COS 597G: Understanding Large Language Models

PRINCETON UNIVERSITY

Lecture 2: BERT (encoder-only models)

Fall 2022

Some slides are adapted from Jacob Devlin

Announcements

- Make sure that you are added to Slack already
- Draft schedule (including presentation and providing feedback) is up
- Changes in lecture preparation meetings
  - **Monday** 3:30-4pm for **Wednesday** lectures (Danqi's appointment-based office hour is moved to 2:30-3:30)
  - **Friday** 4:45-5:15pm for **Monday** lectures

- This Wednesday
  - Presenters: Abhishek Panigrahi, Victoria Graf
  - Feedback providers: Edward Tian, Zihan Ding, Jiatong Yu, Anirudh Ajith

This lecture

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

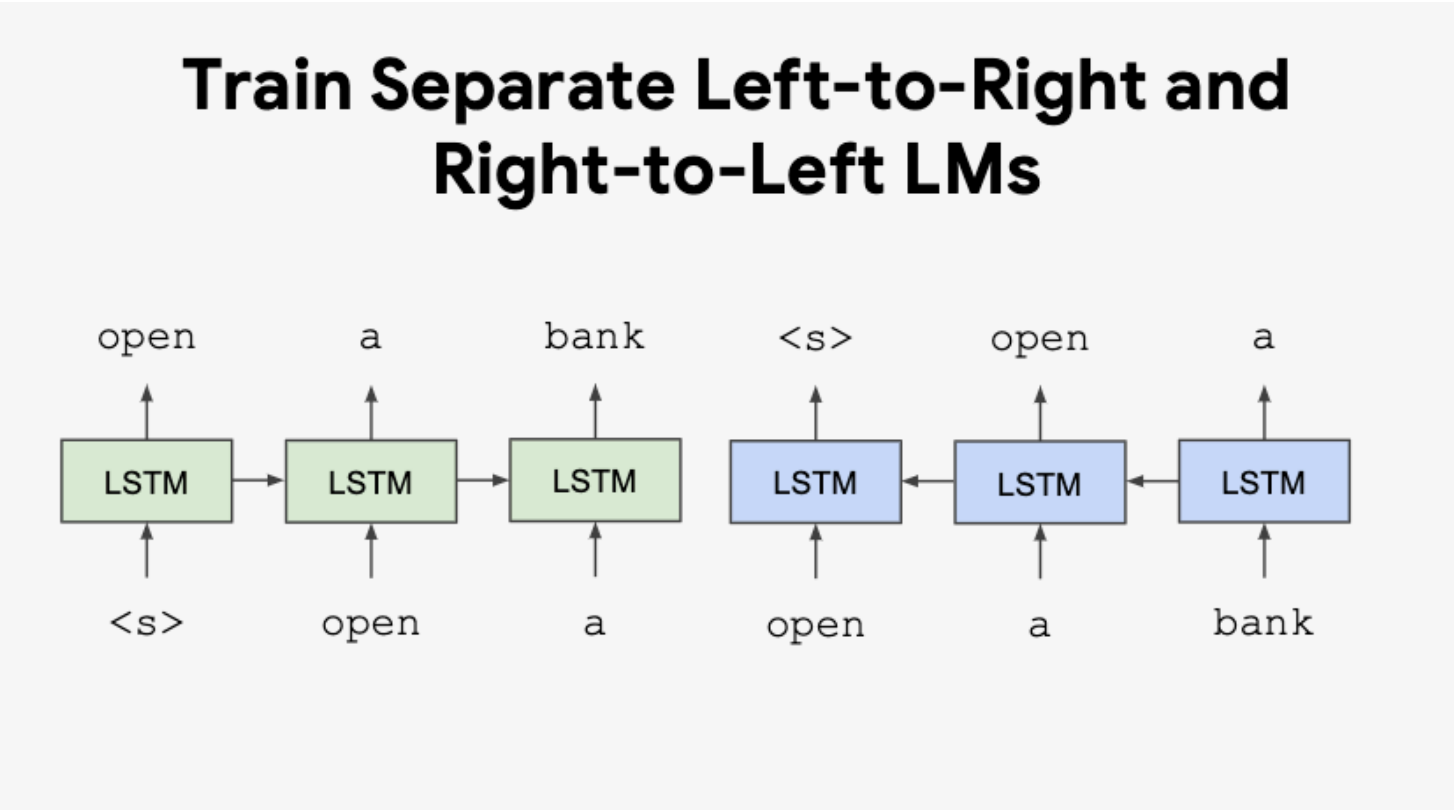
Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

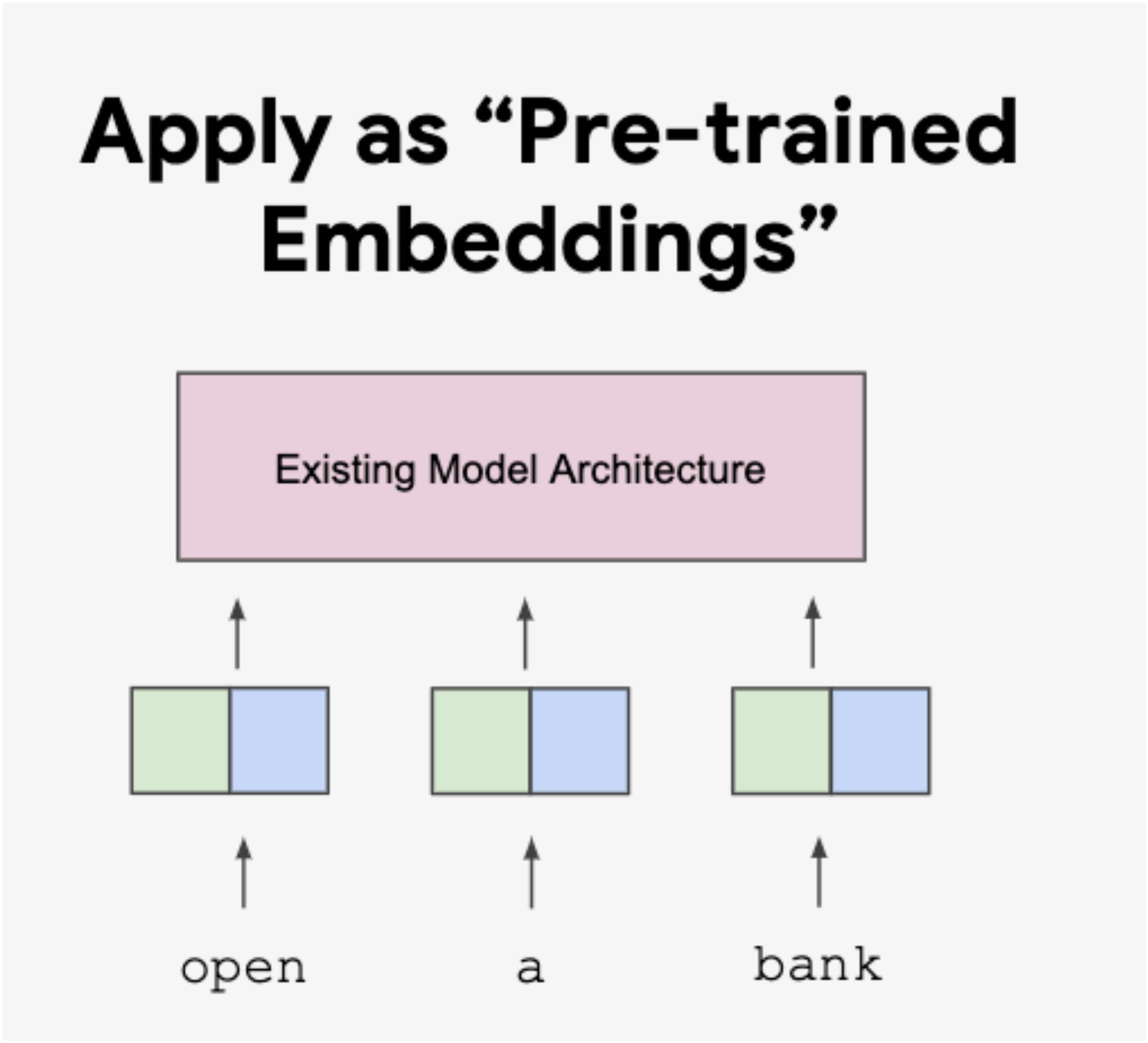
Released in 2018/10, NAACL 2019 best paper

Prior work: ELMo

ELMo (Peters et al., 2018; NAACL 2018 best paper)

- Train two separate **unidirectional** LMs (left-to-right and right-to-left) based on **LSTMs**
- **Feature-based** approach: pre-trained representations used as input to task-specific models
- Trained on **single sentences** from 1B word benchmark (Chelba et al., 2014)

