BERT

Табишева Анастасия
ФКН ПМИ 171
НИС 2020

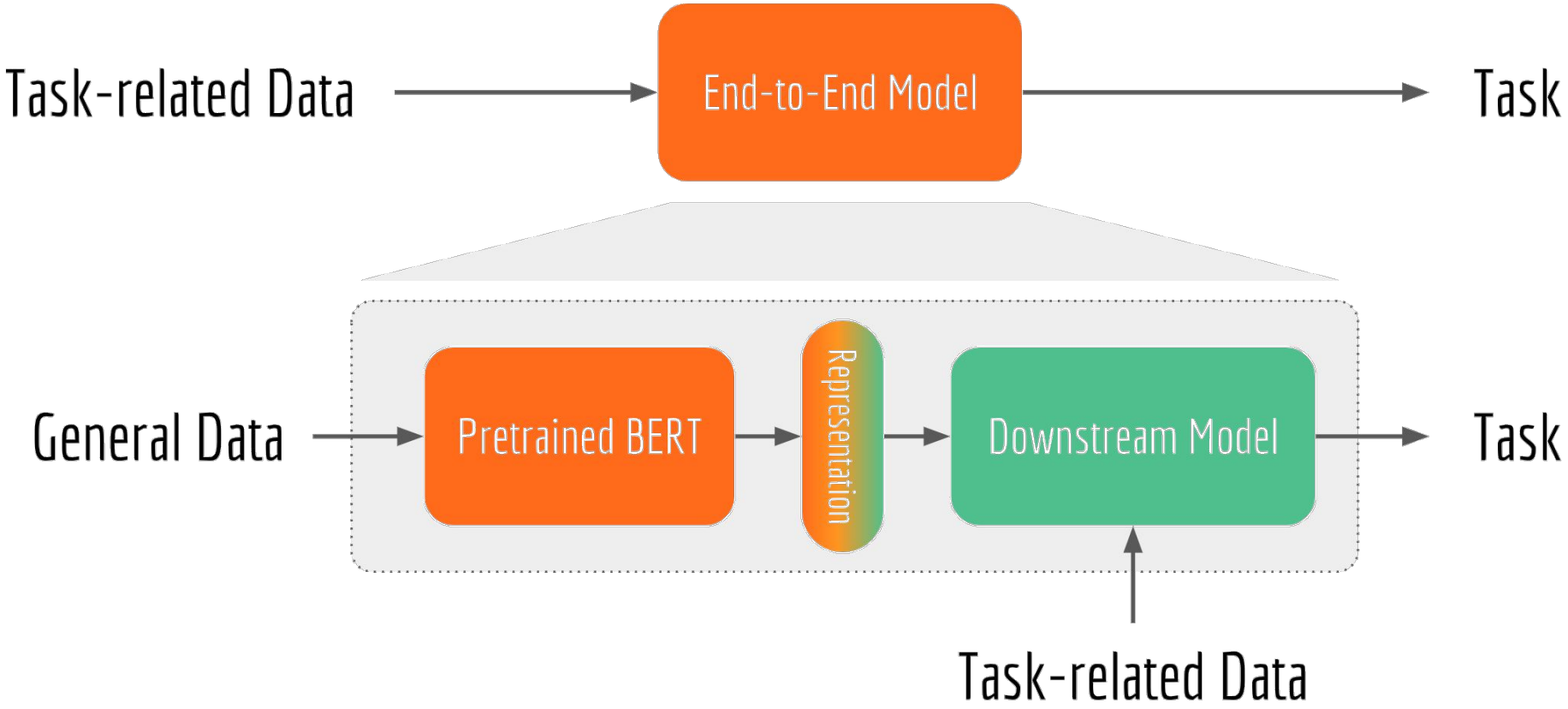
BERT: зачем?

Проблема

Обучать “с нуля” всю модель и собирать данные для конкретной задачи - долго и трудоёмко

Идея

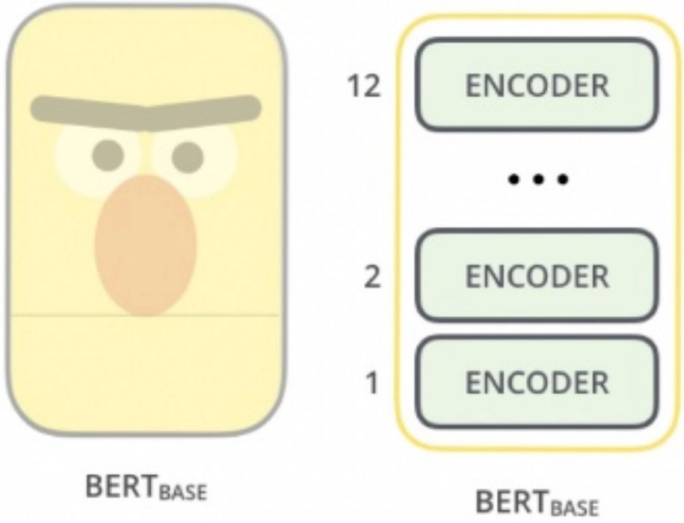
Будем использовать предобученную на огромном корпусе текстов модель и дообучать её под узкую задачу

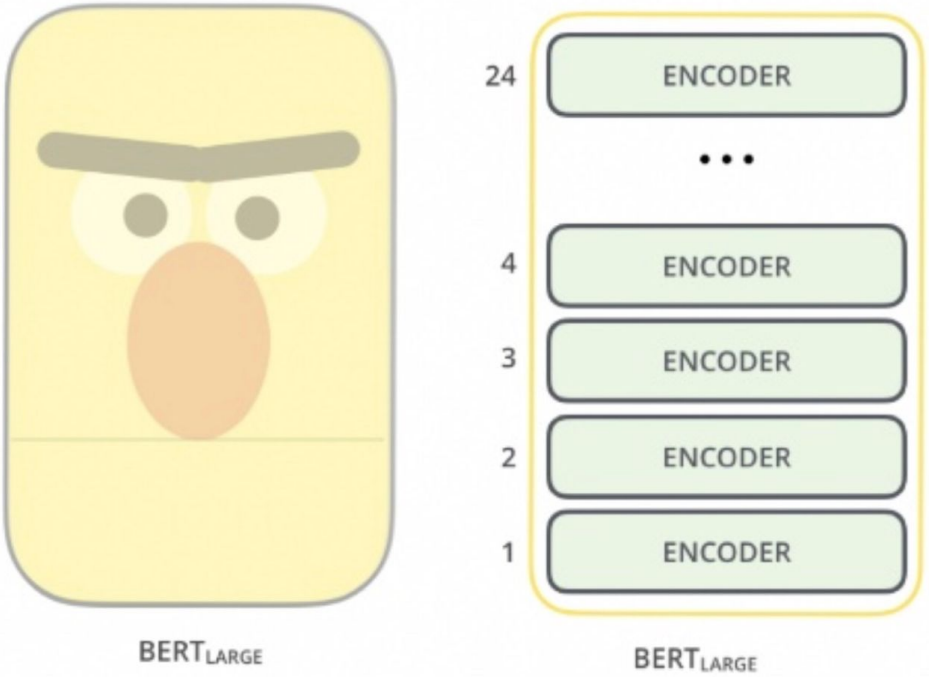

Task-related Data
End-to-End Model
Task
General Data
Pretrained BERT
Representation
Downstream Model
Task
Task-related Data

