Lecture 12: GPT-2

Generative pre-training

Harvard

AC295/CS287r/CSCI E-115B

Chris Tanner


“Are you down with [GPT]?
Yea, you know me!”

ANNOUNCEMENTS

- HW3 has been released! Due Oct 19 (Tues) @ 11:59pm.
- Research Project Phase 2 due Oct 14 (Thurs) @ 11:59pm
- Read "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings" before Oct 14 (Thurs)
- International Collegiate Programming Contest (ICPC) news

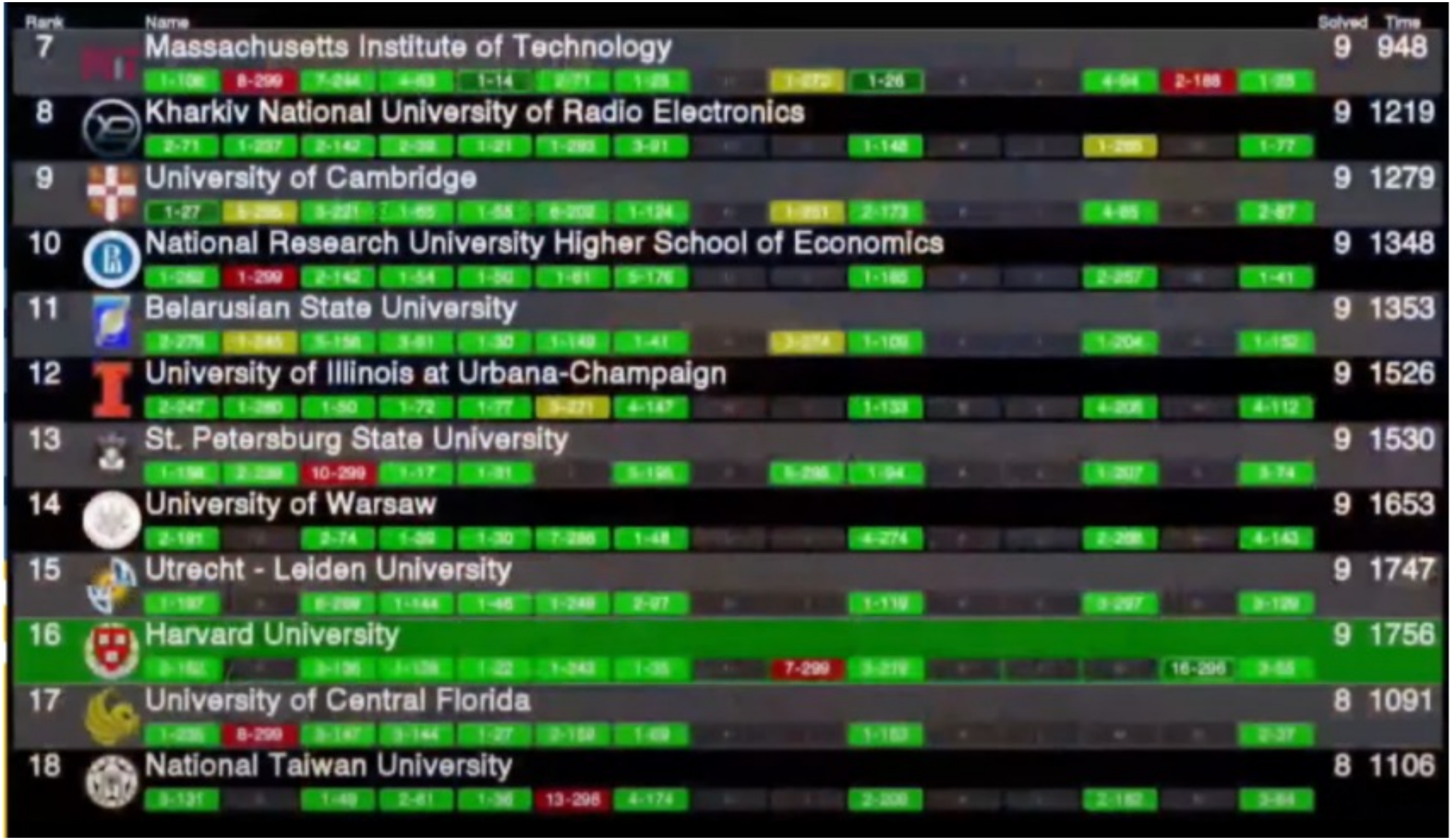
ICPC

The last International Collegiate Programming Contest has hosted over 60,000 students from 3,514 universities in 115 countries that span the globe. October 5, more than 100 teams competed in logic, mental speed, and strategic thinking at Russia’s main Manege Central Conference Hall.

RANK	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Northern Eurasia Nizhny Novgorod State University	12 1714	172 1 try	123 2 tries	99 3 tries	28 2 tries	36 1 try	109 2 tries	76 1 try	287 2 tries	227 3 tries	60 1 try		36 tries	152 3 tries		65 5 tries
2	Asia Pacific Seoul National University	11 1068	85 2 tries	143 2 tries	72 4 tries	17 1 try	31 1 try	31 2 tries	49 1 try	16 tries	217 1 try	76 1 try	1 try		185 2 tries		22 1 try
3	St. Petersburg ITMO University	11 1174	70 3 tries	215 2 tries	59 2 tries	68 2 tries	37 1 try	116 1 try	66 1 try	11 tries	187 1 try	102 1 try		11 tries	117 1 try	1 try	37 1 try
4	Moscow Institute of Physics and Technology	11 1664	31 1 try	204 1 try	203 3 tries	110 1 try	48 1 try	214 3 tries	80 2 tries	3 tries	262 1 try	99 1 try			184 2 tries		69 3 tries
5	Europe University of Wroclaw	11 1772	122 1 try	193 4 tries	187 7 tries	60 2 tries	47 1 try	222 1 try	18 1 try	7 tries	255 2 tries	86 2 tries			173 2 tries		109 3 tries
6	University of Cambridge	11 1905	27 1 try	295 5 tries	221 3 tries	65 1 try	55 1 try	202 6 tries	124 1 try		251 1 try	173 2 tries			85 4 tries		87 2 tries
7	Belarusian State University	11 1912	279 2 tries	245 1 try	158 5 tries	91 3 tries	30 1 try	149 1 try	41 1 try		274 3 tries	109 1 try			204 1 try		152 1 try
8	University of Bucharest	10 1077	153 1 try	200 3 tries	39 1 try	13 3 tries	33 1 try	74 1 try	45 1 try		5 tries	240 3 tries			123 2 tries		17 1 try
9	North America Massachusetts Institute of Technology	10 1220	106 1 try	8 tries	244 7 tries	83 4 tries	14 1 try	71 2 tries	25 1 try		272 1 try	26 1 try			94 4 tries	2 tries	25 1 try
10	Kharkiv National University of Radio Electronics	10 1504	71 2 tries	237 1 try	142 2 tries	39 2 tries	21 1 try	293 1 try	91 3 tries			148 1 try			285 1 try		77 1 try
11	University of Illinois at Urbana-Champaign	10 1837	247 2 tries	280 1 try	50 1 try	72 1 try	77 1 try	271 3 tries	147 4 tries			133 1 try			208 4 tries		112 4 tries
12	National Research University Higher School of Economics	9 1348	262 1 try	1 try	142 2 tries	54 1 try	50 1 try	61 1 try	176 5 tries			185 1 try			257 2 tries		41 1 try
13	St. Petersburg State University	9 1530	158 1 try	239 2 tries	10 tries	17 1 try	31 1 try		195 5 tries		295 5 tries	94 1 try			207 1 try		74 3 tries
14	University of Warsaw	9 1653	191 2 tries		74 2 tries	39 1 try	30 1 try	286 7 tries	48 1 try			274 4 tries			268 2 tries		143 4 tries
15	Utrecht - Leiden University	9 1747	197 1 try		269 6 tries	144 1 try	46 1 try	249 1 try	97 2 tries			119 1 try			297 3 tries		129 3 tries
16	Harvard University	9 1756	182 2 tries		136 3 tries	128 1 try	22 1 try	243 1 try	35 1 try		7 tries	219 3 tries				296 16 tries	55 3 tries
17	University of Central Florida	8 1091	235 1 try	8 tries	147 3 tries	144 3 tries	27 1 try	159 2 tries	69 1 try			153 1 try					37 2 tries
18	National Taiwan University	8 1106	131 3 tries		49 1 try	61 2 tries	36 1 try	13 tries	174 4 tries			209 2 tries			182 2 tries		64 3 tries


Harvard University
First to solve problem N
ICPC
WORLD FINALS
MOSCOW
MIPT
VTB
RUSSIAN POST

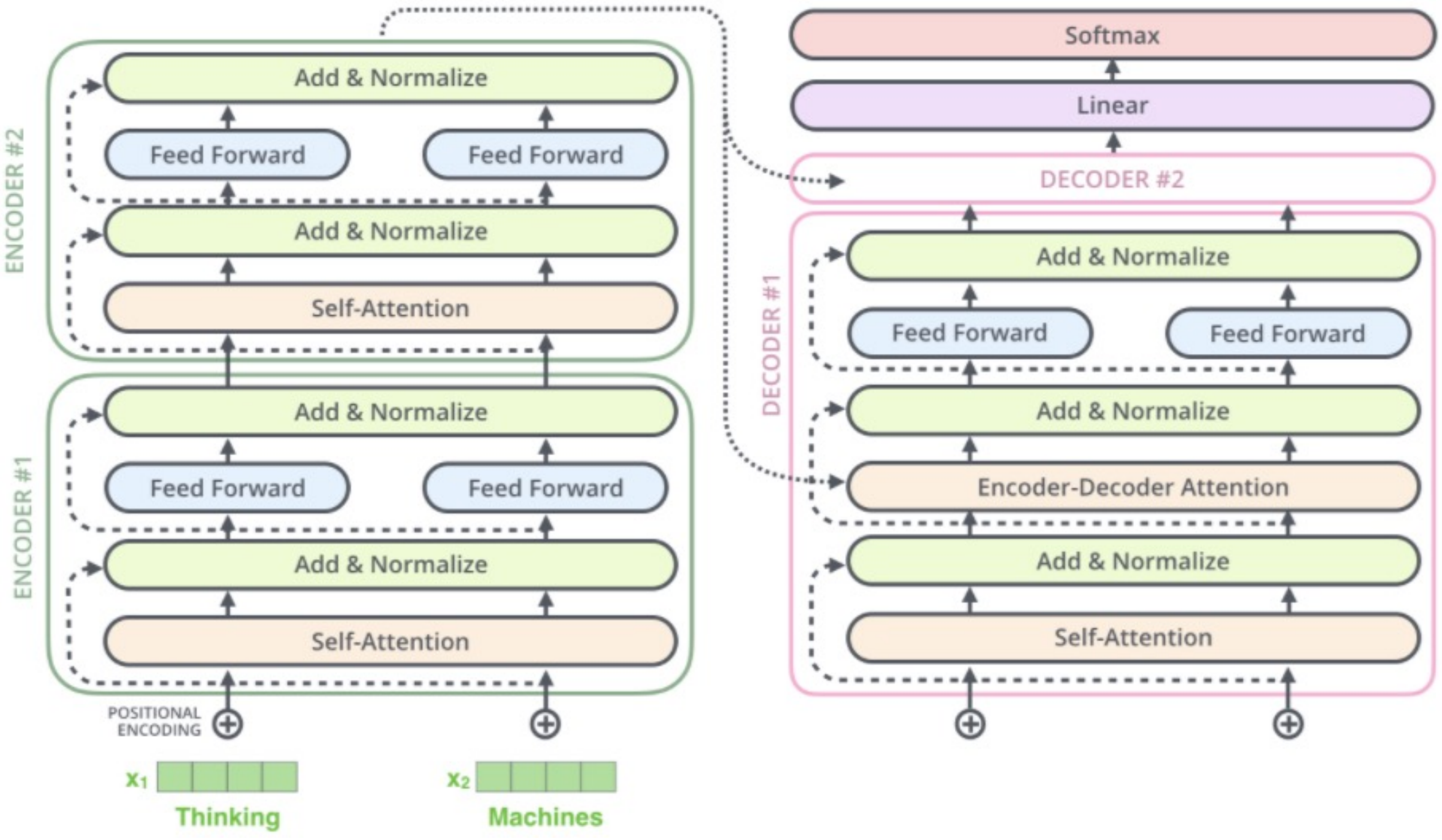

WORLD FINALS
MOSCOW
HOSTED BY MIPT
MANEGE



Rank
Name
Solved
Time
7 Massachusetts Institute of Technology 9 948
8 Kharkiv National University of Radio Electronics 9 1219
9 University of Cambridge 9 1279
10 National Research University Higher School of Economics 9 1348
11 Belarusian State University 9 1353
12 University of Illinois at Urbana-Champaign 9 1526
13 St. Petersburg State University 9 1530
14 University of Warsaw 9 1653
15 Utrecht - Leiden University 9 1747
16 Harvard University 9 1756
17 University of Central Florida 8 1091
18 National Taiwan University 8 1106

