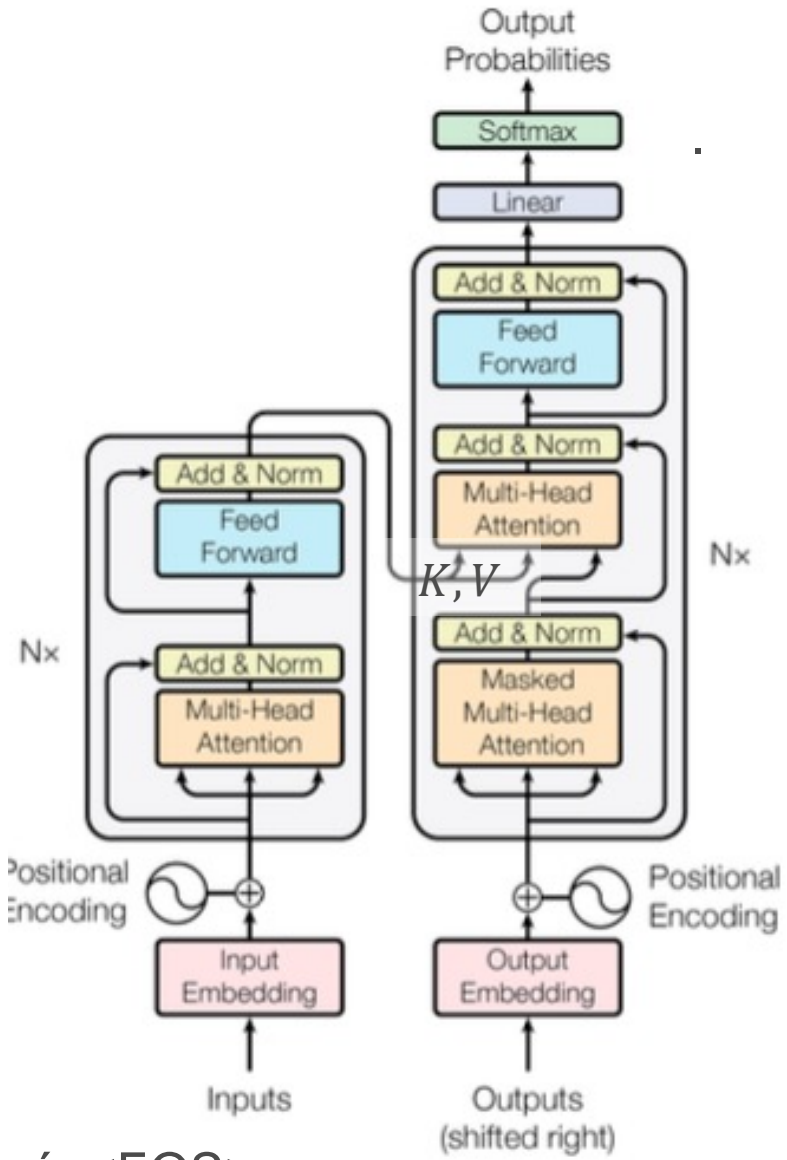
Output: The fox ran away

<SOS> La zorra huyó. <EOS>

<SOS> The fox ran [redacted]

https://ttpoll.com/p/200608

Transformer



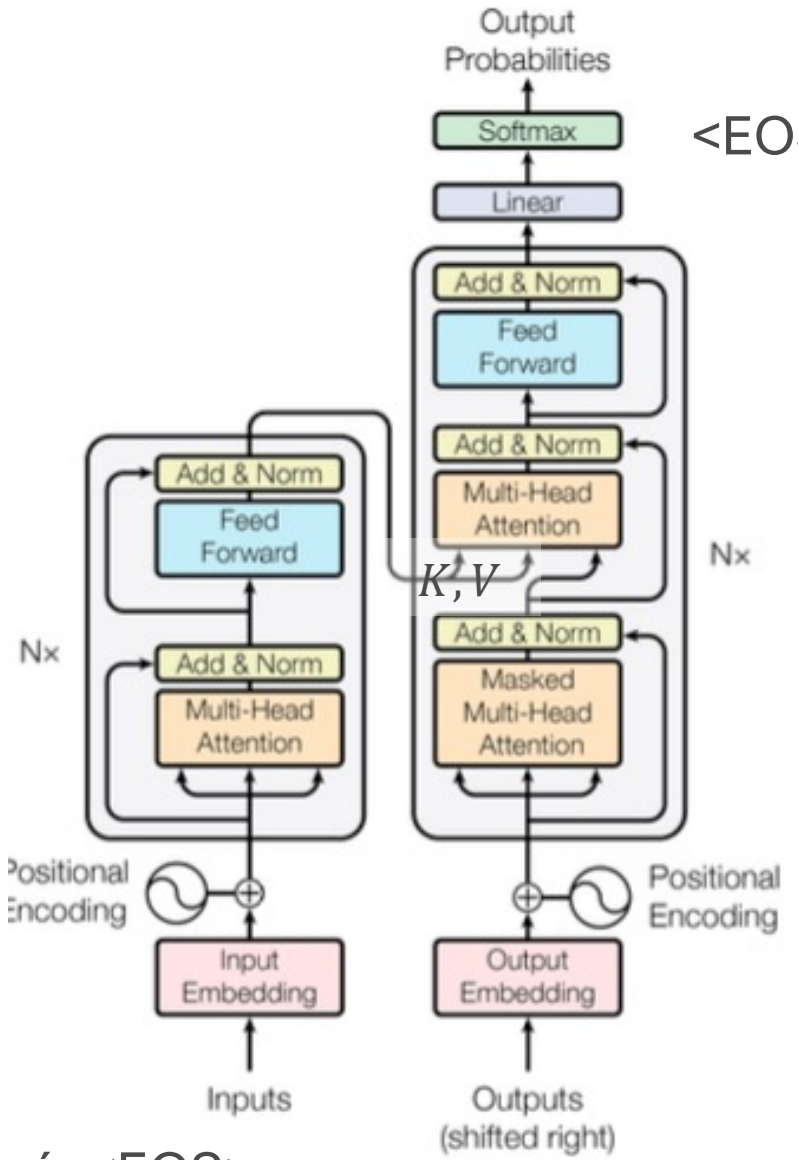
Output: The fox ran away.

<SOS> La zorra huyó. <EOS>

<SOS> The fox ran away [redacted]

https://ttpoll.com/p/200608

Transformer



<EOS>

Output: The fox ran away.

