Lecture 11: BERT

The Power of Transformer Encoders

Harvard

AC295/CS287r/CSCI E-115B

Chris Tanner


Bert, have you seen my waste-paper basket?
Ask me that again and look into my eyes.

ANNOUNCEMENTS

- HW3 has been released! Due Oct 19 (Tues) @ 11:59pm.
- Research Project Selection (Google Form) is now closed.
    - We're making selections now!
- Read "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings" before Oct 14 (Thurs)

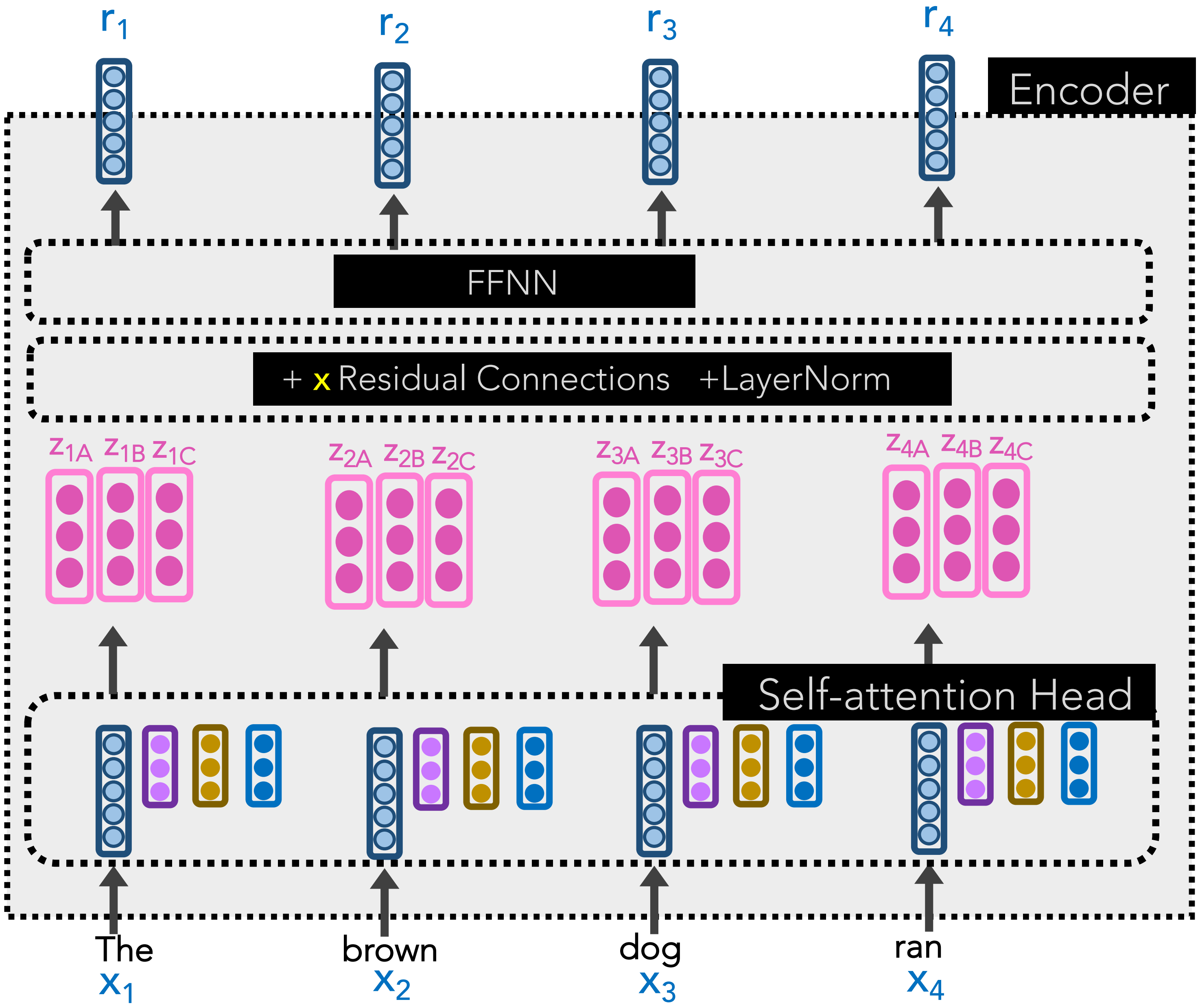
Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

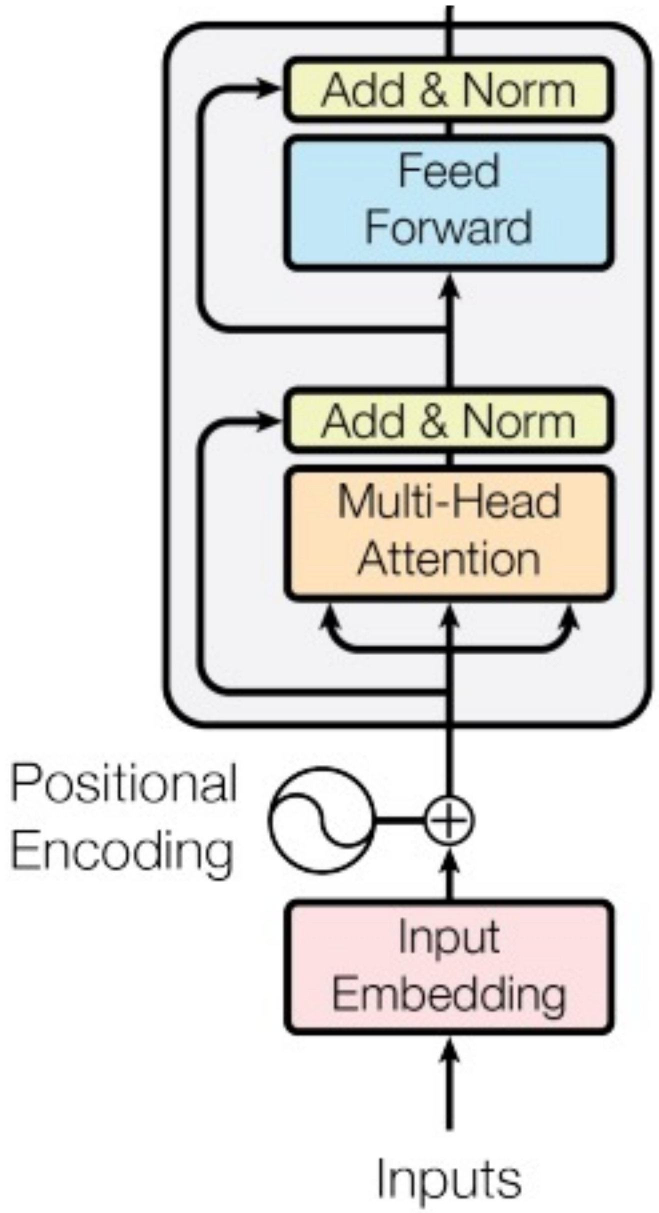
Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

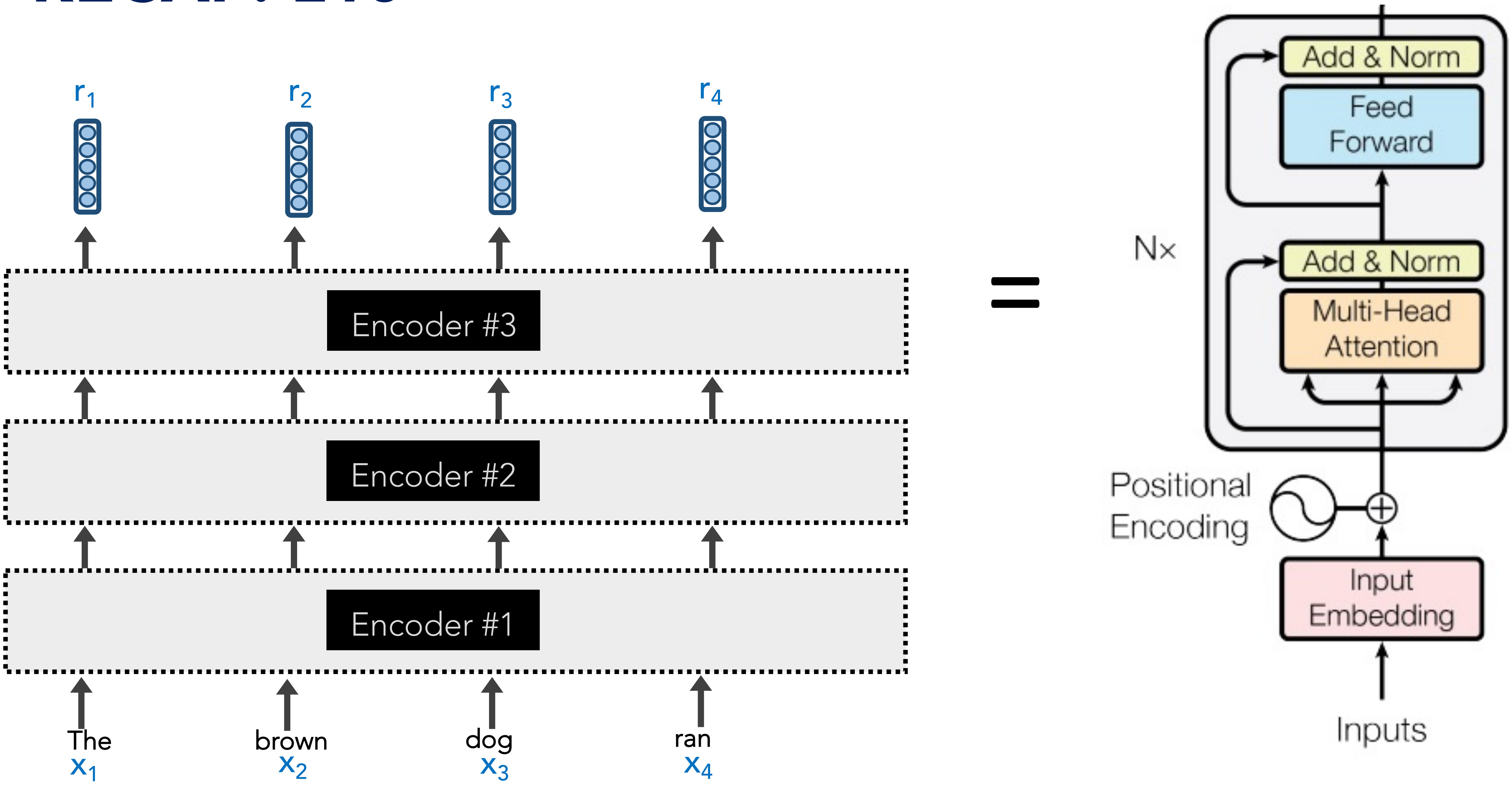
RECAP: L10


r1 r2 r3 r4
Encoder
FFNN
+ x Residual Connections +LayerNorm
z1A z1B z1C z2A z2B z2C z3A z3B z3C z4A z4B z4C
Self-attention Head
The brown dog ran
x1 x2 x3 x4

=


Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Positional Encoding
Input Embedding
Inputs