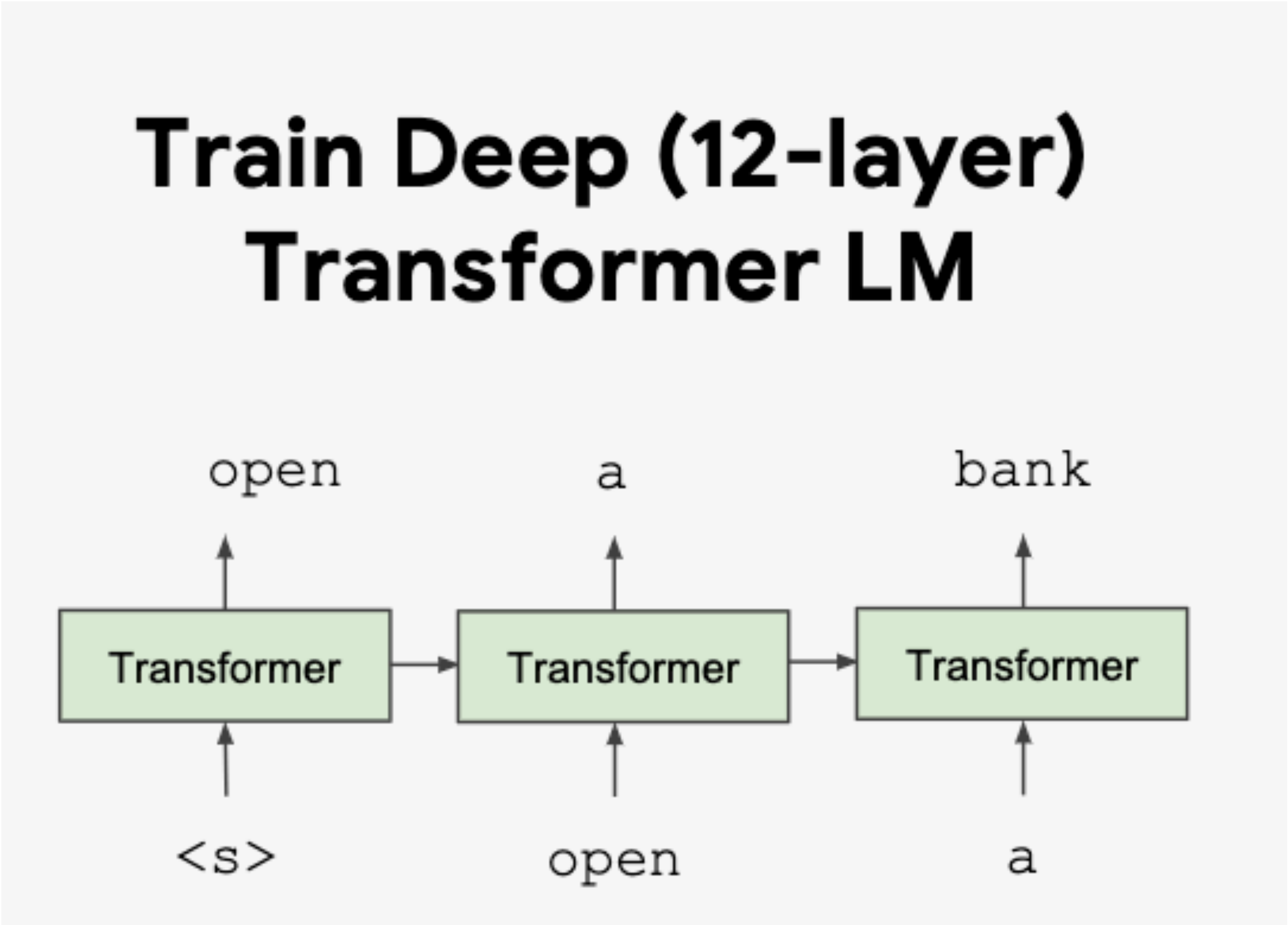


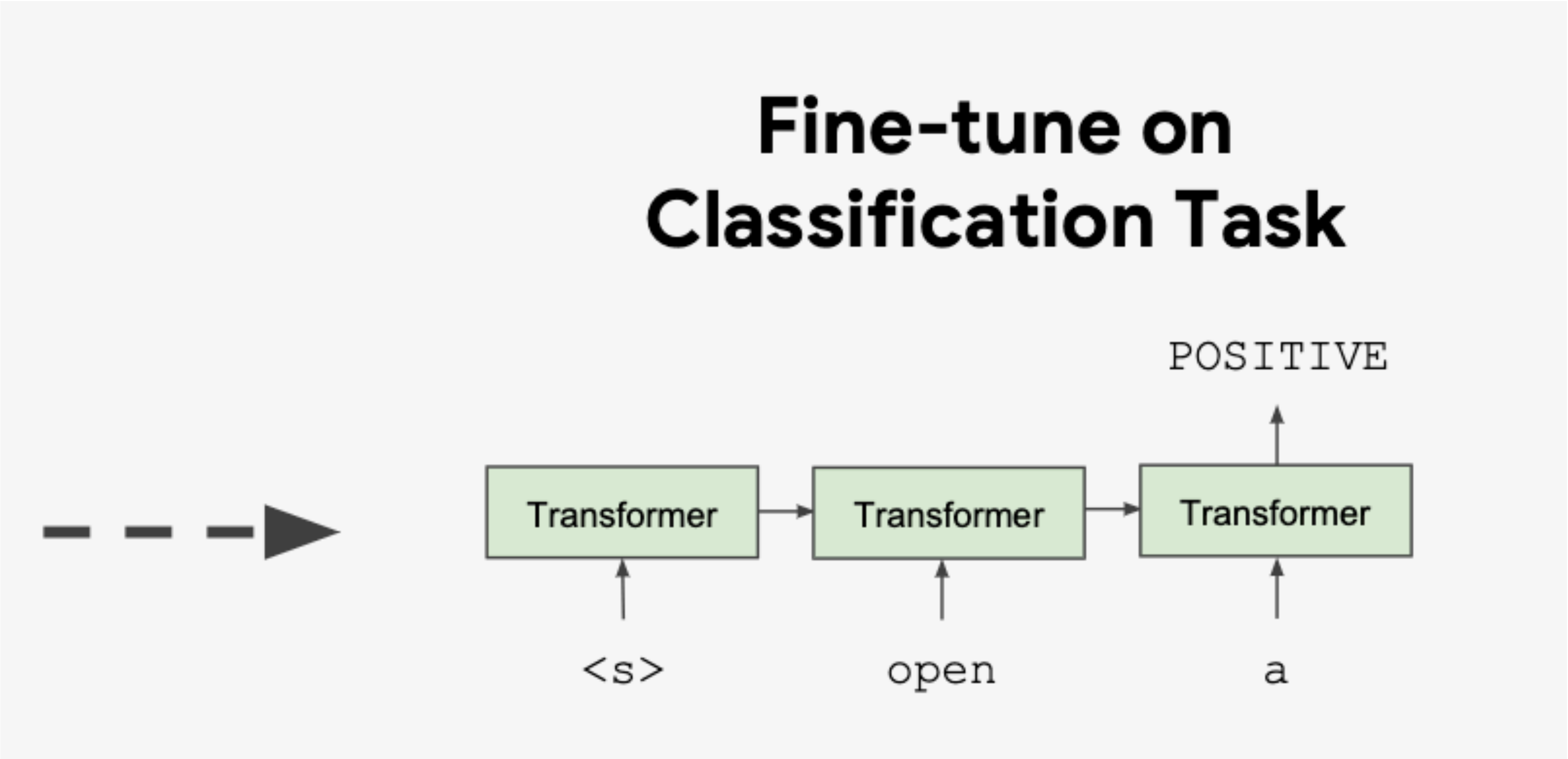


Prior work: OpenAI GPT

OpenAI GPT (Radford et al., 2018; released in 2018/6)

- Train one unidirectional LM (left-to-right) based on a deep **Transformer decoder**
- **Fine-tuning** approach: all pre-trained parameters are re-used & updated on downstream tasks
- Trained on 512-token segments on BooksCorpus — much **longer** context!





BERT: key contributions

- It is a **fine-tuning approach** based on a deep **Transformer encoder**
- The key: learn representations based on **bidirectional context**

Why? Because both left and right contexts are important
to understand the meaning of words.

Example #1: we went to the river bank.

Example #2: I need to go to bank to make a deposit.

- **Pre-training objectives**: masked language modeling + next sentence prediction
- State-of-the-art performance on a large set of **sentence-level** and **token-level** tasks

Masked Language Modeling (MLM)

- Q: Why we can’t do language modeling with bidirectional models?

- Solution: Mask out k% of the input words, and then predict the masked words

store gallon

the man went to [MASK] to buy a [MASK] of milk

MLM: masking rate and strategy

- **Q: What is the value of *k*?**
  - They always use k = 15%.
  - Too little masking: computationally expensive
  - Too much masking: not enough context
  - See (Wettig et al., 2022) for more discussion of masking rates

- **Q: How are masked tokens selected?**
  - 15% tokens are uniformly sampled
  - Is it optimal? See span masking (Joshi et al., 2020) and PMI masking (Levine et al., 2021)

Example: He [MASK] from Kuala [MASK] , Malaysia.

Note: We will see that span masking used in T5 models soon

MLM: 80-10-10 corruption

For the 15% predicted words,

- 80% of the time, they replace it with [MASK] token

went to the store $\longrightarrow$ went to the [MASK]

- 10% of the time, they replace it with a random word in the vocabulary

went to the store $\longrightarrow$ went to the running

- 10% of the time, they keep it unchanged

went to the store $\longrightarrow$ went to the store

Why? Because [MASK] tokens are never seen during fine-tuning

(See Table 8 for an ablation study)

Next Sentence Prediction (NSP)

- Motivation: many NLP downstream tasks require understanding the relationship between two sentences (natural language inference, paraphrase detection, QA)
- NSP is designed to reduce the gap between pre-training and fine-tuning

[CLS]: a special token always at the beginning

[SEP]: a special token used to separate two segments

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]

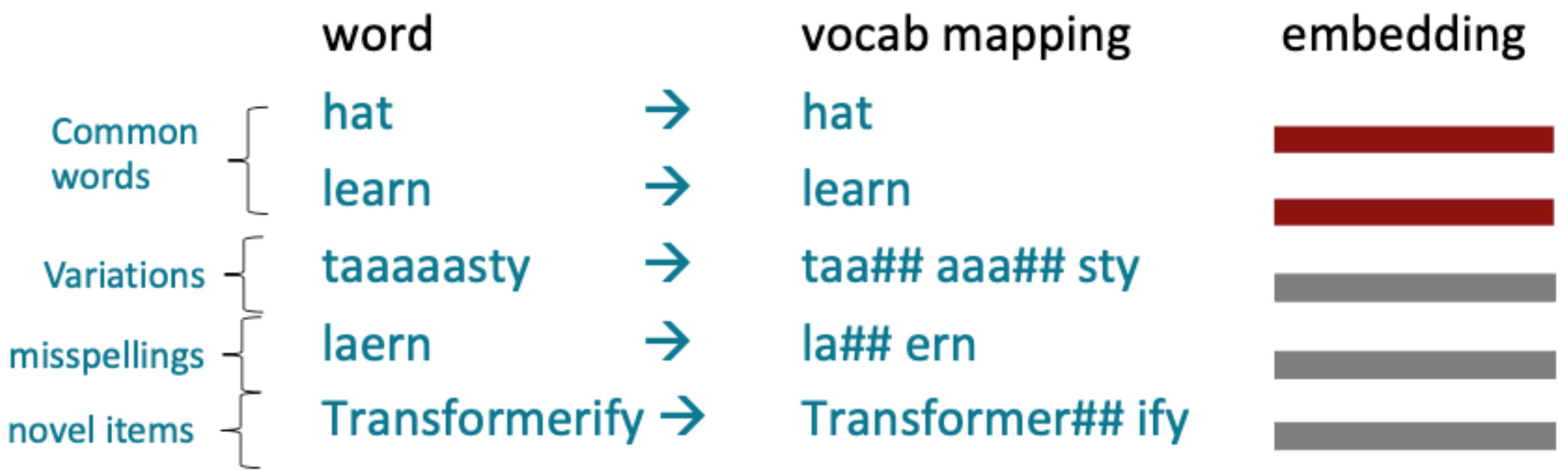
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