Особенности BERT:

- Двунаправленный трансформер
- Дообучение до разных задач
- Универсальный словарный запас
- Лучшие результаты для некоторых бенчмарков

Bert: Model Architecture





Equal to Open AI Transformer

- 12 Blocks (Heads)
- 768 Hidden Unit
- 12 Attention Heads
- 110M parameters

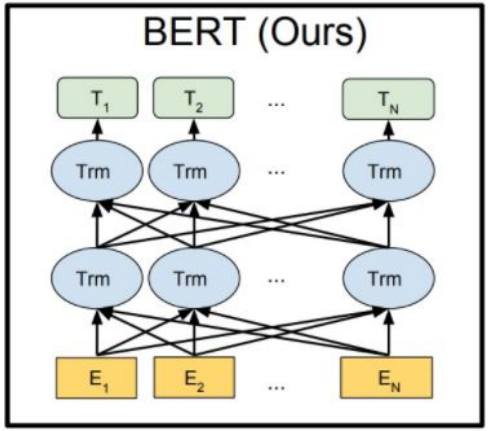
State of the Art Model

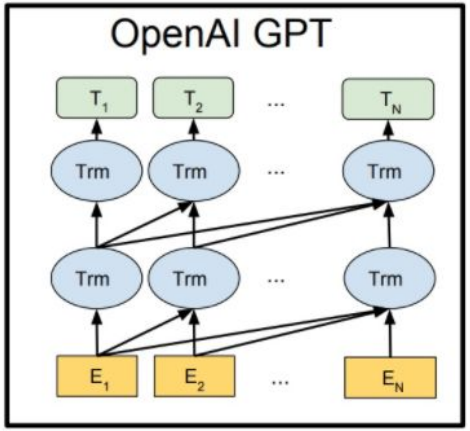
- 24 Blocks
- 1024 Hidden Unit
- 16 Attention Heads
- 340M parameters

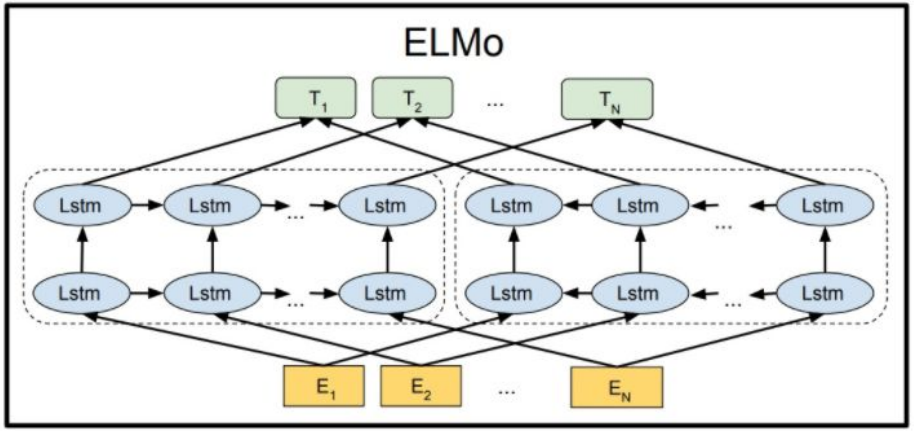
Эволюция эмбеддингов

- Эмбеддинги **word2vec** не учитывают контекст
- **OpenAI GPT** - только левый (правый) контекст
- **ELMO** - независимы обучается на левом и правом контексте
- **BERT -** одновременно учитывает весь контекст

- ELMO - две независимые LSTM
- GPT - однонаправленный трансформер
- BERT - двунаправленный трансформер