RECAP: L10

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



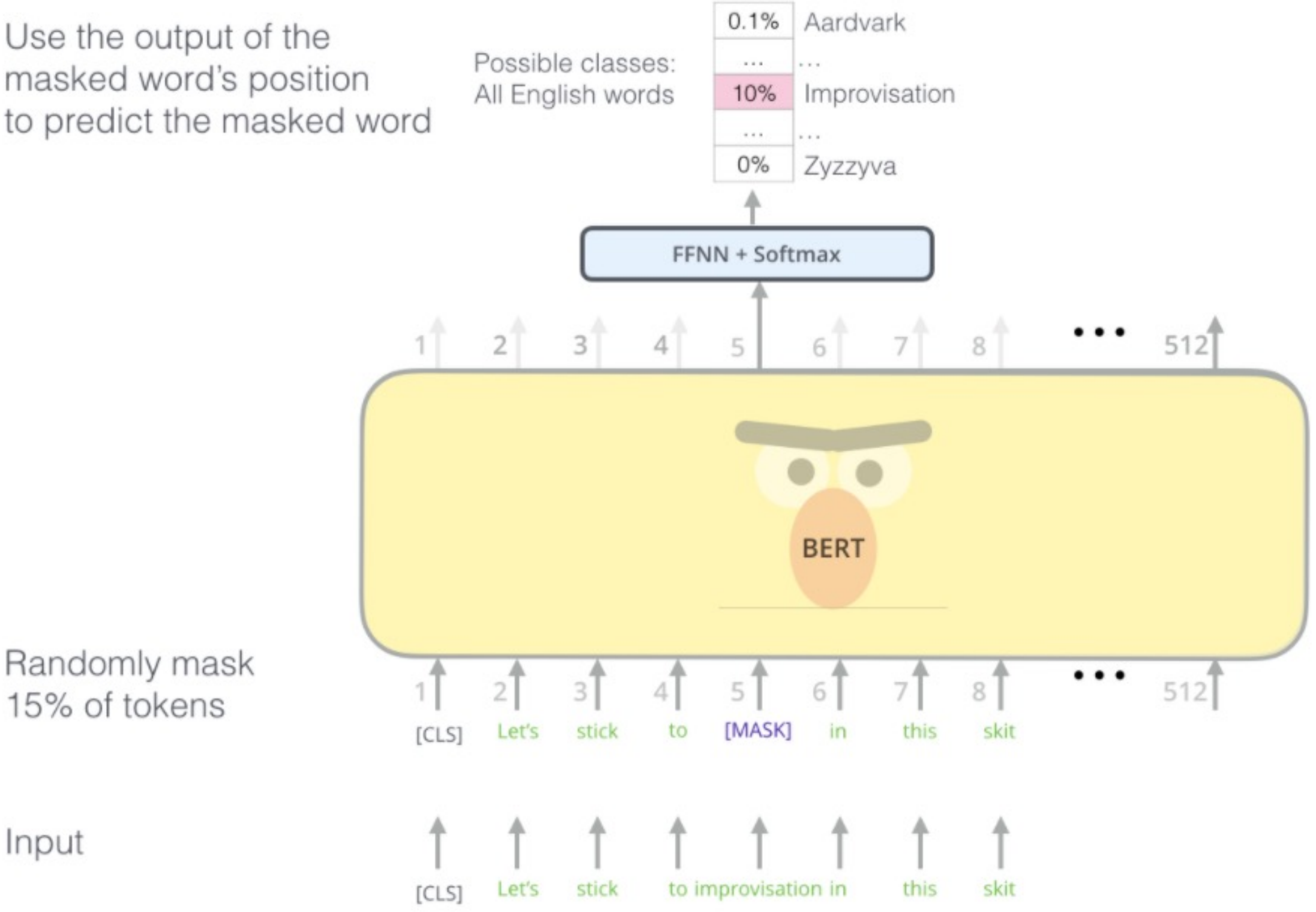
https://jalammar.github.io/illustrated-transformer/

RECAP: L11

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.

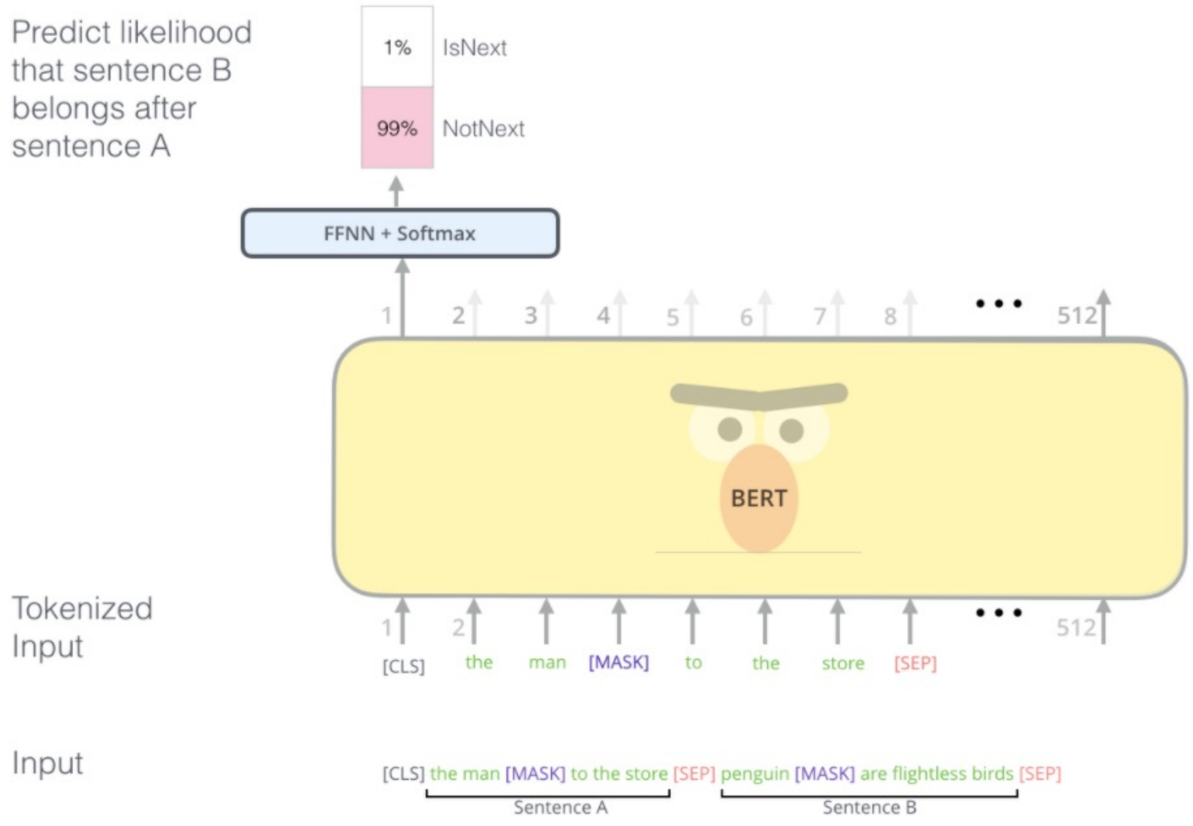
https://jalammar.github.io/illustrated-transformer/

RECAP: L11

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

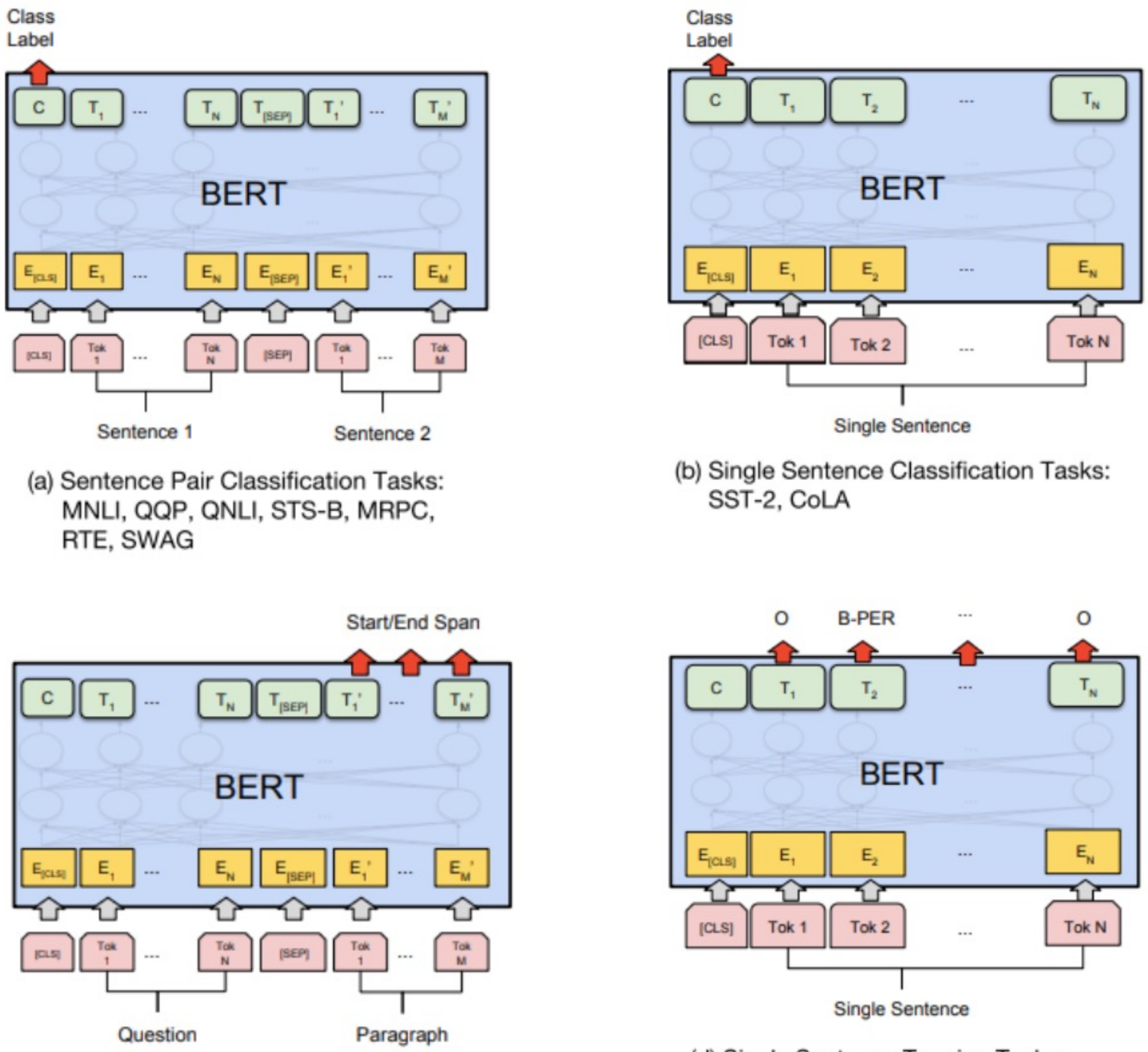


https://jalammar.github.io/illustrated-transformer/

RECAP: L11

BERT is easy to fine-tune on any other classification task

- replace the top layer
- ensure your inputs are tokenized the same way as training, and no OOV tokens
- usually best to allow the original BERT weights to adjust, too (don't freeze)



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

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

Outline

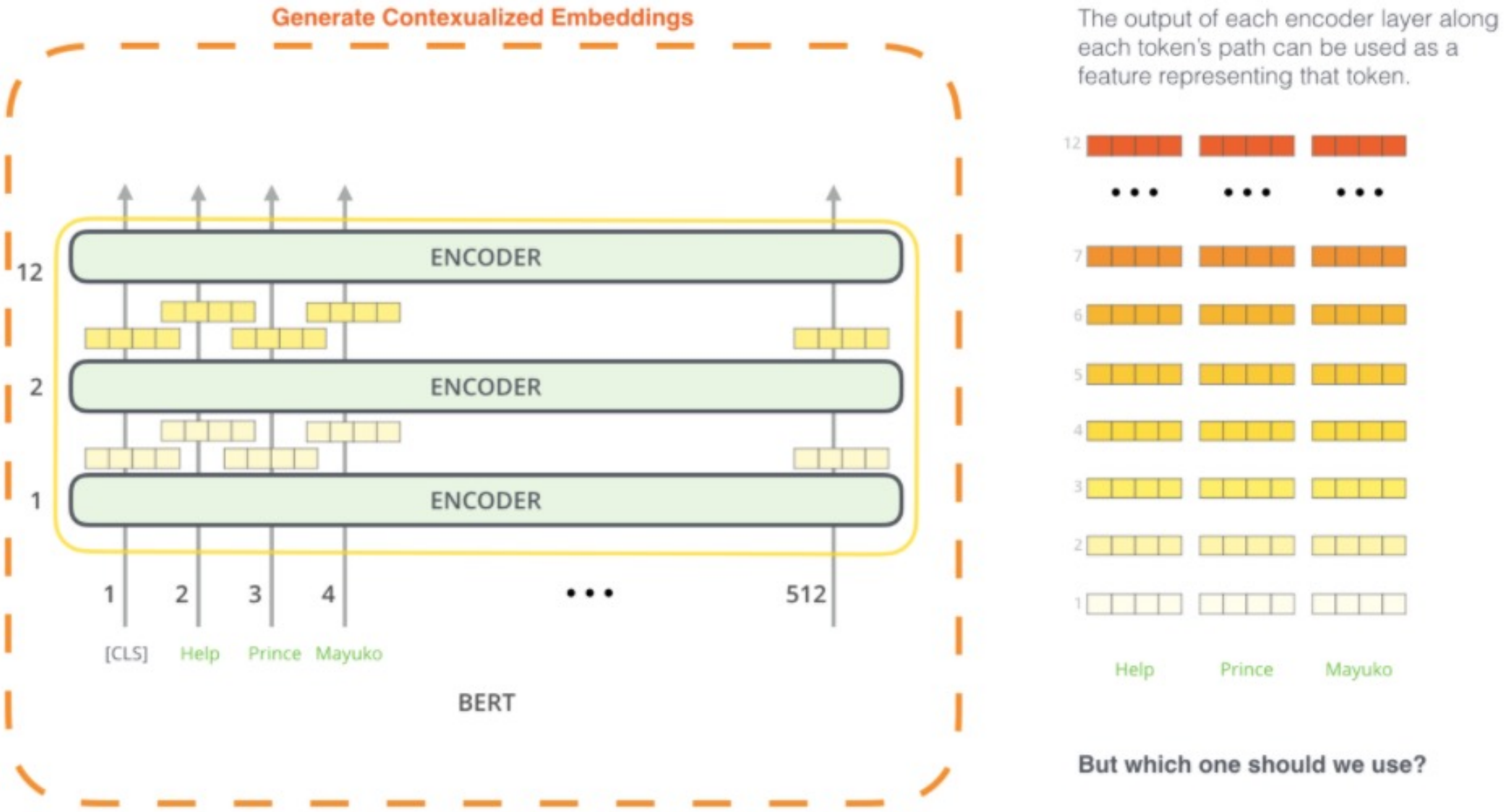
- BERT (finishing up)
- GPT-2
- Issues and remaining work