They sample two contiguous segments for 50% of the time and another random segment from the corpus for 50% of the time

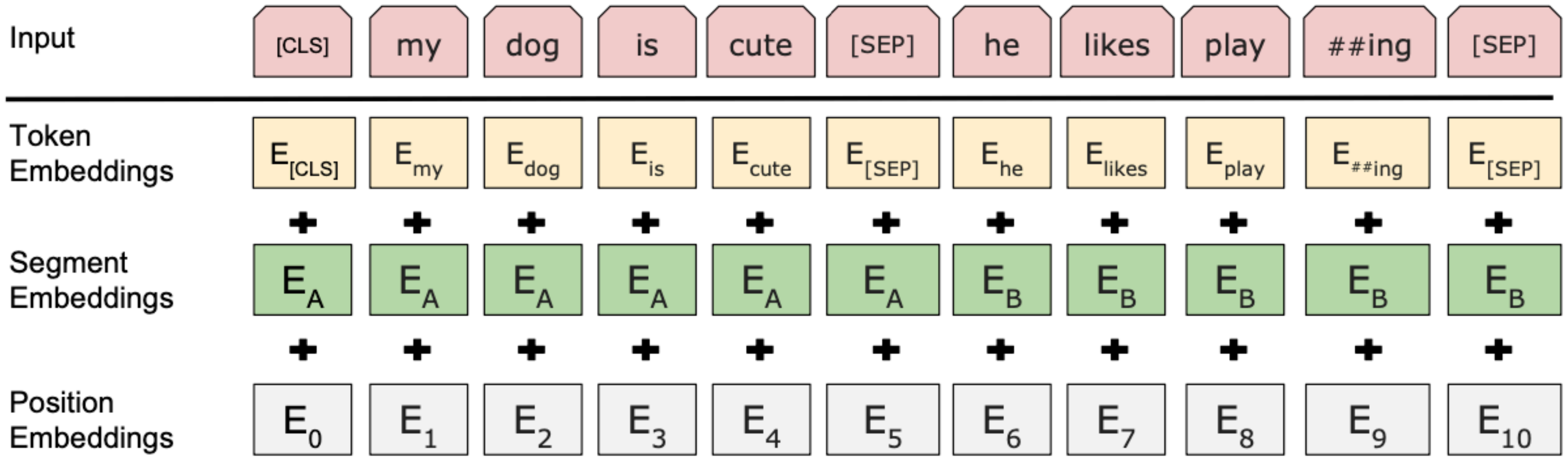
BERT pre-training: putting together

- Vocabulary size: 30,000 workpieces (common sub-word units) (Wu et al., 2016)



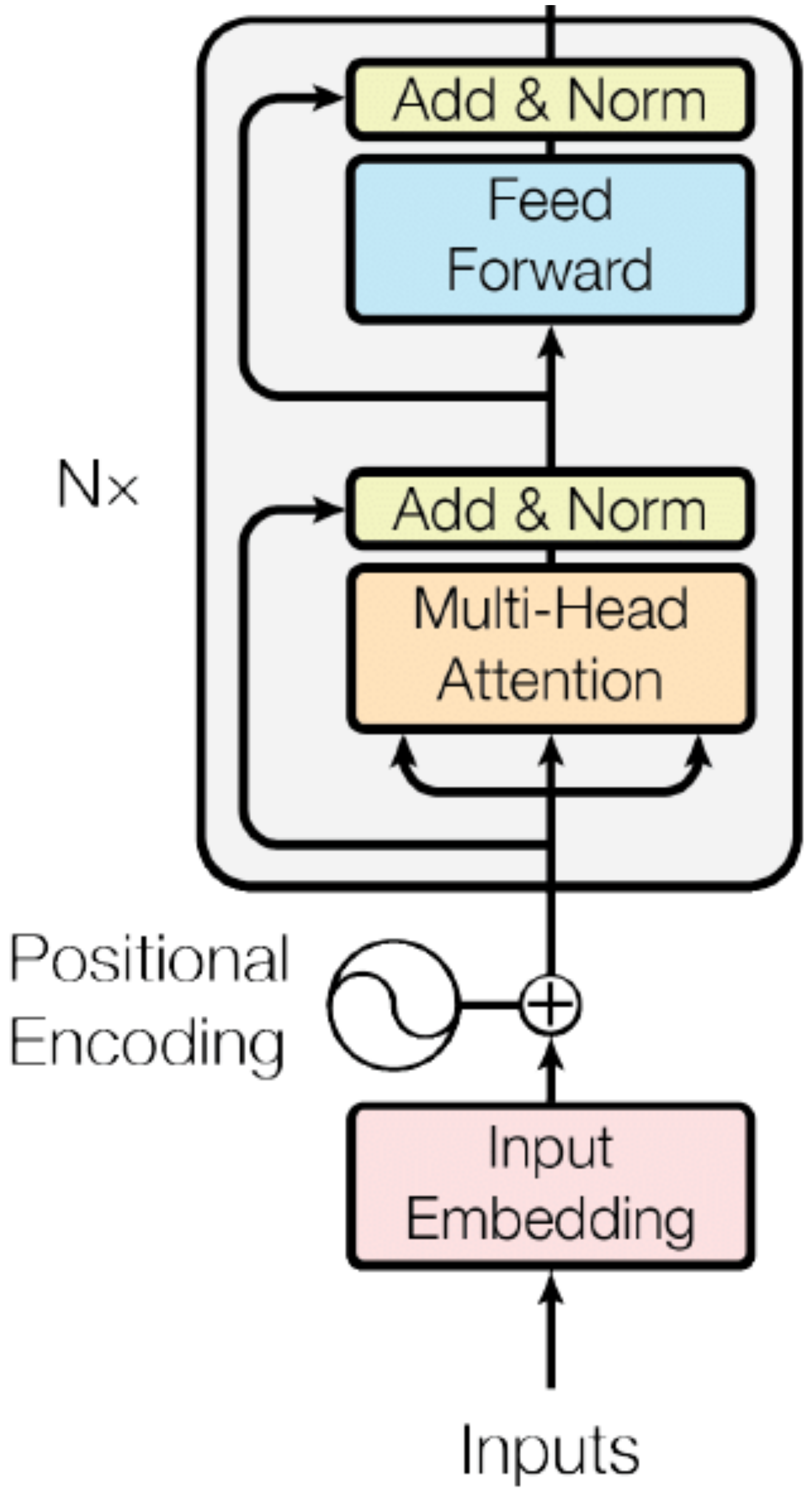
(Image: Stanford CS224N)

- Input embeddings:



Separate two segments

BERT pre-training: putting together



- BERT-base: 12 layers, 768 hidden size, 12 attention heads, 110M parameters

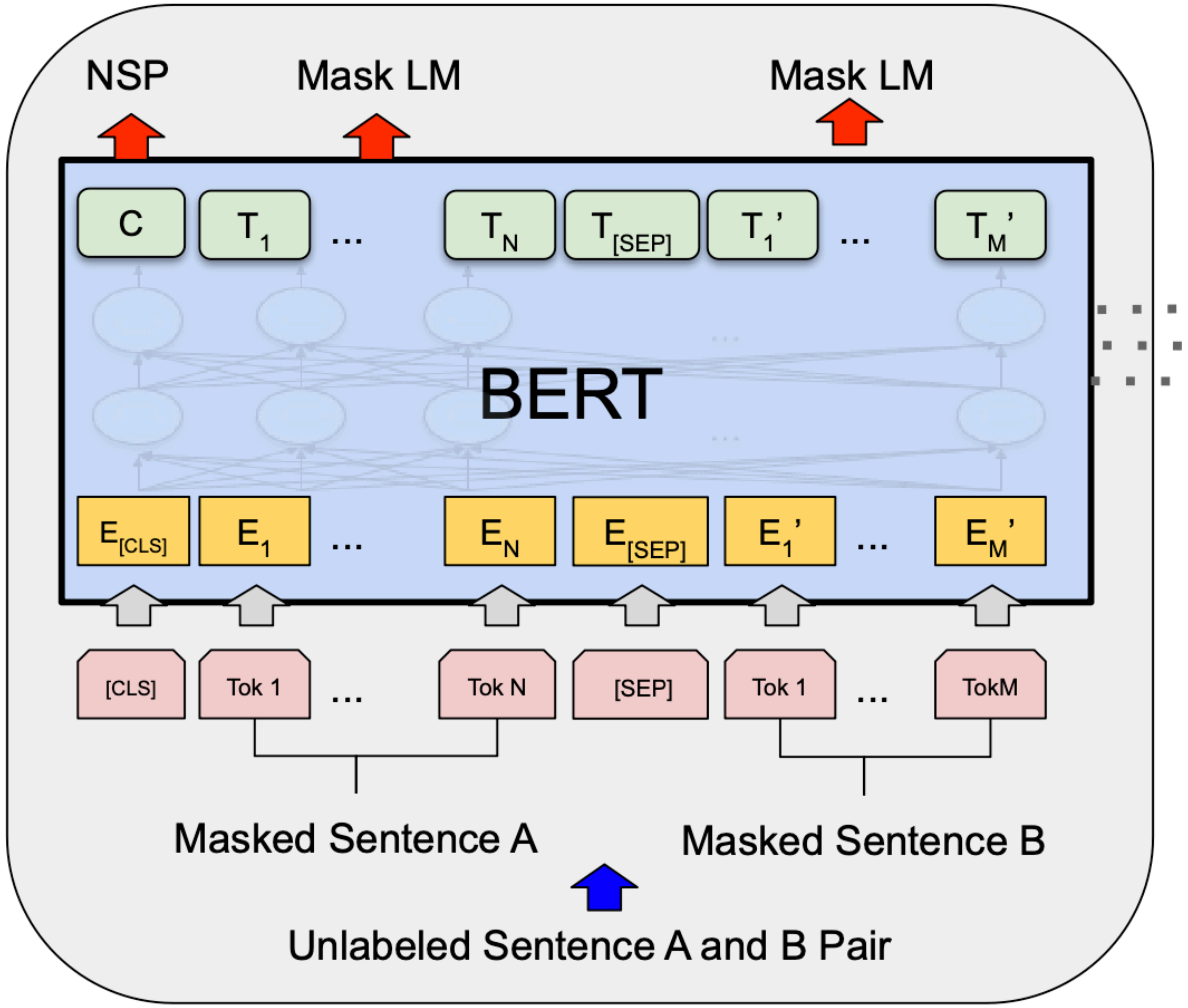
Same as OpenAI GPT

- BERT-large: 24 layers, 1024 hidden size, 16 attention heads, 340M parameters

OpenAI GPT was trained on BooksCorpus only!