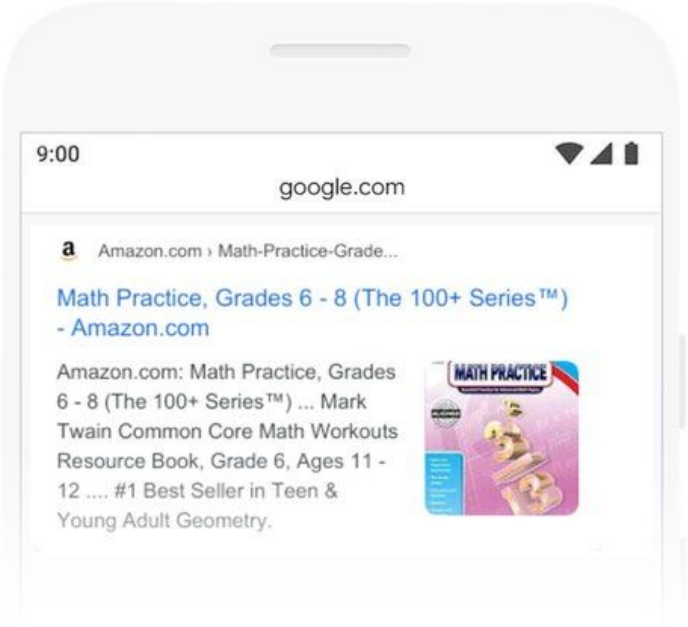




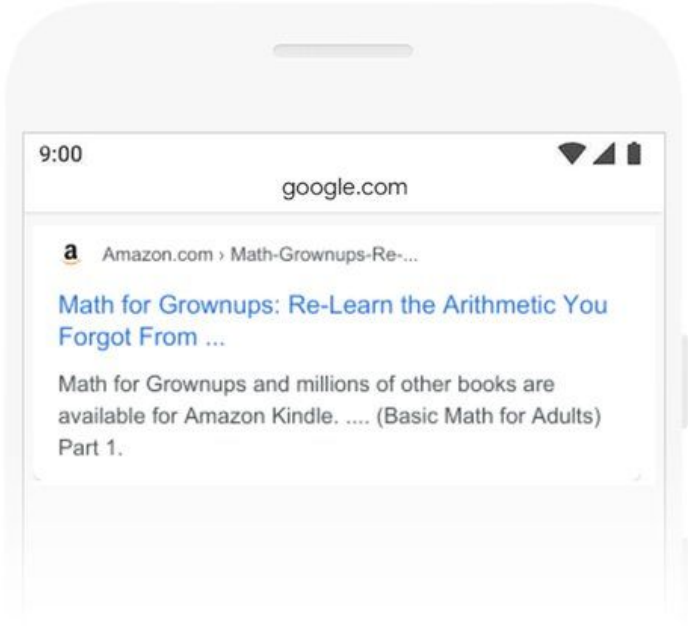


math practice books for adults

BEFORE

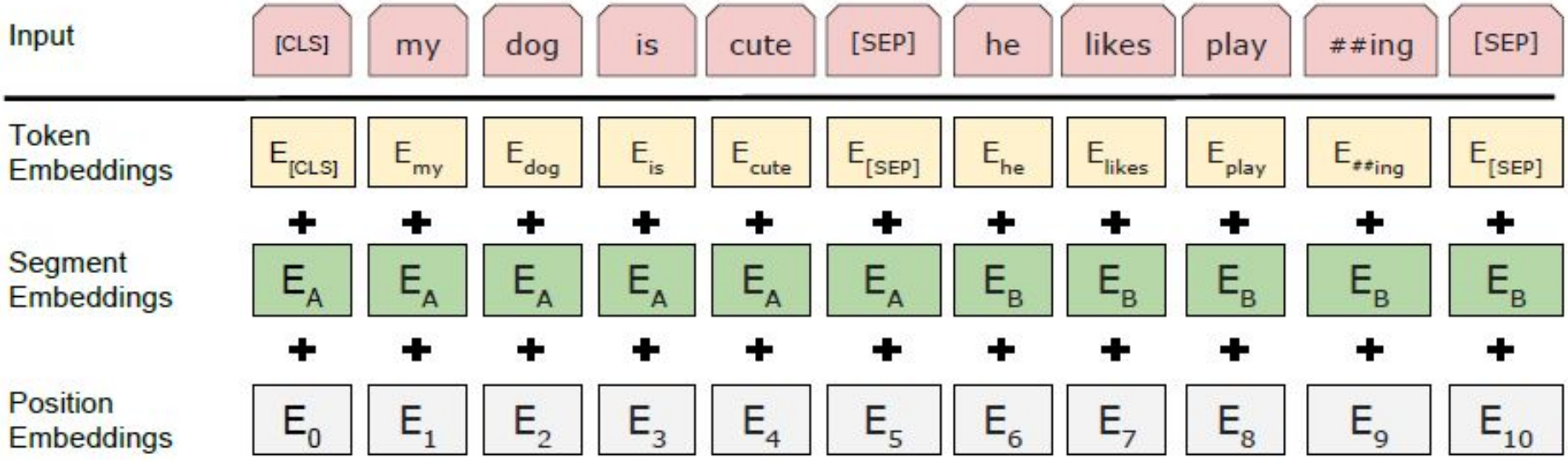


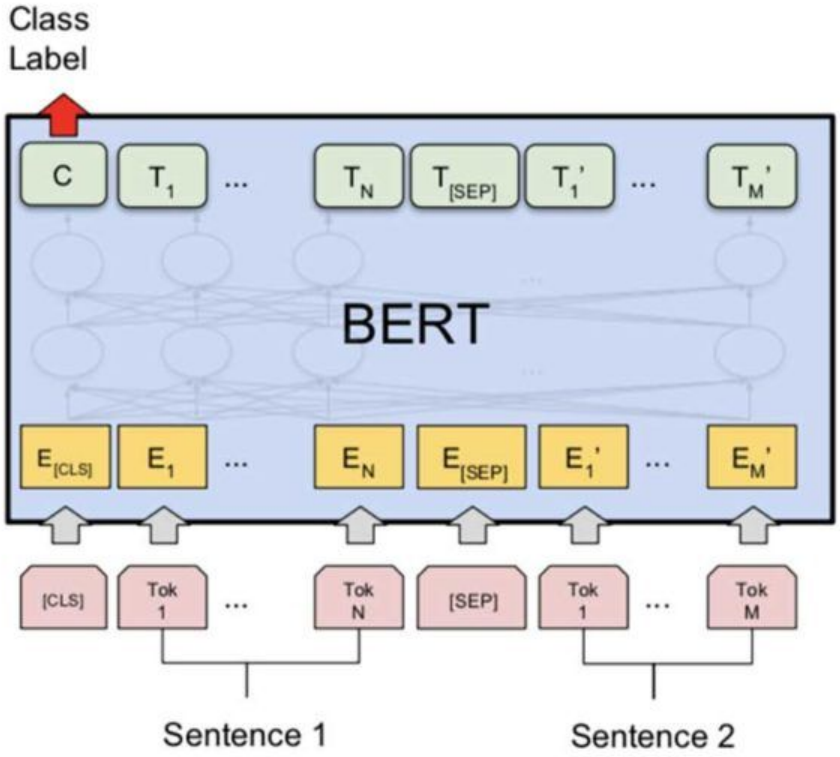
AFTER

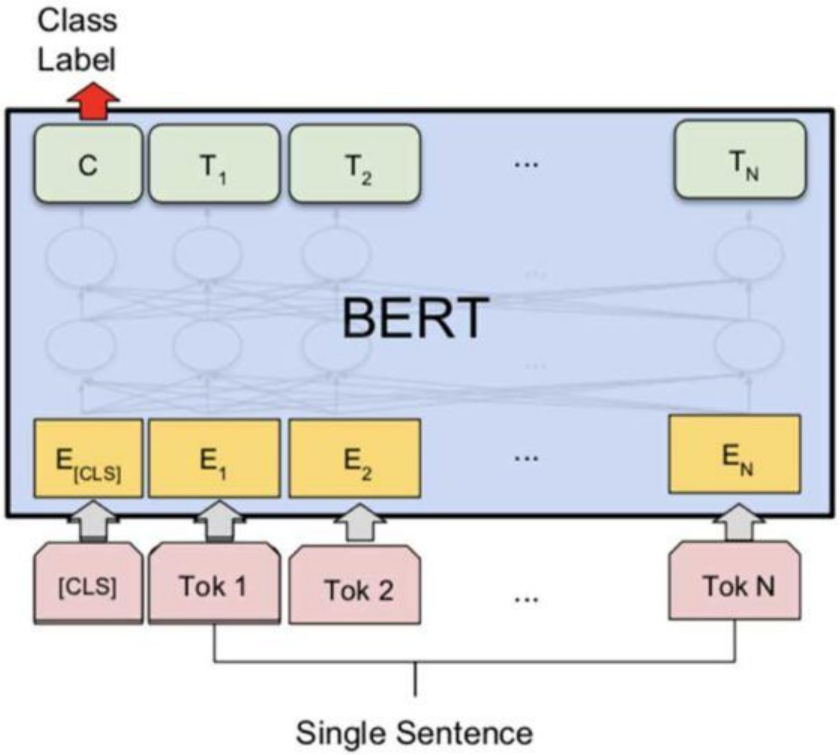


Представление токенов для BERT

- WordPiece токенизация слова
- Индикатор одного из двух предложений
- Позиция слова в предложении


Input
[CLS] my dog is cute [SEP] he likes play ##ing [SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}


Class Label
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
Tok M
Sentence 1
Sentence 2


Class Label
C
T1
T2
...
TN
BERT
E[CLS]
E1
E2
...
EN
[CLS]
Tok 1
Tok 2
...
Tok N
Single Sentence

Стадии обучения

- Pre-training
  - ★ Маскированная языковая модель
  - ★ Предсказание следующего предложения