Outline

- BERT (finishing up)
- GPT-2
- Issues and remaining work

BERT

Instead of fine-tuning, one could extract the contextualized embeddings

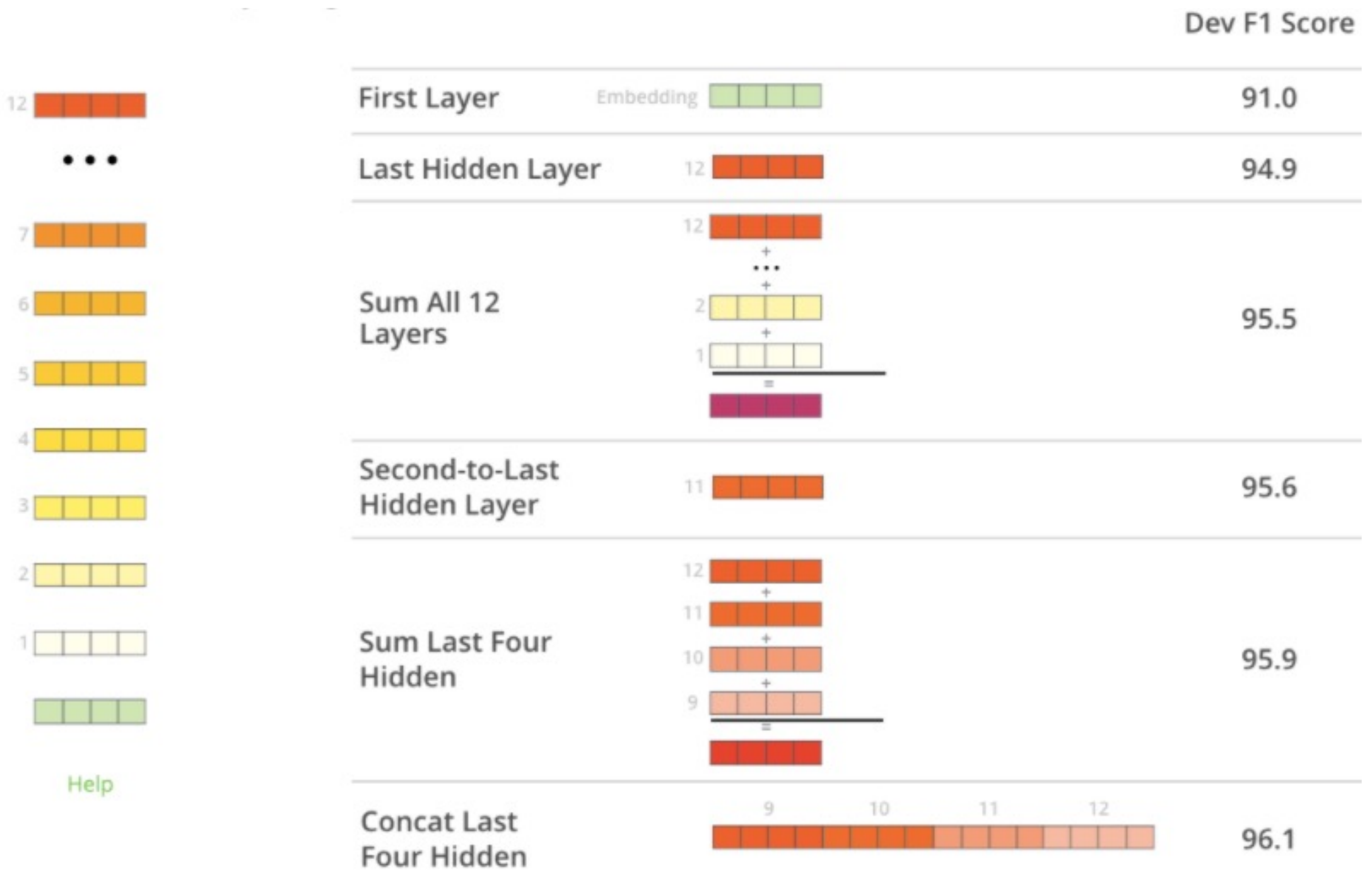


Picture: https://jalammar.github.io/illustrated-bert/

BERT

Later layers have the best contextualized embeddings

(compared to the fine-tuned model which achieved a score of **96.4**)



Picture: https://jalammar.github.io/illustrated-bert/

BERT

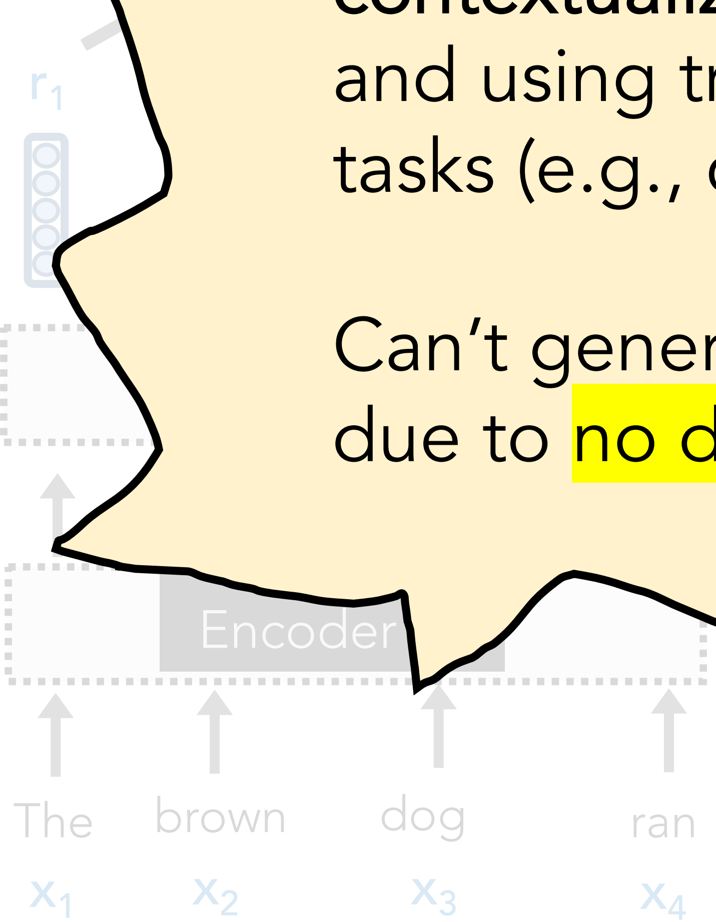
BERT yielded state-of-the-art (SOTA) results on many tasks

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
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**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).

Source: original BERT paper: https://arxiv.org/pdf/1810.04805.pdf

BERT (Tra forme iant)



Takeaway

BERT is incredible for learning **contextualized embeddings** of words and using transfer learning for other tasks (e.g., classification).

Can't generate *new sentences* though, due to no decoders.

some

a

Extensions

Transformer-Encoders

- BERT
- ALBERT (A Lite BERT …)
- RoBERTa (A Robustly Optimized BERT …)
- DistilBERT (small BERT)
- ELECTRA (Pre-training Text Encoders as Discriminators not Generators)
- Longformer (Long-Document Transformer)

Extensions

Autoregressive

- GPT (Generative Pre-training)
- CTRL (Conditional Transformer LM for Controllable Generation)
- Reformer
- XLNet

Outline

- BERT (finishing up)
- GPT-2
- Issues and remaining work

Outline

- BERT (finishing up)
- GPT-2
- Issues and remaining work

Transformer

What if we want to generate a new output sequence?

GPT-2 model to the rescue!

Generative Pre-trained Transformer 2

GPT-2 (a Transformer)

GPT-2 uses only Transformer Decoders (no Encoders) to generate new sequences from scratch or from a starting sequence

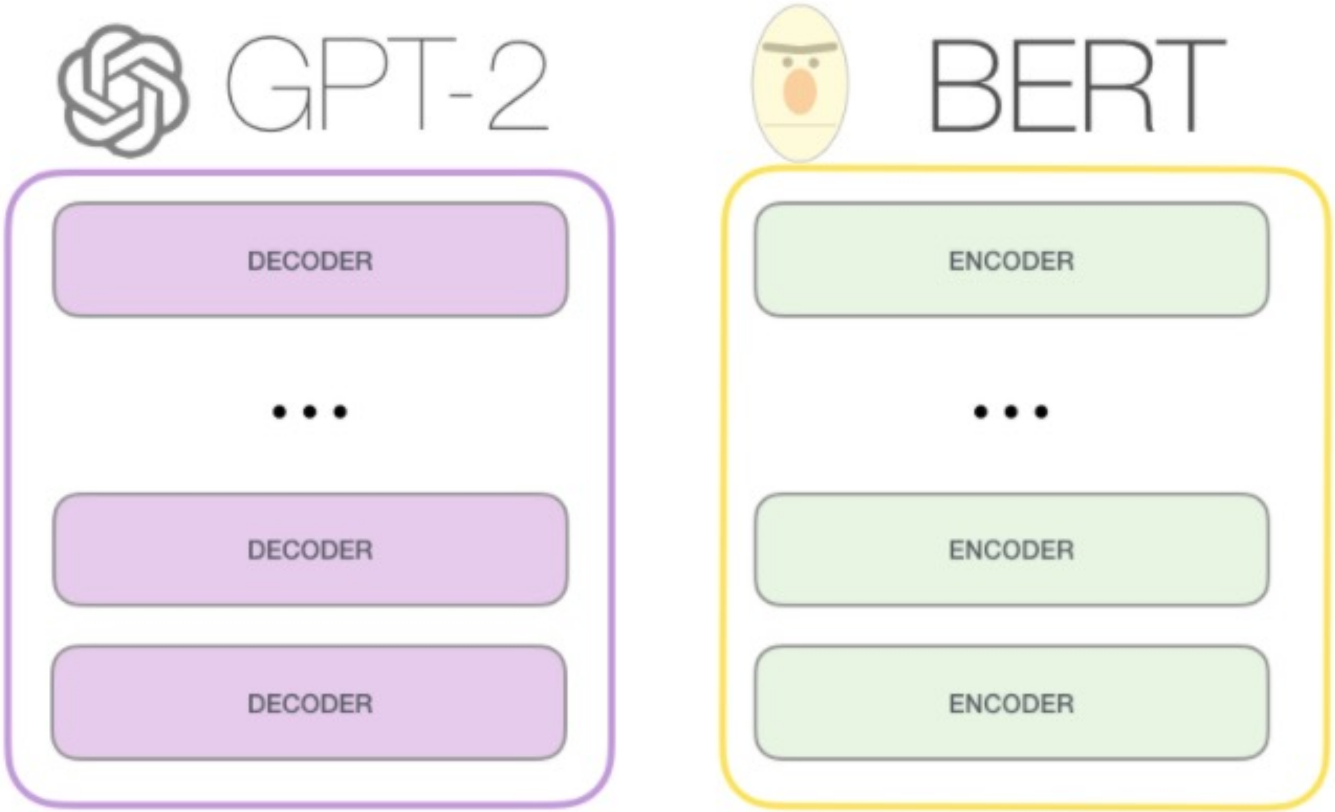


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2 (a Transformer)

- There is no Attention (since there is no **Transformer Encoder** to attend to). So, there is only Self-Attention.
- As it processes each word/token, it **masks** the "future" words and conditions on and attends to the previous words