RECAP: L10

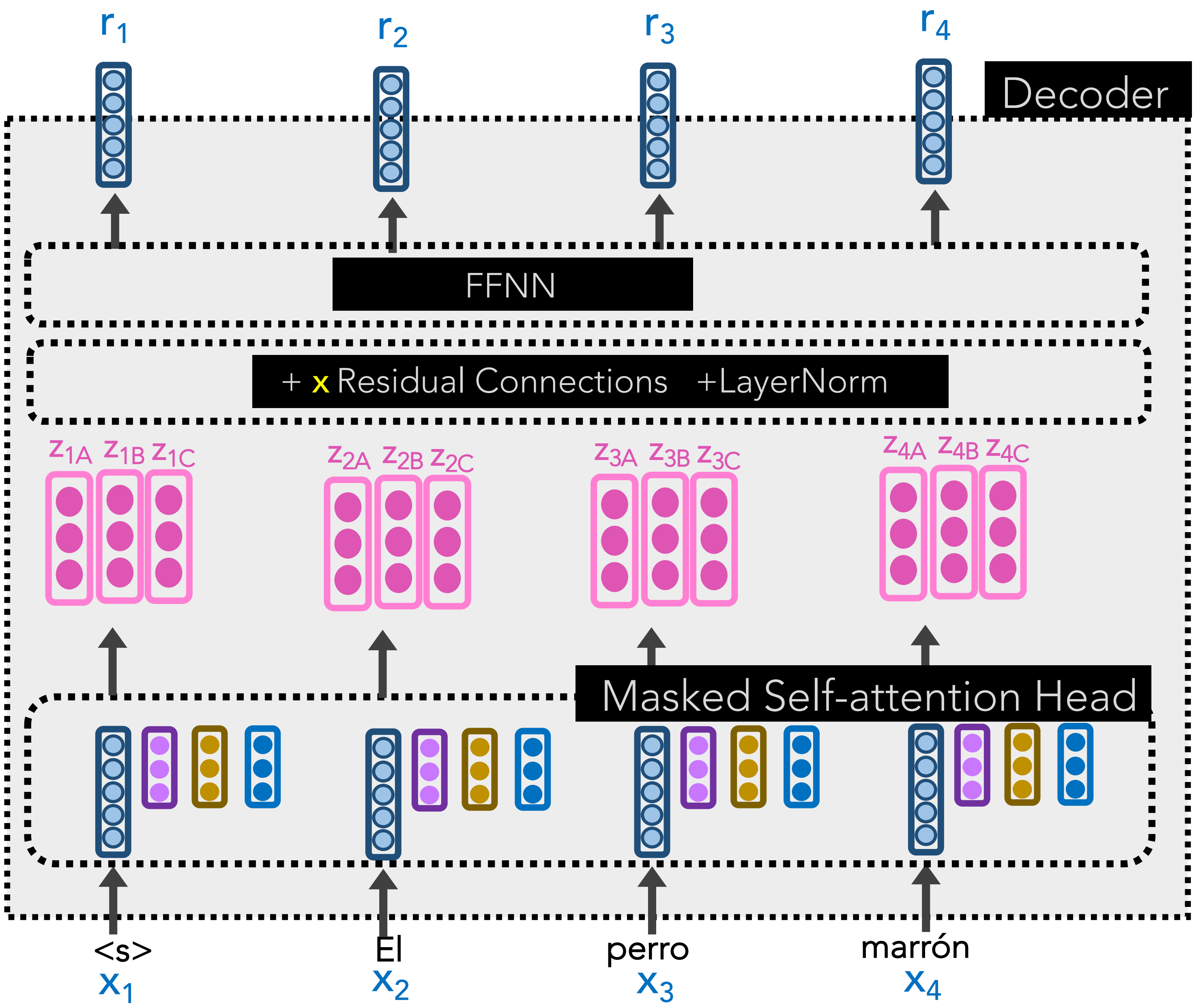


r_1
r_2
r_3
r_4
Encoder #3
Encoder #2
Encoder #1
The
x_1
brown
x_2
dog
x_3
ran
x_4
=
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Positional Encoding
Input Embedding
Inputs

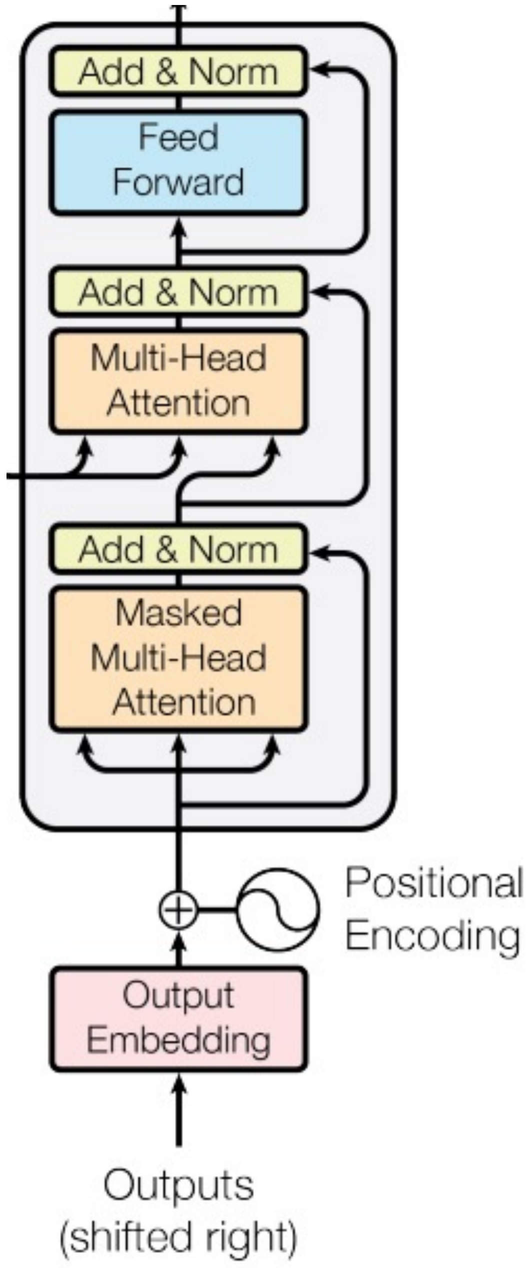
RECAP: L10

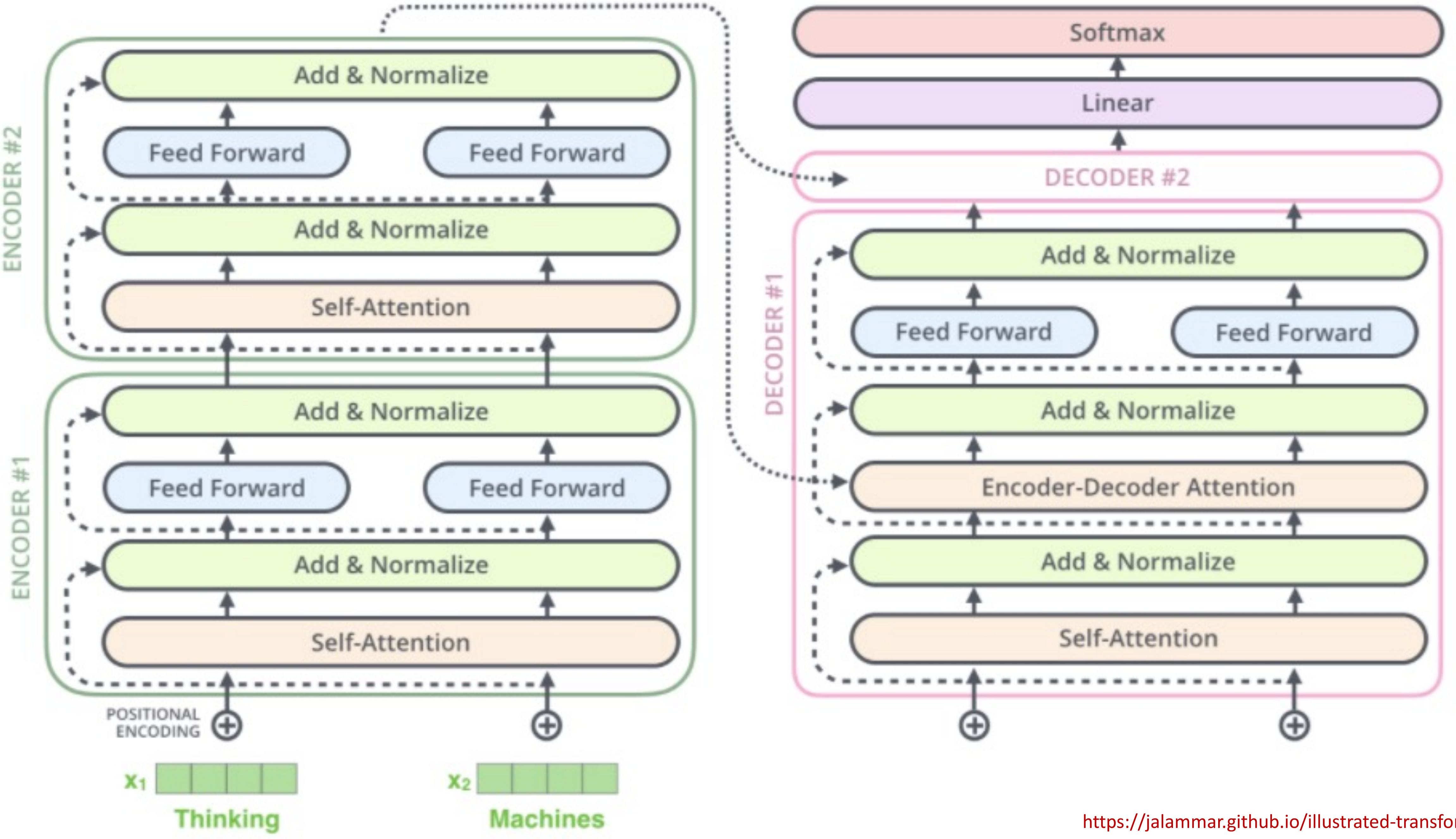
The original Transformer model was intended for Machine Translation, so it had Decoders, too

Transformer Decoder



=

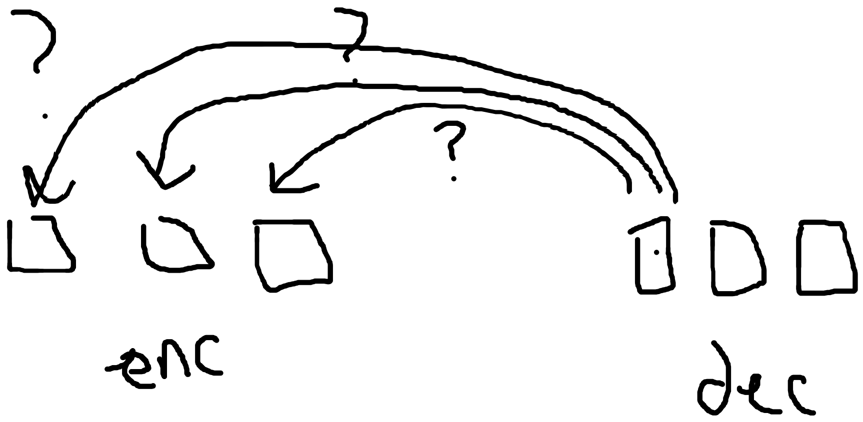




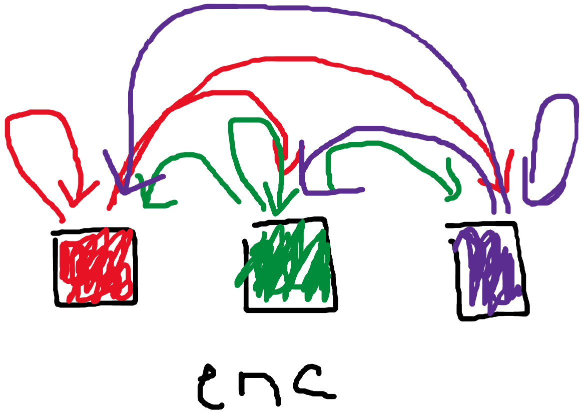
https://jalammar.github.io/illustrated-transformer/