- Fine-tuning

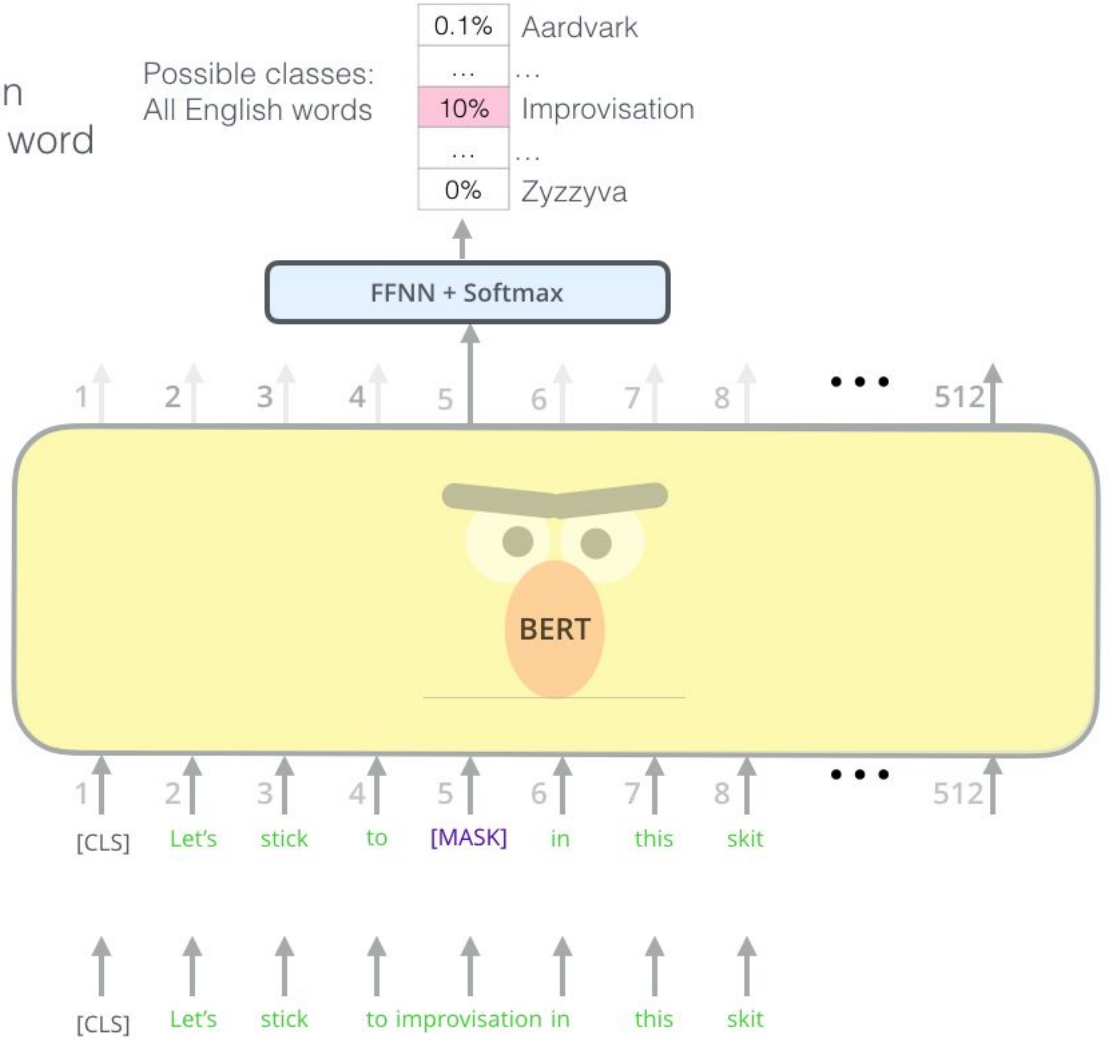
Маскированная языковая модель (MLM)

Хотим научить нашу модель понимать контекст вокруг слов

1. Случайно заменяем 15% токенов на [MASK]
   - 80% времени заменяем токен на [MASK]
   - 10% времени заменяем на случайный токен
   - 10% времени токен остается исходным

2. Модель предсказывает только замаскированные токены, но у нее нет информации, какие именно это токены

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



Randomly mask 15% of tokens

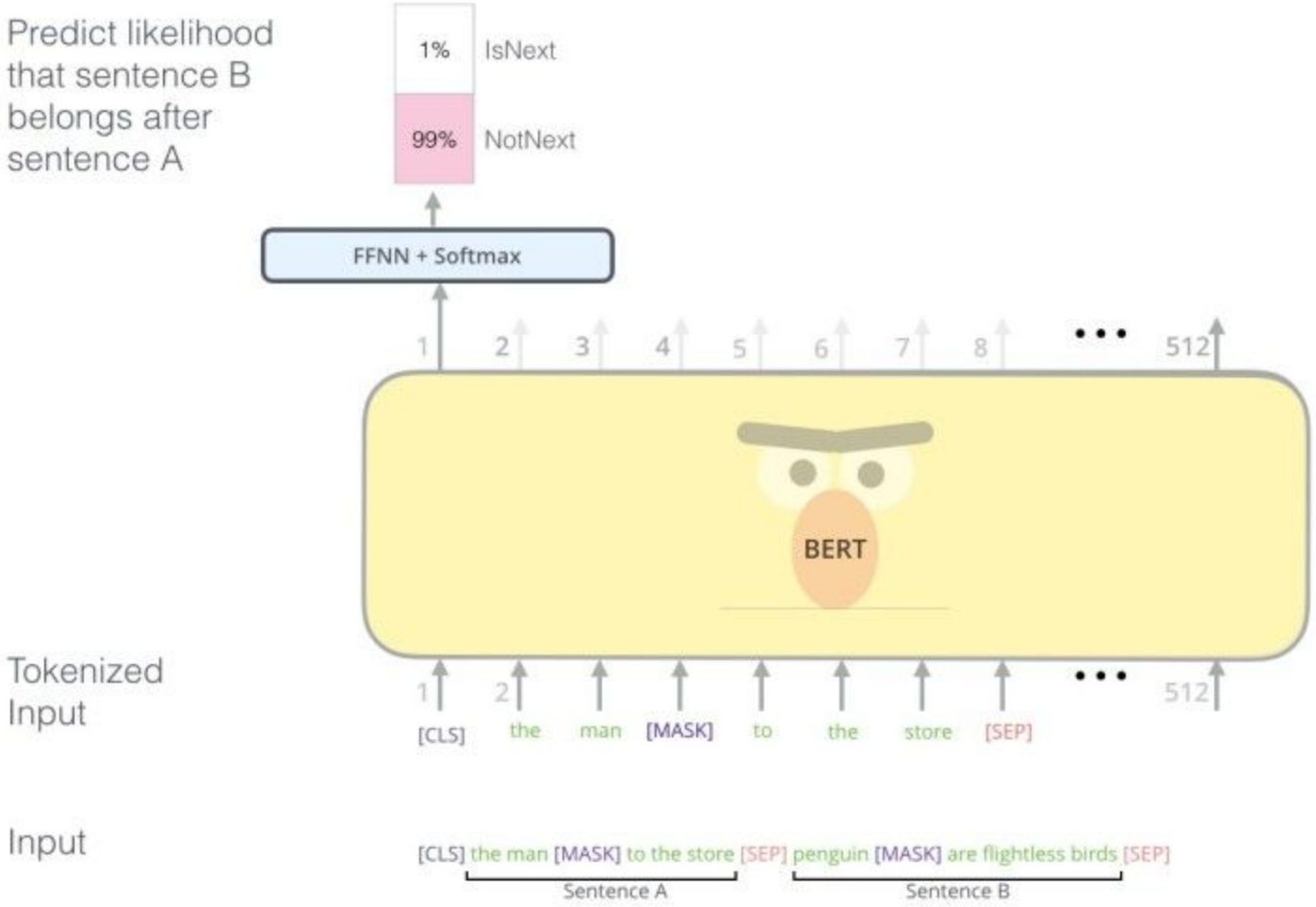
Input

Предсказание следующего предложения (NSP)