- Training corpus: Wikipedia (2.5B) + BooksCorpus (0.8B)
- Max sequence size: 512 word pieces (roughly 256 and 256 for two non-contiguous sequences)
- Trained for 1M steps, batch size 128k

BERT pre-training: putting together



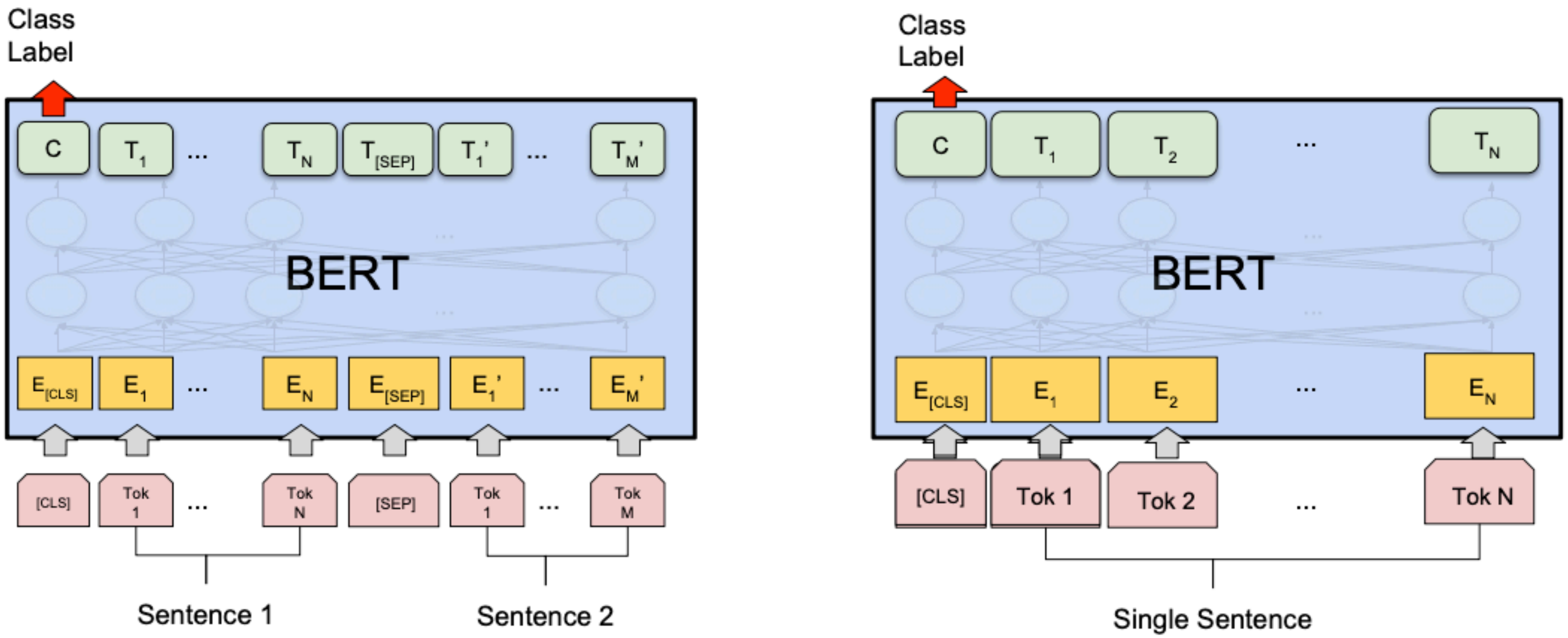
Pre-training

- MLM and NSP are trained together
- [CLS] is pre-trained for NSP
- Other token representations are trained for MLM

Fine-tuning BERT

“Pretrain once, finetune many times.”

sentence-level tasks



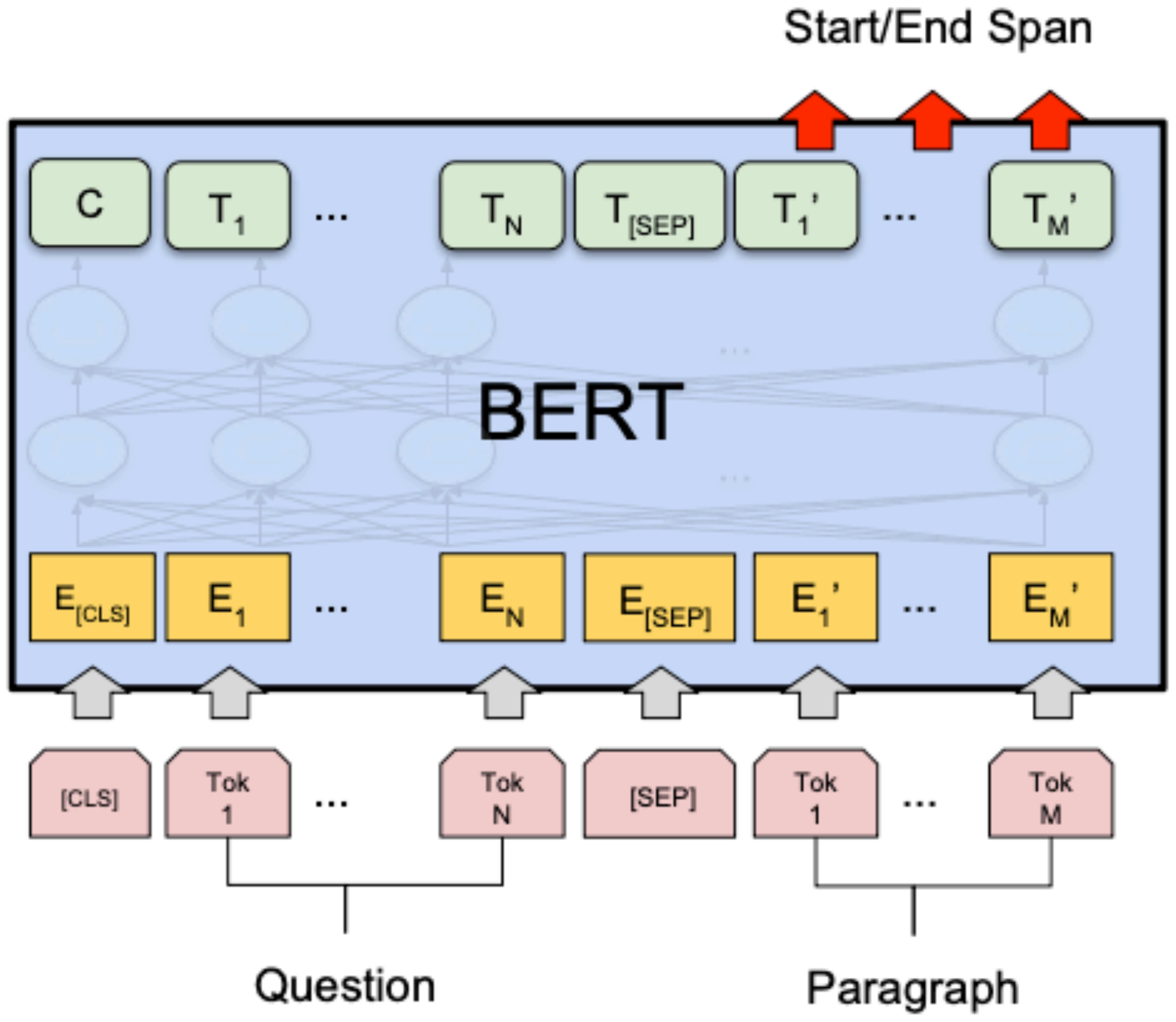
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