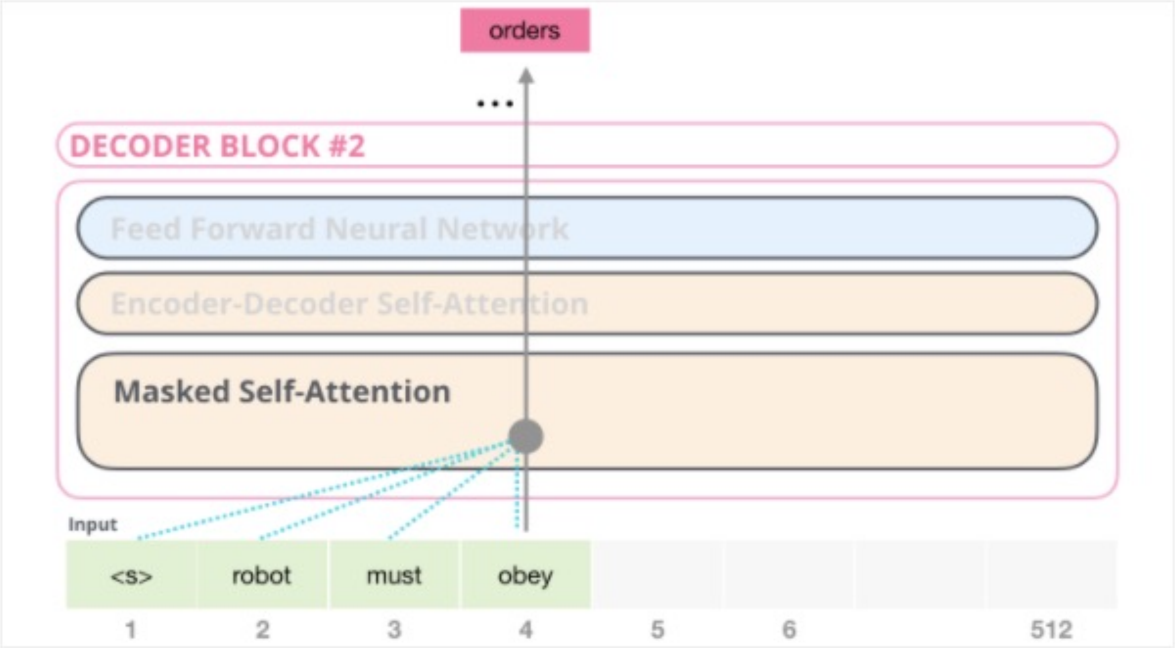


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2 (a Transformer)

As it processes each word/token, it **masks** the "future" words and conditions on and attends to the previous words

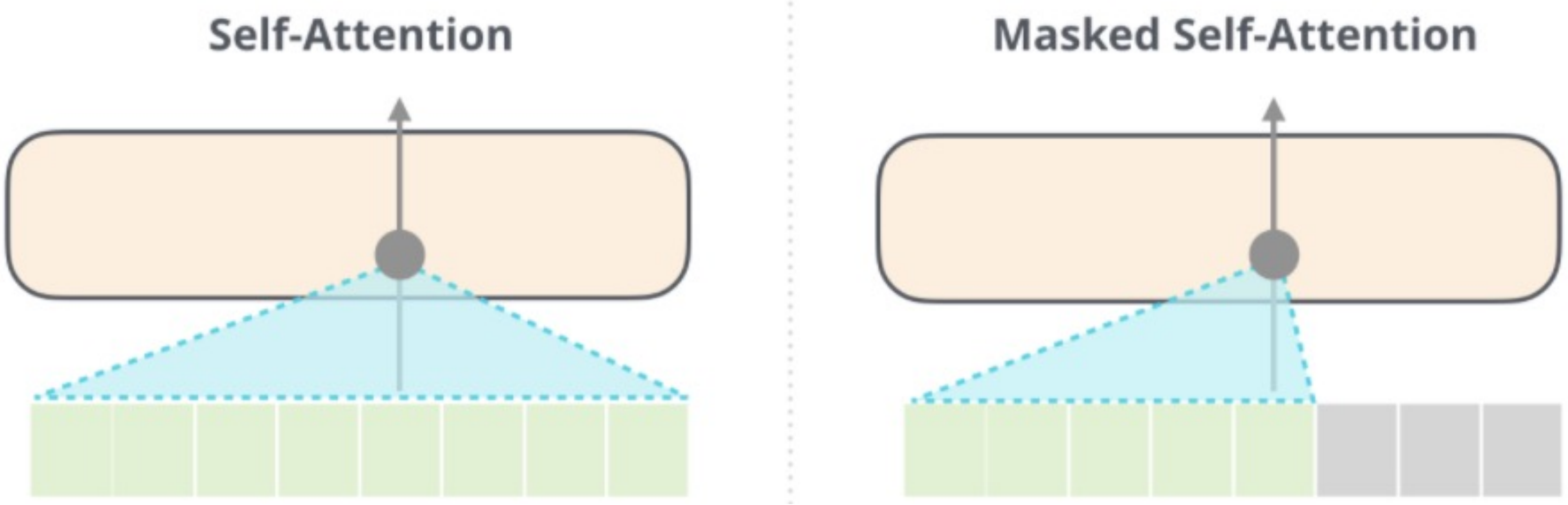


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2 (a Transformer)

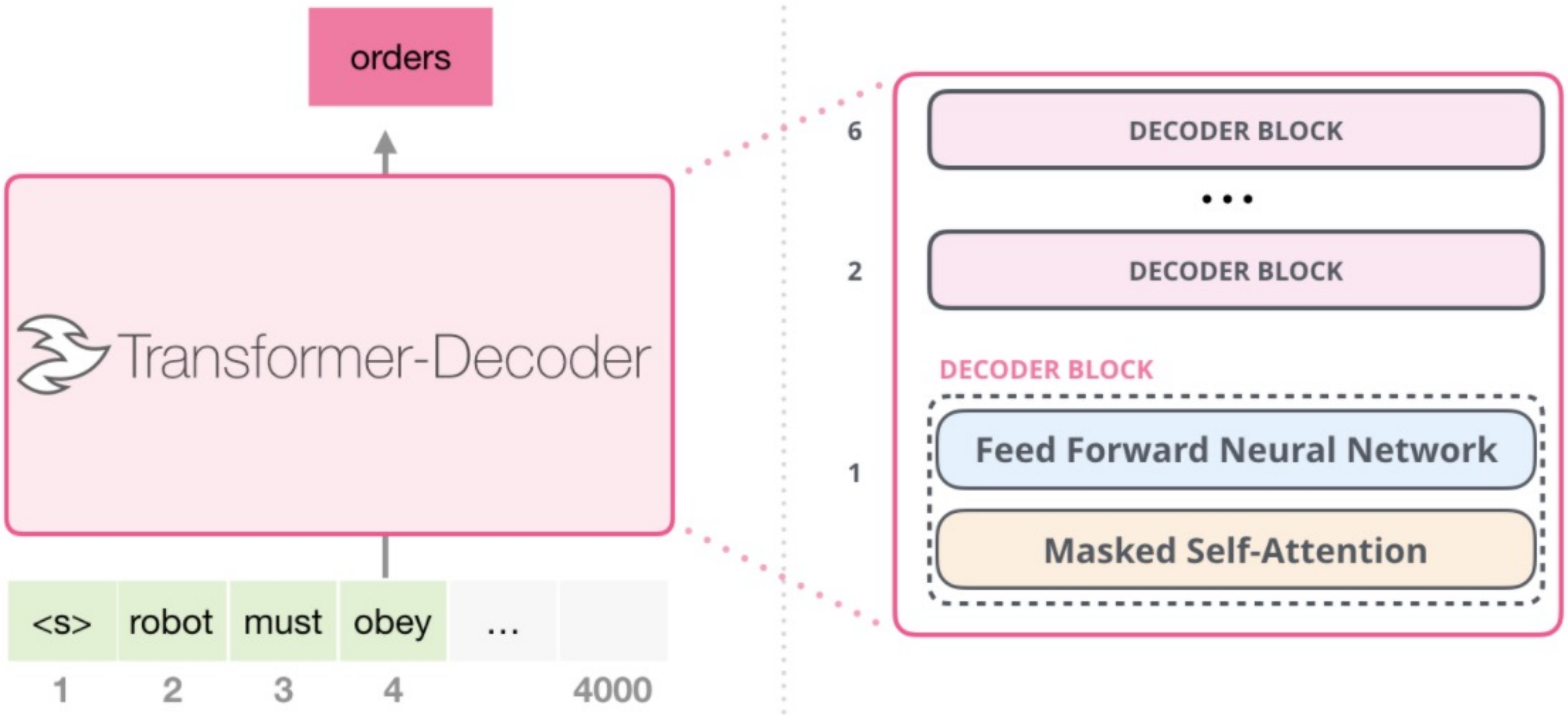


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2 (a Transformer)

- Technically, it doesn't use words as input but Byte Pair Encodings (sub-words), similar to BERT's WordPieces.
- Includes positional embeddings as part of the input, too.
- Easy to fine-tune on your own dataset (language)

GPT-2 (a Transformer)

Image by http://jalammar.github.io/illustrated-gpt2/

Byte Pair Encodings (BPE)

- Invented in 1994 (Gage) and updated in 2015 (Sennrich et al.)
- Looks at the individual symbols (e.g., characters) and repeated merges the most frequent pairs (a la agglomerative clustering)
- Stop after N merges (you specify N). GPT uses N =40k

Philip Gage. 1994. A New Algorithm for Data Compression. C Users J., 12(2):23–38, February

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909, 2015

GPT-2 (a Transformer)

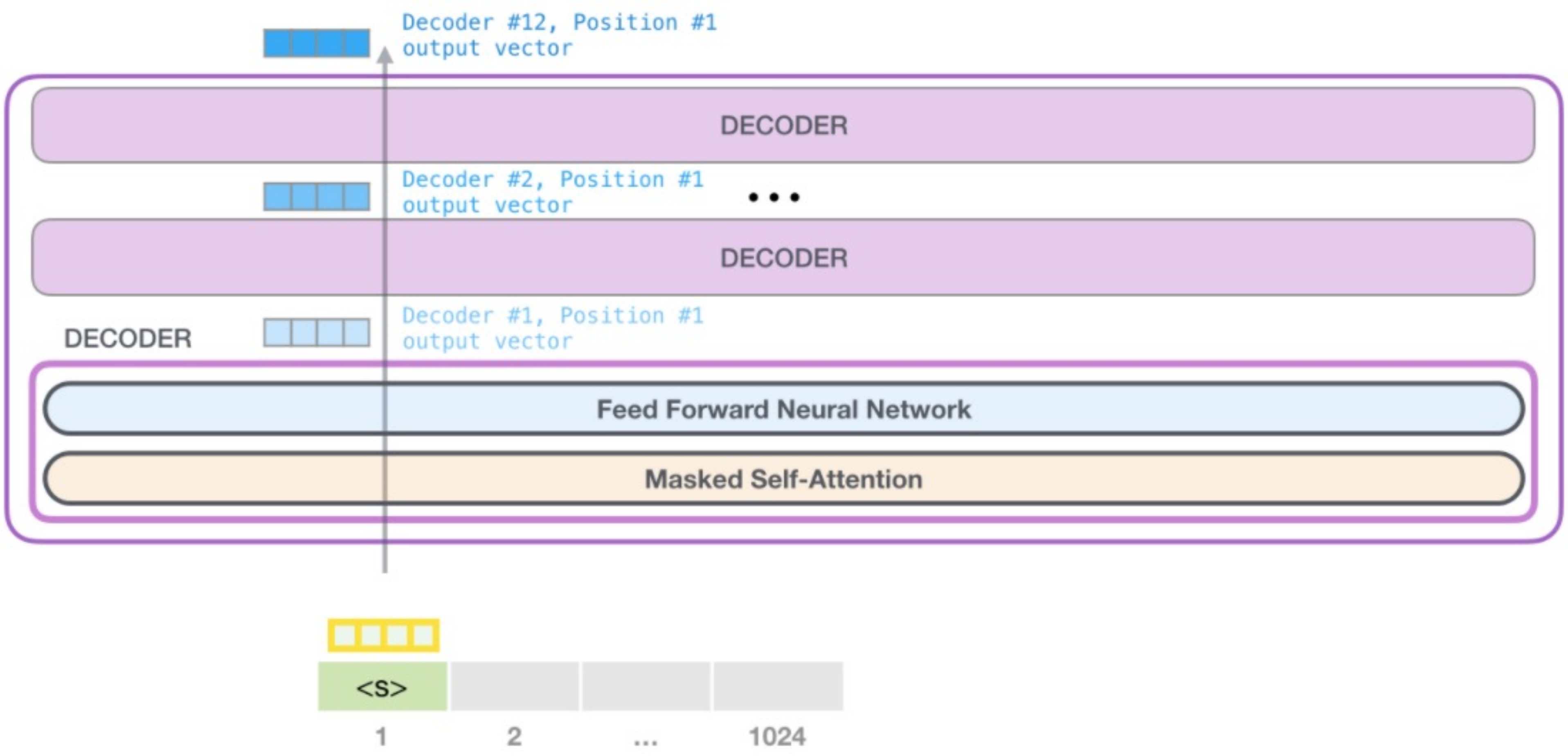


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's Masked Attention

For efficiency, we can still calculate all query-key calculations with matrix multiplications, then mask before softmax'ing.