Хотим понимать связь между двумя предложениями

- 50% времени даем следующее предложение
- 50% времени даем несвязанные предложения

Модель должна предсказать, следует ли второе предложение из первого


Predict likelihood that sentence B belongs after sentence A
1%
IsNext
99%
NotNext
FFNN + Softmax
1
2
3
4
5
6
7
8
512
BERT
Tokenized Input
1
2
[CLS]
the
man
[MASK]
to
the
store
[SEP]
512
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Пример ввода для NSP

Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CSL] the man [MASK] to the store [SEP]

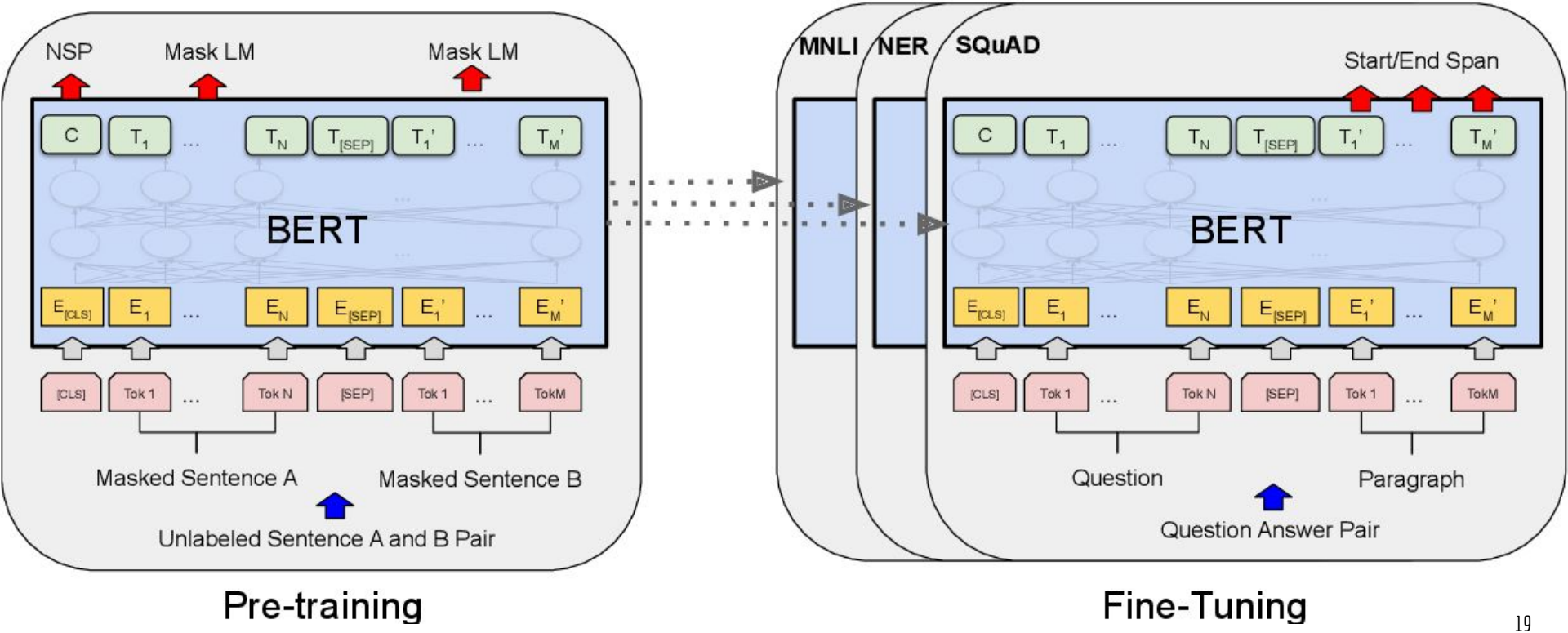
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

Fine-tuning

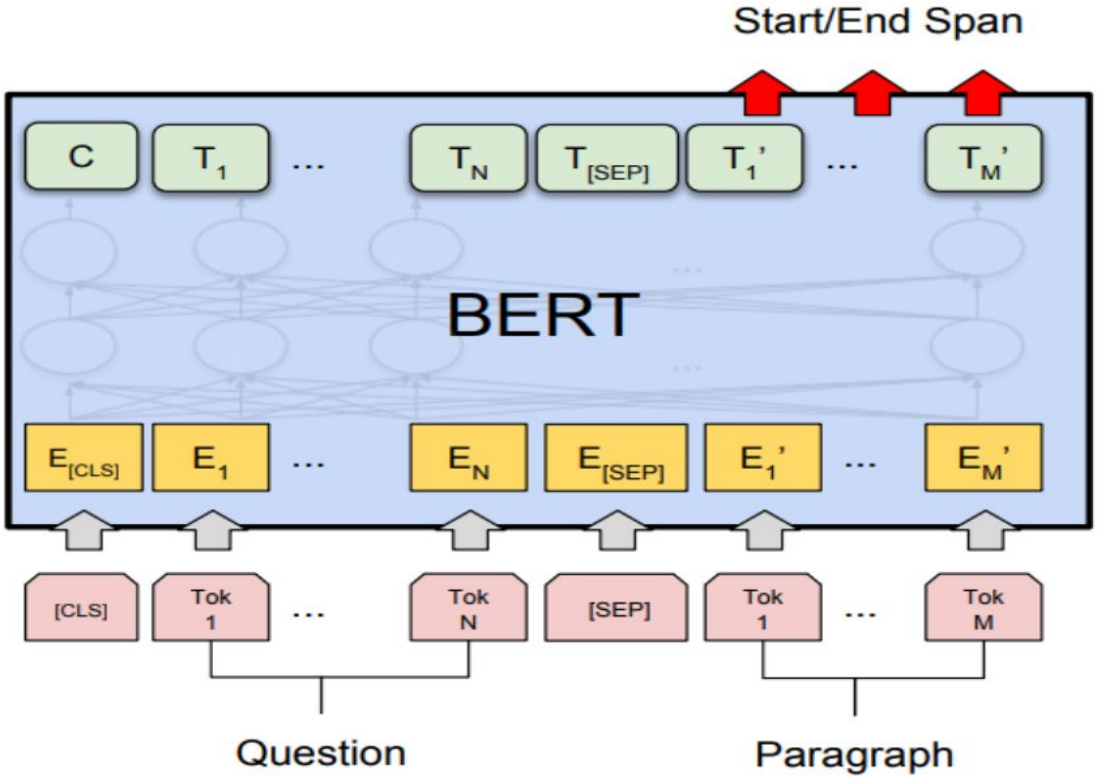
- Классификация предложений (анализ тональности)
- Классификация пар предложений (эквивалентность)
- Создание ответов на вопросы (ответ содержится в тексте)
- Таггинг предложения (предсказание именованных сущностей)