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

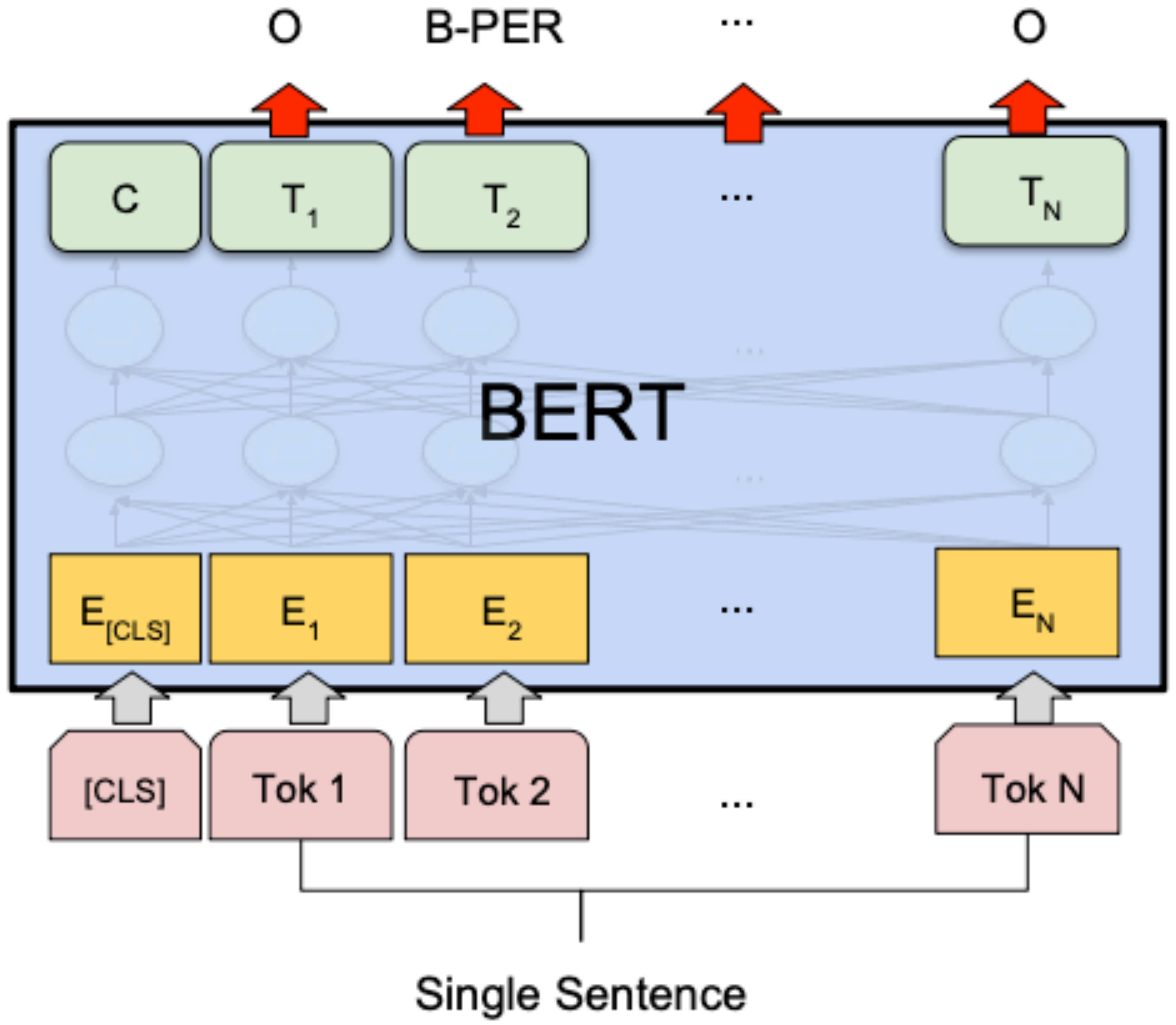
Fine-tuning BERT

"Pretrain once, finetune many times."

token-level tasks



(c) Question Answering Tasks: SQuAD v1.1



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

Sentence-level tasks

- Sentence pair classification tasks:

MNLI

Premise: A soccer game with multiple males playing.

Hypothesis: Some men are playing a sport.

{entailment, contradiction, neutral}

QQP

Q1: Where can I learn to invest in stocks?

Q2: How can I learn more about stocks?

{duplicate, not duplicate}

- Single sentence classification tasks:

SST2

rich veins of funny stuff in this movie

{positive, negative}

GLUE (Wang et al., 2019): 6 sentence pair and 2 single-sentence tasks

Token-level tasks

- Extractive question answering e.g., SQuAD (Rajpurkar et al., 2016)

SQuAD

Question: The New York Giants and the New York Jets play at which stadium in NYC ?

Context: The city is represented in the National Football League by the New York Giants and the New York Jets , although both teams play their home games at MetLife Stadium in nearby East Rutherford , New Jersey , which hosted Super Bowl XLVIII in 2014 .

(Training example 29,883)

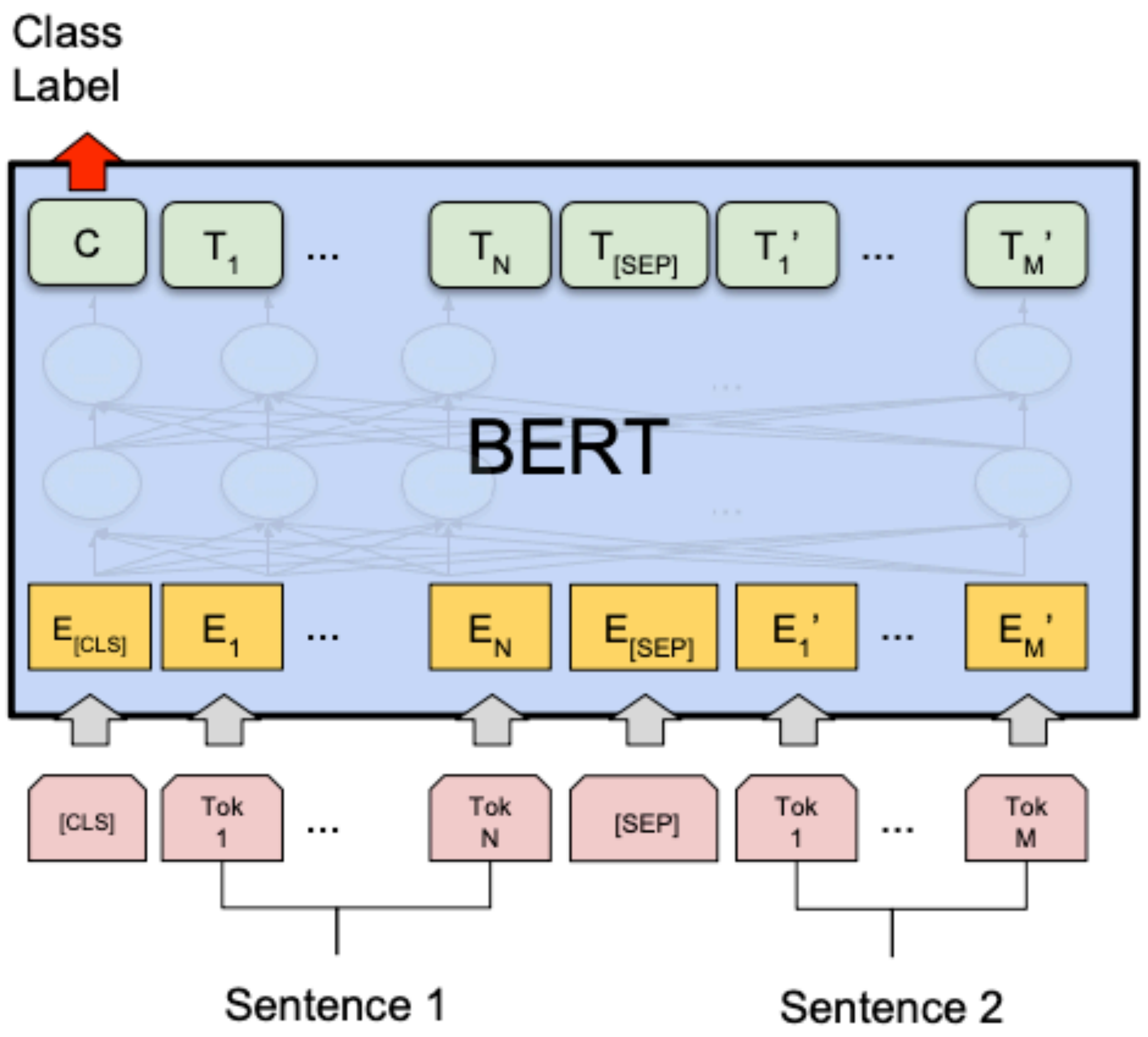
MetLife Stadium

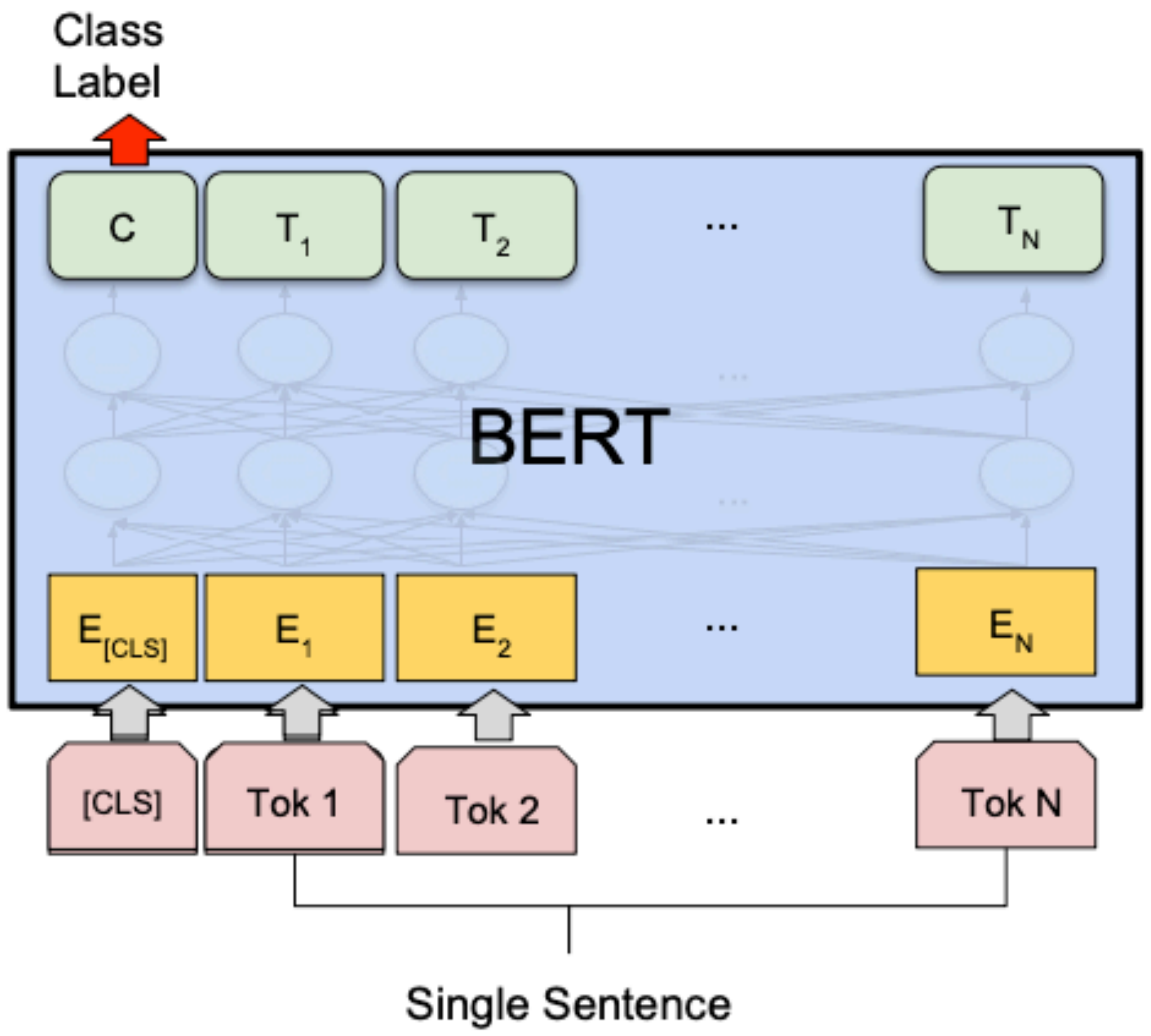
- Named entity recognition (Tjong Kim Sang and De Meulder, 2003)