Three ways to Attend

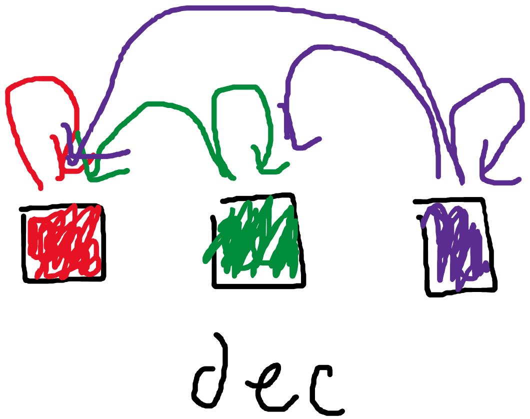
Encoder-Decoder Attention

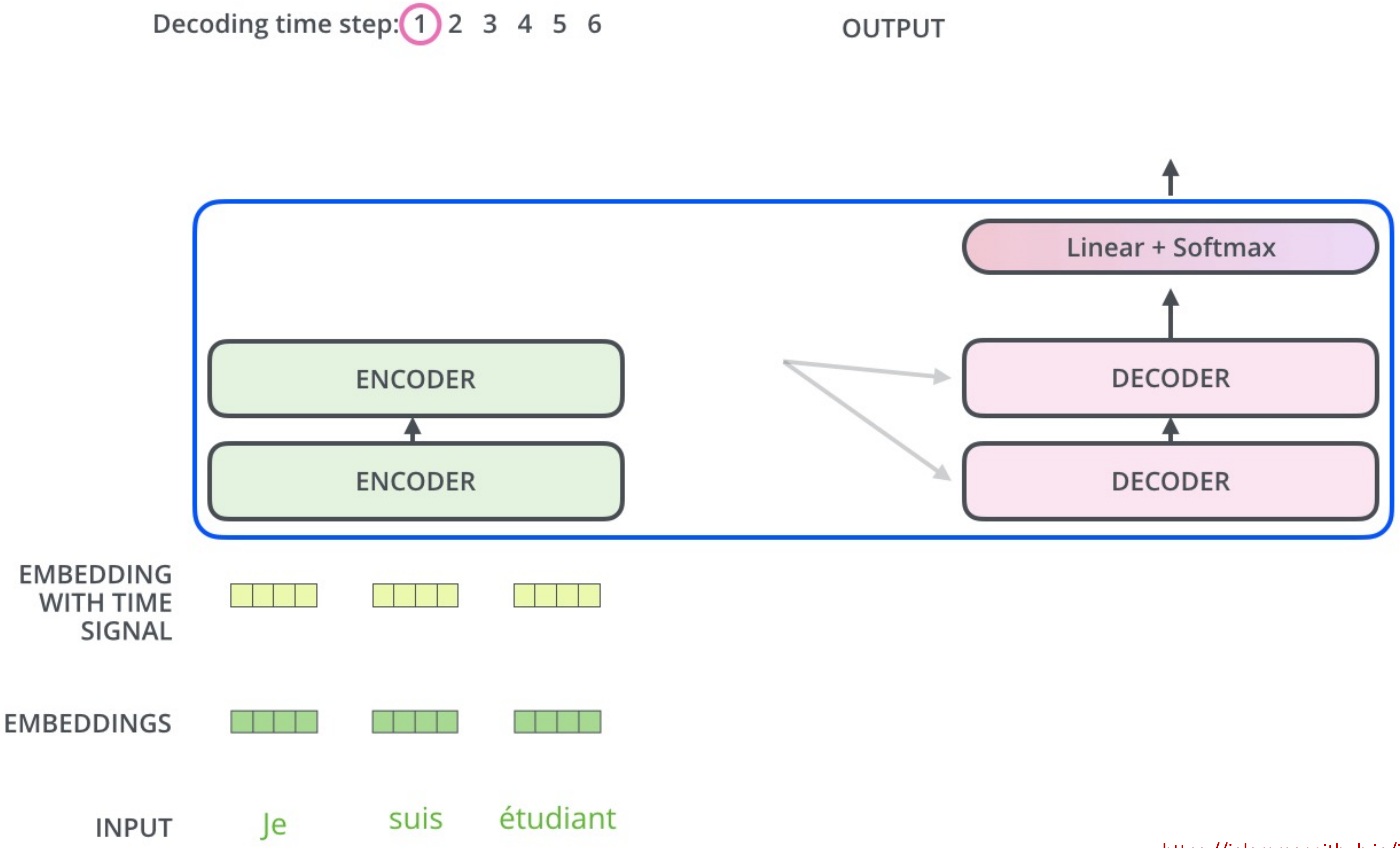


Encoder Self-Attention

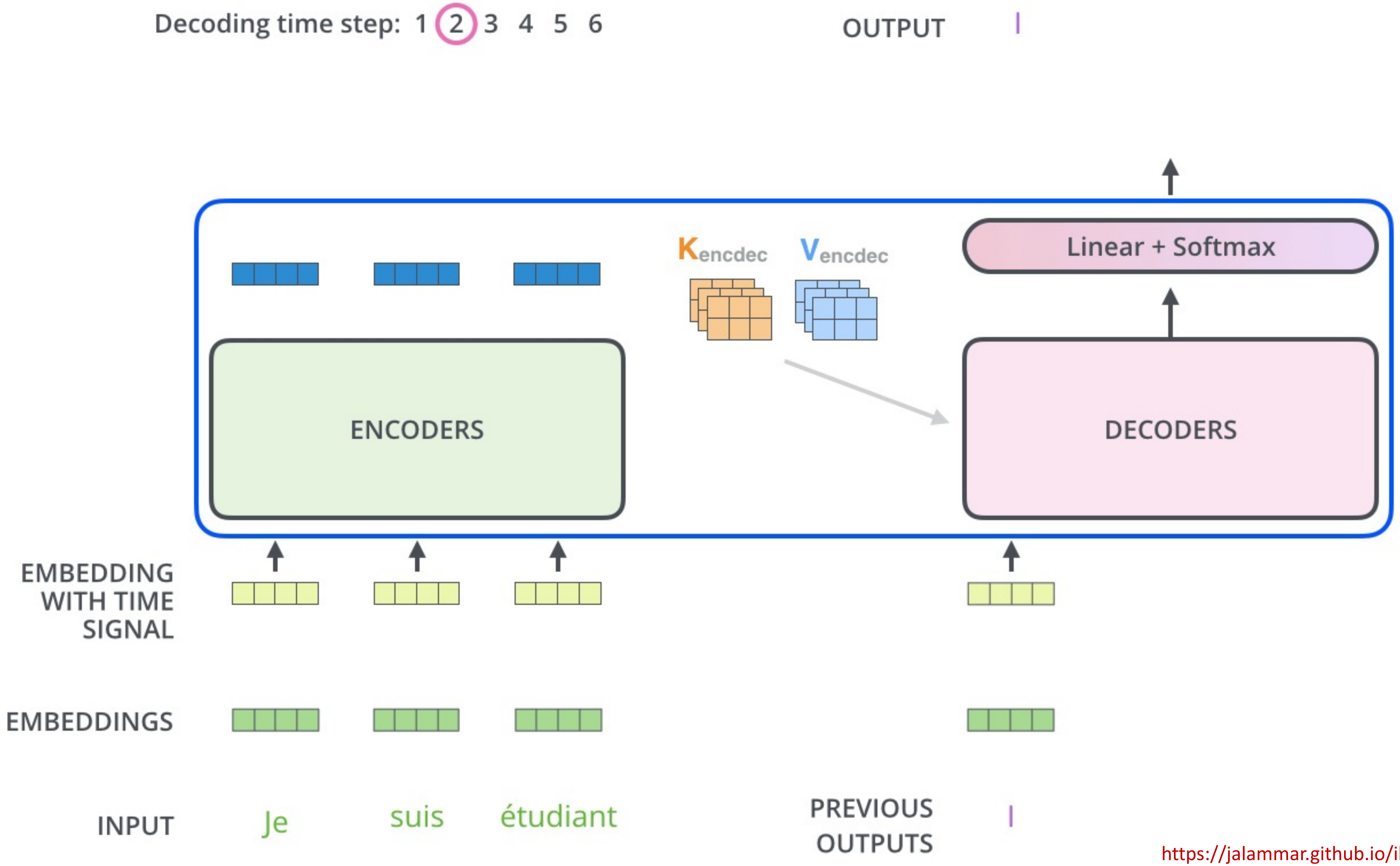


Decoder Masked Self-Attention





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



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

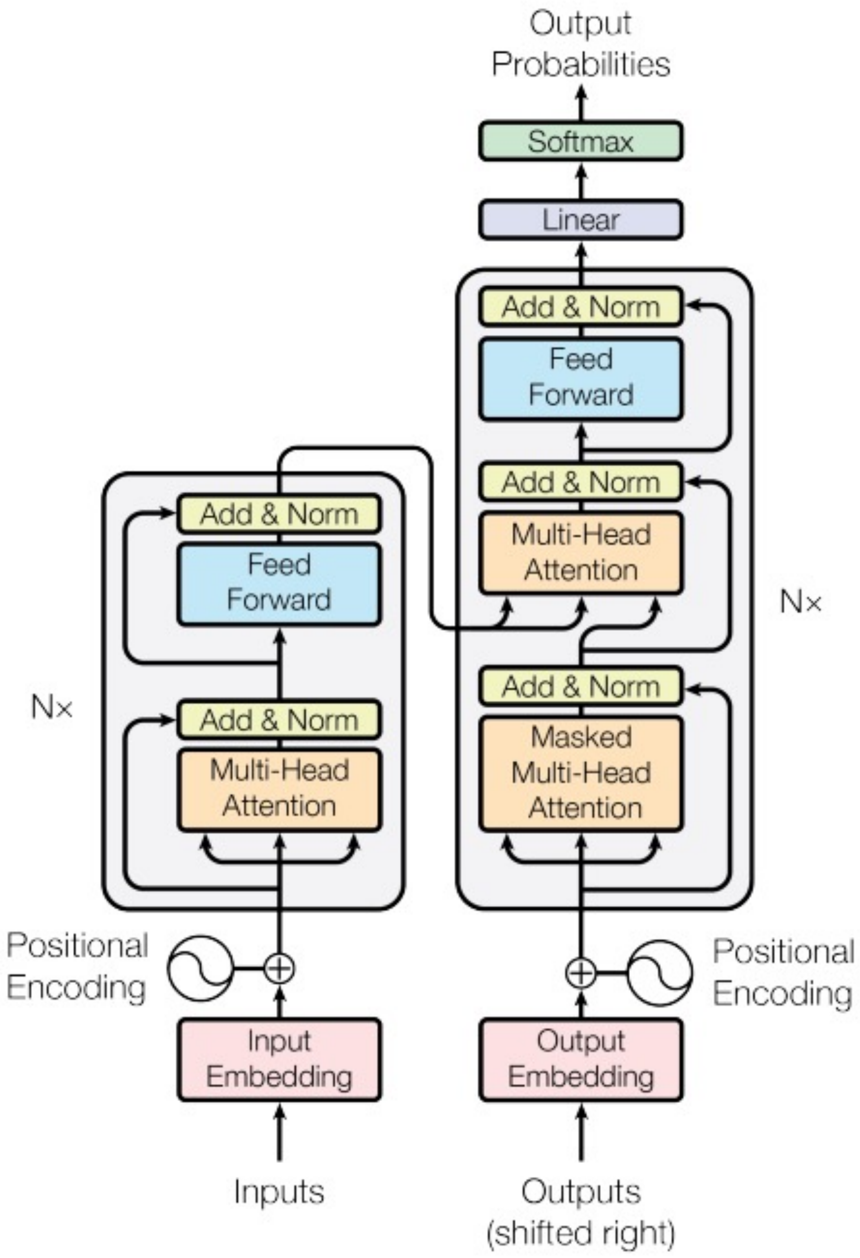


Figure 1: The Transformer - model architecture.

Attention is All you Need (2017) https://arxiv.org/pdf/1706.03762.pdf

Loss Function: cross-entropy (predicting translated word)

Training Time: ~4 days on (8) GPUs

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

n = sequence length

d = length of representation (vector)

Q: Is the complexity of self-attention good?

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Important: when learning dependencies b/w words, you don't want long paths. Shorter is better.

Self-attention connects all positions with a constant # of sequentially executed operations, whereas RNNs require $O(n)$.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

https://arxiv.org/pdf/1706.03762.pdf

Machine Translation results: state-of-the-art (at the time)

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	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Machine Translation results: state-of-the-art (at the time)

You can <u>train</u> to translate from Language A to Language B.

Then <u>train</u> it to translate from Language B. to Language C.

<u>Then, without training,</u> it can translate from Language A to Language C

- What if we don’t want to decode/translate?
- Just want to perform a particular task (e.g., classification)
- Want even more robust, flexible, rich representation!
- Want to explicitly capture fluency, somehow.