Пример задачи для Fine-tuning - SQuAD (Stanford Question Answering Dataset)

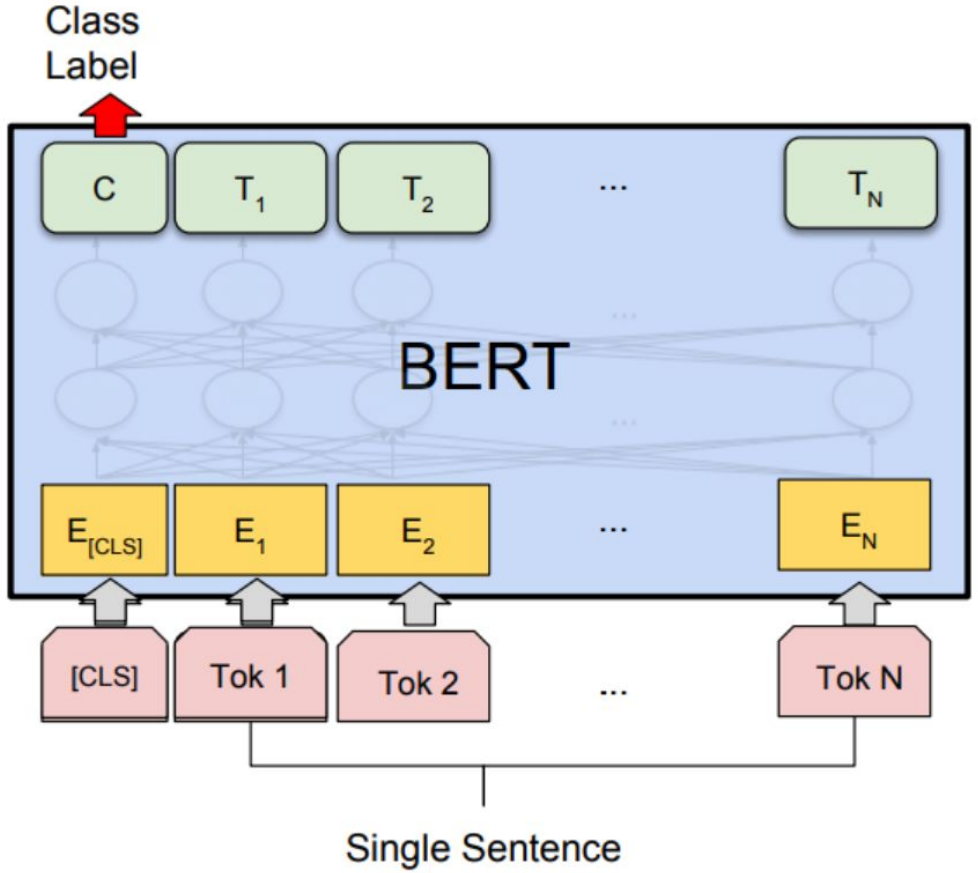


Первая последовательность - вопрос

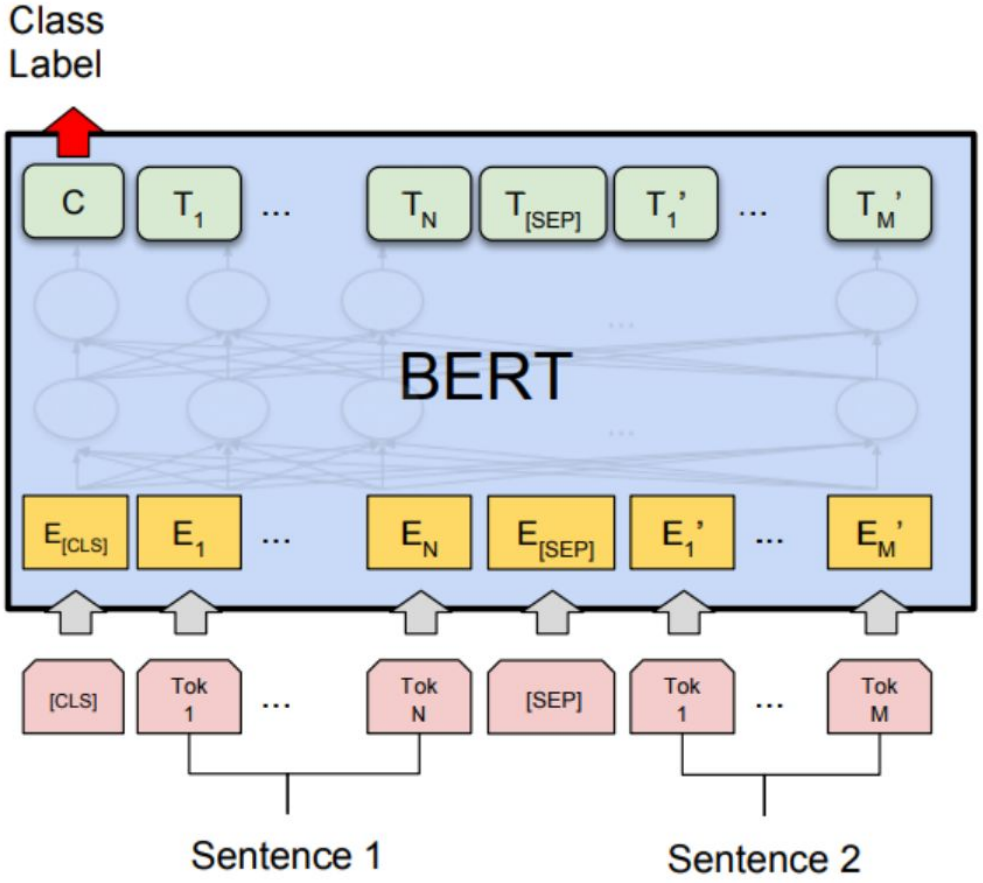
Вторая последовательность - текст с ответом