CoNLL 2003 NER

John	Smith	lives	in	New	York
B-PER	I-PER	O	O	B-LOC	I-LOC

Fine-tuning BERT

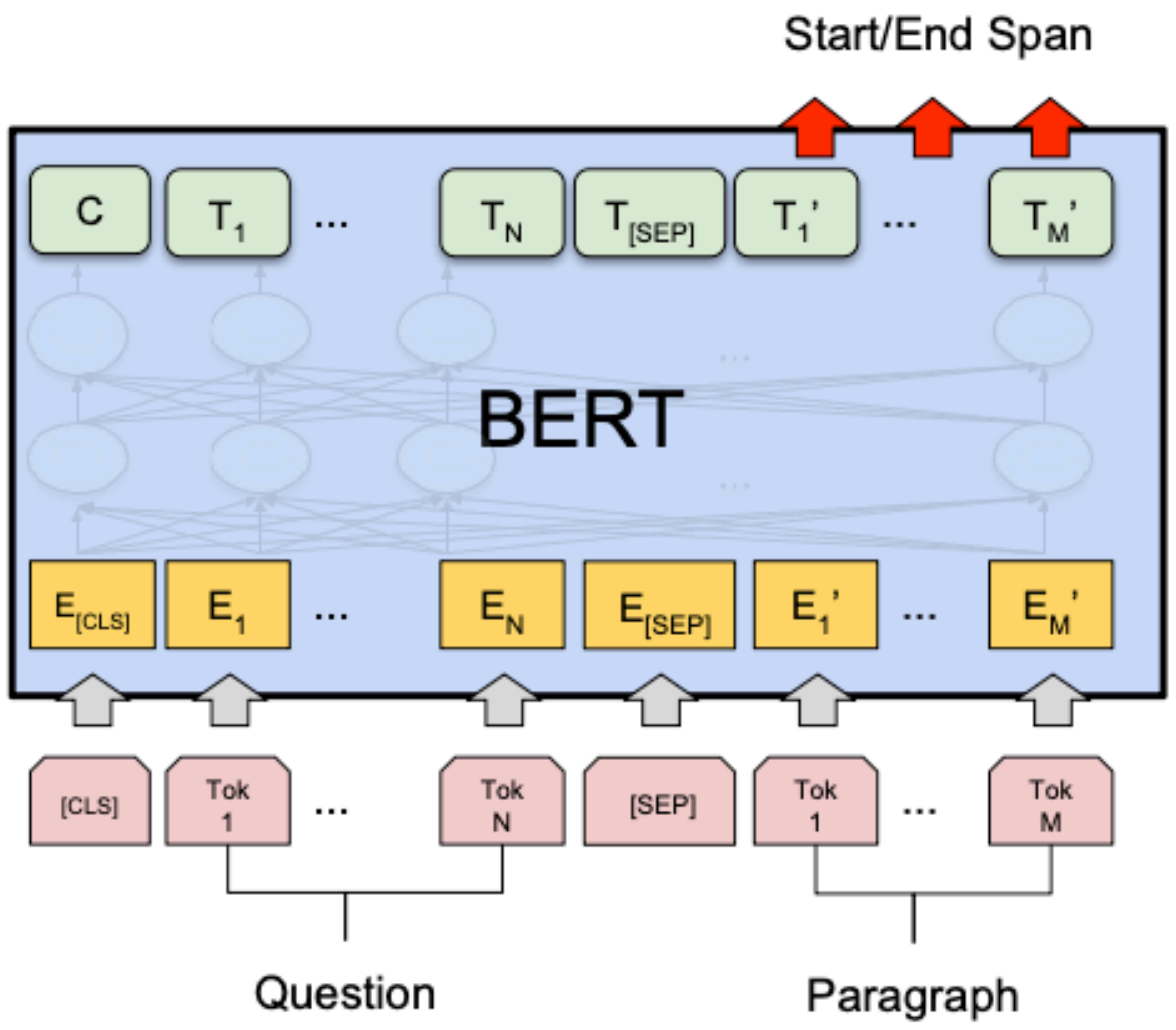


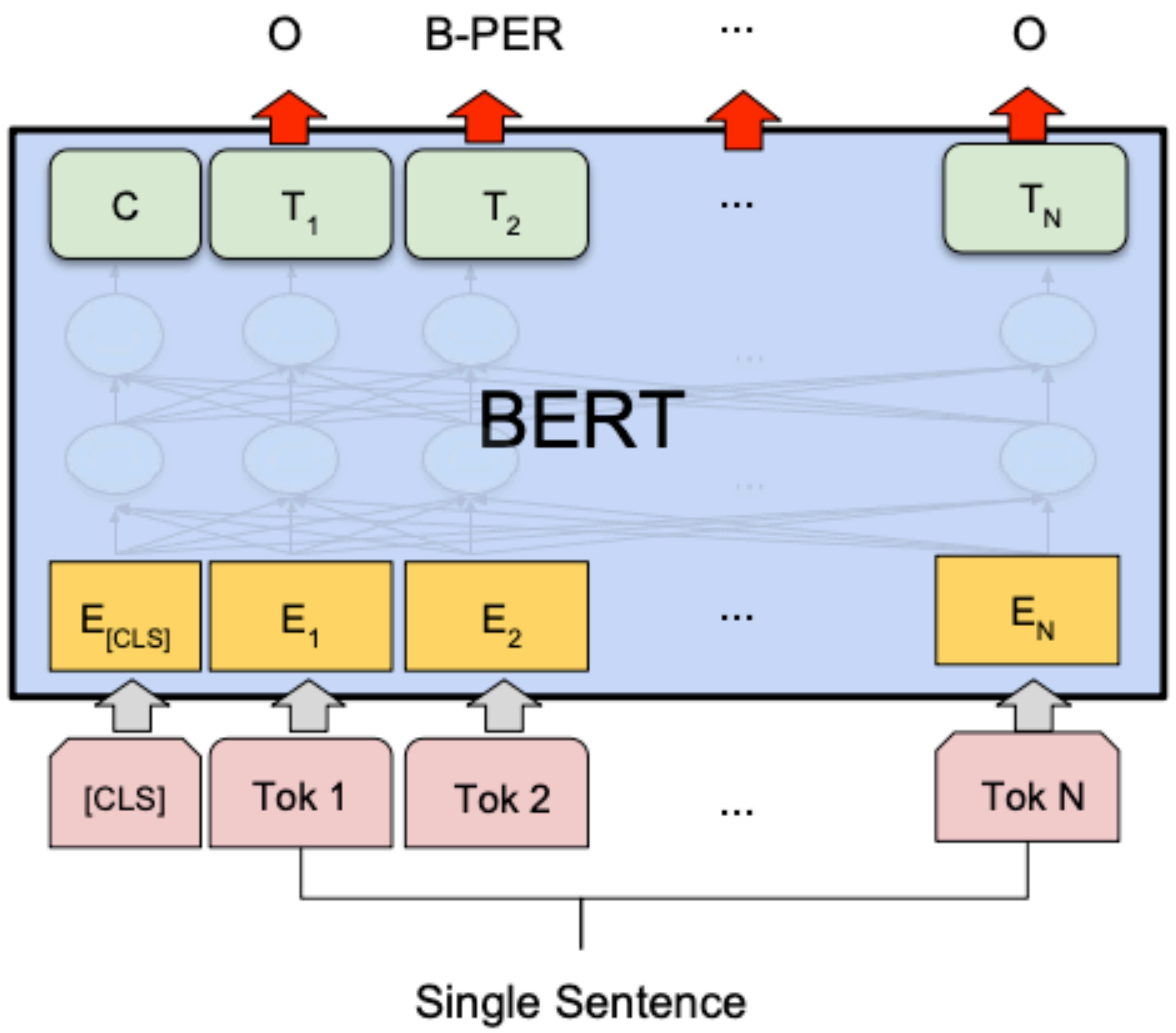


- For sentence pair tasks, use [SEP] to separate the two segments with segment embeddings
- Add a linear classifier on top of [CLS] representation and introduce $C \times h$ new parameters

C: # of classes, h: hidden size

Fine-tuning BERT





- For token-level prediction tasks, add linear classifier on top of hidden representations

Q: How many new parameters?