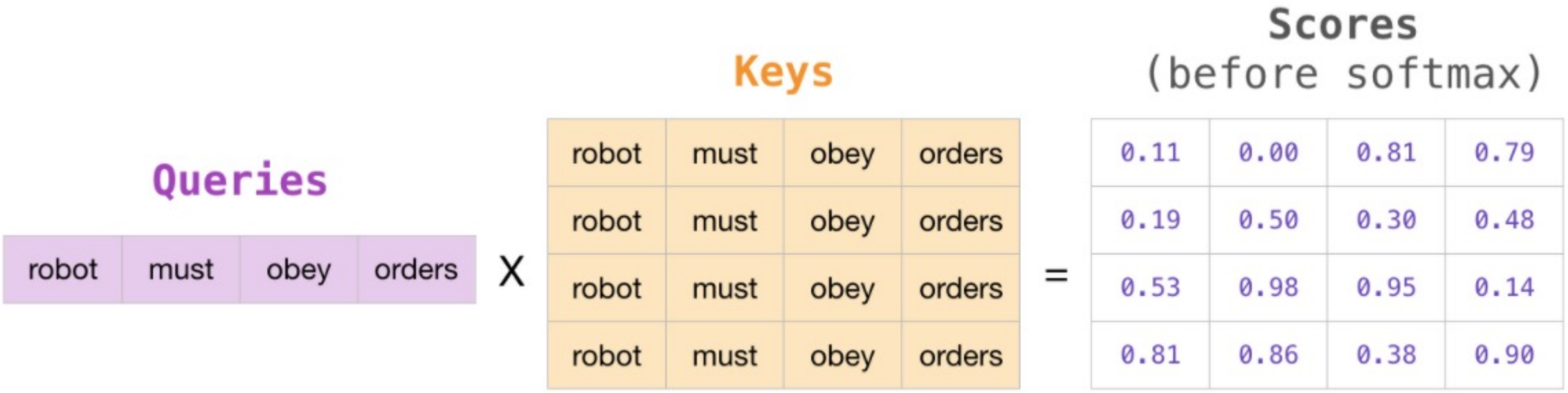


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's Masked Attention

For efficiency, we can still calculate all query-key calculations with matrix multiplications, then mask before softmax'ing.

Scores (before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Apply Attention Mask →

Masked Scores (before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's Masked Attention

For efficiency, we can still calculate all query-key calculations with matrix multiplications, then mask before softmax'ing.

Masked Scores
(before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Softmax
(along rows)

Scores

1	0	0	0
0.48	0.52	0	0
0.31	0.35	0.34	0
0.25	0.26	0.23	0.26

Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

Representations are propagated upwards through the network

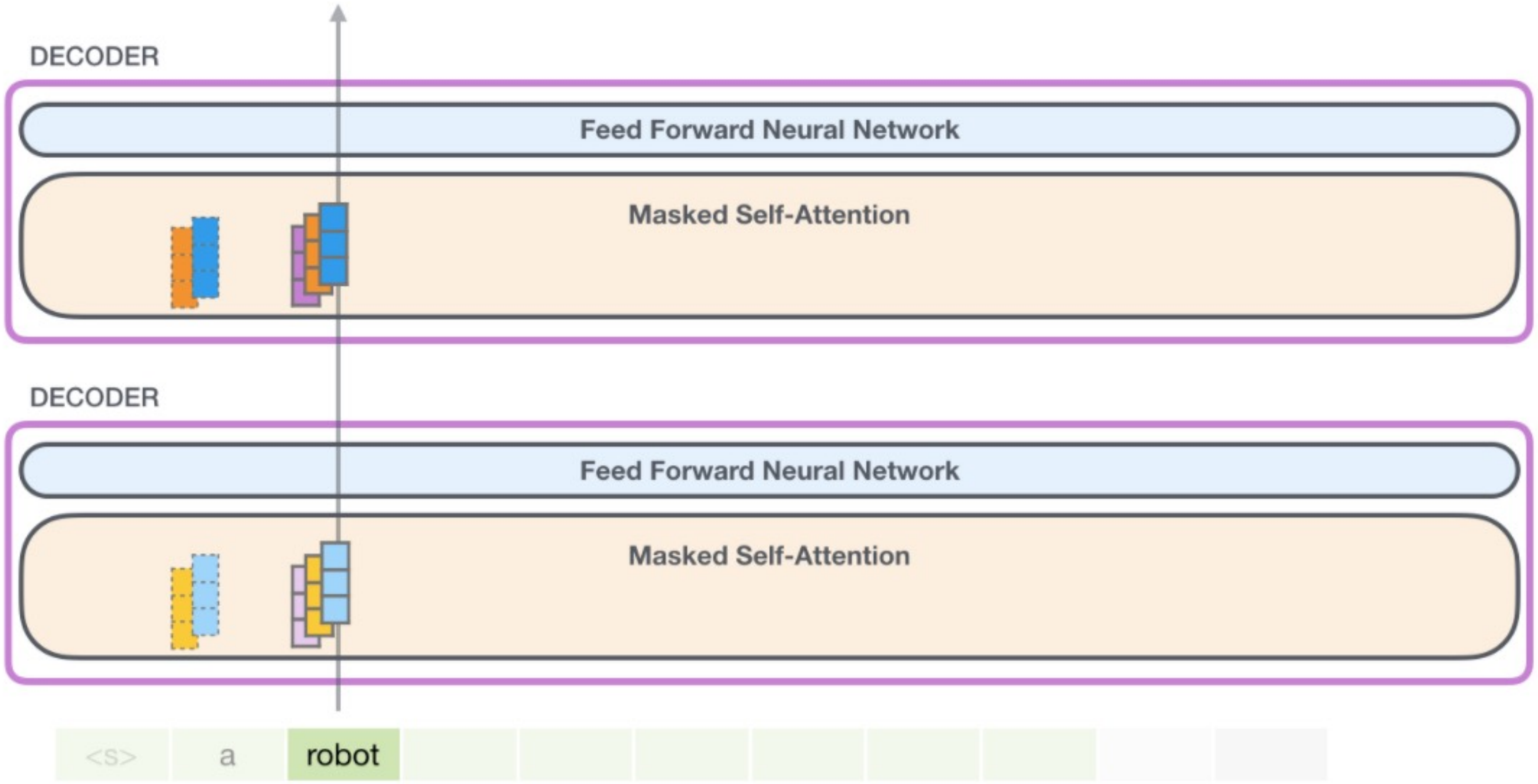


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

Self-attention is otherwise identical to what we saw in BERT

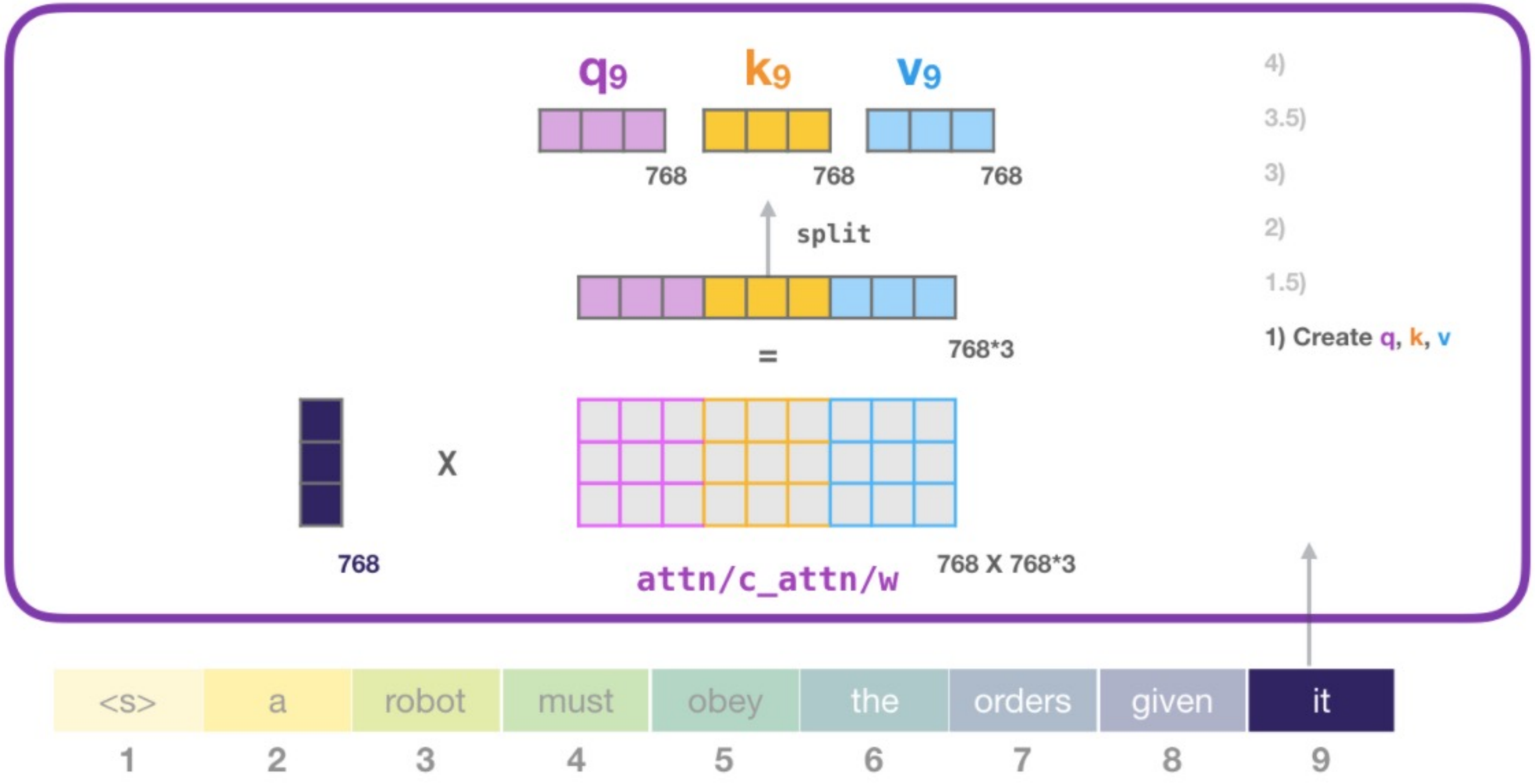


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

Can have Multiple Self-Attention heads

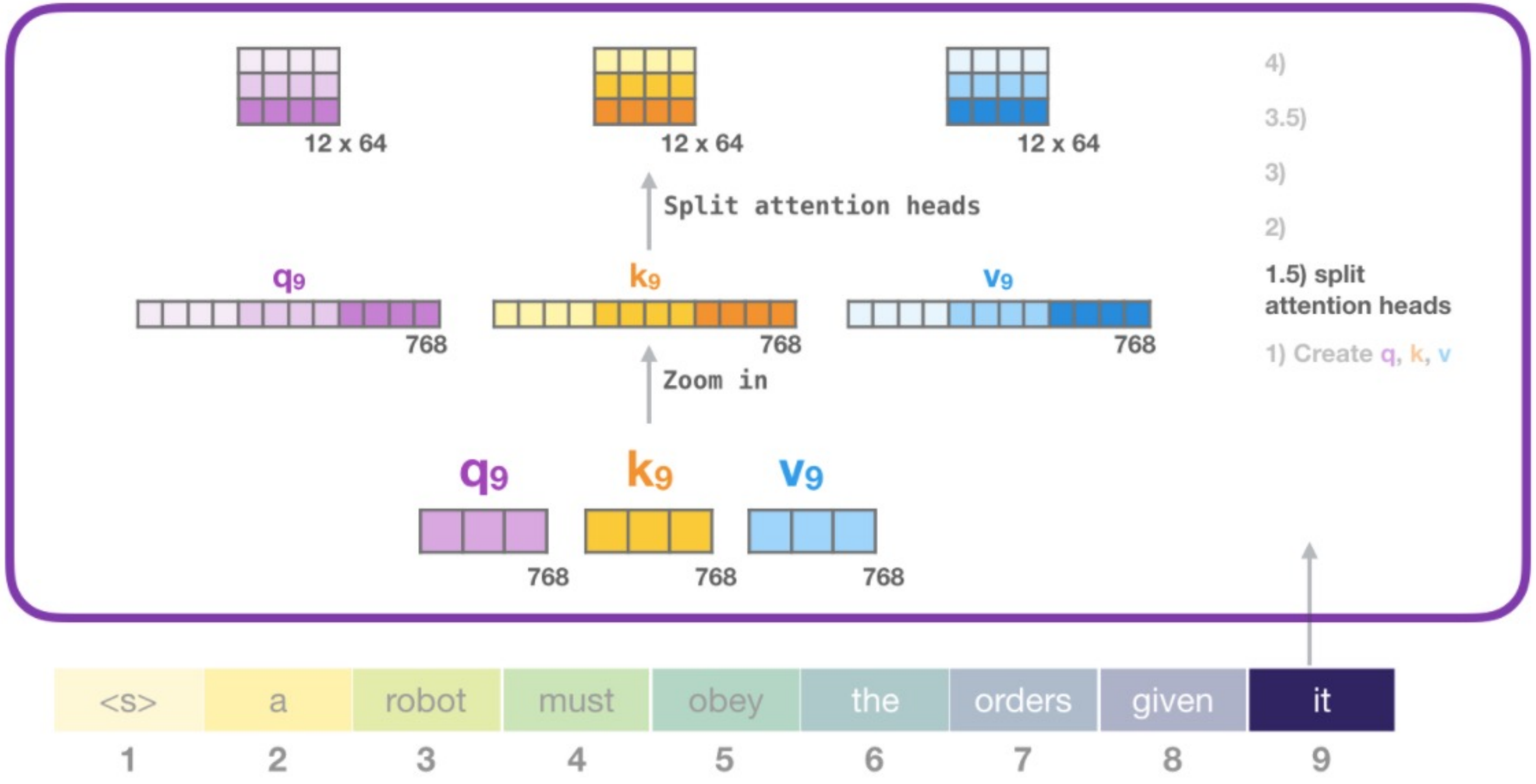


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

Each Self-Attention head is responsible for exactly 1 resulting, output embedding