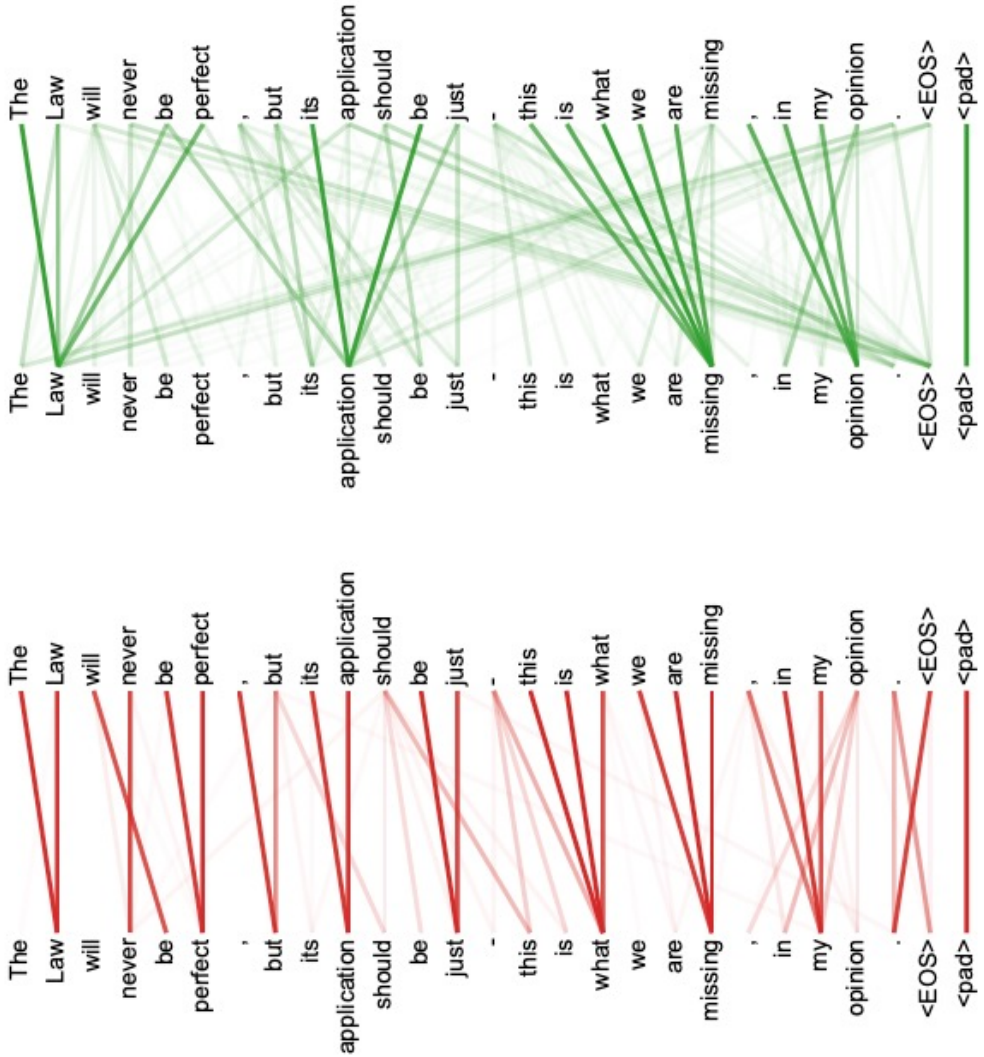
<SOS> La zorra huyó. <EOS>

<SOS> The fox ran away. [redacted]

https://ttpoll.com/p/200608

Attention Example

Different attention heads



https://ttpoll.com/p/200608

3Blue1Brown Visual Tutorials

LLMs Tutorial

Transformer Tutorial

Attention Tutorial

LLM Knowledge Storage Tutorial

https://ttpoll.com/p/200608

Course Project Presentation Details

(

Course Project Presentations

- Presentation slides submitted by Monday, April 28th, 11:59 pm.
  - I will compile a slide deck for each day
  - Large demo videos:
    - Publish the videos on YouTube and add their links to the presentation.
- Presentation day (10 minutes per team)
  - 8 mins for project presentation
    - Each team member with a speaking role
  - 2 mins for Q&A
  - 2 mins for transition and peer score

Rubric

- **Motivation (2%)**: Clear motivation presentation (i.e., problem, importance, solution).
- **Technical Quality (5%)**: Information on model design choices and improvements.
- **Results (3%)**: Clear discussion on model performance and impact of improvement.
- **Conclusion (3%)**: Closing and lessons learned.
- **Q&A (2%)**: Left 2 minutes for questions from peers.
- **Peer score (5%)**: Average score from peer assessment.

https://ttpoll.com/p/200608

Schedule

Course Project Presentation Schedule		
Tuesday, April 29th		
	Team 7	Debris Collection Agent
	Team 8	Super Resolution/Physics
	Team 14	Salary Prediction
	Team 11	Bird classification via sound
	Team 5	Generative Image Detection
	Team 6	Generative Image Detection
Thursday, May 1st		
	Team 4	Tumor Segmentation
	Team 17	MRI GI Tumor segmentation
	Team 15	Knee MRI Classification
	Team 13	Bench press counter
	Team 12	Squat phase classification
	Team 18	Animal Pose Estimation
Tuesday May 6th		
	Team 1	Facial Expressions
	Team 2	Facial Emotions
	Team 9	Image Caption Generation
	Team 3	Poker Agent
	Team 10	Bioinformatics/Malaria Treat. Resist.
	Team 16	Building footprint estimation

https://ttpoll.com/p/200608

Course Project Presentation Details

)

Key success behind transformers

- Self-attention mechanism to capture long-term dependencies
- Self-supervision for large-scale model training
- Typical transformer training steps
  - Pre-training on large unlabeled datasets (Self-supervised)
  - Task-specific training on labeled data (Supervised)
    - Fine-tuning (Retraining existing architecture)
    - Feature-based (Use L-layer output as embedding for a downstream task)

Foundational Model

GPT (Generative Pre-trained Transformer)

- Developed by OpenAI
- GPT-1 Unidirectional: Predicts the next word in a sentence

- Models
  - GPT-1 (2018): 110 million parameters
  - GPT-2 (2019): 1.5 billion parameters
  - GPT-3 (2020): 1.75 billion parameters
  - GPT-4 (2023): 8x220 billion parameters (unofficial)
    - Mixture of experts (MoE)

GPT-1 Key Contributions

- Remove labeling bottleneck
  - 2-Step training
    - Generative pre-training (unlabeled, self-supervised learning)
    - Discriminative fine-tuning (labeled, supervised learning)
- Pre-training
  - BookCorpus with 7,000 books
- Architecture
  - Based on the original Transformer decoder architecture

GPT-1 (2018)