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

Everything we've discussed so far

GOALS/TASKS:

- Learn distributed representations
  - word2vec (type-based word embeddings)
  - RNNs/LSTMs (token-based contextualized)
- Machine Translation
- Text classification

MODELS:

- n-gram (not neural)
- RNNs/LSTMs
- seq2seq
- Transformer Encoder/Decoder

Everything we've discussed so far

GOALS/TASKS:

- Learn distributed representations
  - word2vec (type-based word embeddings)
  - RNNs/LSTMs (token-based contextualized)
- Machine Translation
- Text classification

MODELS:

- n-gram (not neural)
- RNNs/LSTMs
- seq2seq
- Transformer Encoder/Decoder

Aside from text classification (e.g., IMDb sentiments), we've only worked with unlabelled, naturally occurring data so far.

There's a vast ocean of interesting tasks that require labelled data, and we often perform different types of learning in order to better leverage our *limited* labelled data.

Types of Data

UNLABELLED

- Raw text (e.g., web pages)
- Parallel corpora (e.g., for translations)

LABELLED

- Linear/unstructured
  - N-to-1 (e.g., sentiment analysis)
  - N-to-N (e.g., POS tagging)
  - N-to-M (e.g., summarization)
- Structured
  - Dependency parse trees
  - Constituency parse trees
  - Semantic Role Labelling

Types of Data

UNLABELLED

- Raw text (e.g., web pages)
- Parallel corpora (e.g., for translations)

We most often about this type of data

LABELLED

- Linear/unstructured
  - N-to-1 (e.g., sentiment analysis)
  - N-to-N (e.g., POS tagging)
  - N-to-M (e.g., summarization)
- Structured
  - Dependency parse trees
  - Constituency parse trees
  - Semantic Role Labelling

Types of Data

Labelled data is a scarce commodity.

How can we get more of it?

How can we leverage more plentiful, other data (either labelled or unlabelled) so as to make better use of our limited labelled data?

Types of Learning

One axis that refers to our style of using/learning our data:

- Multi-task Learning
- Transfer Learning
- Pre-training

One axis that hinges upon the type of data we have:

- Supervised Learning
- Unsupervised Learning
- Self-supervised Learning
- Semi-supervised Learning

Types of Learning

One axis that refers to our style of using/learning our data:

Multi-task Learning = general term for training on **multiple tasks**

Transfer Learning = type of multi-task learning where **we only care about one of the tasks**

Pre-training = type of transfer learning where we **first focus on one objective**

See chalkboard for example

Multi-task heuristics

- Ideally, your tasks should be closely related (e.g., constituency parsing and dependency parsing)
- Multi-task learning may help improve the task that has limited data
  - General domain → specific domain (e.g., all of the web's text -> law text)
  - High-resourced language → low-resourced language (e.g., English -> Igbo)
  - Unlabelled text → labelled text (e.g., language model -> named entity recognition)

Inspired by or based on http://www.phontron.com/class/anlp2021/assets/slides/anlp-08-pretraining.pdf

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

BERT

Bidirectional Encoder Representations from Transformers

Like *Bidirectional LSTMs*, let's look in **both** directions

BERT

Bidirectional Encoder Representations from Transformers

Let's only use Transformer *Encoders*, no Decoders

BERT

Bidirectional Encoder Representations from Transformers

It's a language model that builds rich representations

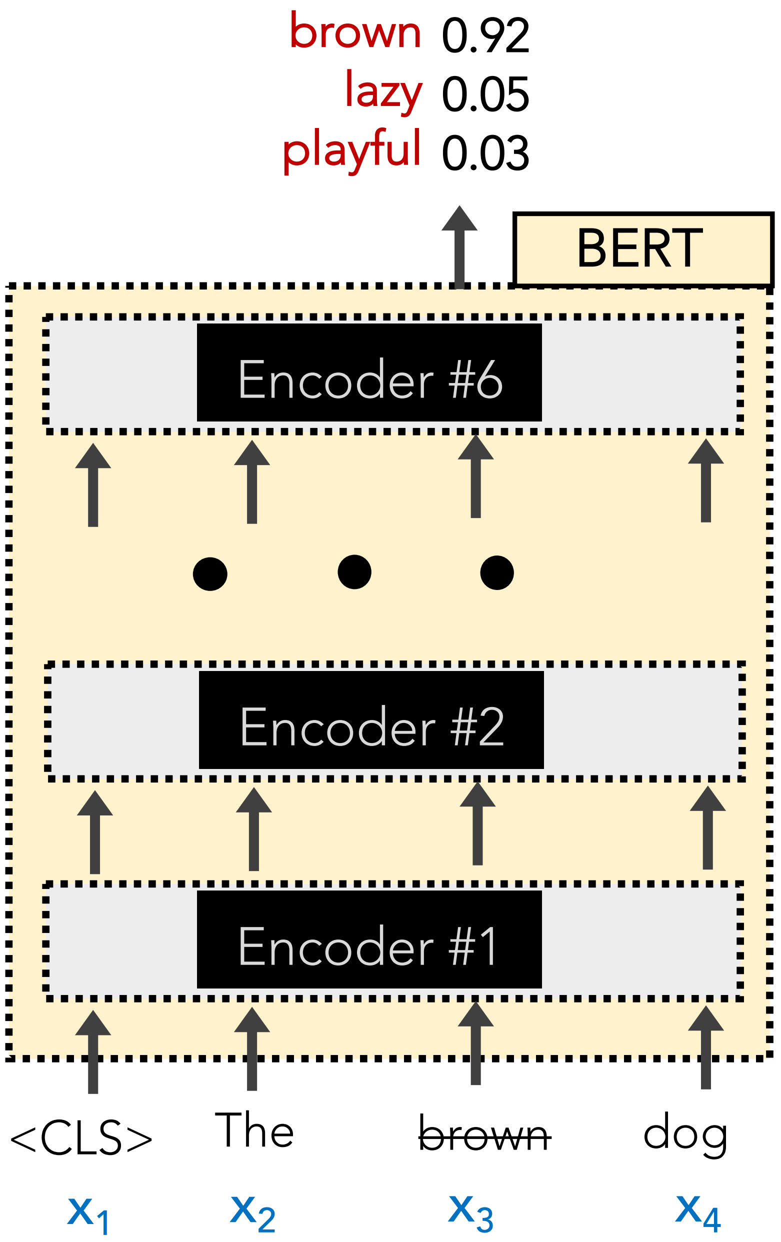
Naming convention

Many deep learning models, including pre-trained ones with cute names (e.g., ELMo, BERT, ALBERT, GPT-3), refer to an exact combination of:

- The model's architecture
- The training objective to pre-train (e.g., MLM prediction)
- The data (e.g., Google BooksCorpus, Wikipedia)

Many people abuse the terms and swap out components.

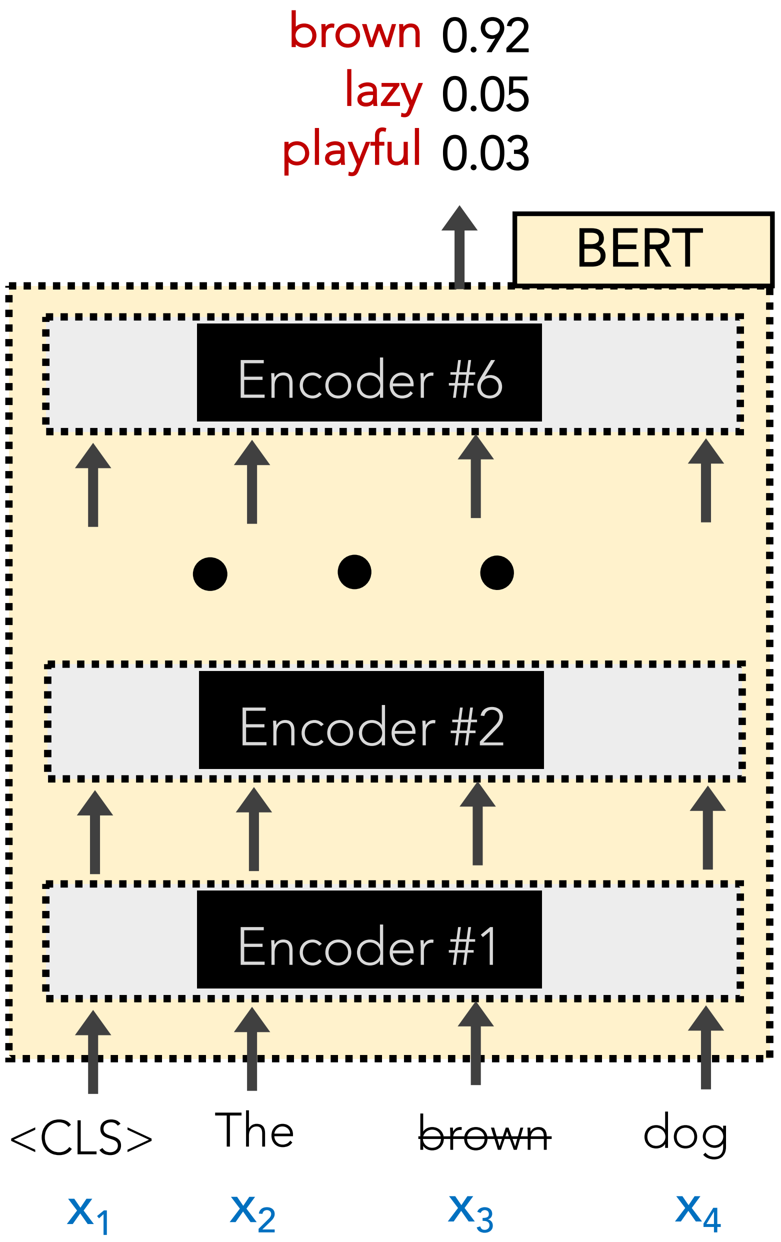
BERT



BERT:

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

BERT



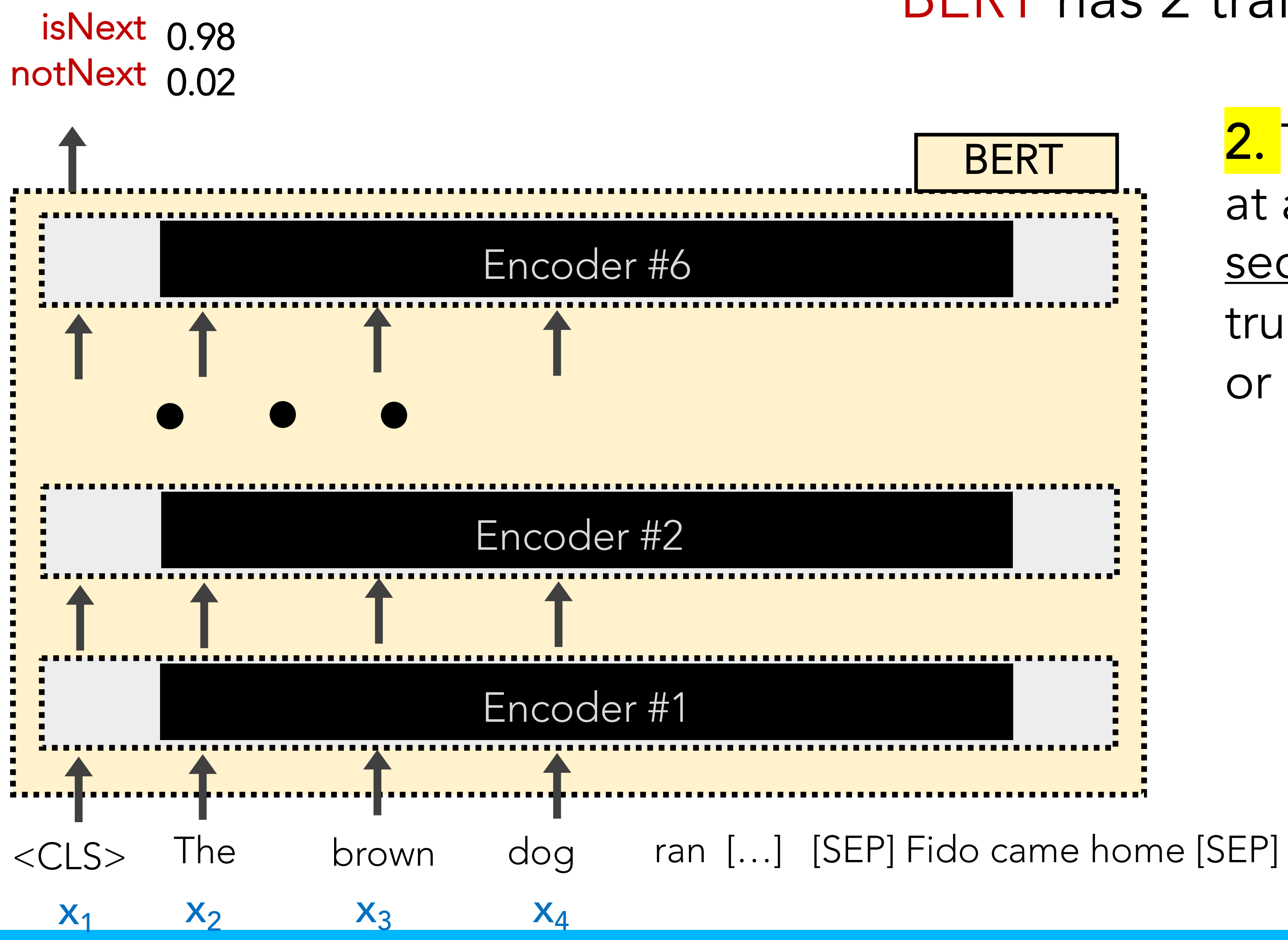
BERT has 2 training objectives:

1. Predict the **Masked word** (a la CBOW)

15% of all input words are randomly masked.