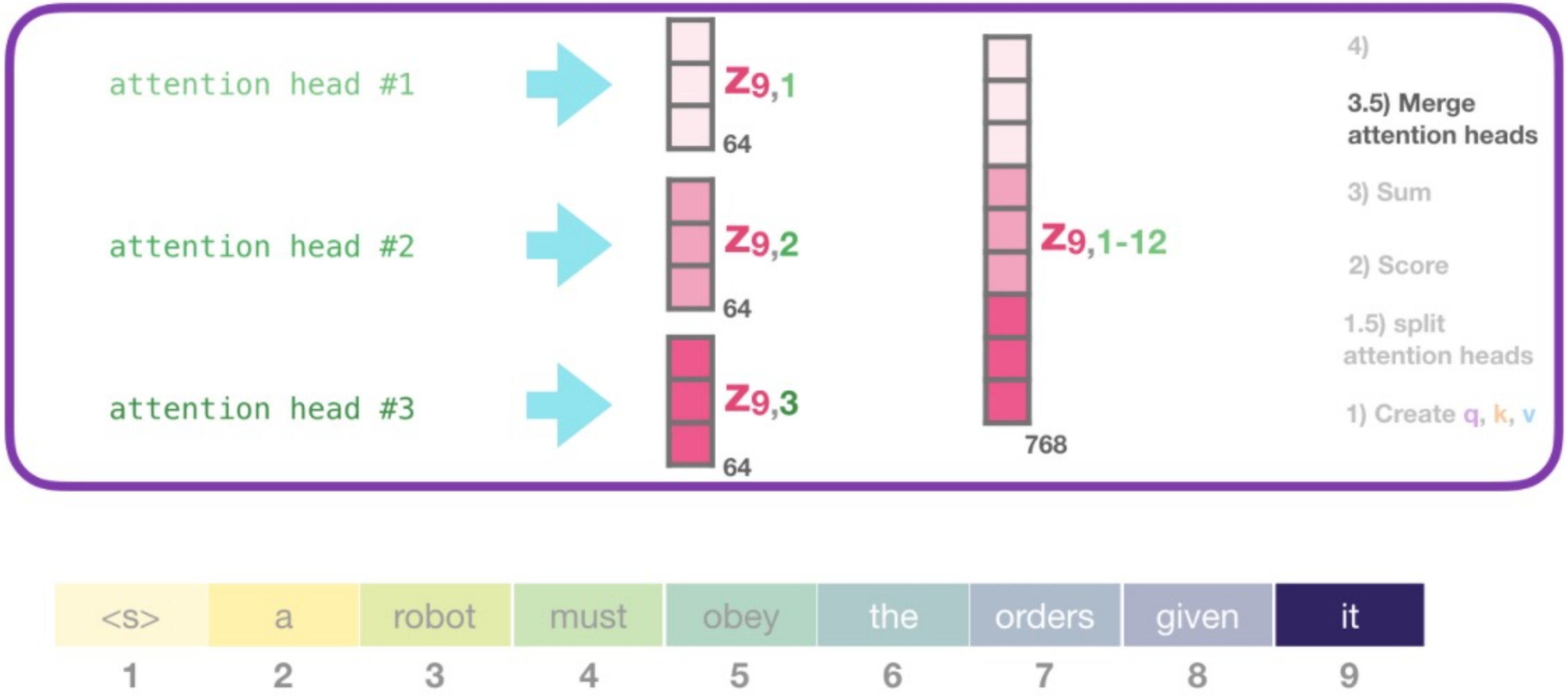


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

Remember, these Masked Self-Attention layers are fed into a FFNN

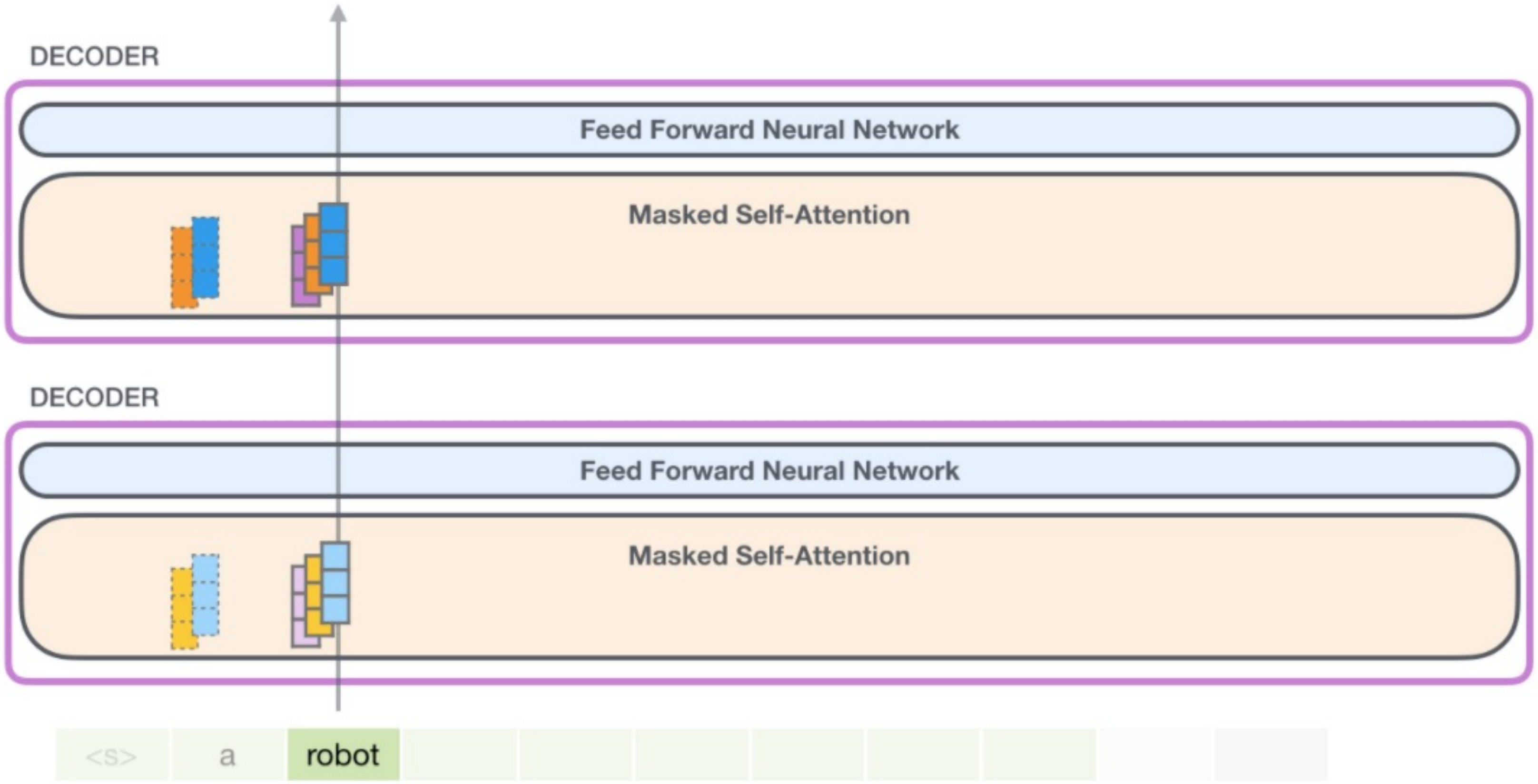
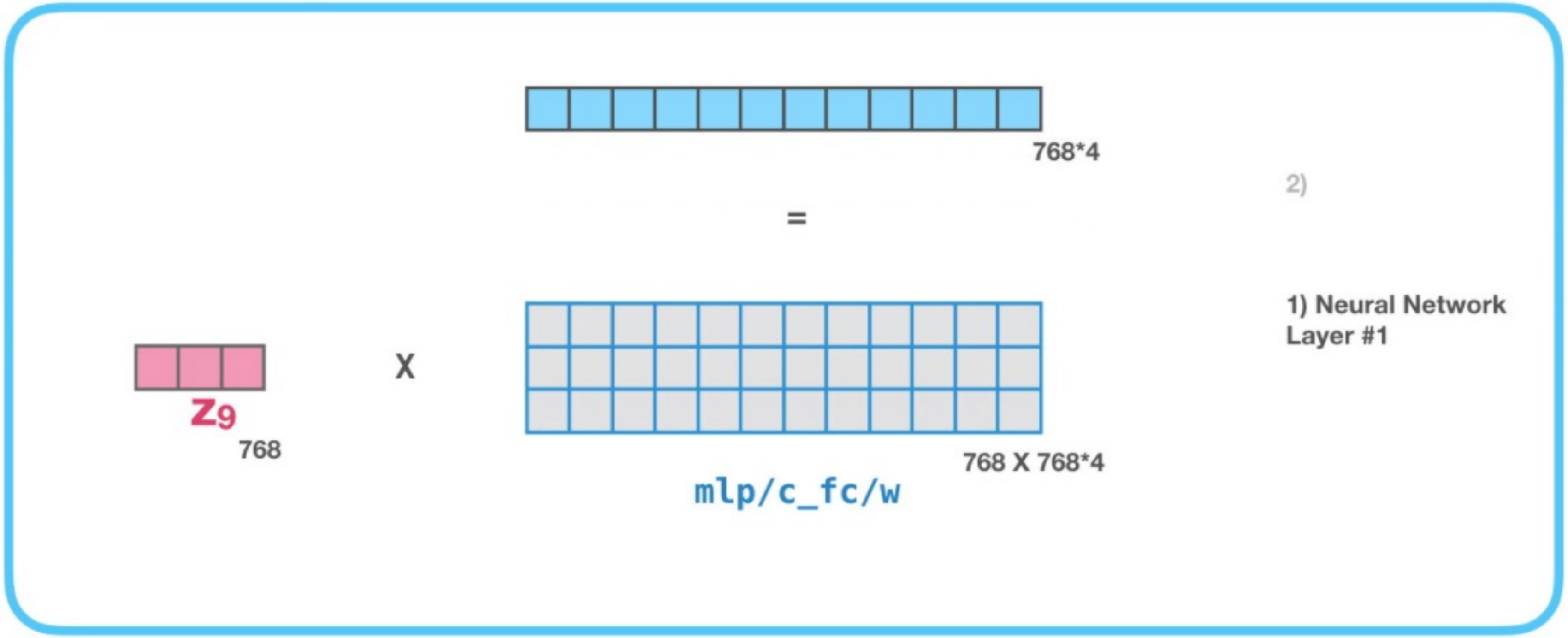