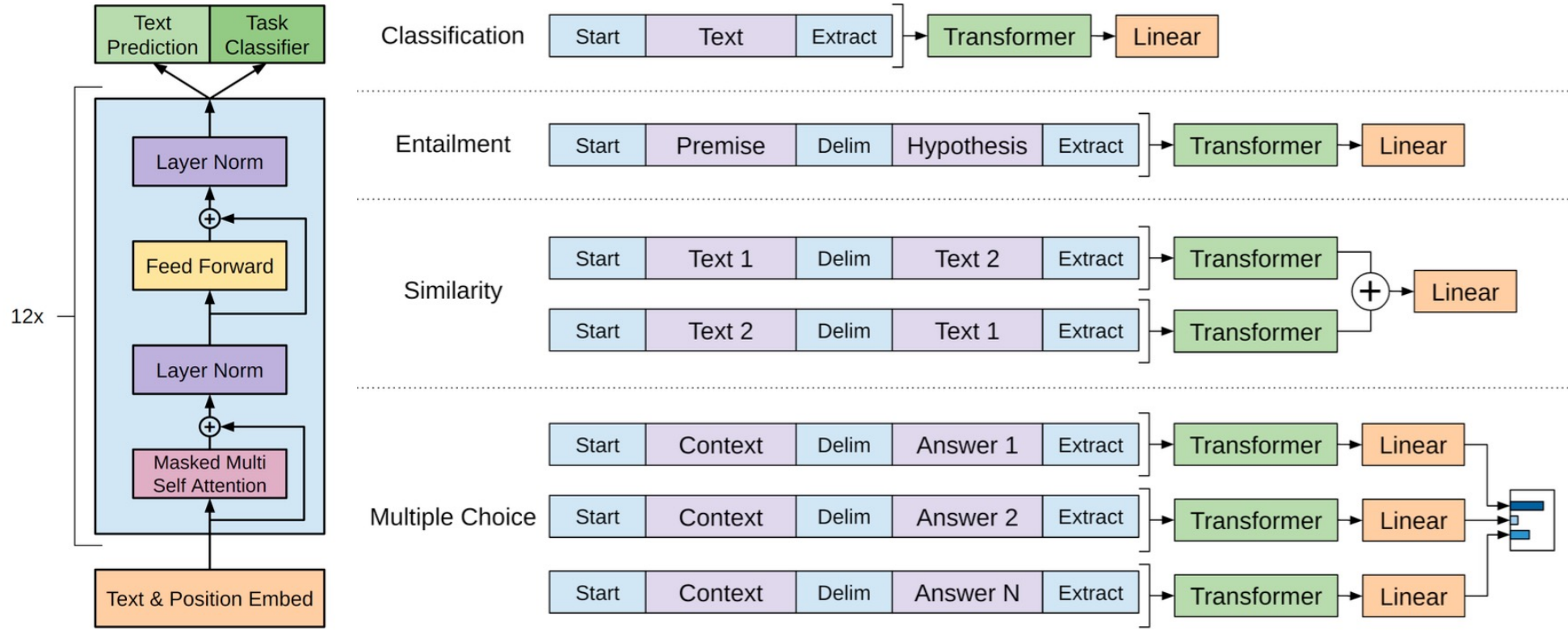


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

GPT-1 (2018)

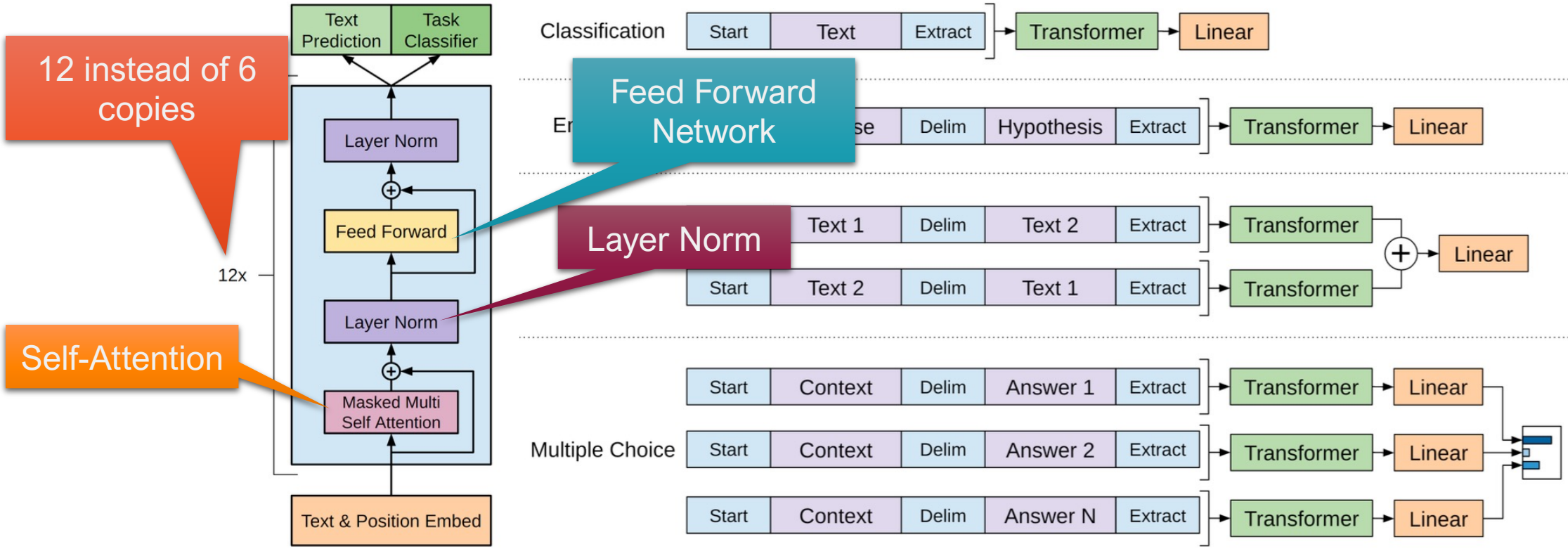


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

https://ttpoll.com/p/200608

GPT-1 (2018)