- 80% become [MASK]
- 10% become revert back
- 10% become are deliberately corrupted as wrong words

BERT



BERT has 2 training objectives:

2. Two sentences are fed in at a time. Predict the if the second sentence of input truly follows the first one or not.

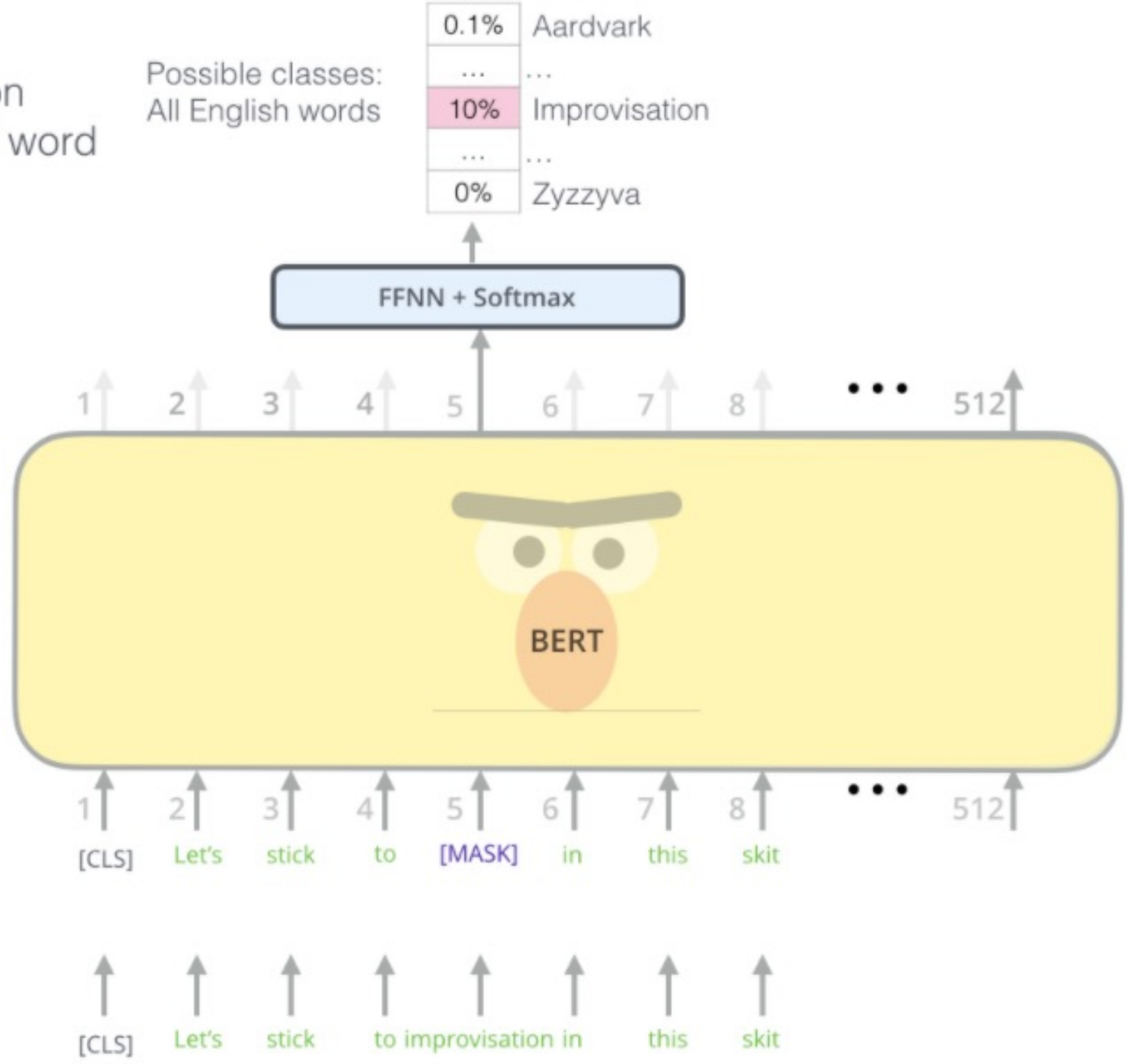
BERT (alternate view)

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

Use the output of the masked word's position to predict the masked word

Randomly mask 15% of tokens

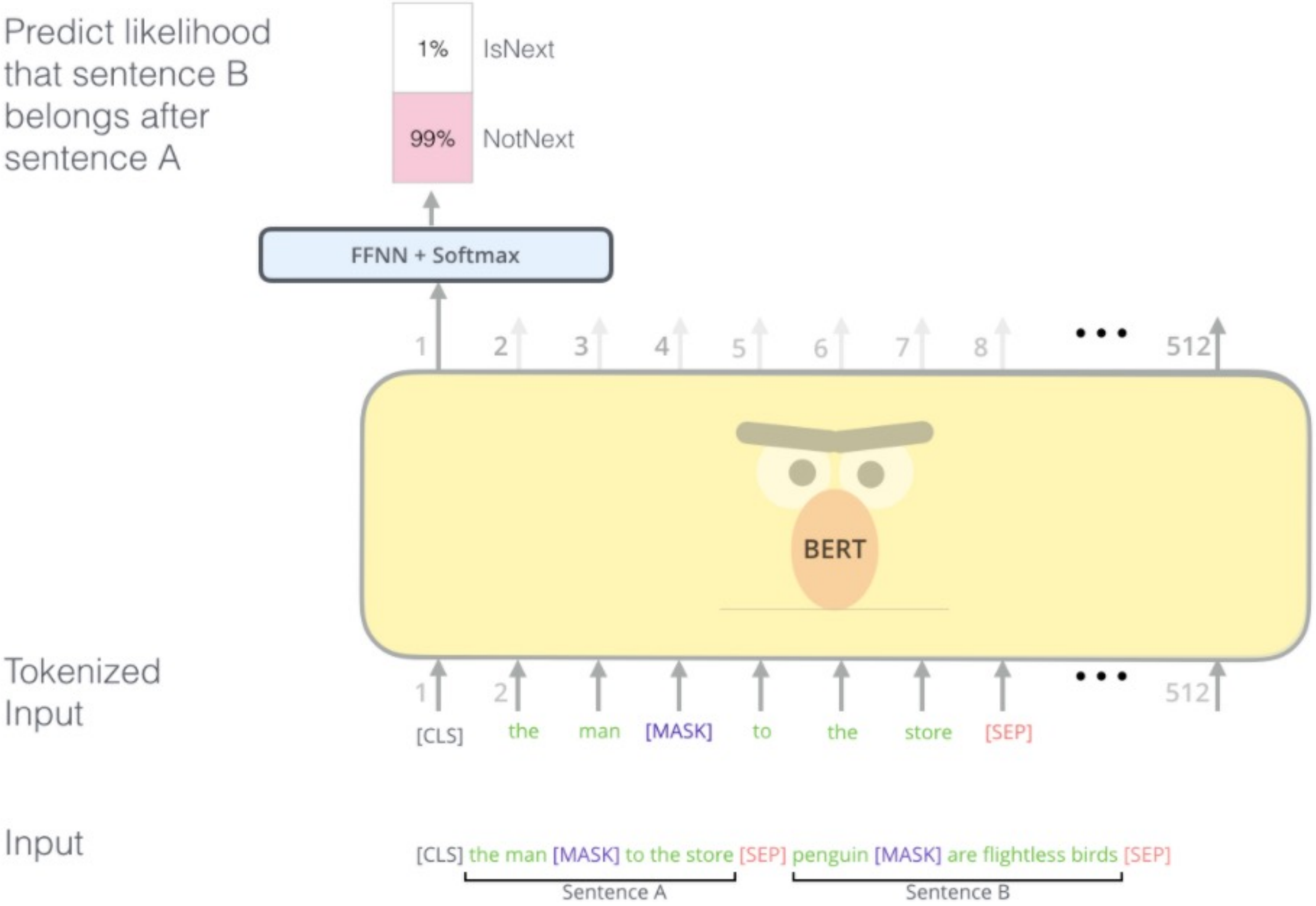
Input



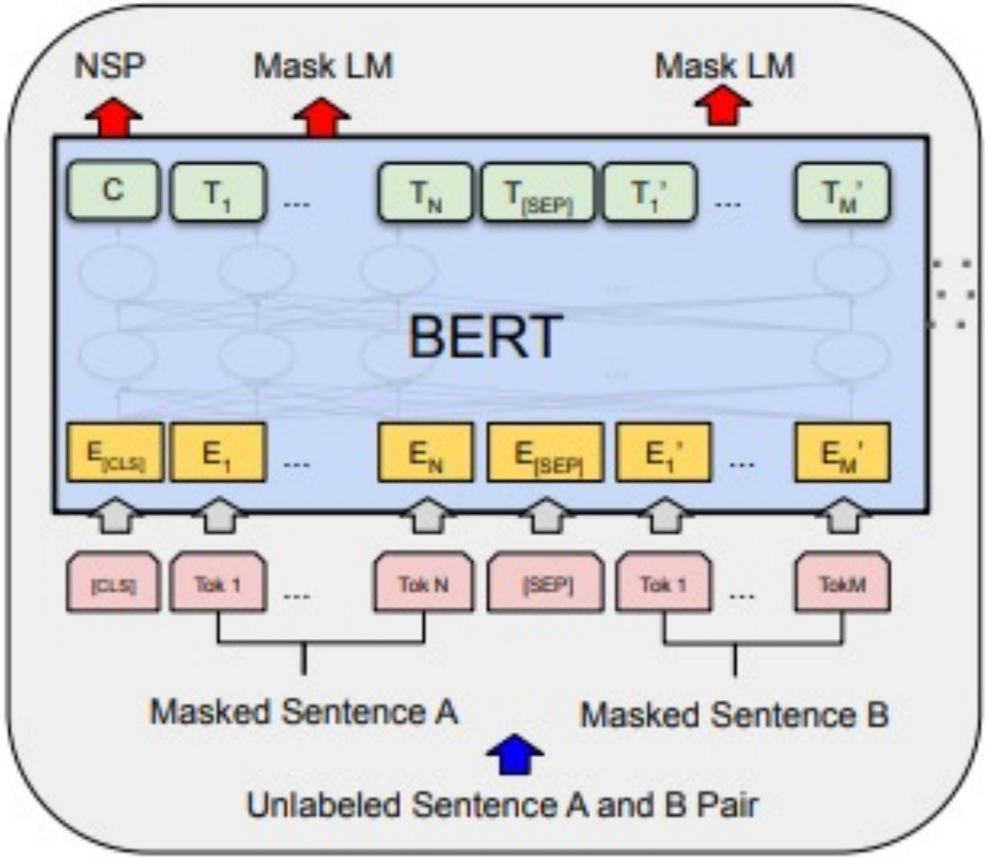
BERT's clever language modeling task masks 15% of words in the input and asks the model to predict the missing word.