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

Remember, these Masked Self-Attention layers are fed into a FFNN



First hidden layer expands to 4x in size of the input

Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

2nd (final) layer of the FFNN projects it back to the original size

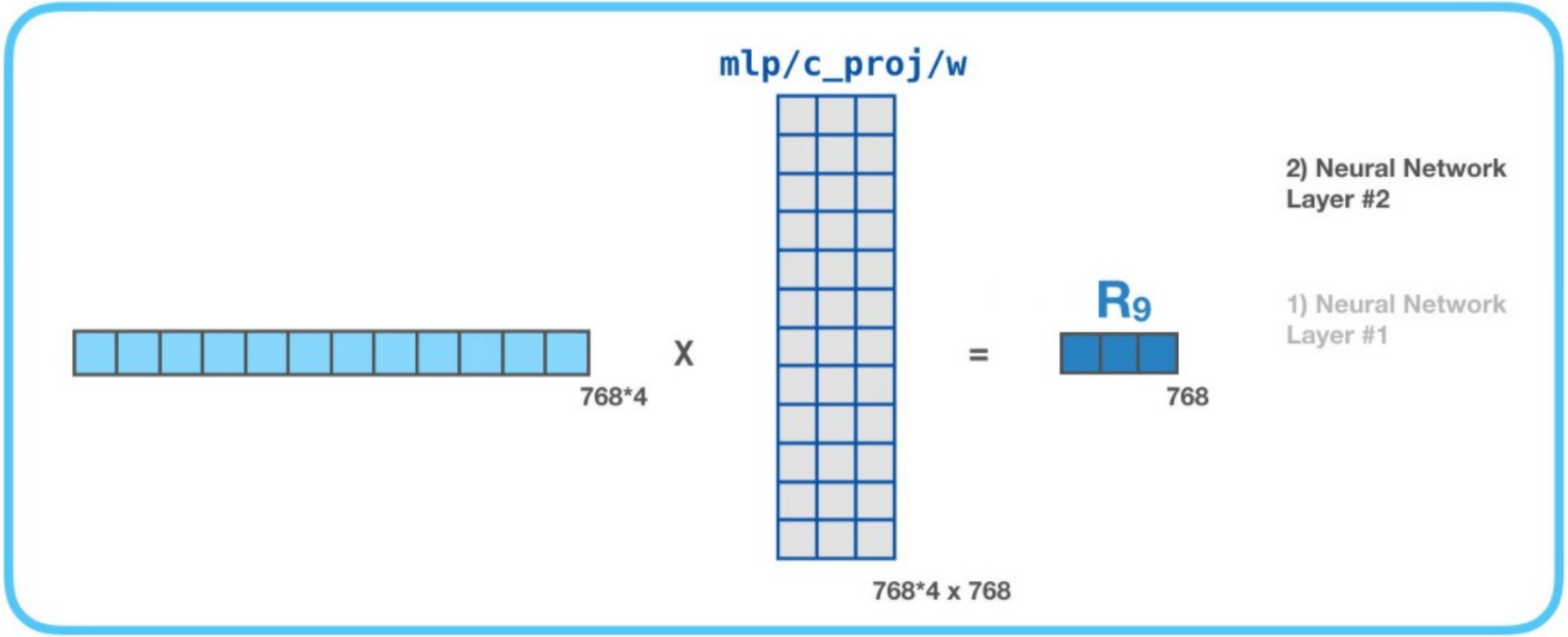


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-2's

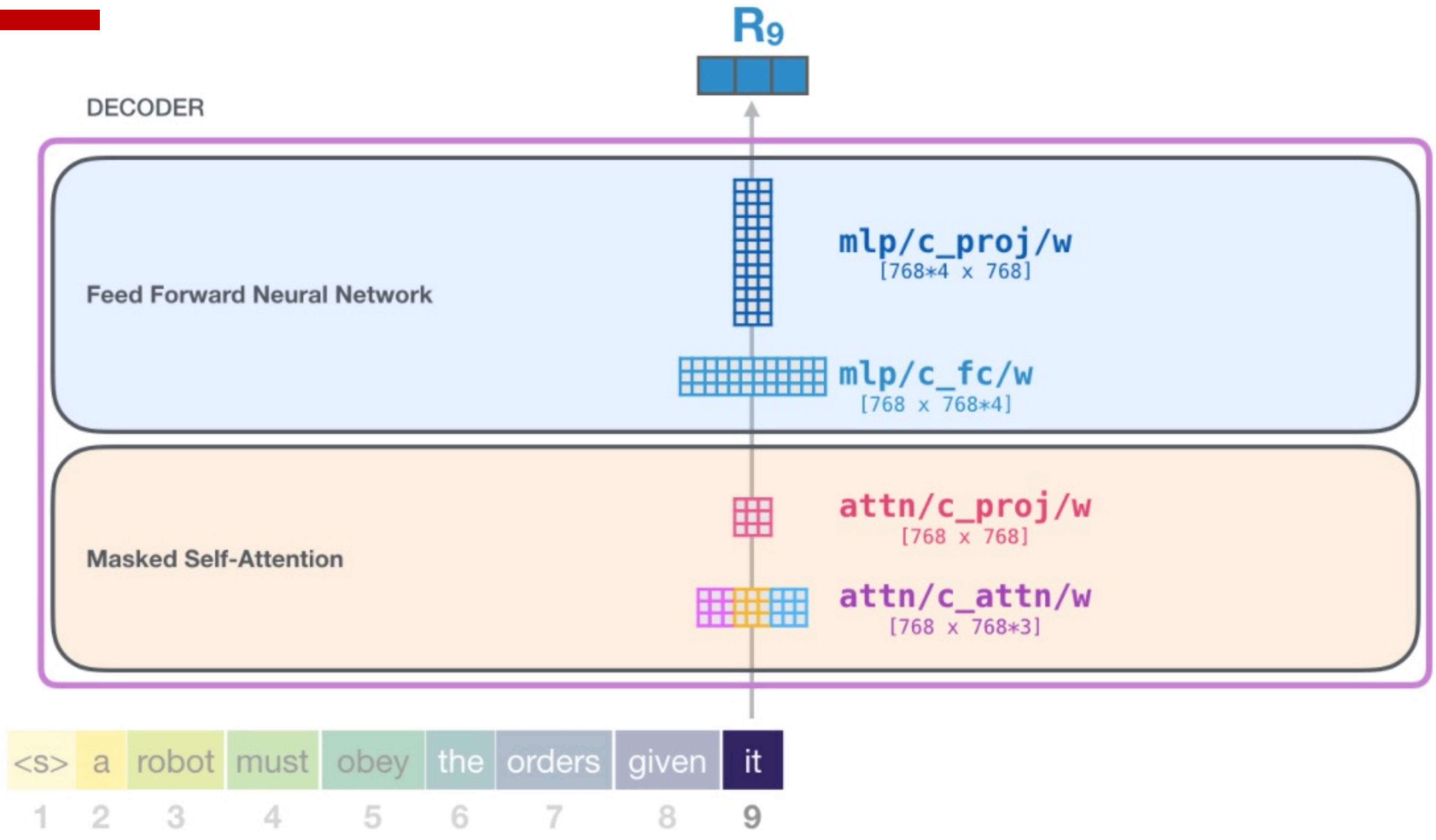


Image by http://jalammar.github.io/illustrated-gpt2/

Each Decoder block has its own weights (e.g., W_k, W_q, W_v)

But the entire model only has 1 token-embedding weight matrix and positional encoding weight matrix. This helps all the blocks to work together and supplement their captured aspects

Image by http://jalammar.github.io/illustrated-gpt2/

The authors of GPT-2 created 4 different version (sizes) of the model

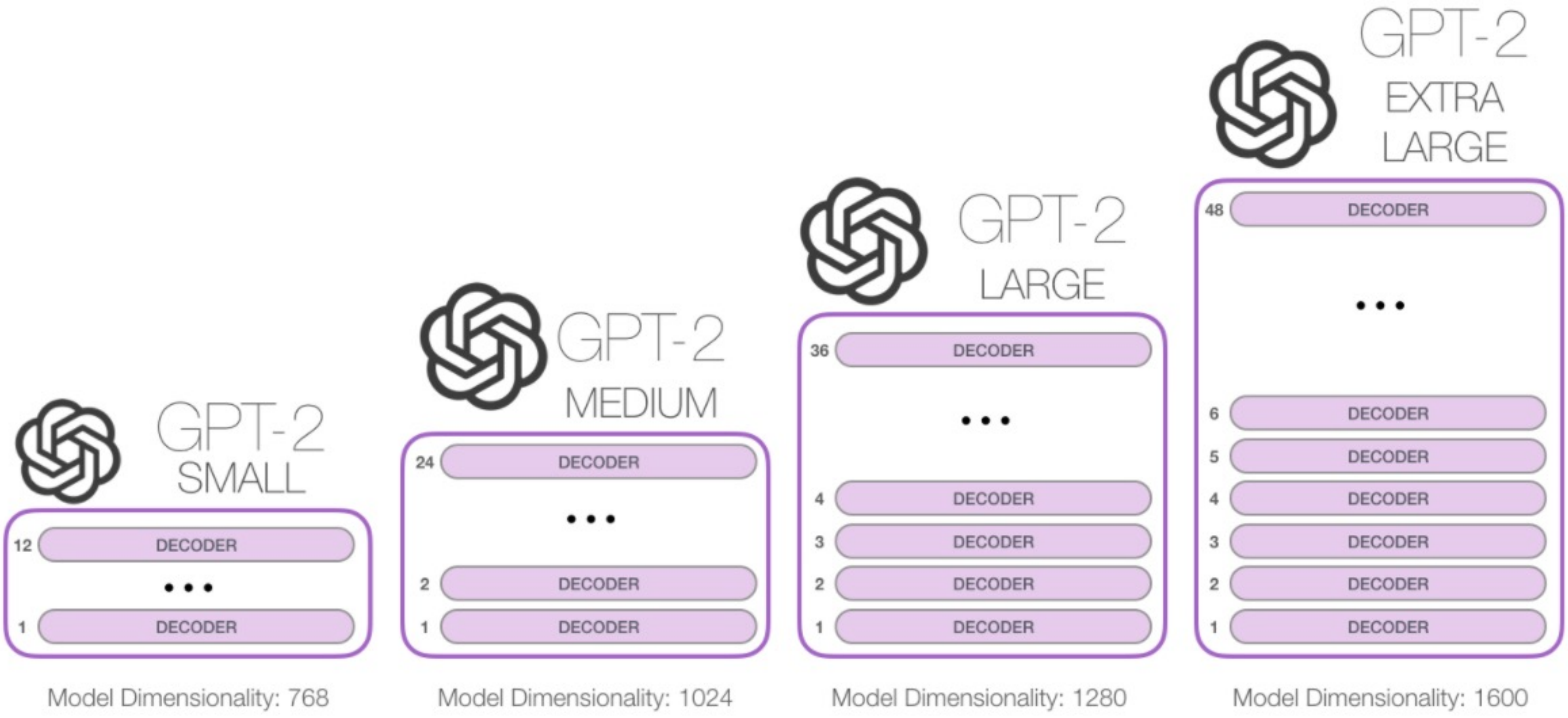


Image by http://jalammar.github.io/illustrated-gpt2/

GPT-1

- Model: Transformer Decoders we just described
- Objective: next word prediction (cross-entropy loss)
- Data: BooksCorpus (7k books from a variety of genres, such as Adventure, Fantasy, and Romance)

GPT-1

Authors were primarily focused on demonstrating that you could fine-tune this LM on <u>supervised tasks</u> and get SOTA results

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com

Karthik Narasimhan OpenAI karthikn@openai.com

Tim Salimans OpenAI tim@openai.com

Ilya Sutskever OpenAI ilyasu@openai.com

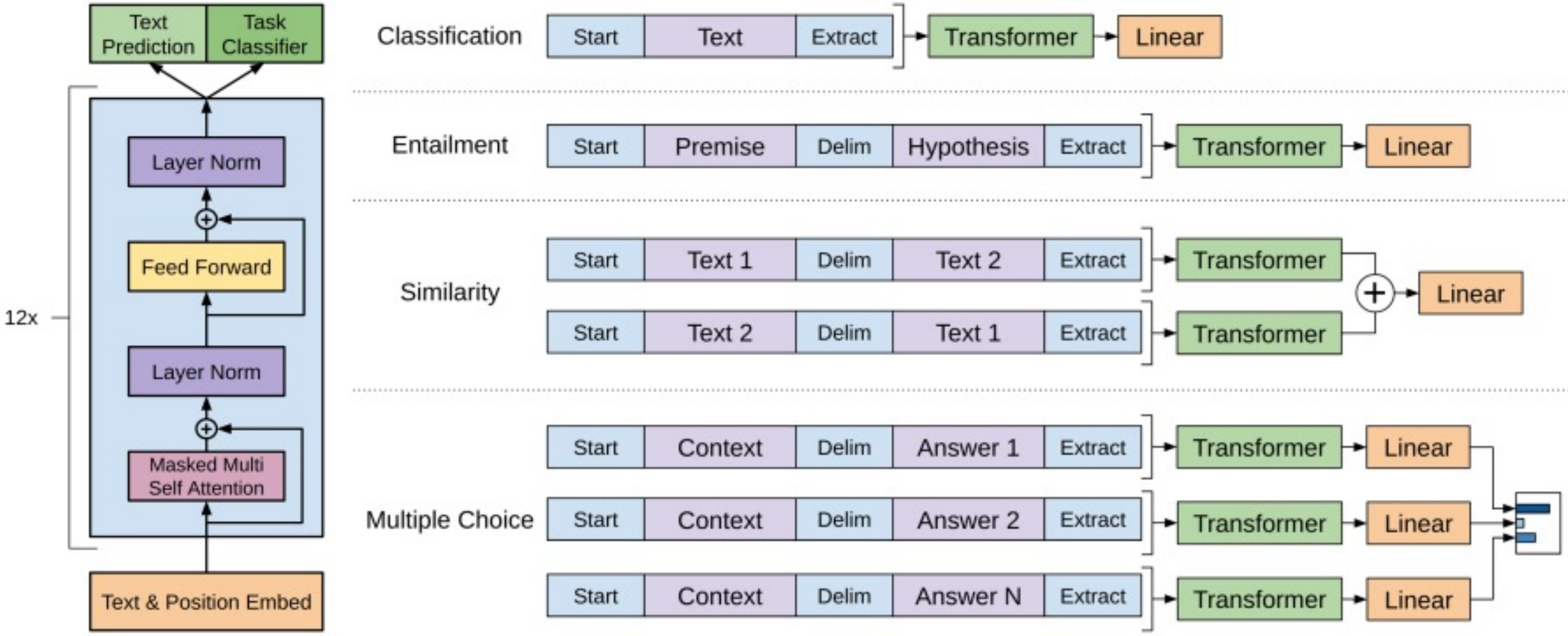


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Given an unsupervised corpus of tokens $\mathcal{U} = \{u_1, \ldots, u_n\}$, we use a standard language modeling objective to maximize the following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta) \tag{1}$$

After training the model with the objective in Eq. 1, we adapt the parameters to the supervised target task. We assume a labeled dataset $\mathcal{C}$, where each instance consists of a sequence of input tokens, $x^1, \ldots, x^m$, along with a label y. The inputs are passed through our pre-trained model to obtain the final transformer block's activation h_l^m, which is then fed into an added linear output layer with parameters W_y to predict y:

$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y). \tag{3}$$

This gives us the following objective to maximize:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m). \tag{4}$$

We additionally found that including language modeling as an auxiliary objective to the fine-tuning helped learning by (a) improving generalization of the supervised model, and (b) accelerating convergence. This is in line with prior work [50, 43], who also observed improved performance with such an auxiliary objective. Specifically, we optimize the following objective (with weight λ):

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C}) \tag{5}$$

https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-1

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

NLI is when you predict if the hypothesis phrase is entailed, neutral, or contradicts the preceding premise phrase.

https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-1

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Story Cloze is like MLM, by predicting the blank

https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-1

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Overall, our approach achieves new state-of-the-art results in 9 out of the 12 datasets we evaluate on, outperforming ensembles in many cases. Our results also indicate that our approach works well across datasets of different sizes, from smaller datasets such as STS-B (≈5.7k training examples) – to the largest one – SNLI (≈550k training examples).

https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-2 is identical to GPT-1, but:

- has Layer normalization in between each sub-block (as we've already seen)
- Vocab extended to 50,257 tokens and context size increased from 512 to 1024
- Data: 8 million docs from the web (Common Crawl), minus Wikipedia

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

You can finagle the system to yield synthetic predictions.