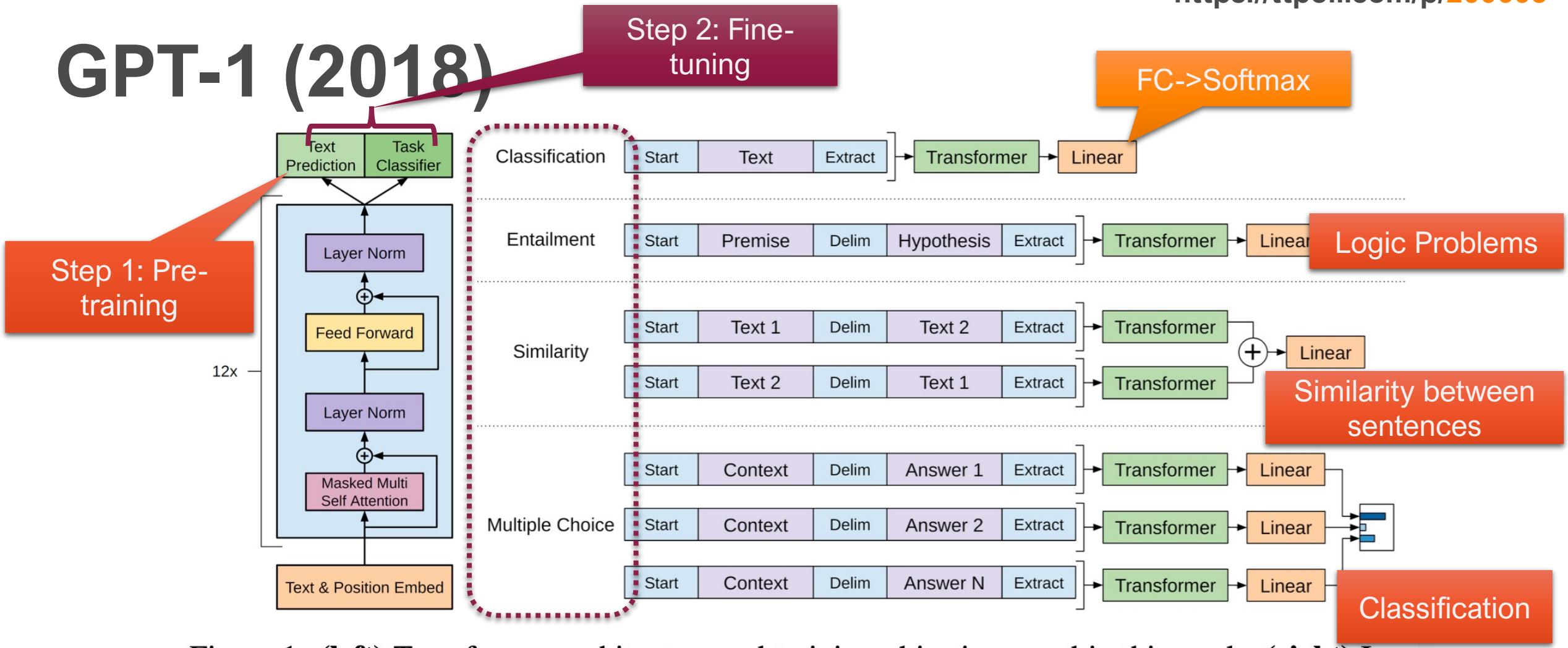


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

GPT-1 Results

Table 1: A list of the different tasks and datasets used in our experiments.

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

GPT-1 Results

The Corpus of Linguistic Acceptability (CoLA) in its full form consists of 10657 sentences from 23 linguistics publications, expertly annotated for acceptability (grammaticality) by their original authors. The public version provided here contains 9594 sentences belonging to training and development sets, and excludes 1063 sentences belonging to a held out test set. Contact alexwarstadt [at] gmail [dot] com with any questions or issues. Read the paper or check out the source code for baselines.

Table 1: A list of the different tasks and datasets used in our ex

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

https://ttpoll.com/p/200608

GPT-1 Results

Table 1: A list of the different tasks and datasets use

Task	Datas
Natural language inference	SNLI [5], MultiNLI [66], Question
Question Answering	RACE [30], Stor
Sentence similarity	MSR Paraphrase Corpus [14], Quora Qu
Classification	Stanford Sentiment Treeb

Table 5: Analysis
of all the results. (

Method	
Transformer w/ aux L	
Transformer w/o pre-	
Transformer w/o aux	
LSTM w/ aux LM	

The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. The corpus is modeled on the SNLI corpus, but differs in that covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation. The corpus served as the basis for the shared task of the RepEval 2017 Workshop at EMNLP in Copenhagen.

Examples

Premise	Label	Hypothesis
Fiction		
The Old One always comforted Ca'daan, except today.	*neutral*	Ca'daan knew the Old One very well.
Letters		
Your gift is appreciated by each and every student who will benefit from your generosity.	*neutral*	Hundreds of students will benefit from your generosity.
Telephone Speech		
yes now you know if if everybody like in August when everybody's on vacation or something we can dress a little more casual or	*contradiction*	August is a black out month for vacations in the company.
9/11 Report		
At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	*entailment*	People formed a line at the end of Pennsylvania Avenue.

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

GPT-1 Results

Table 1: A list of the different tasks and datasets used in our experiments.

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

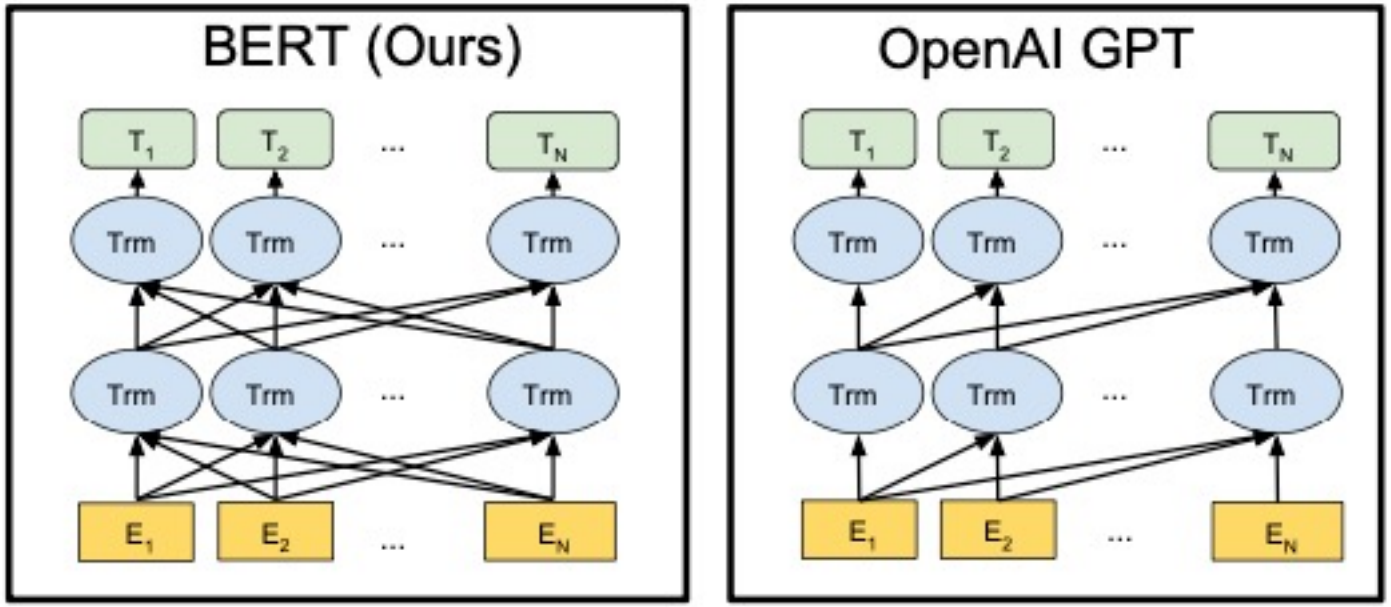
Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

Radford, 2018, Improving Language Understanding by Generative Pre-Training.