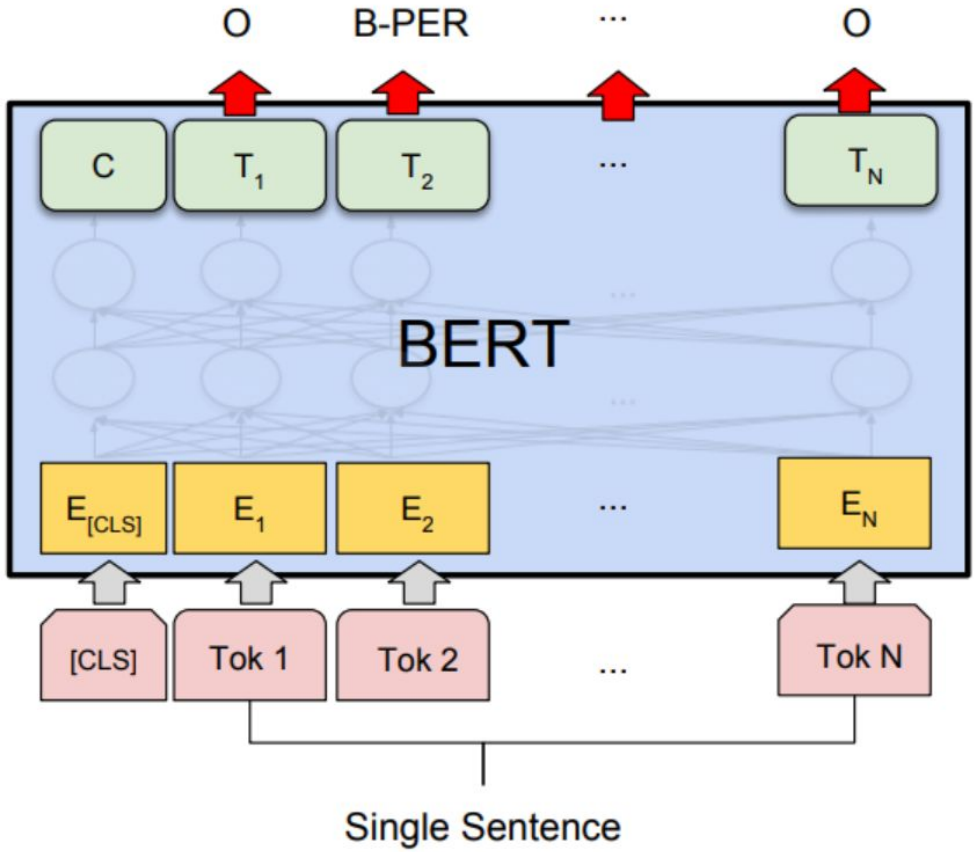
Классификация предложений



Классификация пар предложений



Таггинг предложения



Недостатки BERT

- Каждое скрытое слово предсказывается в отдельности. Мы теряем информацию о возможных связях между маскированными словами
- Несоответствие между тренировкой модели (есть [MASK] токены) и использованием предобученной модели (таких токенов нет)
- Слишком много параметров

Результаты

- Показал state-of-the-art результаты в 11 бенчмарках NLP

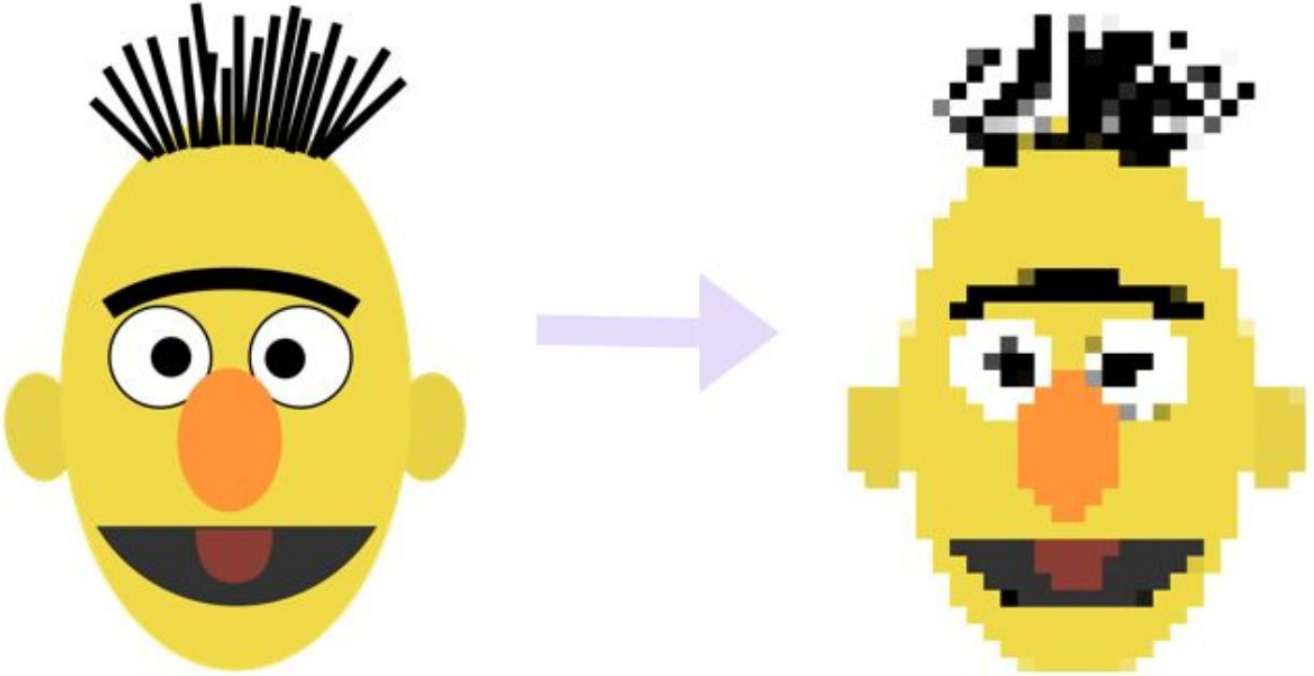
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	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**

Не слишком ли много?

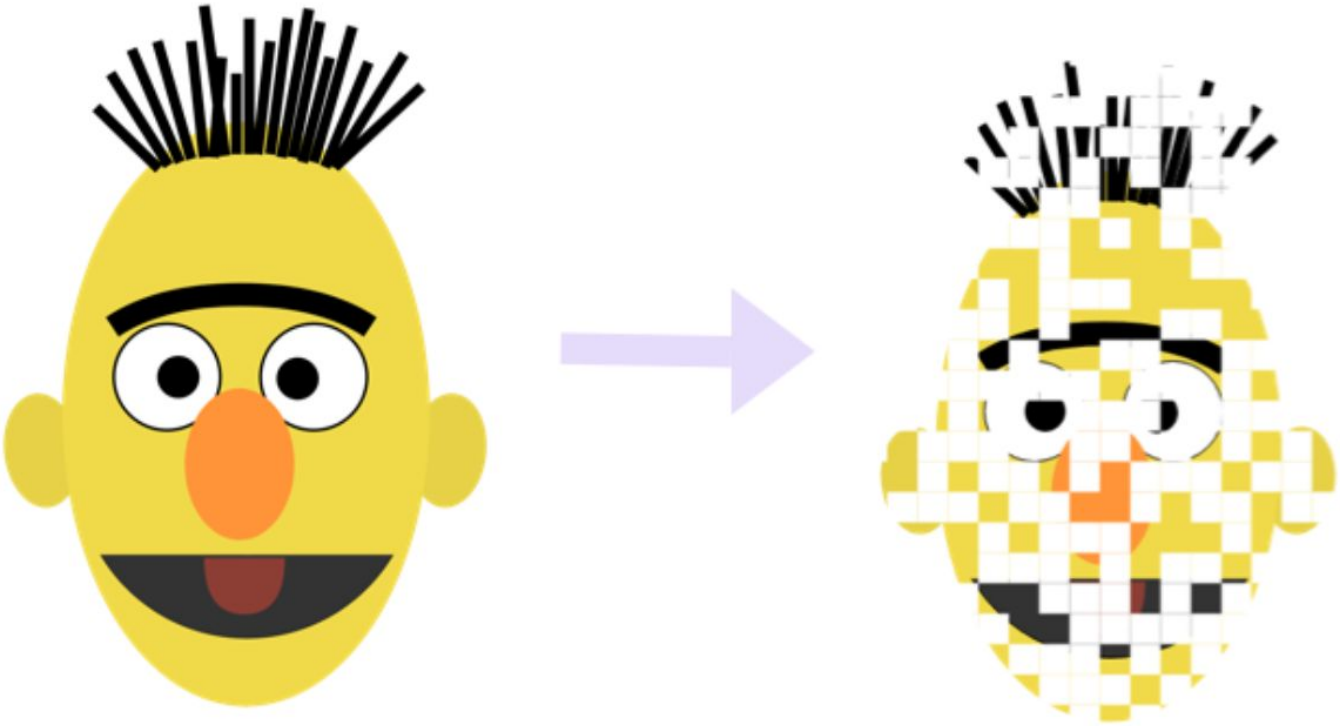
- BERT base – 12 слоев (блоков трансформера), 12 attention heads, 110 млн параметров
- BERT Large – 24 слоя, 16 attention heads, 340 млн параметров