BERT (alternate view)

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



BERT



Pre-training

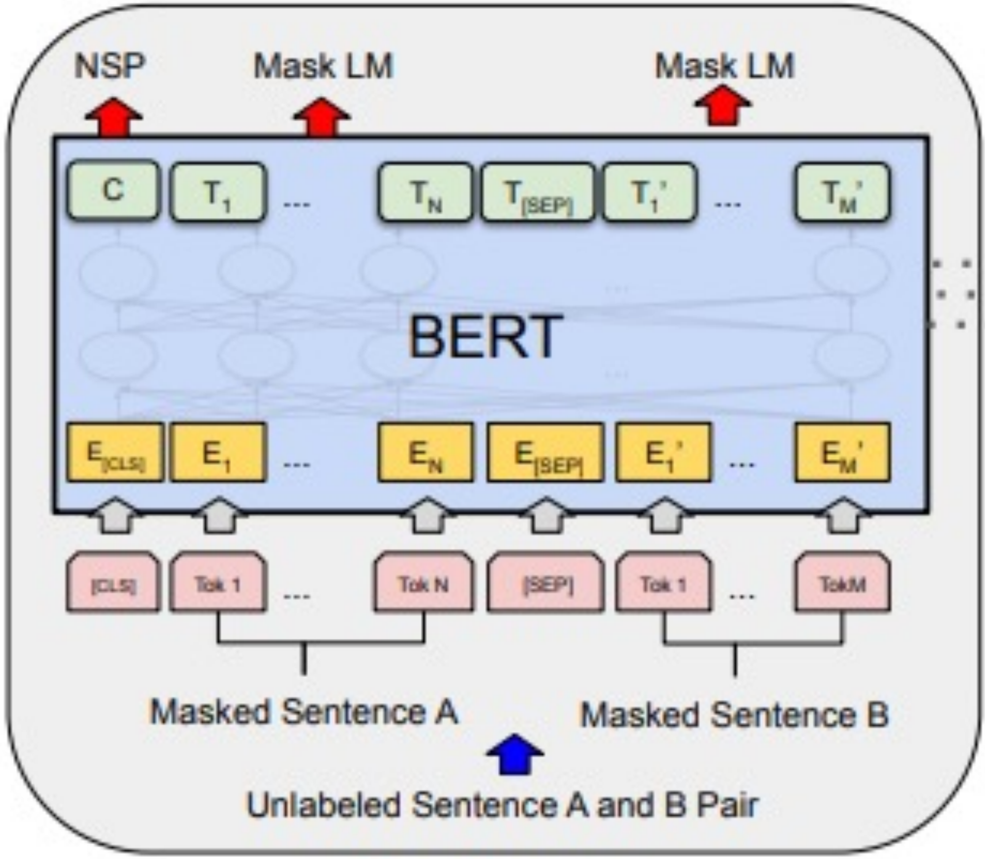
The two sentences are separated by a <SEP> token.

50% of the time, the 2nd sentence is a randomly selected sentence from the corpus.

50% of the time, it truly follows the first sentence in the corpus.

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

BERT



Pre-training

NOTE: BERT also embeds the inputs by their WordPiece embeddings.

WordPiece is a <u>sub-word tokenization</u> learns to merge and use characters based on which pairs maximize the likelihood of the training data if added to the vocab.

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

BERT's inputs

https://arxiv.org/pdf/1810.04805.pdf

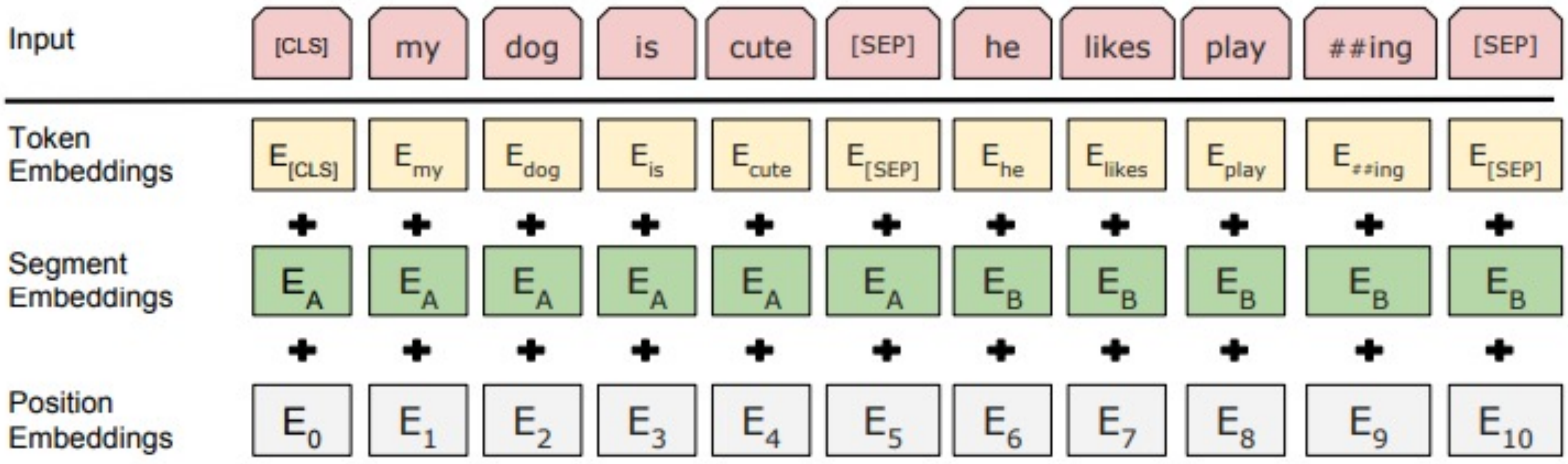


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

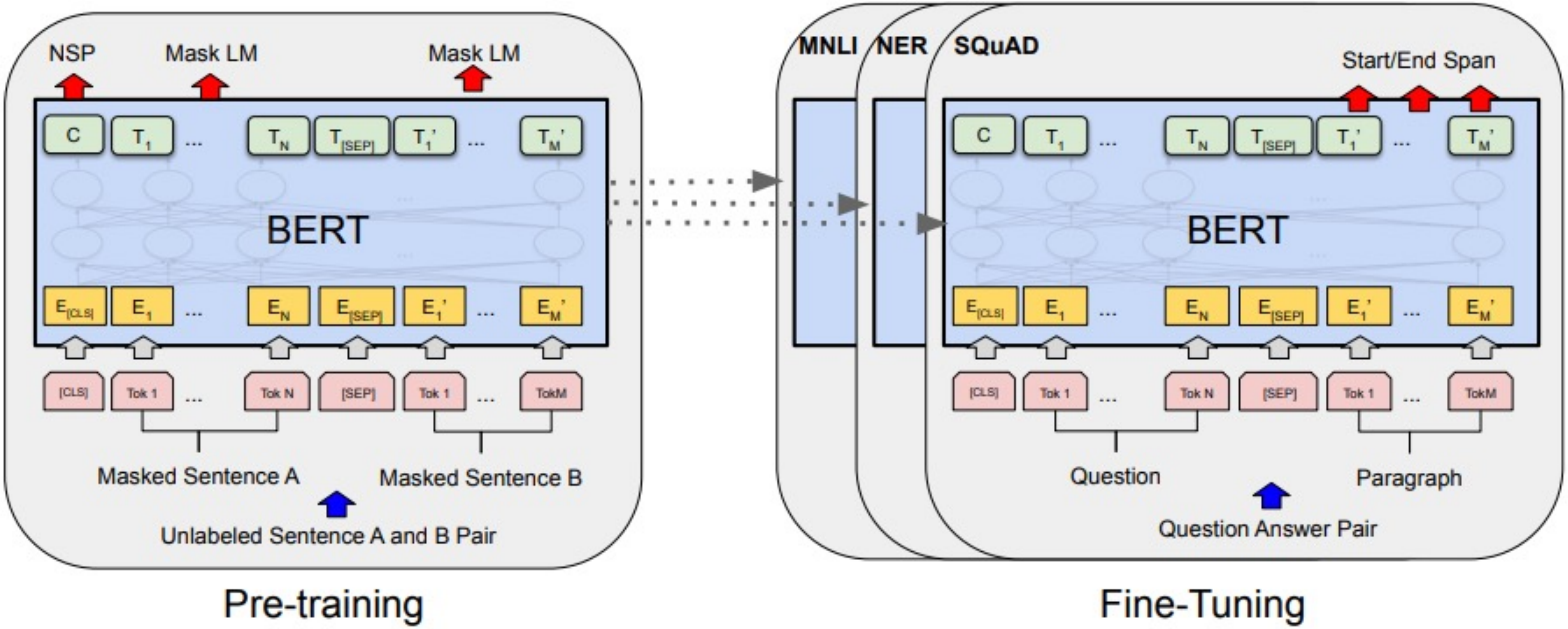
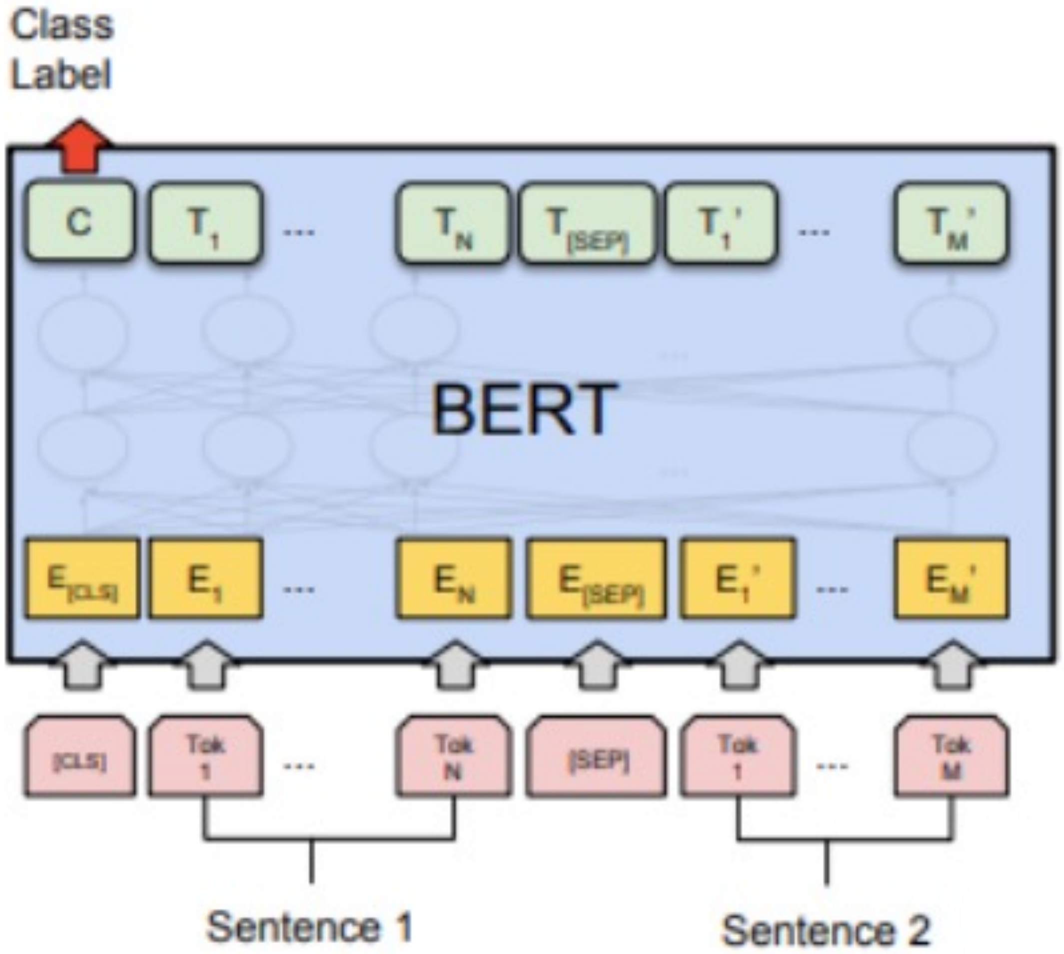


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are 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).