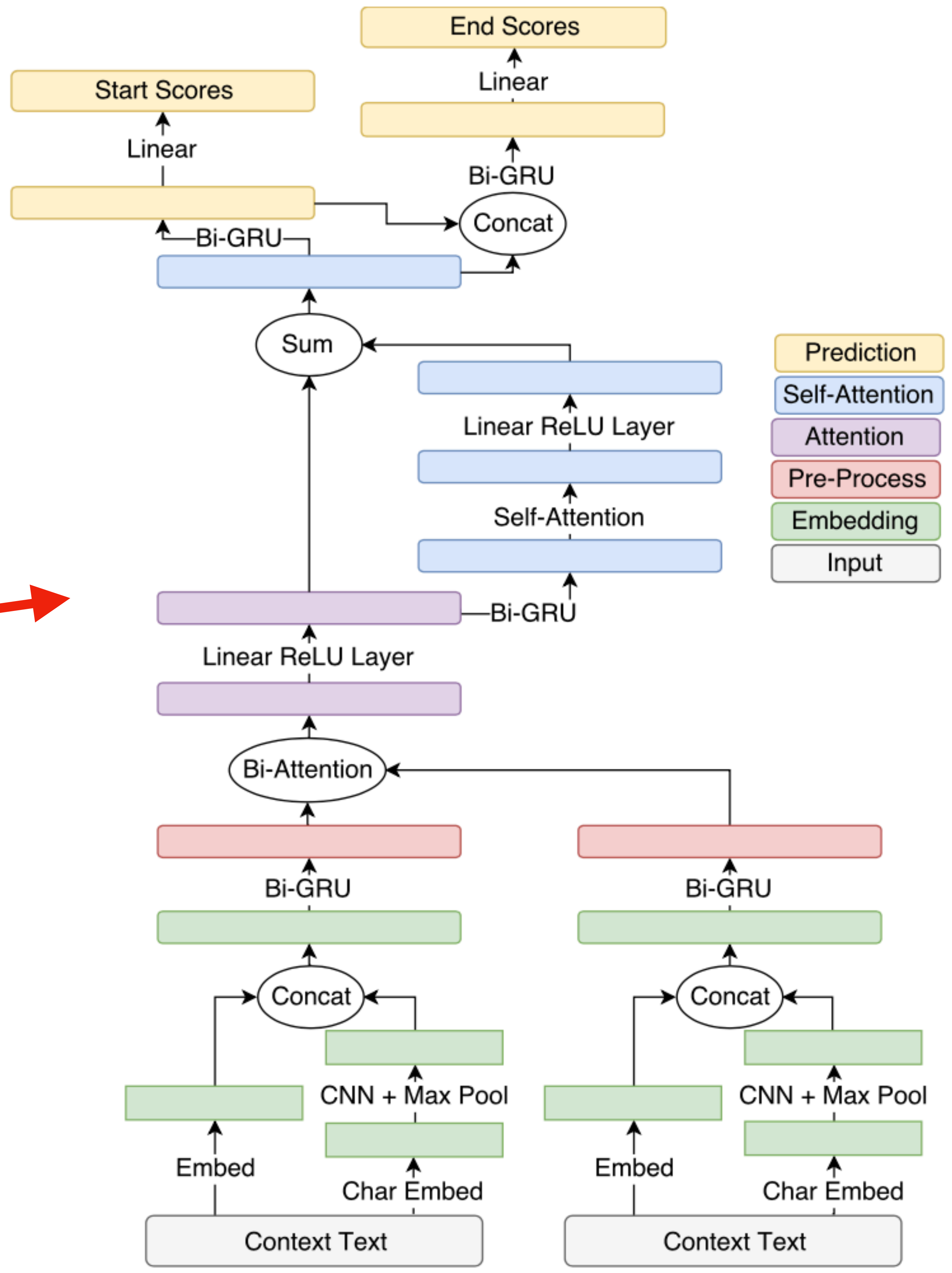
Experimental results: GLUE

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	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**

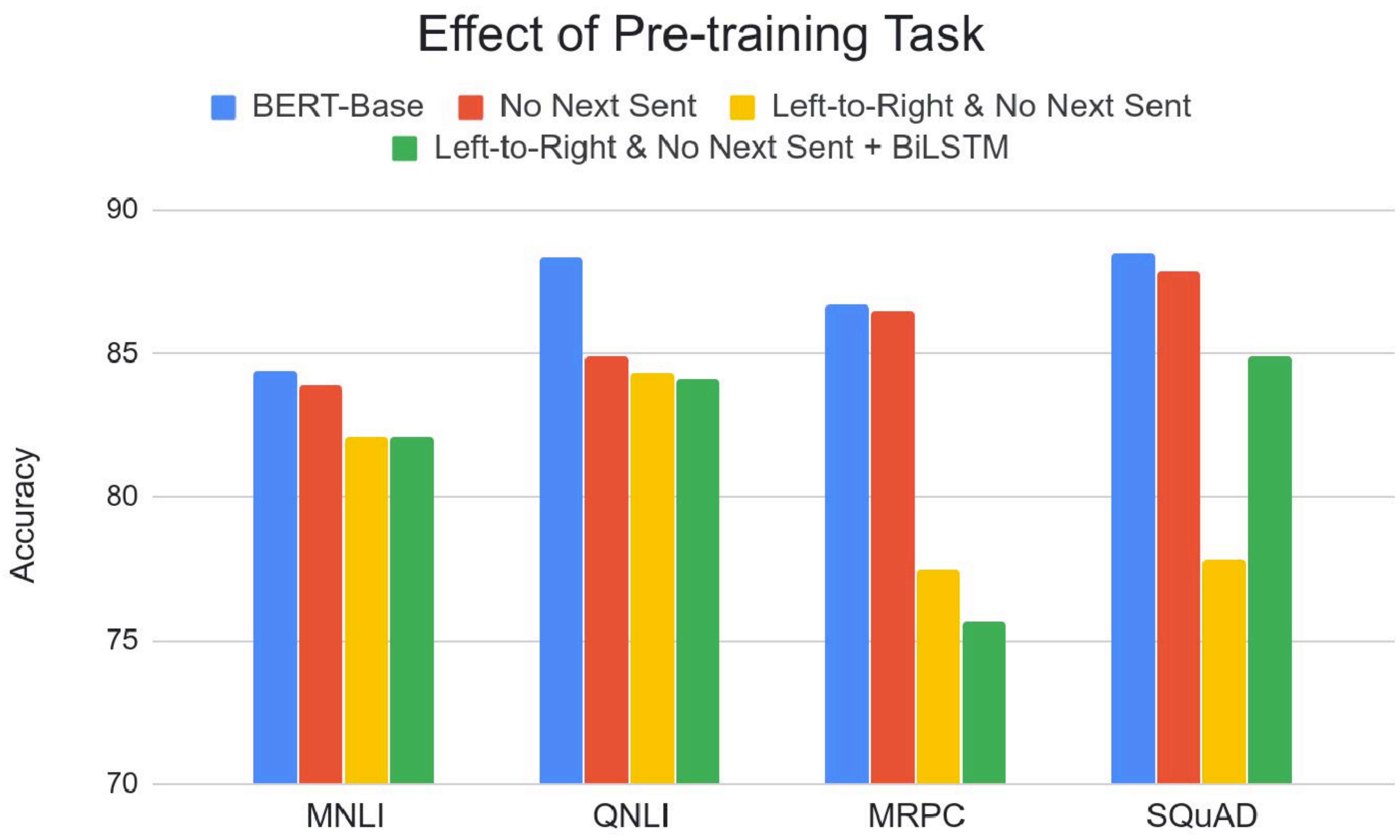
See Appendix A.4 for detailed differences between BERT and OpenAI GPT

Experimental results: SQuAD

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**



Ablation study: pre-training tasks



- MLM >> left-to-right LMs
- NSP improves on some tasks
- Note: later work (Joshi et al., 2020; Liu et al., 2019) argued that NSP is not useful

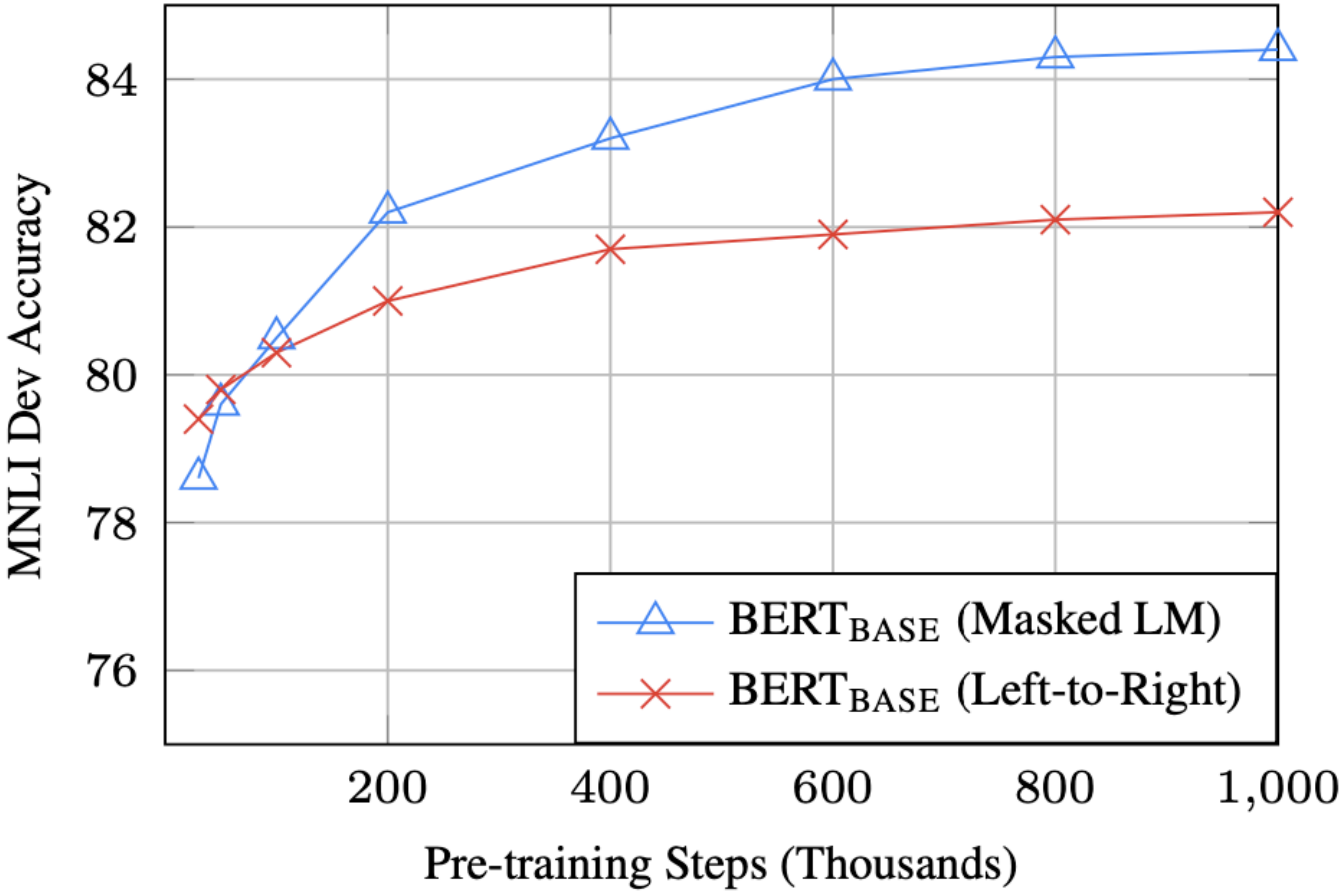
Ablation study: model sizes

\# layers hidden size # of heads

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

The bigger, the better!