Children's Book Test (CBT) is a classification task. Fill-in-the-blank, and you predict which of the 10 possible choices is correct.

You can compute the probability of each choice + its ending.

You can finagle the system to yield synthetic predictions.

LAMBADA dataset tests model's ability to understand long-range dependencies.

Task: predict the final word of sentences which humans need 50+ tokens of context in order to accurately predict.

GPT-2 Results

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

You can finagle the system to yield synthetic predictions.

Summarization. The add the text “TL;DR:” after an article, then generate 100 tokens with top-2 random sampling, then extract the first 3 sentences.

GPT-2 Results

	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 `TL;DR:`	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Table 4. Summarization performance as measured by ROUGE F1 metrics on the CNN and Daily Mail dataset. Bottom-Up Sum is the SOTA model from (Gehrmann et al., 2018)

GPT-2 Results

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%

GPT-2 Results

Easy to fine-tune on your own dataset (language)

SYSTEM PROMPT (HUMAN-WRITTEN)

A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.

MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)

The incident occurred on the downtown train line, which runs from Covington and Ashland stations.

In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief.

“The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,” said Tom Hicks, the U.S. Energy Secretary, in a statement. “Our top priority is to secure the theft and ensure it doesn’t happen again.”
The stolen material was taken from the University of Cincinnati’s Research Triangle Park nuclear research site, according to a news release from Department officials.
The Nuclear Regulatory Commission did not immediately release any information.

Context (WebText test)

Pinky

The pink ghost's AI is designed to "feel" opposite of the red ghost's behavior. Pinky actually attempts to get out in front of Pac-Man. This is accomplished by setting the target 4 tiles ahead of Pac-Man's current location in the direction that Pac-Man is travelling. One exception to this is when Pac-Man is traveling up. Due to an overflow bug in the code, the calculation includes a left offset equal to the expected up offset.

The use of Pac-Man's expected position allows the player to exploit the navigation code for Pinky. This was most often seen in situations where Pac-Man and Pinky are in close proximity to each other. For example, if they were headed toward one another, once they are less than 4 tiles apart, the target tile is now behind Pinky. If he encounters an intersection while in this configuration, he will attempt to double-back (by turning) in order to reach the target tile.

Player's would often perform what was referred to as a "head fake" by changing directions rapidly just as Pinky was approaching an intersection. By displacing the target tile just as Pinky

Context (WebText test)

was about to cross the intersection, Pac-Man would be able to dodge the ghosts' projectiles and return to the safe location of the safe house. Buster The yellow ghost's AI is specifically designed to try to avoid Pac-Man from leaving the safe house. The player must press the SPACE bar to reveal the safe house's doors, then go into them to find the yellow ghost hiding in the next room. The player must also note that the ghost will not attempt to exit the safe house if it sees Pac-Man already there. In fact, the ghost will not attempt an exit in the first place unless they're already in the safe house. These two behaviors are based on the fact that the player does not always attempt to reach the safe house before Pac-Man (or Blinky). The player must also beware of this glitch if a ghost (or another player) is already in the safe house and attempting to get out as soon as Pac-Man or Blinky is gone.

LIVE
DEMO

BERT (a Transformer Encoder)

BERT is trained on a lot of text data:

- BooksCorpus (800M words)
- English Wikipedia (2.5B words)

Yay, for transfer learning!

BERT-Base model has 12 transformer blocks, 12 attention heads,

110M parameters!

BERT-Large model has 24 transformer blocks, 16 attention heads,

340M parameters!

GPT-2 (a Transformer Decoder)

GPT-2 is:

- trained on 40GB of text data (8M webpages)!
- 1.5B parameters

GPT-3 is an even bigger version (175B parameters) of GPT-2, but isn't open-source

Yay, for transfer learning!

Outline

- BERT (finishing up)
- GPT-2
- Issues and remaining work

Outline

- BERT (finishing up)
- GPT-2
- Issues and remaining work

Concerns

There are several issues to be aware of:

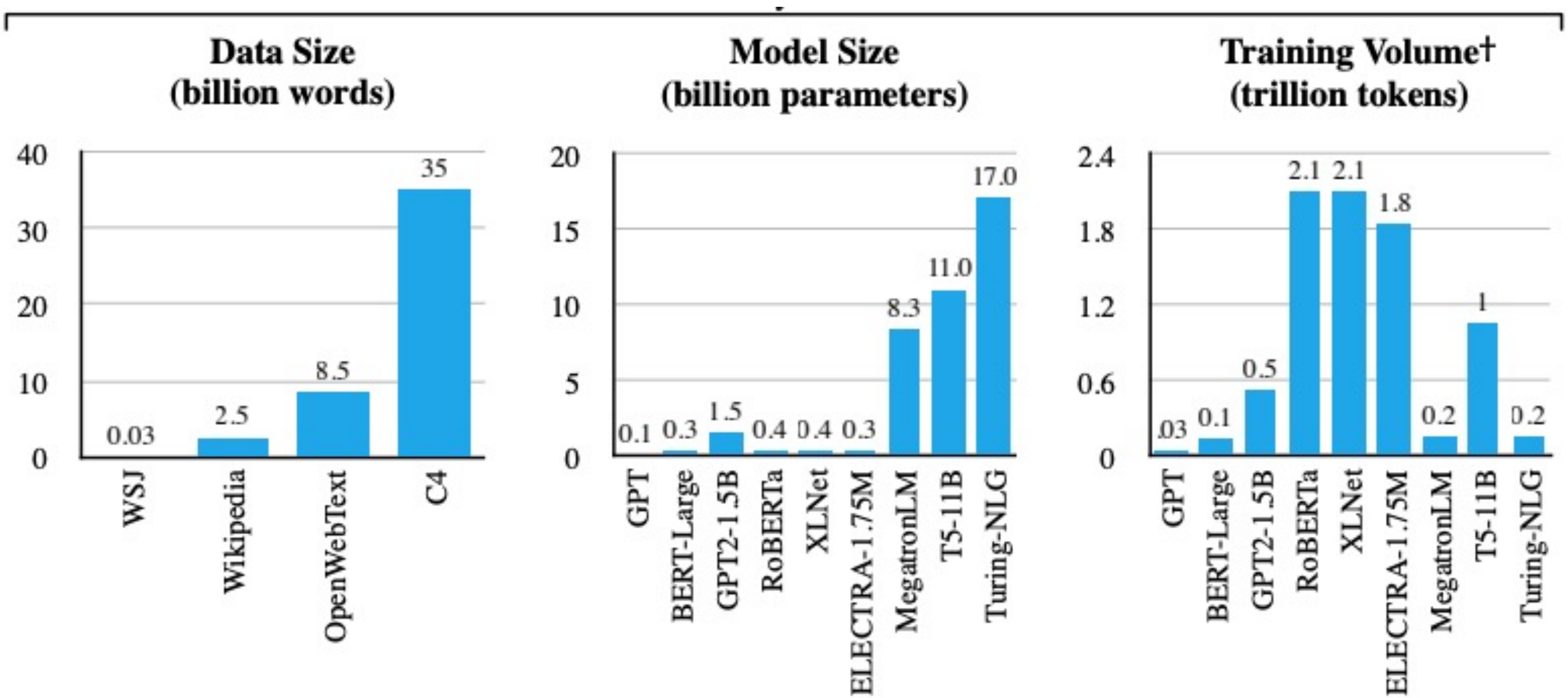
- It is very costly to train these large models. The companies who develop these models easily spend an entire month training one model, which uses incredible amounts of electricity.
- BERT alone is estimated to cost over $1M for their final models

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

Source: https://arxiv.org/pdf/2004.08900.pdf

Concerns

It is very costly to train these large models.



Source: https://arxiv.org/pdf/2004.08900.pdf

Concerns

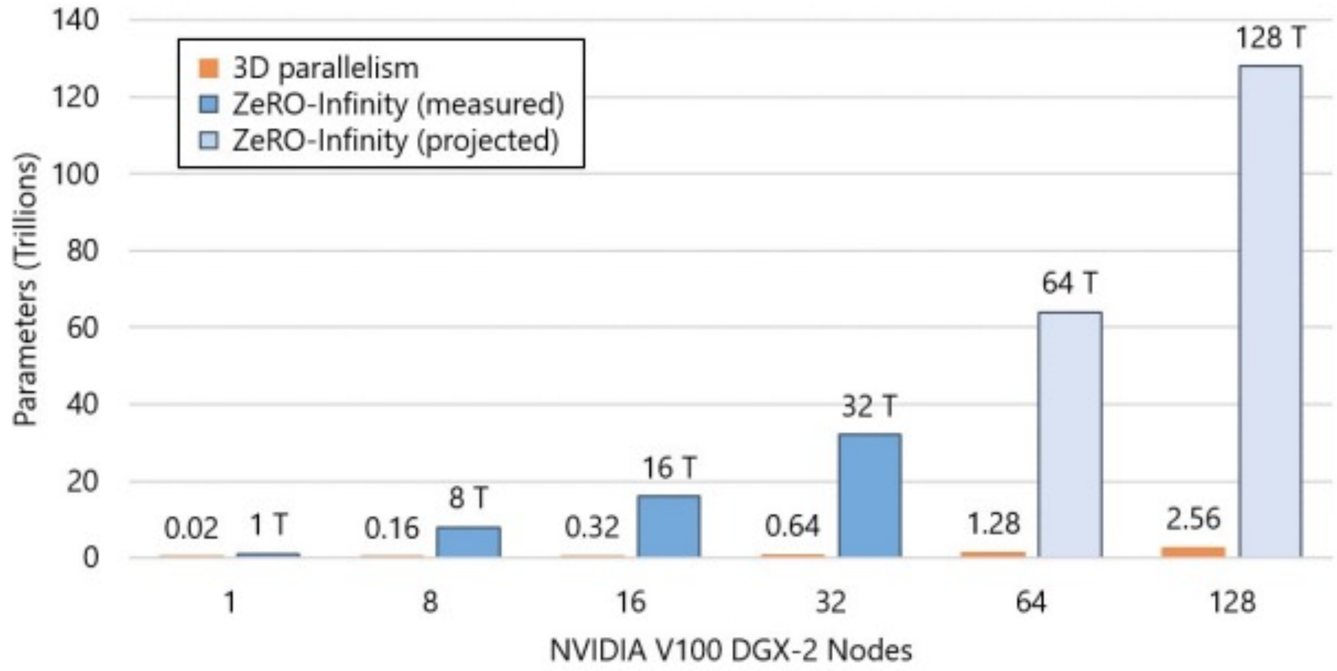


Figure 1: ZeRO-Infinity can train a model with 32 trillion parameters on 32 NVIDIA V100 DGX-2 nodes (512 GPUs), 50x larger than 3D parallelism, the existing state-of-the-art.

ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, Yuxiong He

{samyamr, olruwase, jerasley, shsmit, yuxhe}@microsoft.com

Concerns

- Further, these very large language models have been shown to be biased (e.g., in terms of gender, race, sex, etc).
- Converting from one language to another often converts gender neutral pronouns to sexist stereotypes
- Using these powerful LMs comes with risks of producing such text and/or evaluating/predicting tasks based on these biased assumptions.
- People are researching how to improve this

Concerns

- As computer-generated text starts to become indistinguishable from authentic, human-generated text, consider the ethical impact of fraudulently claiming text to be from a particular author.
- If used maliciously, it can easily contribute toward the problem of Fake News

Summary

- There has been significant NLP progress in the past few years.
- Some of the complex models are incredible, but rely on having a lot of data and computational resources (e.g., Transformers)
- With all data science and machine learning, it's best to understand your data and task very well, clean your data, and start with a simple model (instead of jumping to the most complex model)

Summary

- NLP is incredibly fun, with infinite number of problems to work on
- Neural models allow us to easily represent words as distributed representations
  - Input unique word (or sub-words) as tokens
  - Recurrent models can be for capturing the sequential nature, but it puts too much responsibility on the model to keep track of the entire meaning and to pass it onwards

Summary

- Transformers allow for more complete, free access to everything (unless masked) at once
- It's very useful to pre-train a large unsupervised/self-supervised LM then fine-tune on your particular task (replace the top layer, so that it can work)

Outstanding Questions

- What is the model *actually* learning → probing tasks/interpretability
- biases exist within data & model. How can we improve this? → debiasing
- How can we make models faster, smaller, more robust? → distillation, robustness
- Can we better understand the sensitivity of models and protect against vulnerabilities? → adversarial NLP
- How can we better handle low-resource/scarce/unlabelled data?
- How can we get better at complex tasks? (e.g., coreference resolution, tasks that require commonsense reasoning and leveraging real-world knowledge)
- How can we get better at long-form documents, mixed-mediums? (e.g., tabular data, images, structured text such as scientific papers)