BERT: Bidirectional Encoder Representation from Transformers

- Multi-layer bidirectional transformer encoder
- Bidirectional masking
- Next sentence prediction (NSP)
- Datasets
  - BookCorpus (As GPT)
  - Wikipedia



Devlin, et. al., 2019, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Inputs

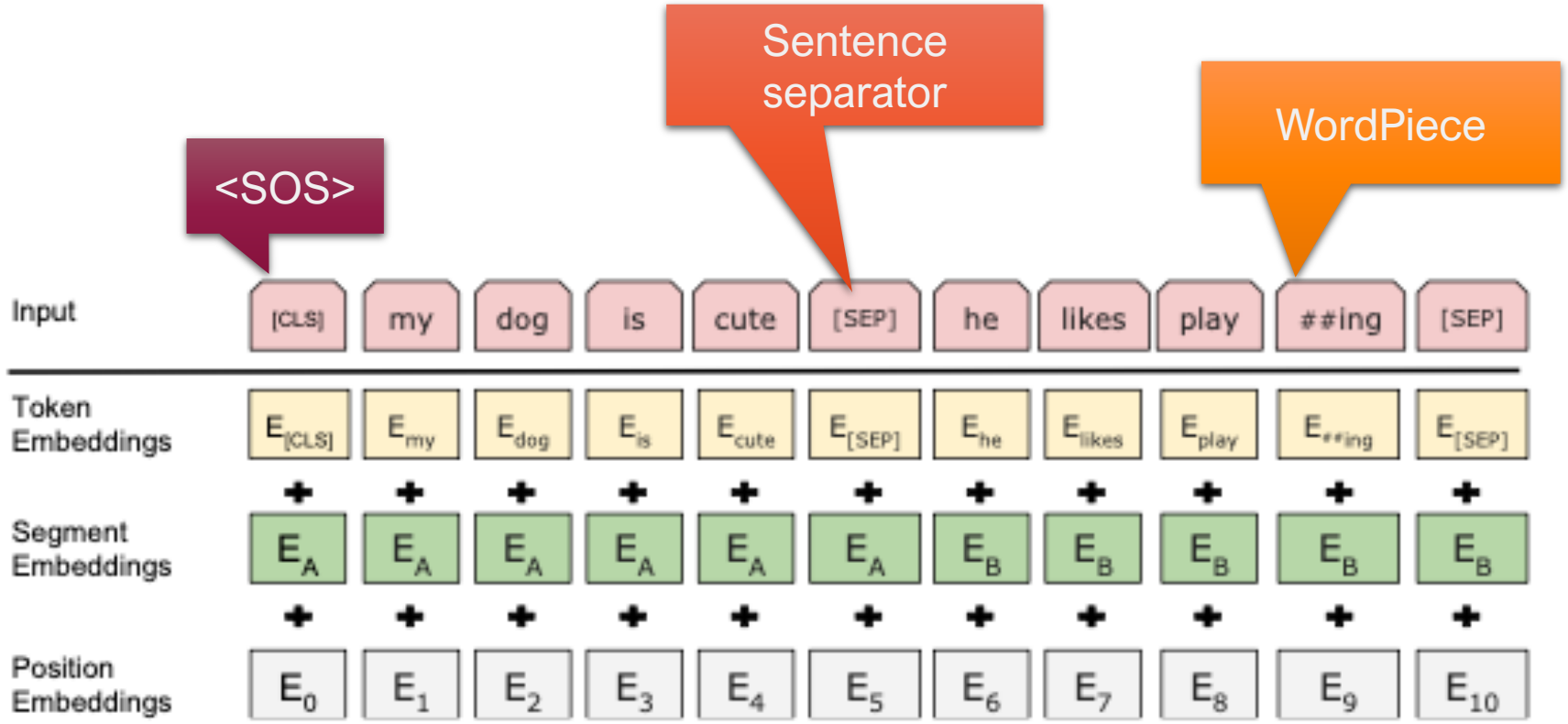
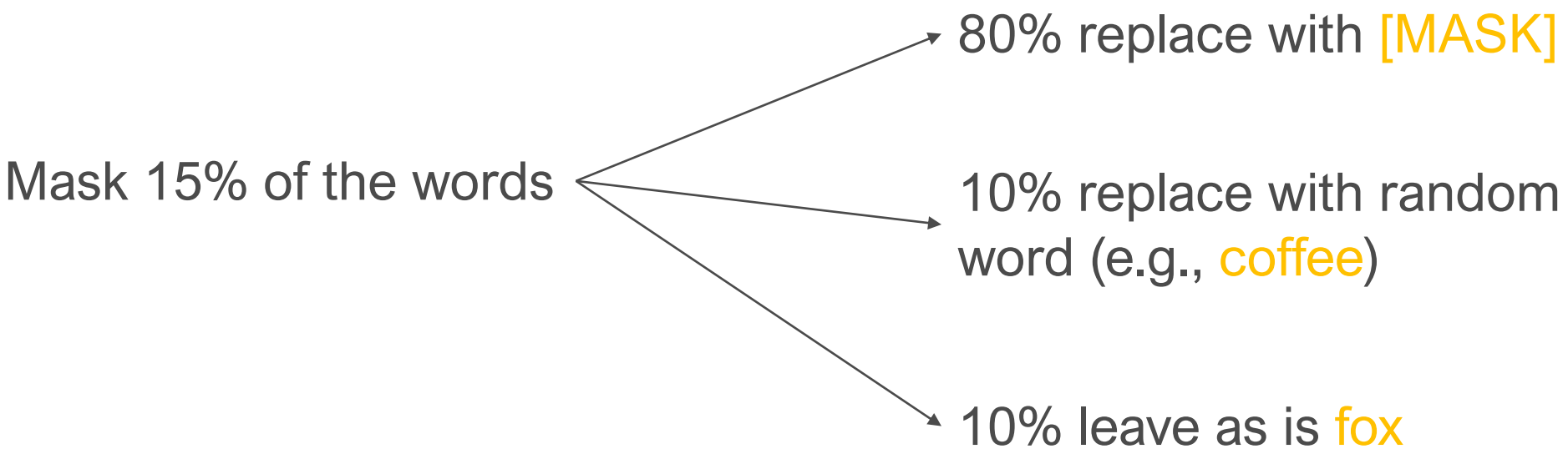