Ablation study: training efficiency



MLM takes slightly longer to converge because it only predicts 15% of tokens

Conclusions (in early 2019)

From Jacob Devlin's talk in 2019/1:

- Is modeling "solved" in NLP? I.e., is there a reason to come up with novel model architectures?
  - But that's the most fun part of NLP research :(

- Personal belief: Near-term improvements in NLP will be mostly about making clever use of "free" data.
  - Unsupervised vs. semi-supervised vs. synthetic supervised is somewhat arbitrary.
  - "Data I can get a lot of without paying anyone" vs. "Data I have to pay people to create" is more pragmatic distinction.

Conclusions (in early 2019)

From Jacob Devlin’s talk in 2019/1:

- Empirical results from BERT are great, but biggest impact on the field is:
- With pre-training, bigger == better, without clear limits (so far).

What happened after BERT?

Lots of people are trying to understand what BERT has learned and how it works

A Primer in BERTology: What We Know About How BERT Works

Anna Rogers
Center for Social Data Science
University of Copenhagen
arogers@sodas.ku.dk

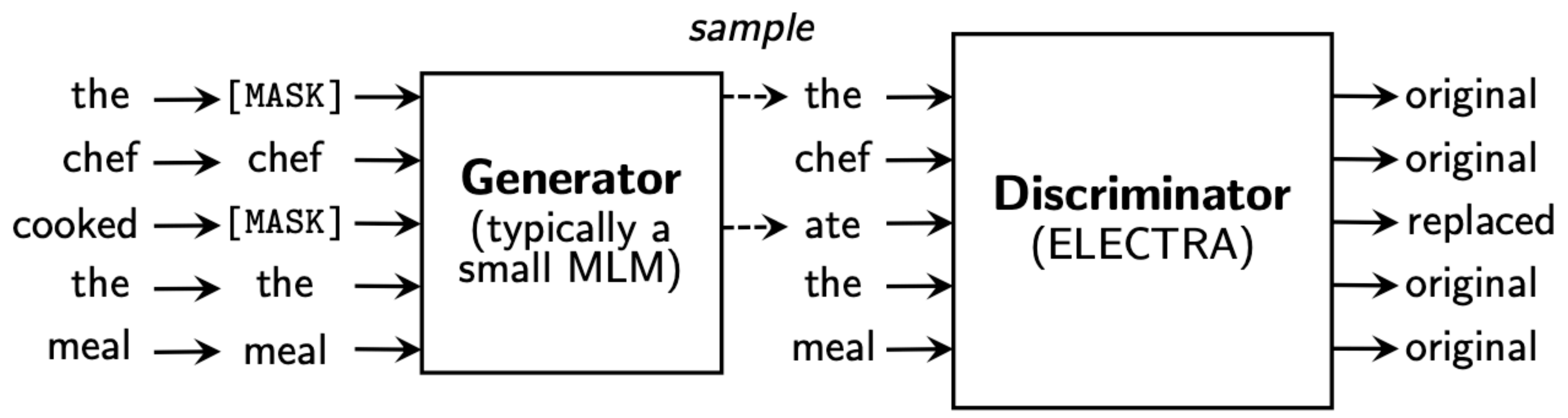
Olga Kovaleva
Dept. of Computer Science
University of Massachusetts Lowell
okovalev@cs.uml.edu

Anna Rumshisky
Dept. of Computer Science
University of Massachusetts Lowell
arum@cs.uml.edu

- Syntactic knowledge, semantic knowledge, world knowledge…
- How to mask, what to mask, where to mask, alternatives to masking..

What happened after BERT?

- RoBERTa (Liu et al., 2019)
  - Trained on 10x data & longer, no NSP
  - Much stronger performance than BERT (e.g., 94.6 vs 90.9 on SQuAD)
  - Still one of the most popular models to date
- ALBERT (Lan et al., 2020)
  - Increasing model sizes by sharing model parameters across layers
  - Less storage, much stronger performance but runs slower..
- ELECTRA (Clark et al., 2020)
  - It provides a more efficient training method by predicting 100% of tokens instead of 15% of tokens



What happened after BERT?

- Models that handle long contexts ($\gg$ 512 tokens)
  - Longformer, Big Bird, …
- Multilingual BERT
  - Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary
- BERT extended to different domains
  - SciBERT, BioBERT, FinBERT, ClinicalBERT, …
- Making BERT smaller to use
  - DistillBERT, TinyBERT, …

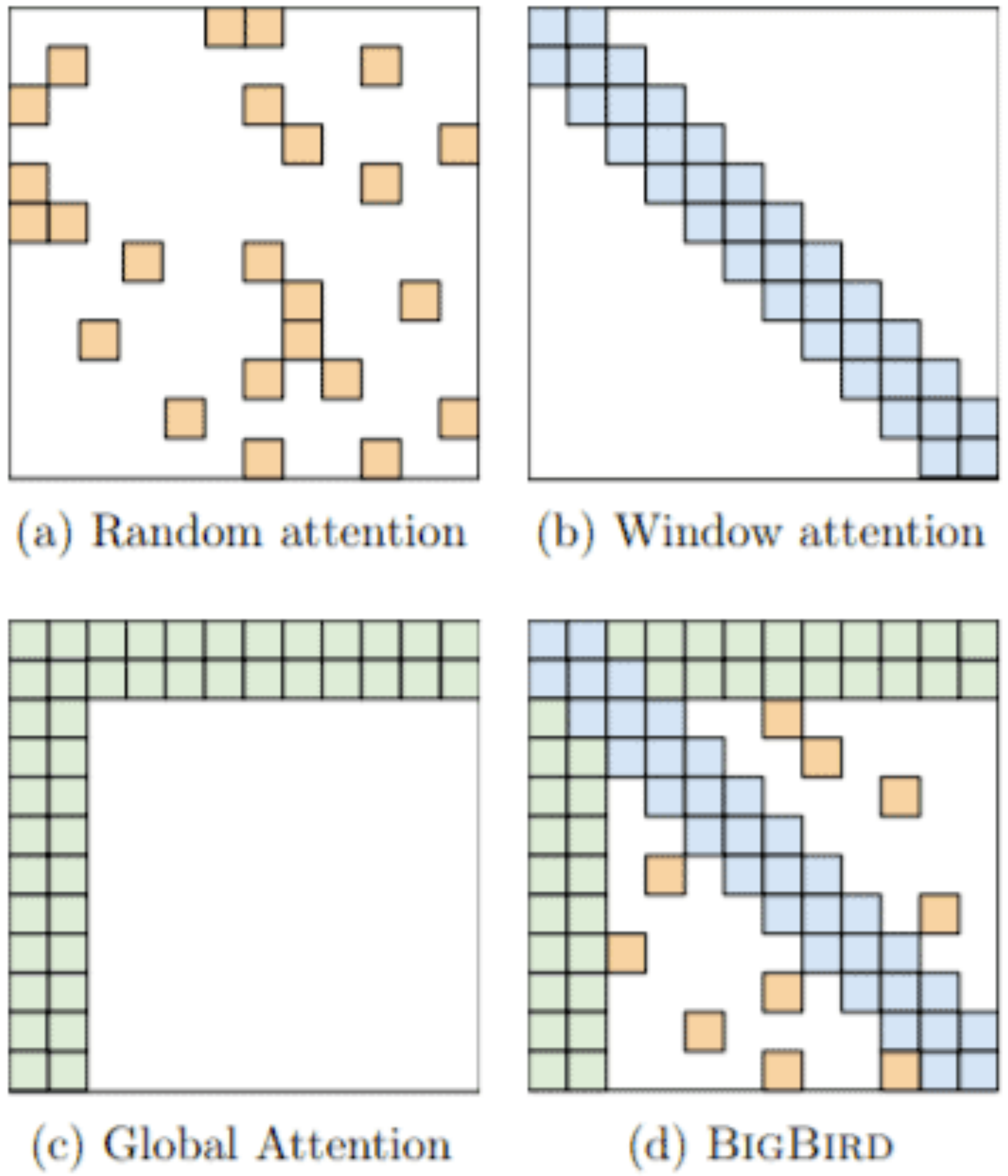


Image from the original paper

Text generation using BERT

BERT has a Mouth, and It Must Speak:
BERT as a Markov Random Field Language Model

Alex Wang
New York University
alexwang@nyu.edu

Kyunghyun Cho
New York University
Facebook AI Research
CIFAR Azrieli Global Scholar
kyunghyun.cho@nyu.edu

Mask-Predict: Parallel Decoding of
Conditional Masked Language Models

Marjan Ghazvininejad* **Omer Levy*** **Yinhan Liu*** **Luke Zettlemoyer**
Facebook AI Research
Seattle, WA

Exposing the Implicit Energy Networks behind Masked Language Models via Metropolis--Hastings

Kartik Goyal, Chris Dyer, Taylor Berg-Kirkpatrick

src	Der Abzug der franzsischen Kampftruppen wurde am 20. November abgeschlossen .
$t = 0$	The departure of the French combat completed completed on 20 November .
$t = 1$	The departure of French combat troops was completed on 20 November .
$t = 2$	The withdrawal of French combat troops was completed on November 20th .

Leveraging Pre-trained Checkpoints for Sequence Generation Tasks

Sascha Rothe, Shashi Narayan, Aliaksei Severyn

Q1. Feature-based vs fine-tuning approaches

- Feature-based: task-specific architectures that uses pre-trained representations as features
- Fine-tuning: introduces minimal task-specific parameters and trains on downstream examples by simply fine-tuning all the parameters

Fine-tuning is more appealing
1) no task-specific engineering
2) re-using most pre-trained weights leads to stronger performance

Q2. BERT’s masking strategy

- 15% uniform masking - Why?
- 80-10-10 strategy - Why?

Q3: If we scale up BERT by 1000x, would it be still better than unidirectional models? Why do you think the largest models to date are always unidirectional?