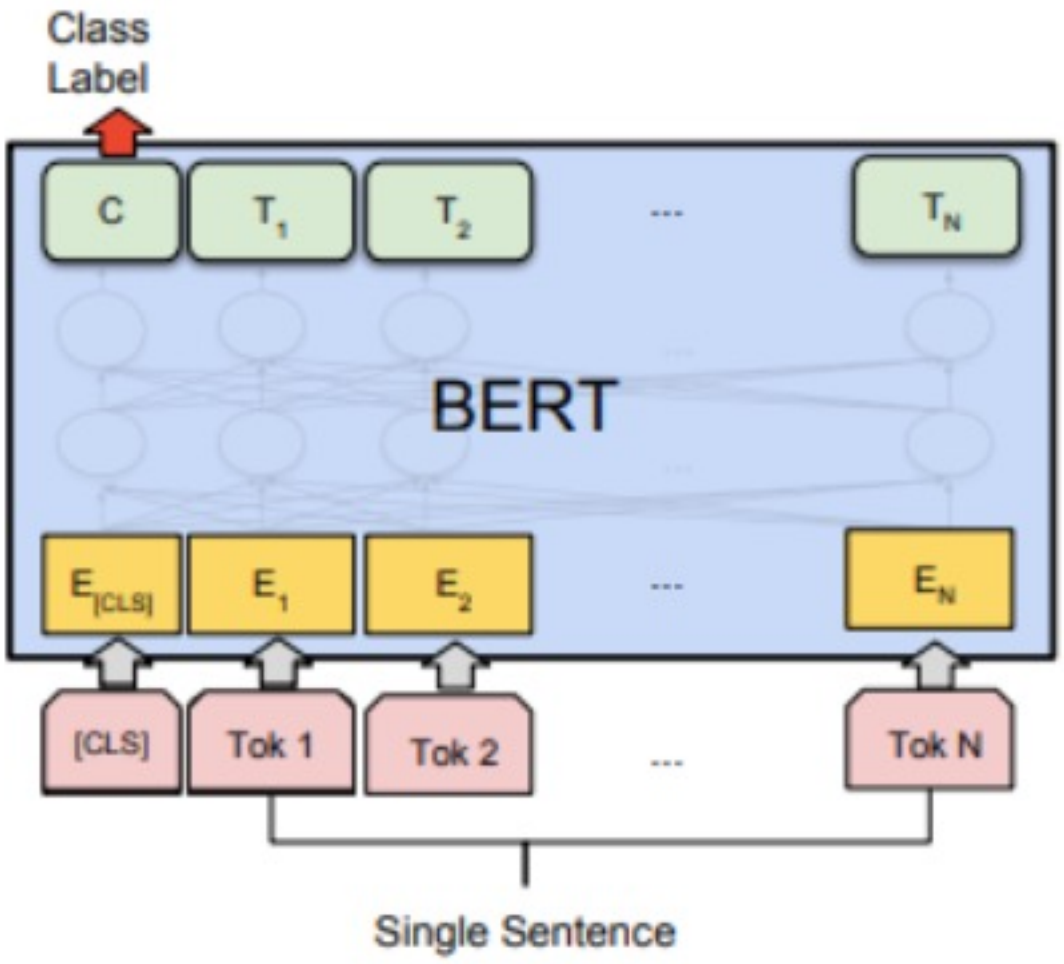
https://aclanthology.org/N19-1423.pdf

BERT fine-tuning



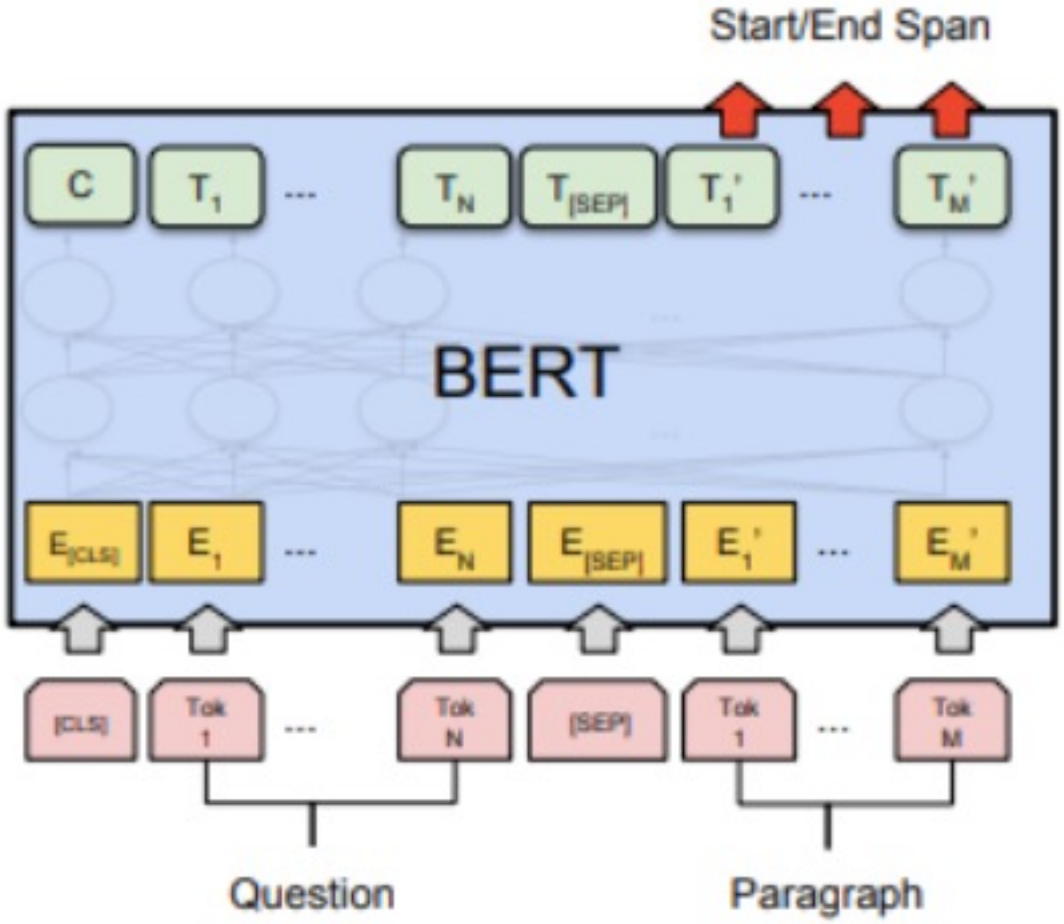
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

BERT fine-tuning



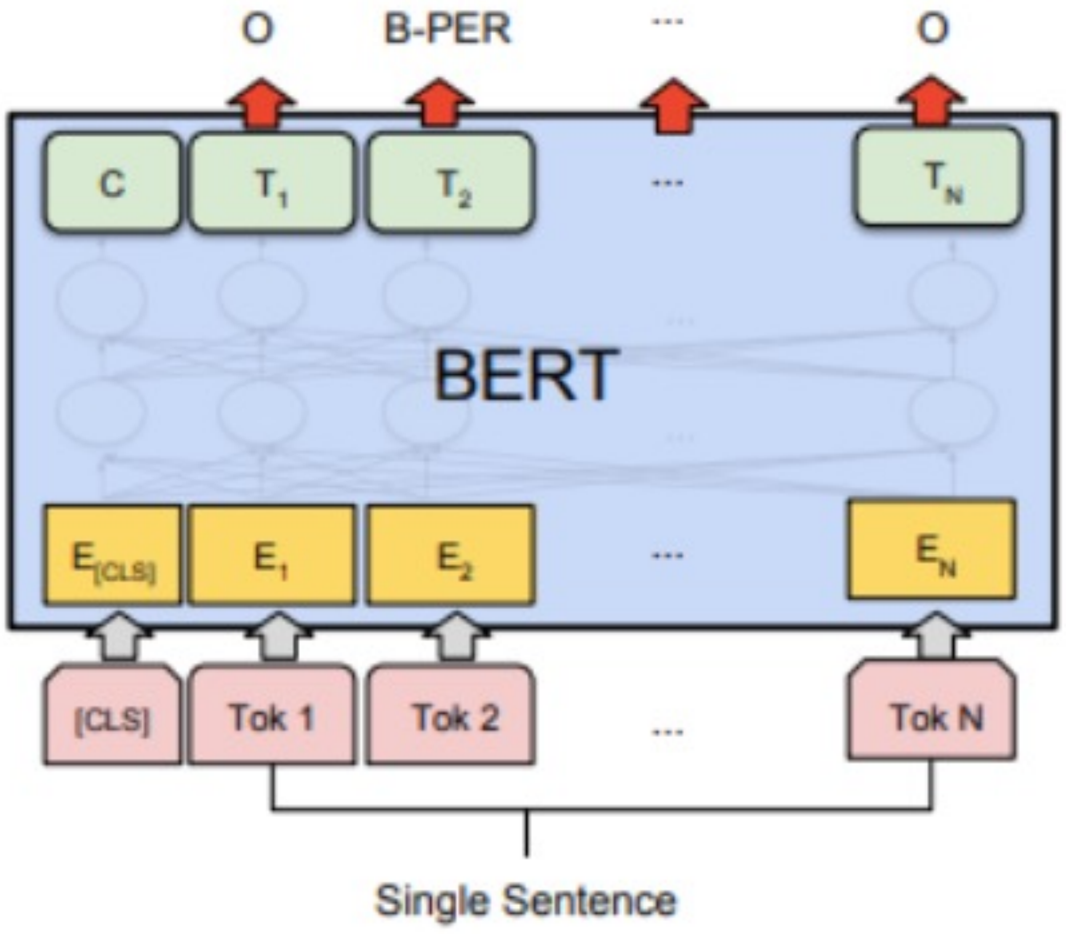
(b) Single Sentence Classification Tasks: SST-2, CoLA