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Pre-Training Tasks

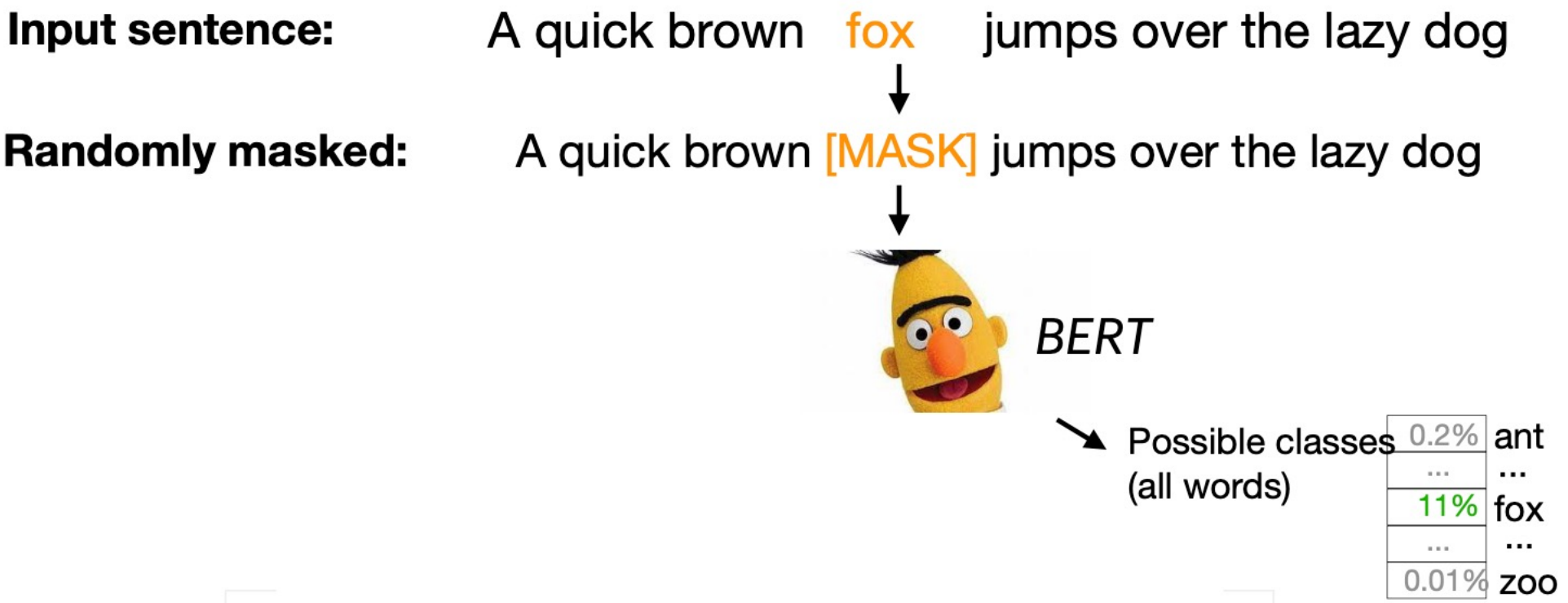
Masked Language Model

Input Sentence: A quick brown fox jumps over the lazy dog.



Slide credit: Sebastian Raschka

BERT Training MLM



Slide credit: Sebastian Raschka

Pre-Training Tasks

Predict Next Sentence

Balanced binary classification task (50% IsNext, 50% NotNext)

Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]
Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]
Label = NotNext

Slide credit: Sebastian Raschka

BERT: Bidirectional Encoder Representation from Transformers

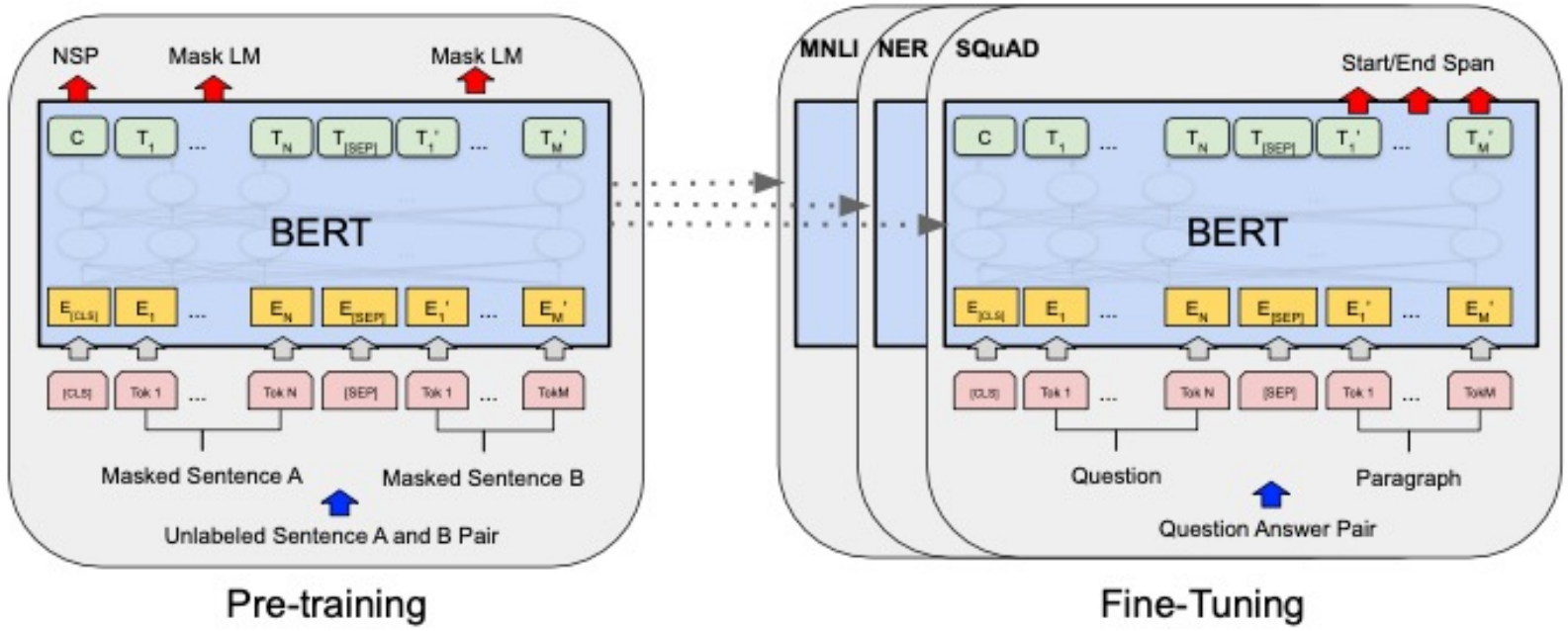
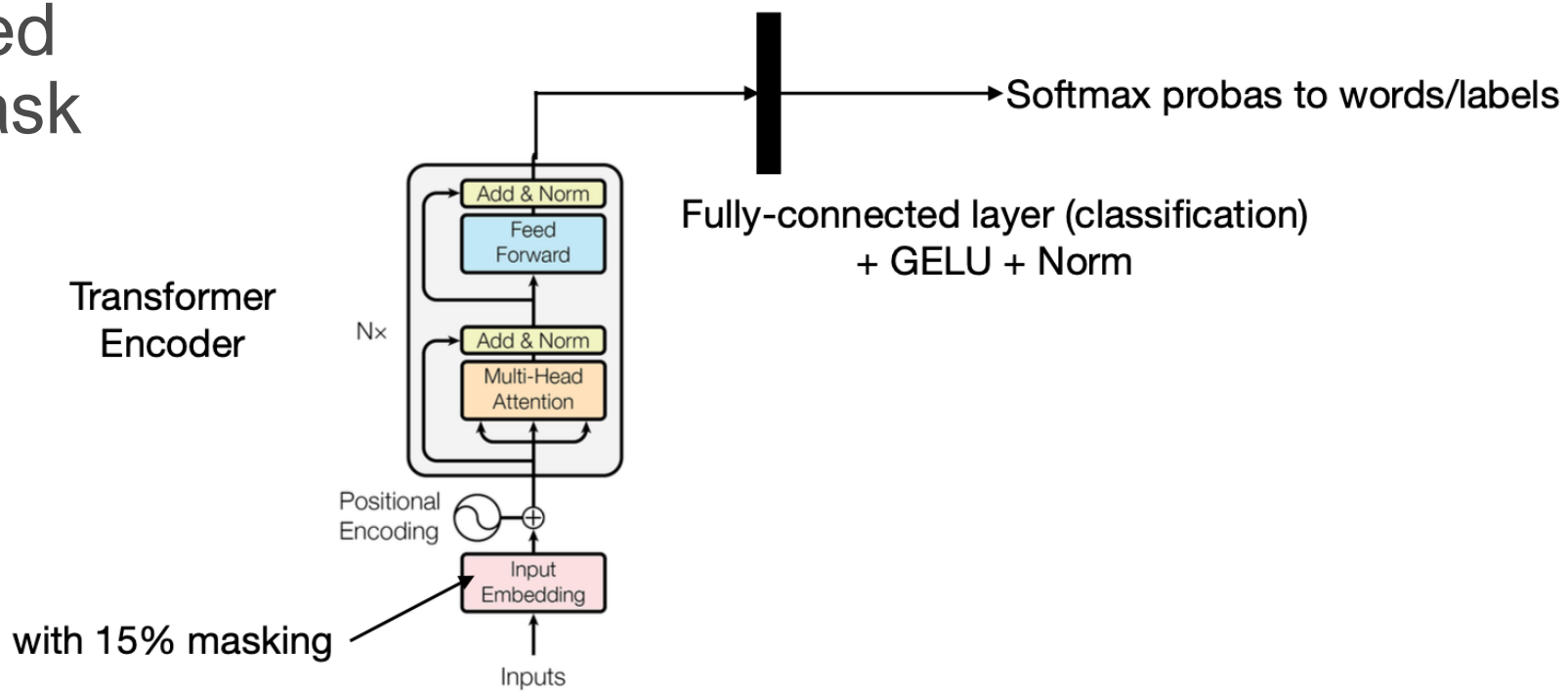