Можно ли меньше?

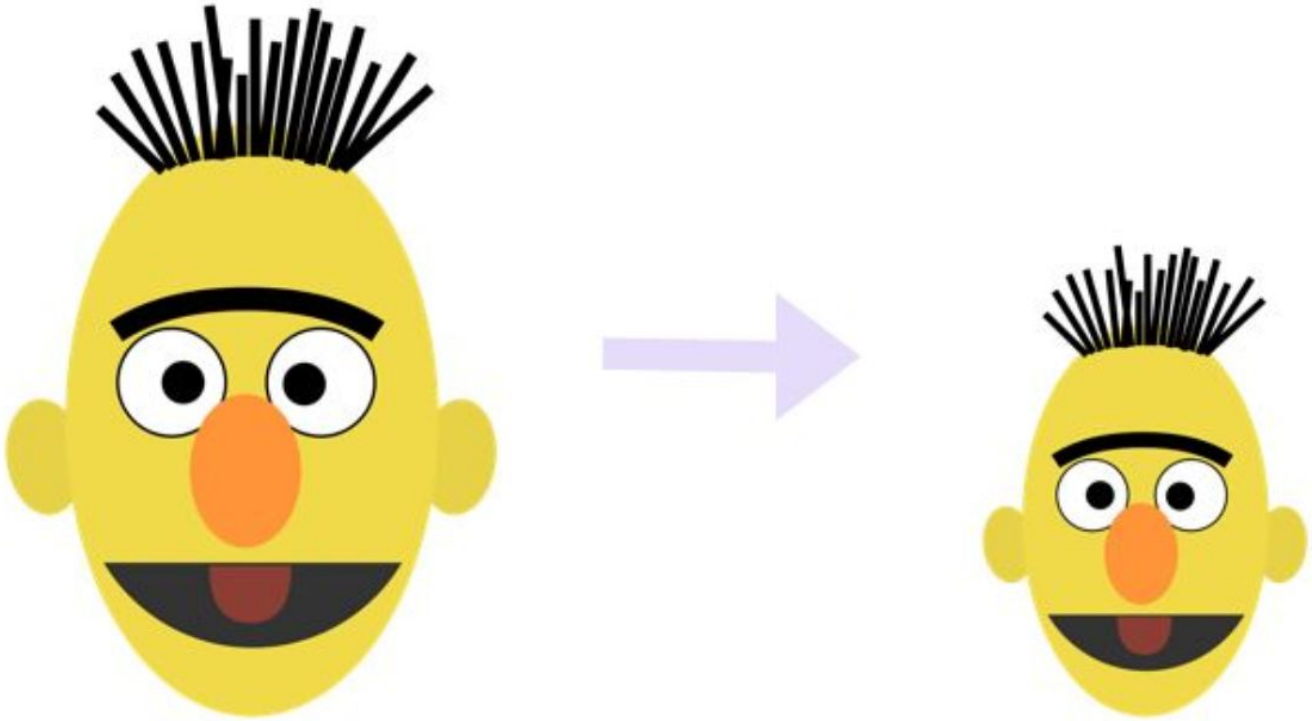
Квантизация



Прунинг



Дистилляция знаний



		Compression	Performance	Speedup	Model	Evaluation
Distillation	DistilBERT (Sanh et al., 2019)	×2.5	90%	×1.6	$BERT_6$	All GLUE tasks
	$BERT_6$-PKD (Sun et al., 2019a)	×1.6	97%	×1.9	$BERT_6$	No WNLI, CoLA and STS-B
	$BERT_3$-PKD (Sun et al., 2019a)	×2.4	92%	×3.7	$BERT_3$	No WNLI, CoLA and STS-B
	(Aguilar et al., 2019)	×2	94%	-	$BERT_6$	CoLA, MRPC, QQP, RTE
	BERT-48 (Zhao et al., 2019)	×62	87%	×77	$BERT_{12}^{*\dagger}$	MNLI, MRPC, SST-2
	BERT-192 (Zhao et al., 2019)	×5.7	94%	×22	$BERT_{12}^{*\dagger}$	MNLI, MRPC, SST-2
	TinyBERT (Jiao et al., 2019)	×7.5	96%	×9.4	$BERT_4^{*\dagger}$	All GLUE tasks
	MobileBERT (Sun et al.)	×4.3	100%	×4	$BERT_{24}^{\dagger}$	No WNLI
	PD (Turc et al., 2019)	×1.6	98%	$\times 2.5^3$	$BERT_6^{\dagger}$	No WNLI, CoLA and STS-B
	MiniBERT(Tsai et al., 2019)	$\times 6^{\S}$	98%	$\times 27^{\S}$	$mBERT_3^{\dagger}$	CoNLL-2018 POS and morphology
	BiLSTM soft (Tang et al., 2019)	×110	91%	$\times 434^{\ddagger}$	$BiLSTM_1$	MNLI, QQP, SST-2
Quant.	Q-BERT (Shen et al., 2019)	×13	99%	-	$BERT_{12}$	MNLI, SST-2
	Q8BERT (Zafrir et al., 2019)	×4	99%	-	$BERT_{12}$	All GLUE tasks
Other	ALBERT-base (Lan et al., 2019)	×9	97%	×5.6	$BERT_{12}^{**}$	MNLI, SST-2
	ALBERT-xxlarge (Lan et al., 2019)	×0.47	107%	×0.3	$BERT_{12}^{**}$	MNLI, SST-2
	BERT-of-Theseus (Xu et al., 2020)	×1.6	98%	-	$BERT_6$	No WNLI

NO, you cannot understand the meaning of a text without explicitly evaluating its linguistic constituents and defining grammar rules!

haha gpus go brrrrrrrrr

Вопросы

- Из каких частей состоят эмбеддинги для модели?
- Опишите задачи, на которых предобучается BERT
- Для каких задач может быть дообучен BERT, приведите конкретный пример, опишите, что подается на вход и какие нужно добавить выходные слои

Всем спасибо!