BERT fine-tuning



(c) Question Answering Tasks:
SQuAD v1.1

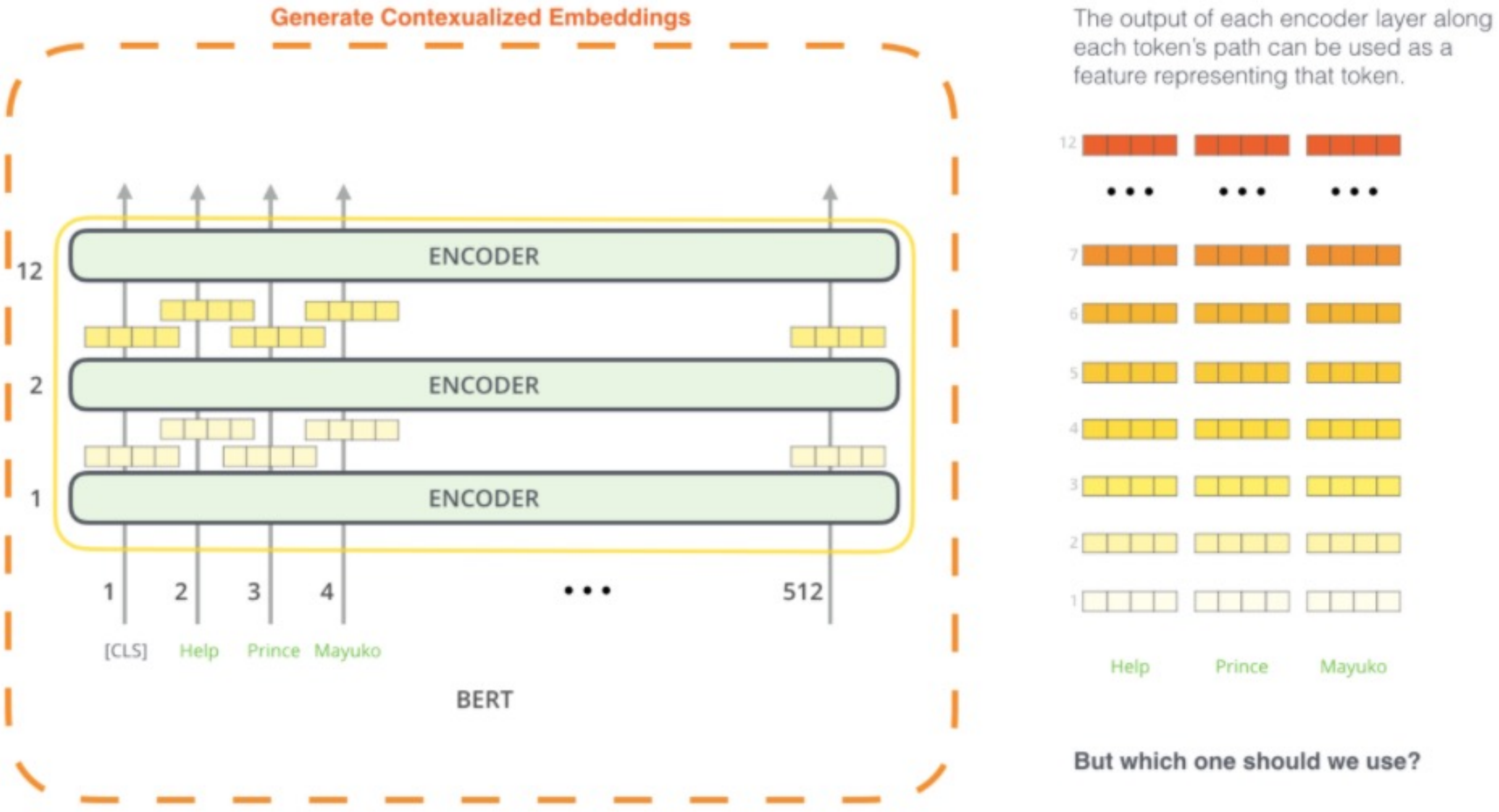
BERT fine-tuning



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

BERT

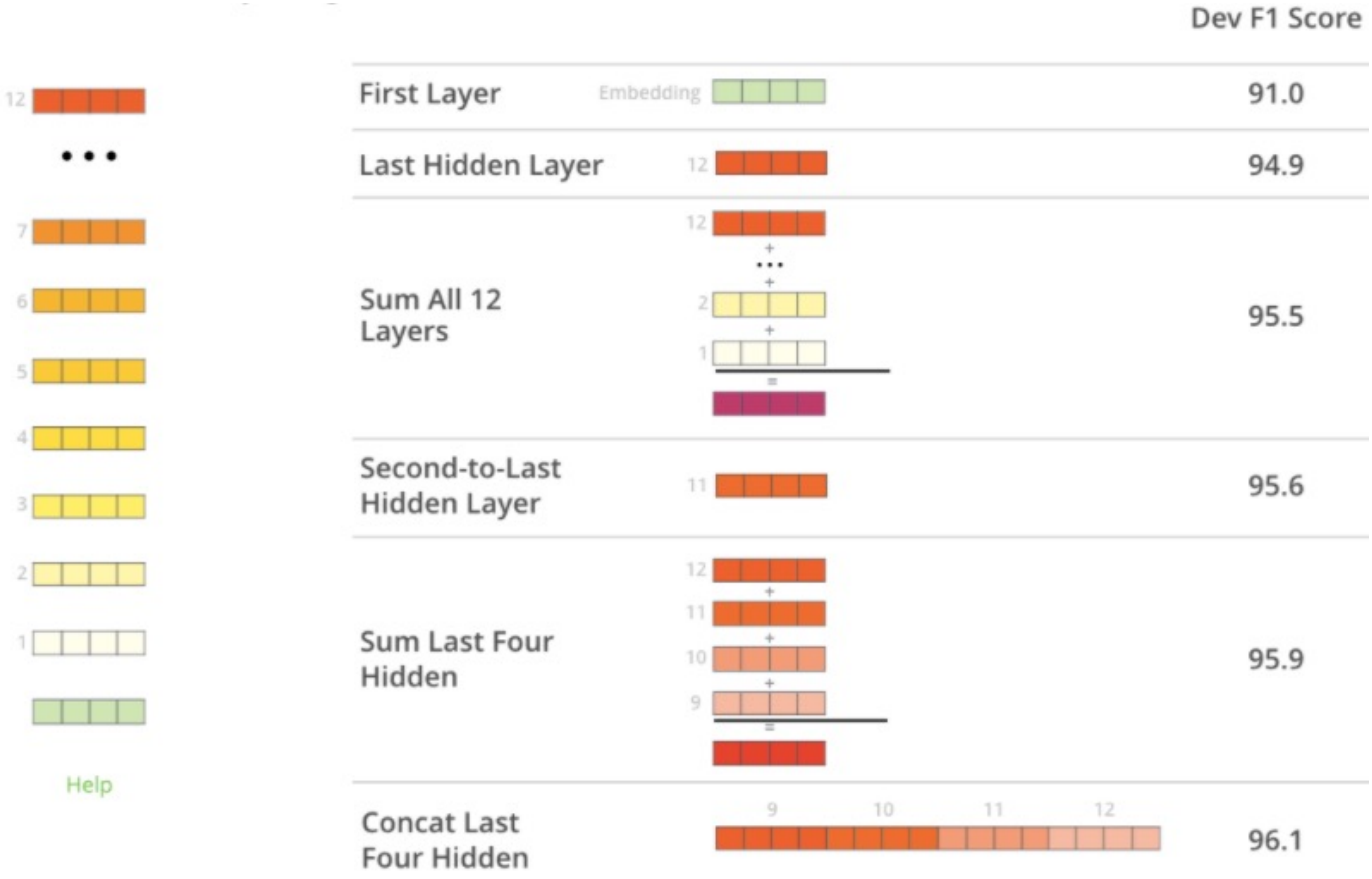
Or, one could extract the contextualized embeddings



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

BERT

Later layers have the best contextualized embeddings



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

BERT

BERT yields 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
$\text{BERT}_{\text{BASE}}$	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).

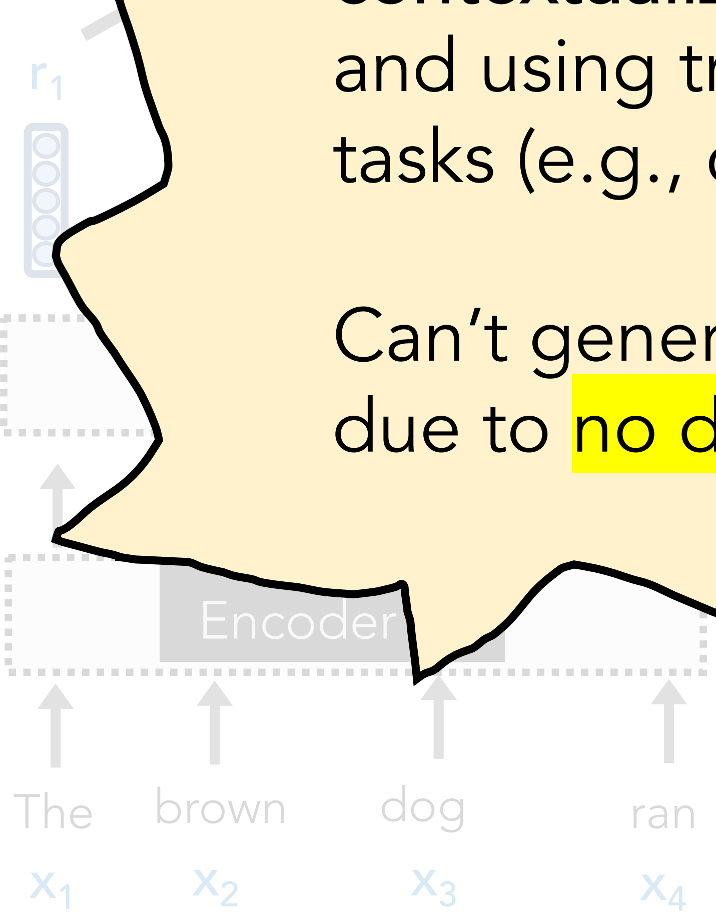
Source: original BERT paper: https://arxiv.org/pdf/1810.04805.pdf

BERT (a Transformer variant)

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.



Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

Outline

- Transformer Decoder
- Learning/Data/Tasks
- BERT
- BERT Fine-Tuning
- Extensions

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

ICPC

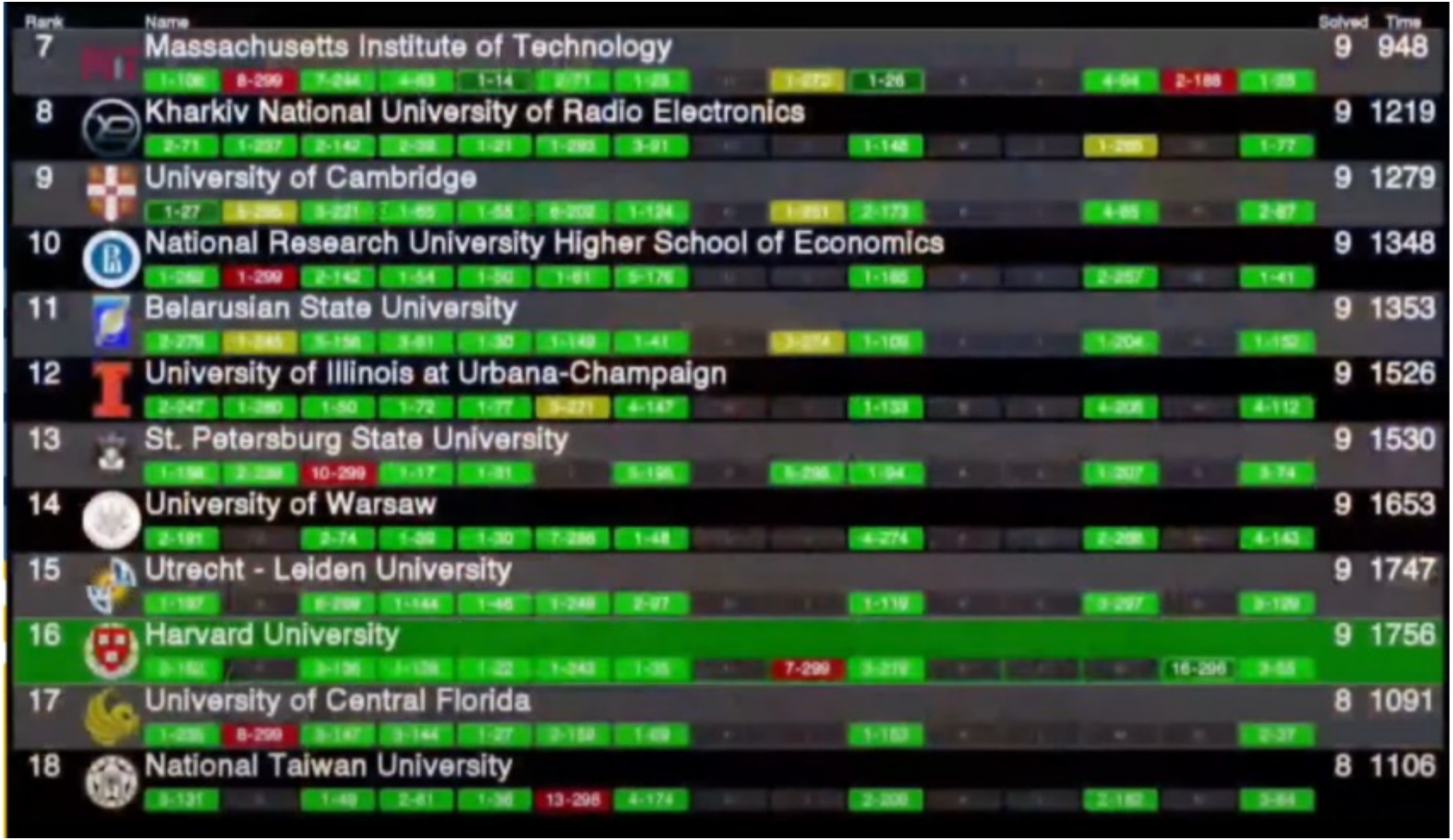
ICPC

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 will compete 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


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