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Devlin, et. al., 2019, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Finetuning

- Add classification layer
- End-to-end supervised training on specific task



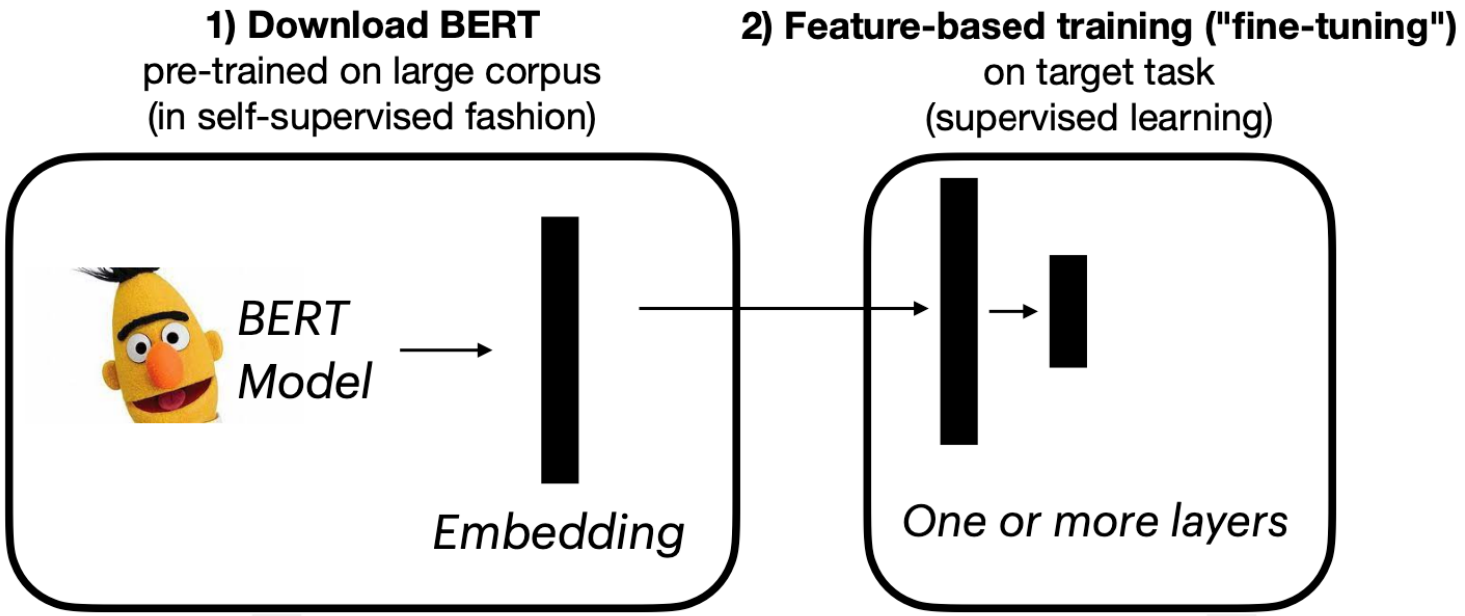
Results

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
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	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Featured-based Training

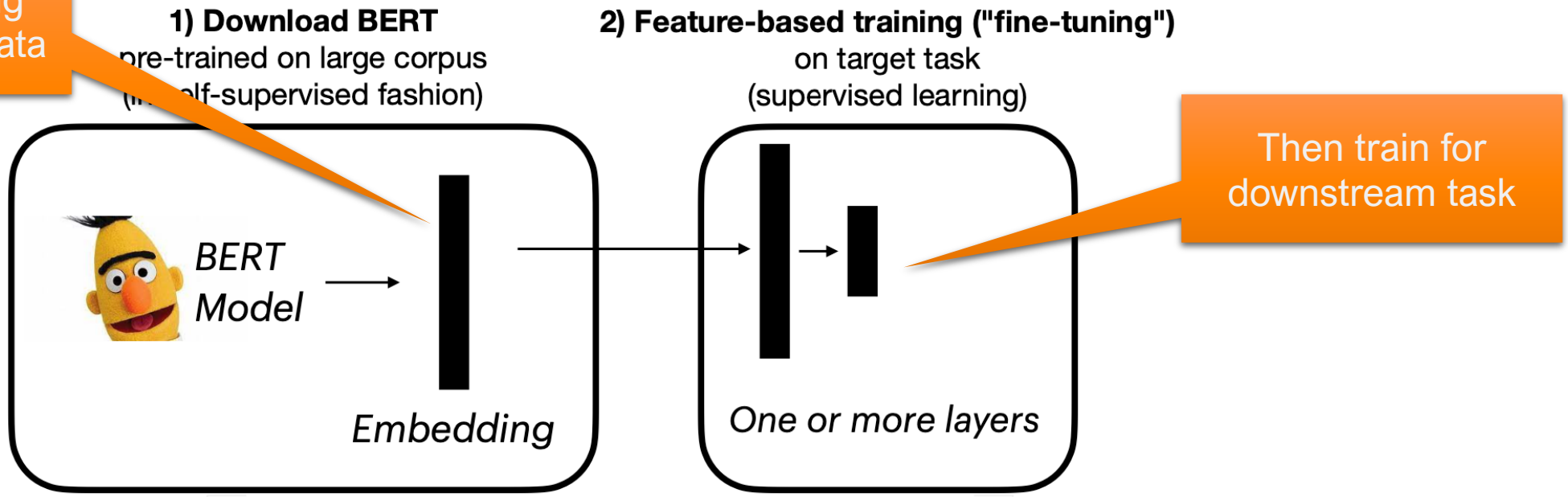
- Keep BERT frozen after pre-training
- Create BERT embeddings for labeled dataset for downstream task and train new model on these embeddings
  (in original paper,
  2-layer biLSTM on embeddings from concatenated last 4 layers performed best)



Featured-based Training

- Keep BERT frozen after pre-training
- Create BERT embeddings for labeled dataset for downstream task and train new model on these embeddings
  (in original paper,
  2-layer biLSTM on embeddings from concatenated last 4 layers performed best)

Train embedding layer with your data



Then train for downstream task

Featured-based Training Results

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

Table 7: CoNLL-2003 Named Entity Recognition results. Hyperparameters were selected using the Dev set. The reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

Other remarks

- GPT-2, and GPT-3 are virtually GPT-1, with more parameters and trained on larger datasets
  - Context window sizes of 512, 1024, and 2048, respectively
- GPT and BERT are foundational models
  - Recall our feature-based training discussion
  - Enable zero-, one-, few-shot learning

Bommasani et. al., 2021, On the Opportunities and Risks of Foundation Models

https://ttpoll.com/p/200608

THE COST OF TRAINING NLP MODELS
A CONCISE OVERVIEW

Or Sharir
AI21 Labs
ors@ai21.com

Barak Peleg
AI21 Labs
barakp@ai21.com

Yoav Shoham
AI21 Labs
yoavs@ai21.com

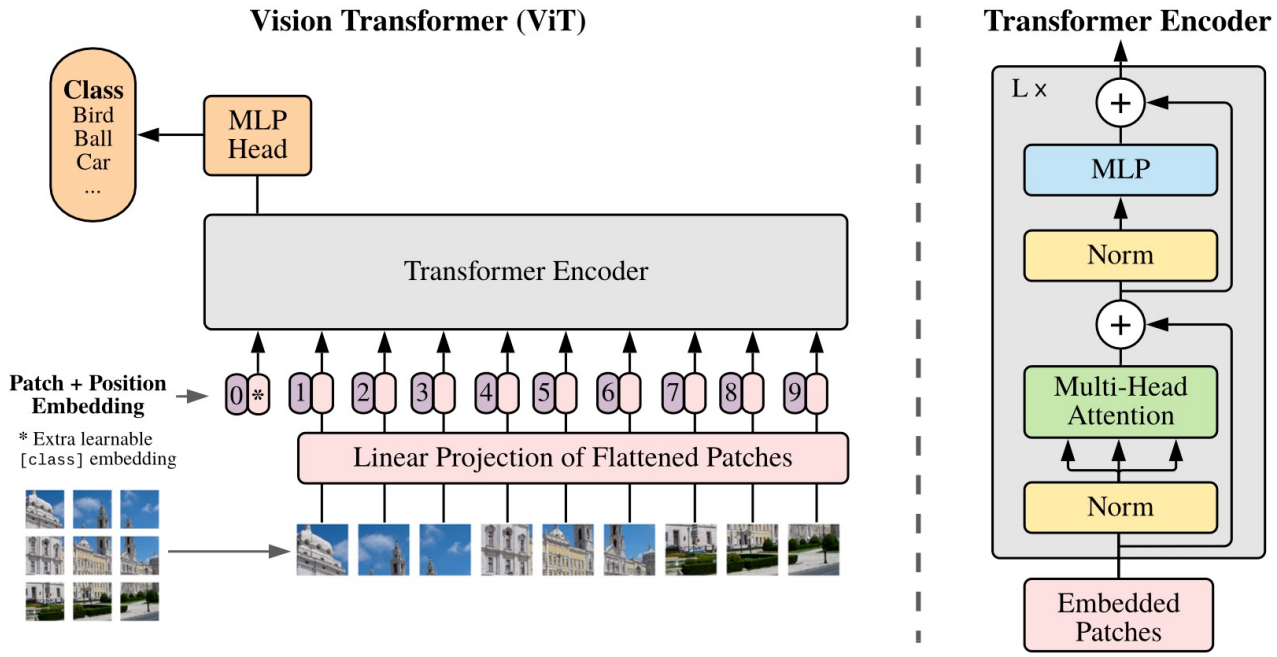
April 2020

http://arxiv.org/abs/2004.08900

Costs: Not for the faint hearted

- $2.5k - $50k (110 million parameter model)
- $10k - $200k (340 million parameter model)
- $80k - $1.6m (1.5 billion parameter model)

Vision Transformers (ViT)



Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Dosovitskiy, 2021, AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Vision Transformers (ViT)

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

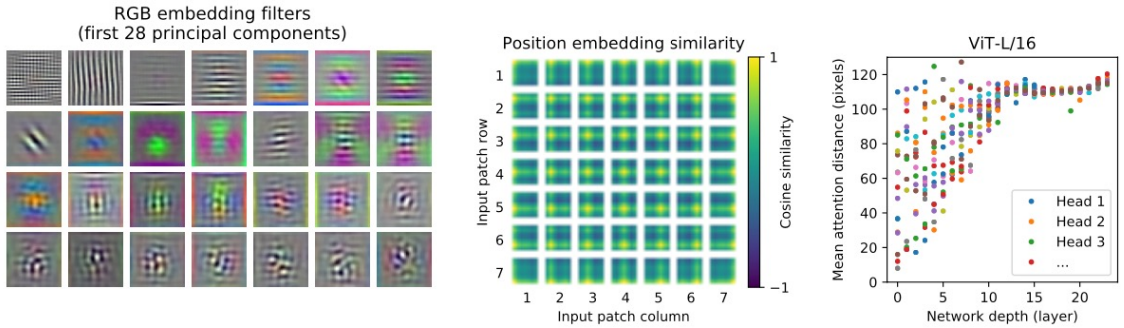


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.7 for details.

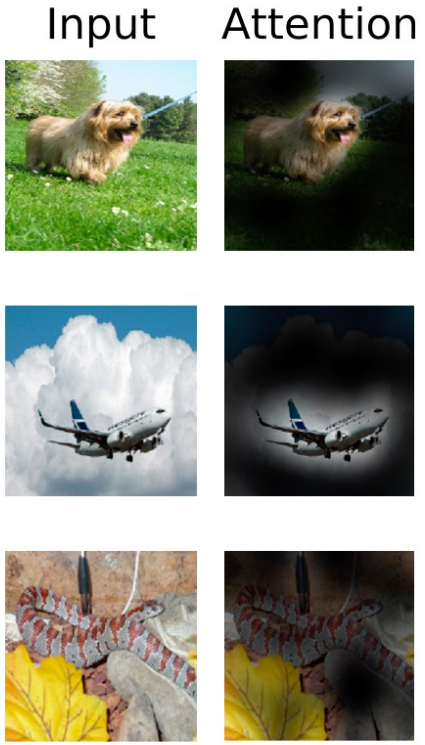


Figure 6: Representative examples of attention from the output token to the input space. See Appendix D.7 for details.

Dosovitskiy, 2021, AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE