ТЕХНОСФЕРА

Лекция 8
Современные модели для NLP

Байгушев Данила

16 апреля 2021 г.

Современные проблемы NLP

- Поиск ответа в тексте
- Суммаризация
- Генерация продолжения
- Заполнение пропущенных частей
- Ответ на вопросы

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jan 10, 2020	Retro-Reader on ALBERT (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	**90.115**	**92.580**
7 Mar 06, 2020	ELECTRA (single model) *Google Brain & Stanford*	88.716	91.365
8 Feb 24, 2020	ALBERT (Single model) *SRCB_DML*	88.592	91.286
8 Sep 16, 2019	ALBERT (single model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	88.107	90.902
8 Jul 26, 2019	UPM (ensemble) *Anonymous*	88.231	90.713
8 Feb 10, 2020	SkERT-Large (single model) *Skelter Labs*	87.994	90.944
9 Nov 15, 2019	XLNet (single model) *Google Brain & CMU*	87.926	90.689

Пример: SQuAD

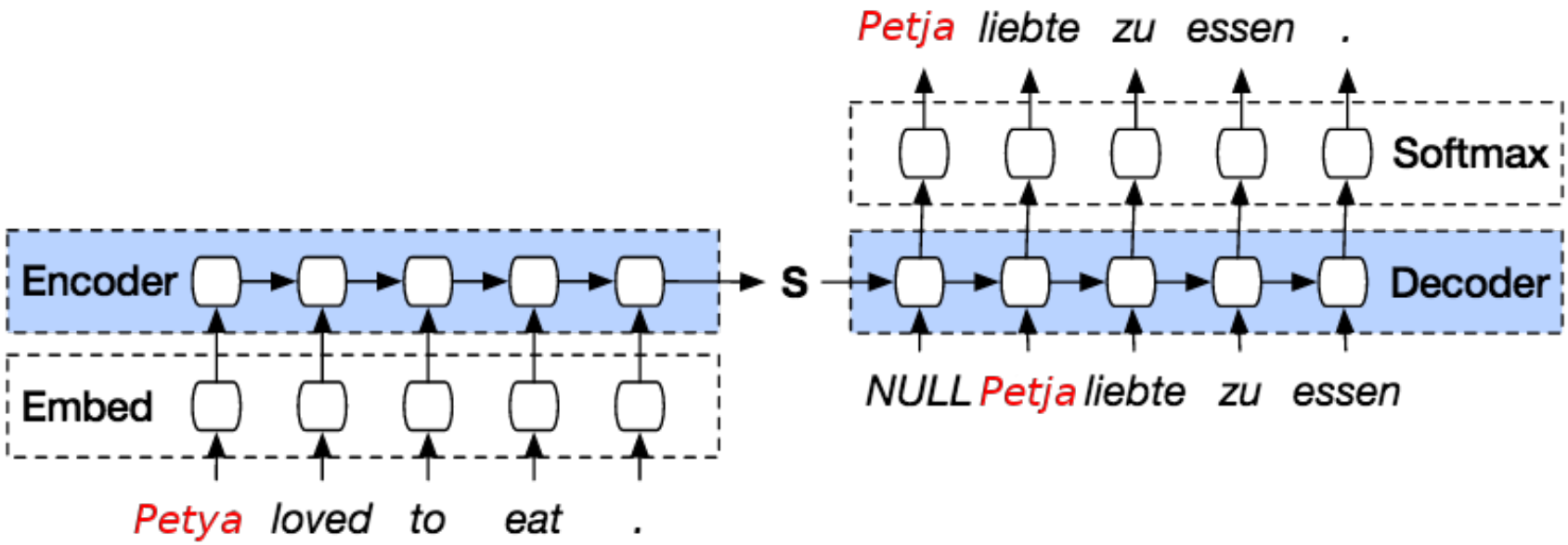
The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

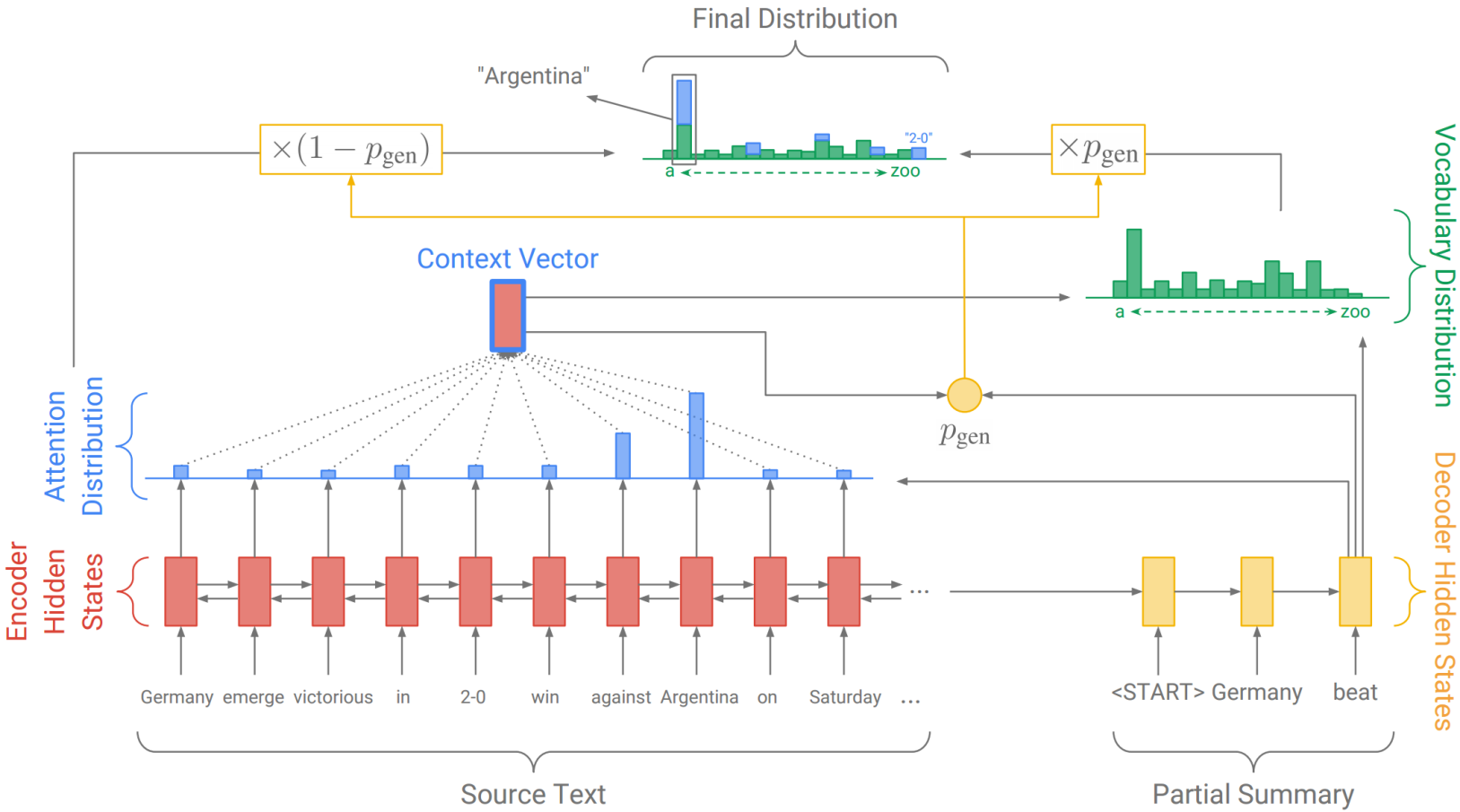
RNN

Проблема: надо запоминать точные сущности из текста, например, имена, названия, ..., также для цитирования надо запомнить точный текст, при ограниченном размере эмбеддинга это невозможно.



Attention[1]

Добавим Attention.



[1] https://arxiv.org/abs/1409.0473

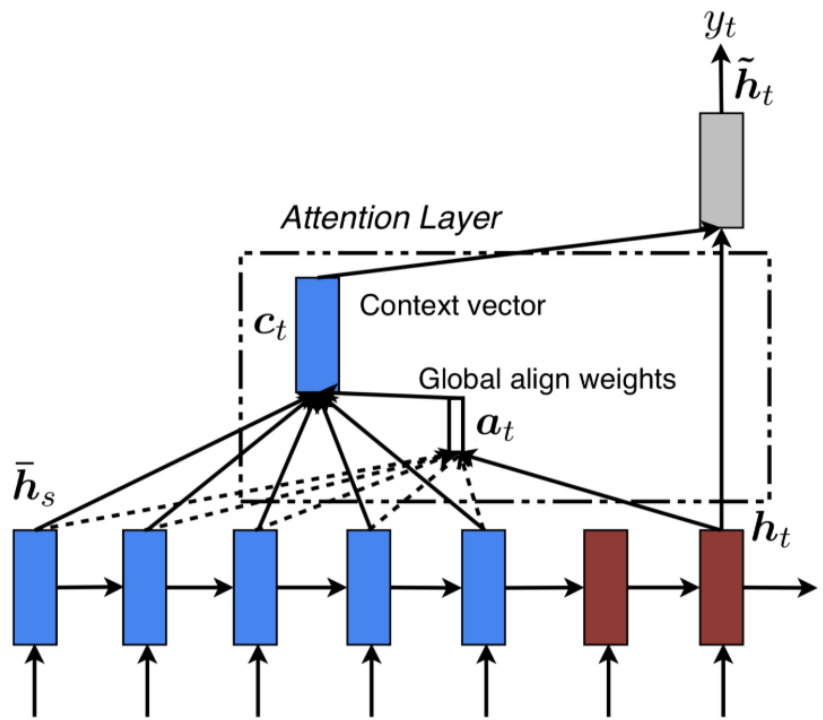
Attention

$$\text{decoder}_i = \text{RNN}(...)$$

$$\text{attention\_score}_{ij} = \text{softmax}_j(\text{attention}(\text{decoder}_i, \text{encoder\_output}_j))$$

$$\text{context}_i = \sum_j \text{attention\_score}_j \cdot \text{encoder\_output}_j$$

$$\text{decoder\_output}_i = \text{softmax}(f(\text{decoder}_i, \text{context}_i))$$



Attention to images

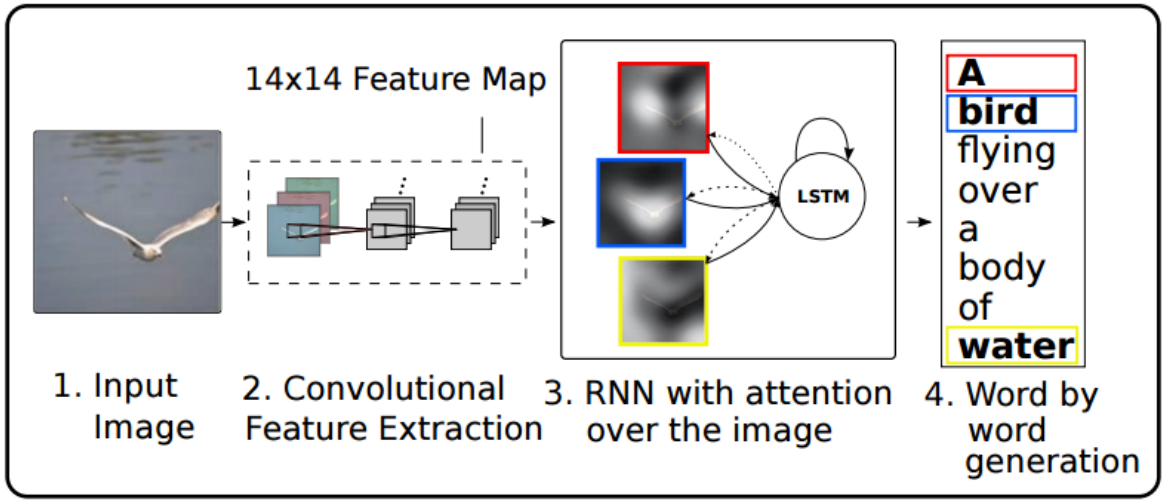
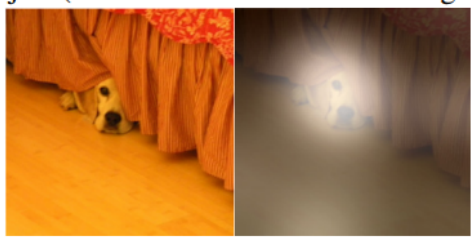


Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

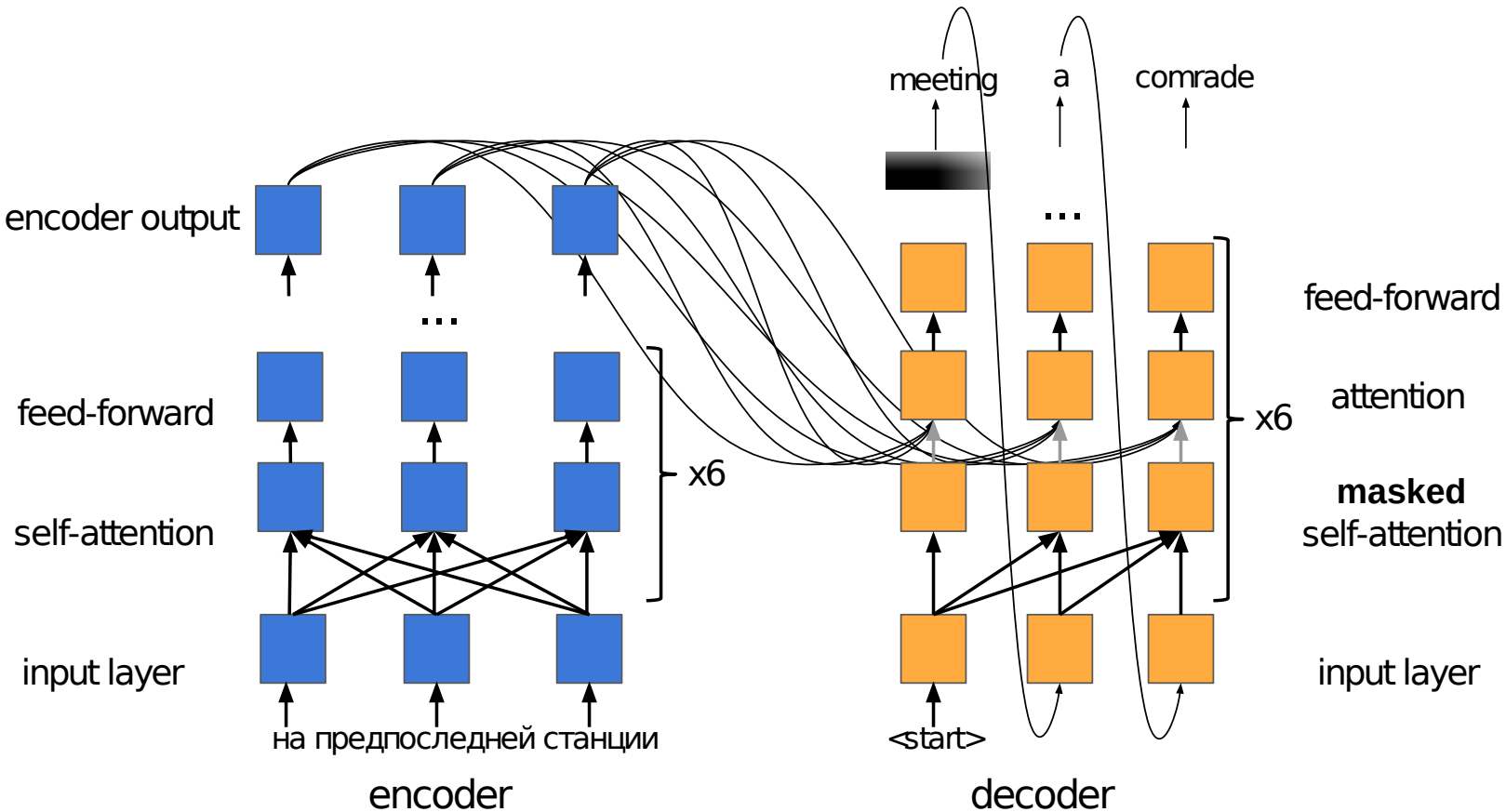
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

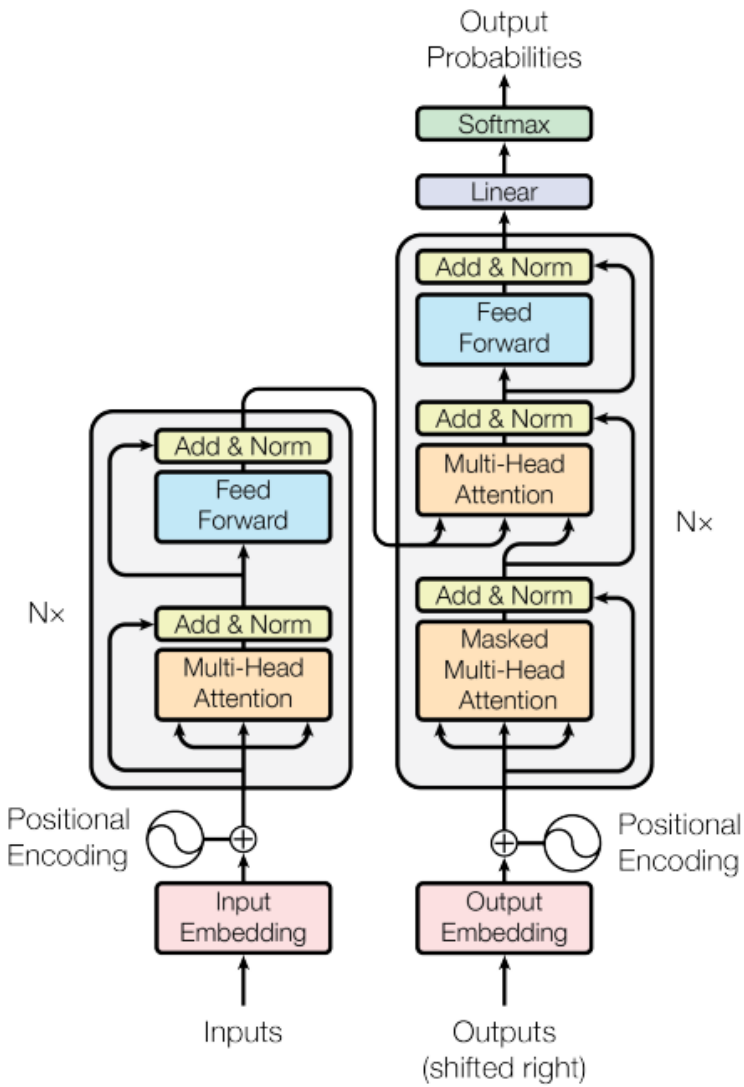
Transformer (Attention is all you need)[2]

Визуализация (gif)

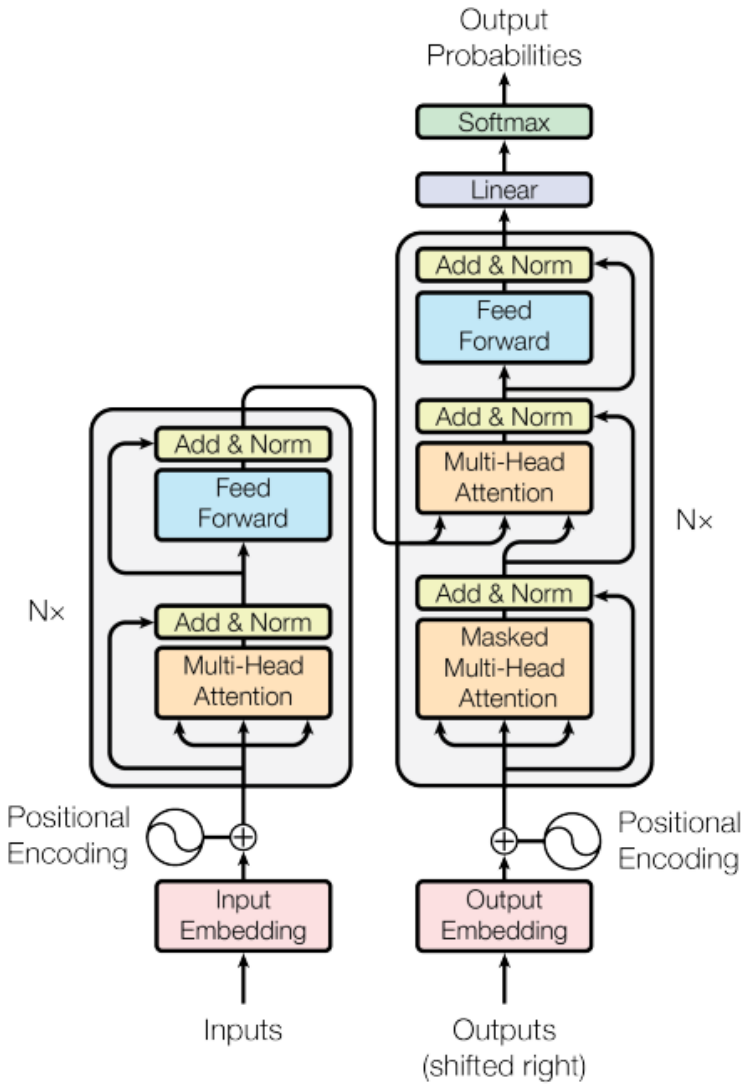


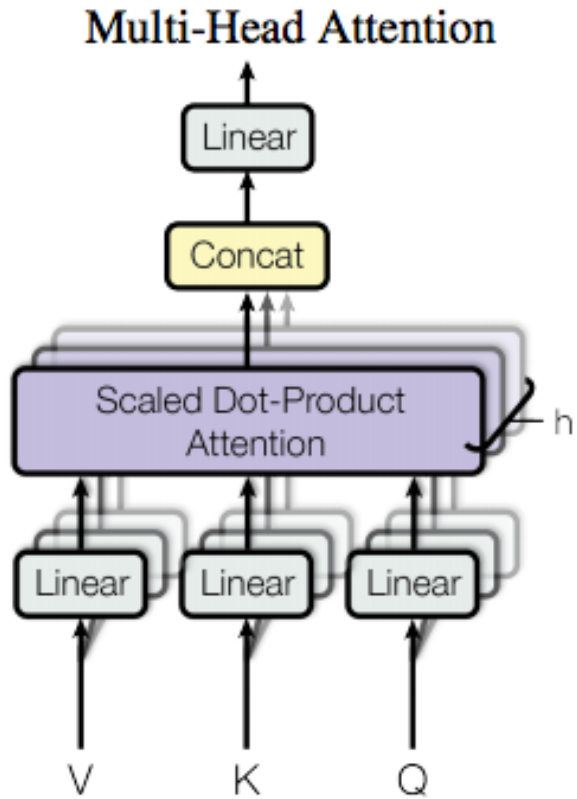
[2] https://arxiv.org/abs/1706.03762

Transformer (Attention is all you need)



Transformer (Attention is all you need)





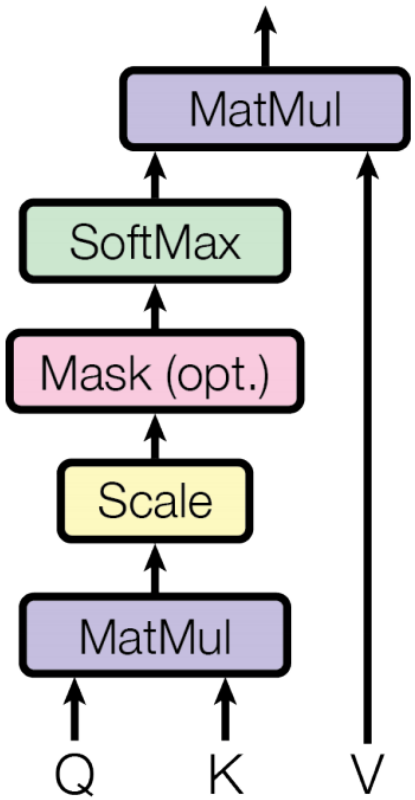
Attention (Attention is all you need)

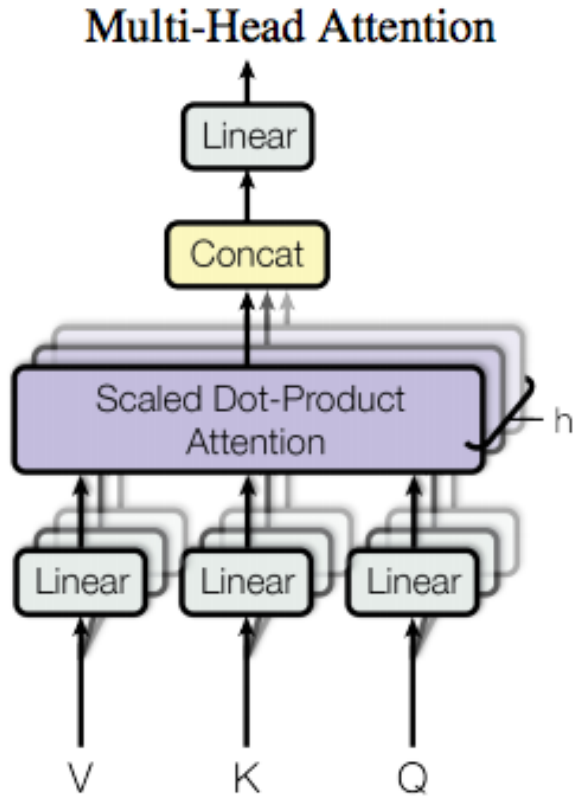
Одна голова: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$.

Параллелим: $\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$

Где $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

One Head Attention

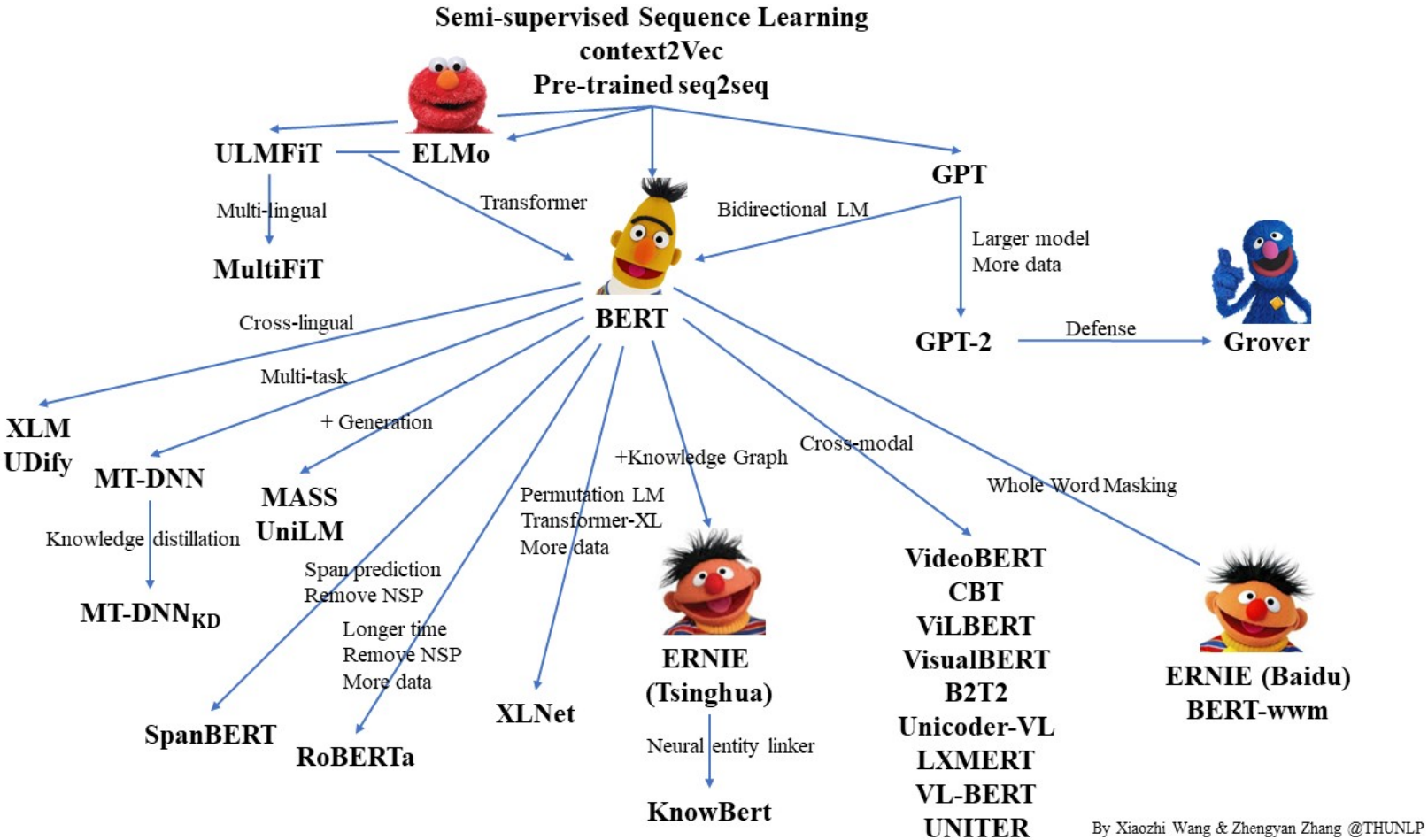




Transformer (Attention is all you need)

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

Transformer Family

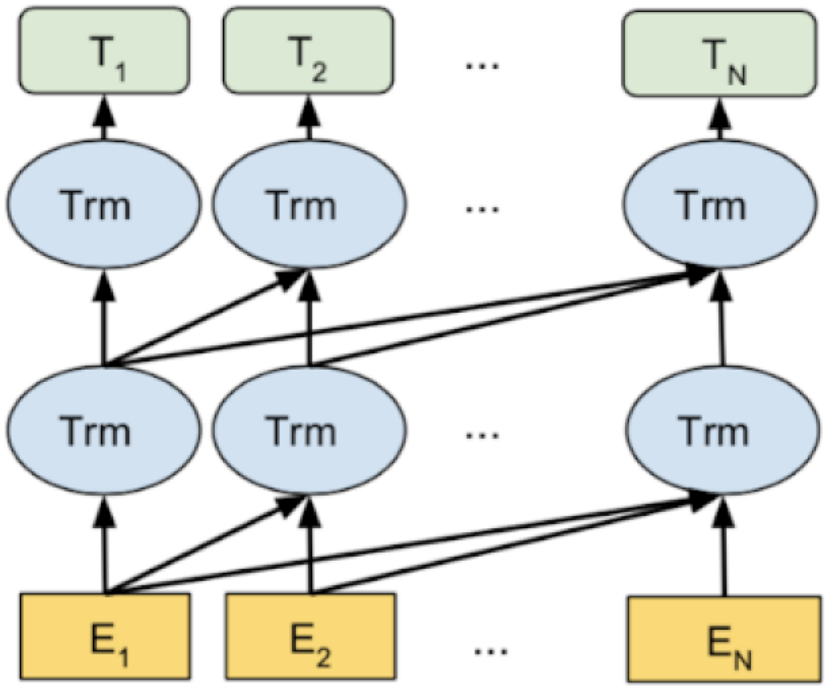


GPT[3]

Language model:

$$P(w_{n+1}|w_1, \ldots, w_n) = f(w_1, \ldots, w_n)$$

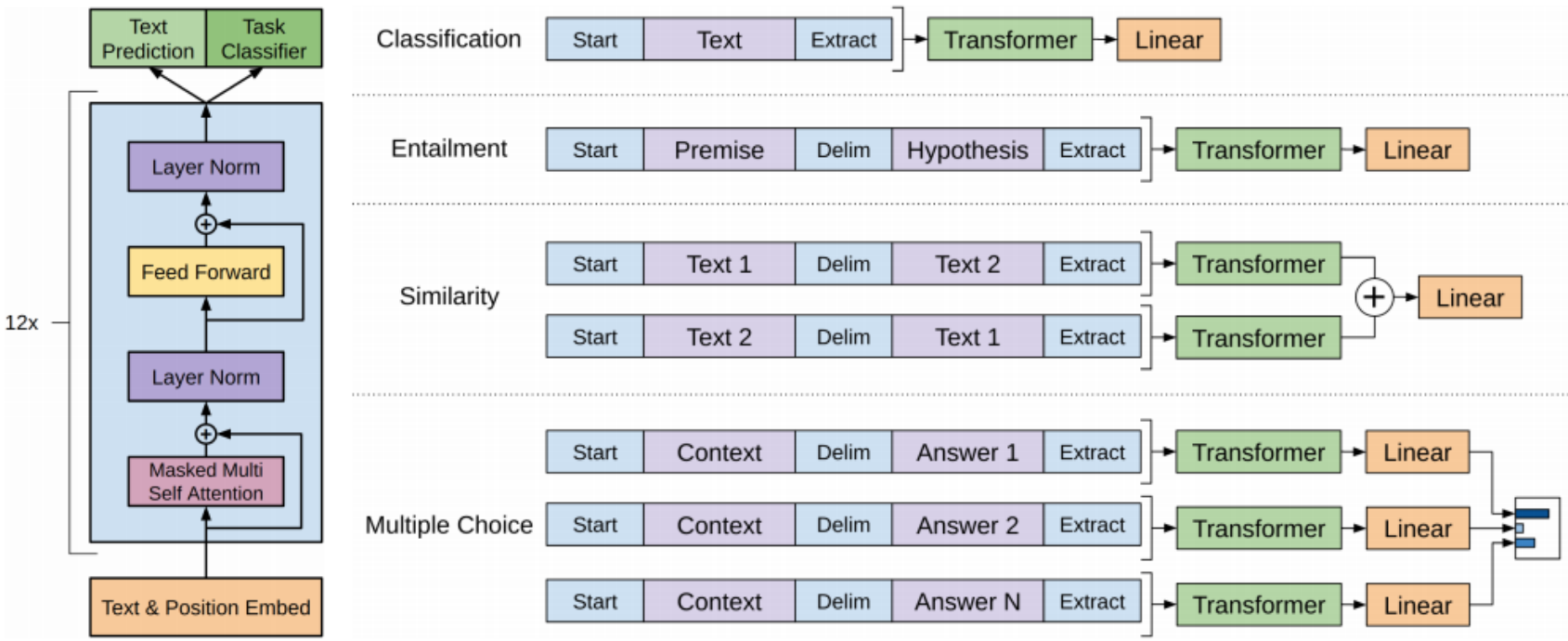
В случае GPT используются слои трансформера.
Также для ускорения обучения мы можем за один проход учить сразу несколько предсказаний за счет масок. (каждая позиция может "смотреть" только назад)



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

Multitask pretraining

Предобучаем модель на огромном корпусе $\rightarrow$
fine-tune на конкретную задачу.



GPT-2[4] (TL;DR)

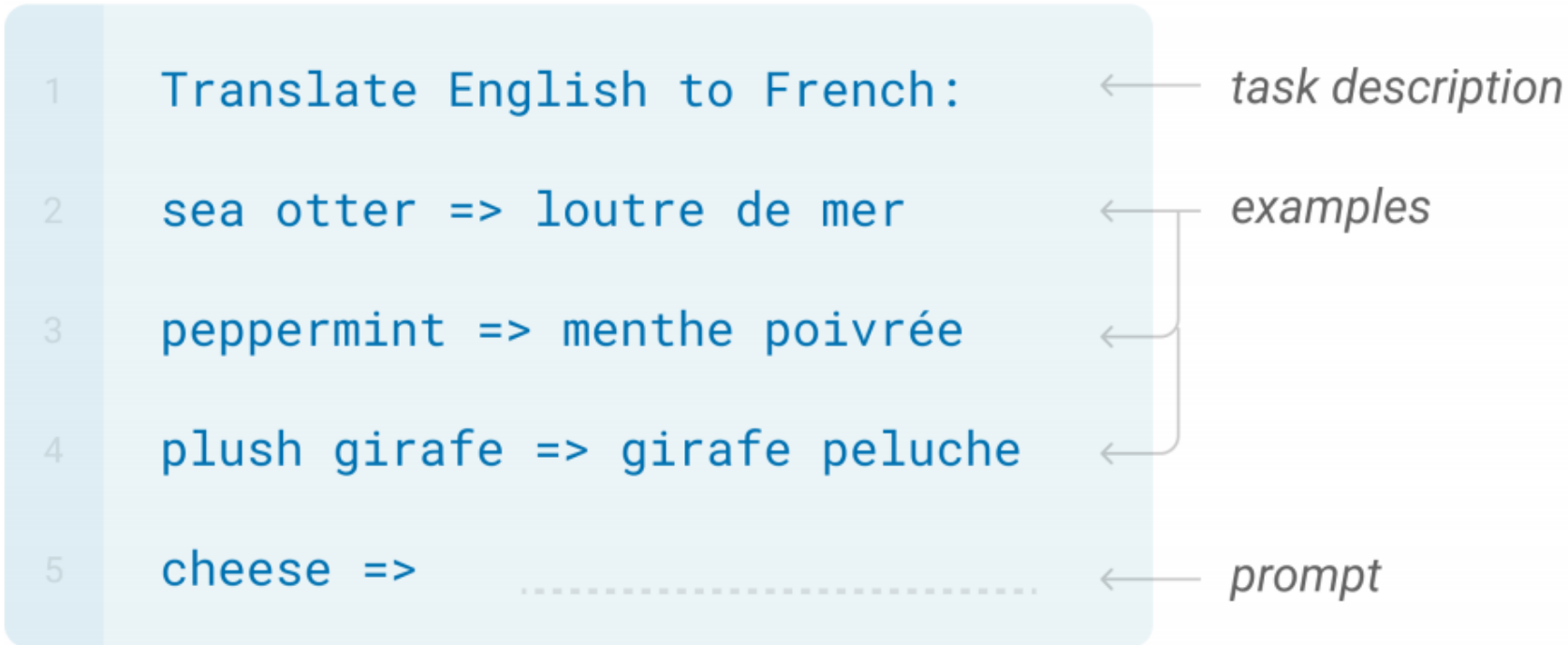
SYSTEM PROMPT (HUMAN-WRITTEN)	*A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.*
MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)	The incident occurred on the downtown train line, which runs from Covington and Ashland stations. In an email 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. According to the release, the U.S. Department of Energy's Office of Nuclear Material Safety and Security is leading that team's investigation. "The safety of people, the environment and the nation's nuclear stockpile is our highest priority," Hicks said. "We will get to the bottom of this and make no excuses.

[4] https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-3[5] (zero-shot learning!)

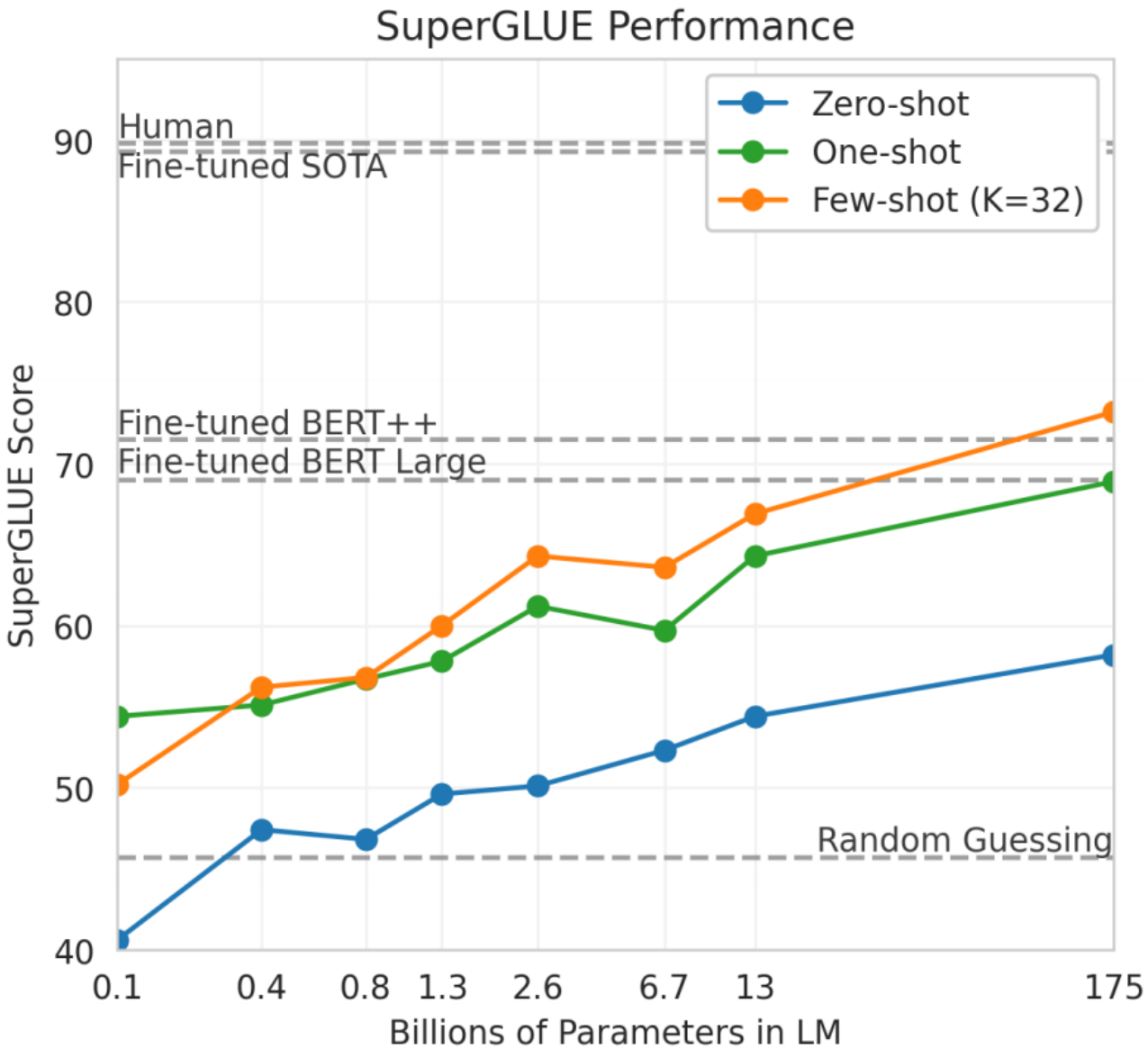
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

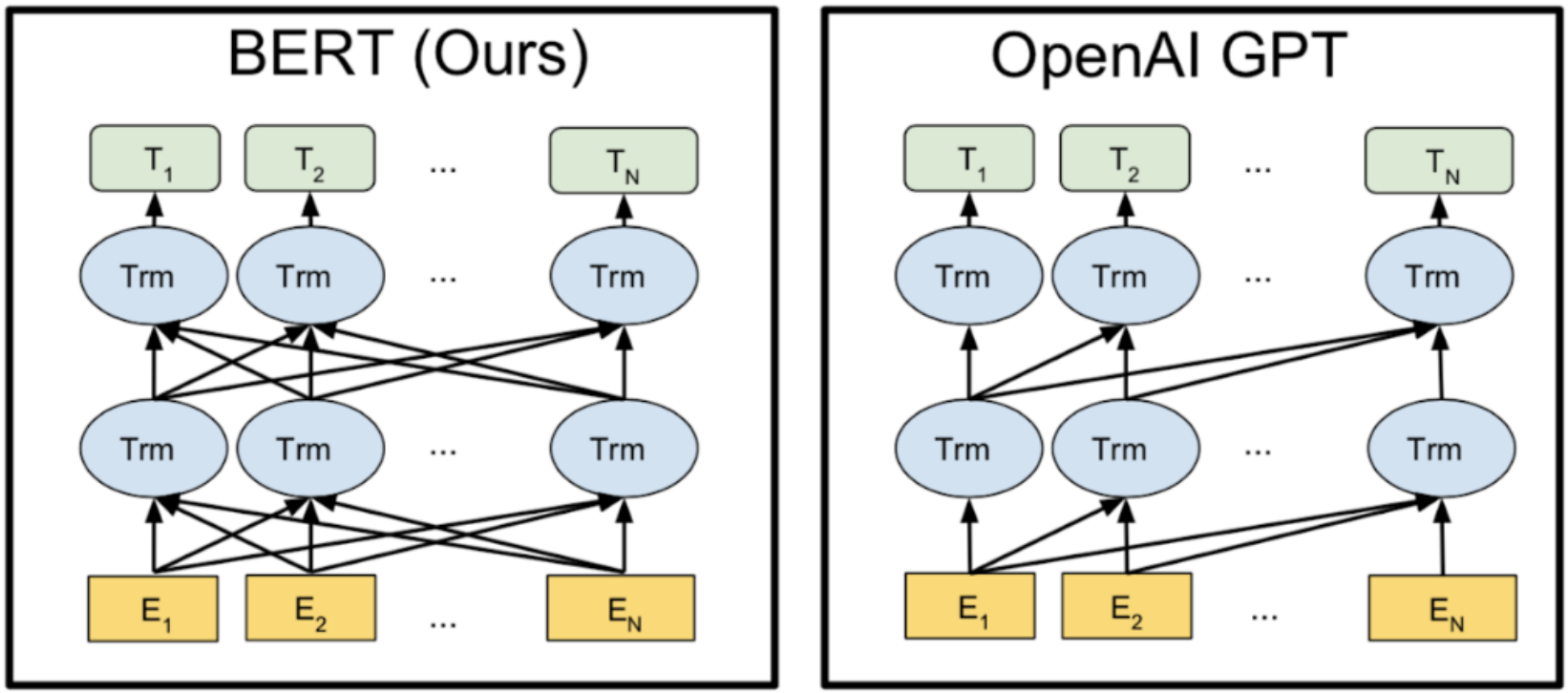


[5] https://arxiv.org/abs/2005.14165

GPT-3 (zero-shot learning!)

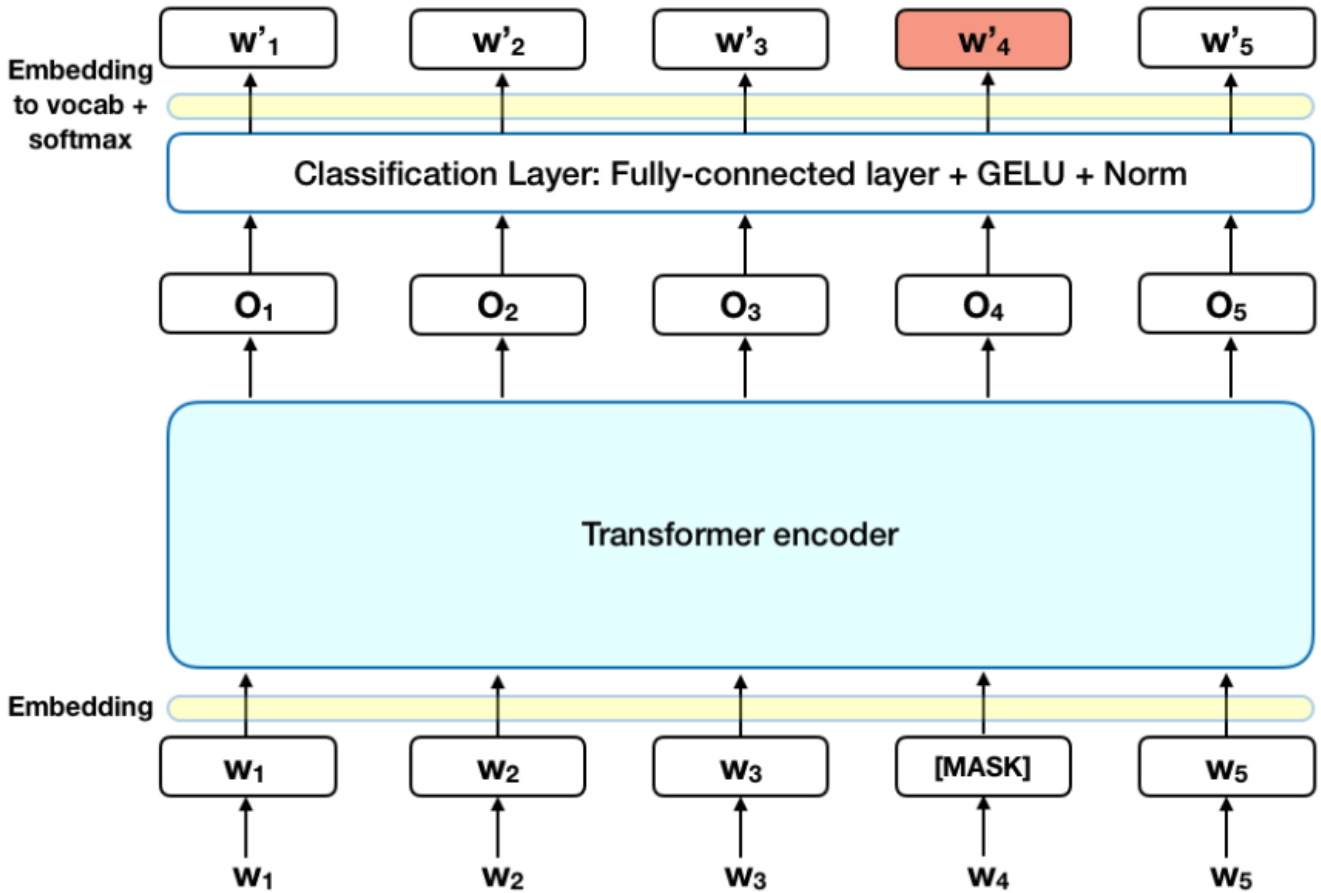


BERT vs GPT


BERT (Ours)
T1
T2
...
TN
Trm
Trm
...
Trm
Trm
Trm
...
Trm
E1
E2
...
EN
OpenAI GPT
T1
T2
...
TN
Trm
Trm
...
Trm
Trm
Trm
...
Trm
E1
E2
...
EN

BERT[6]

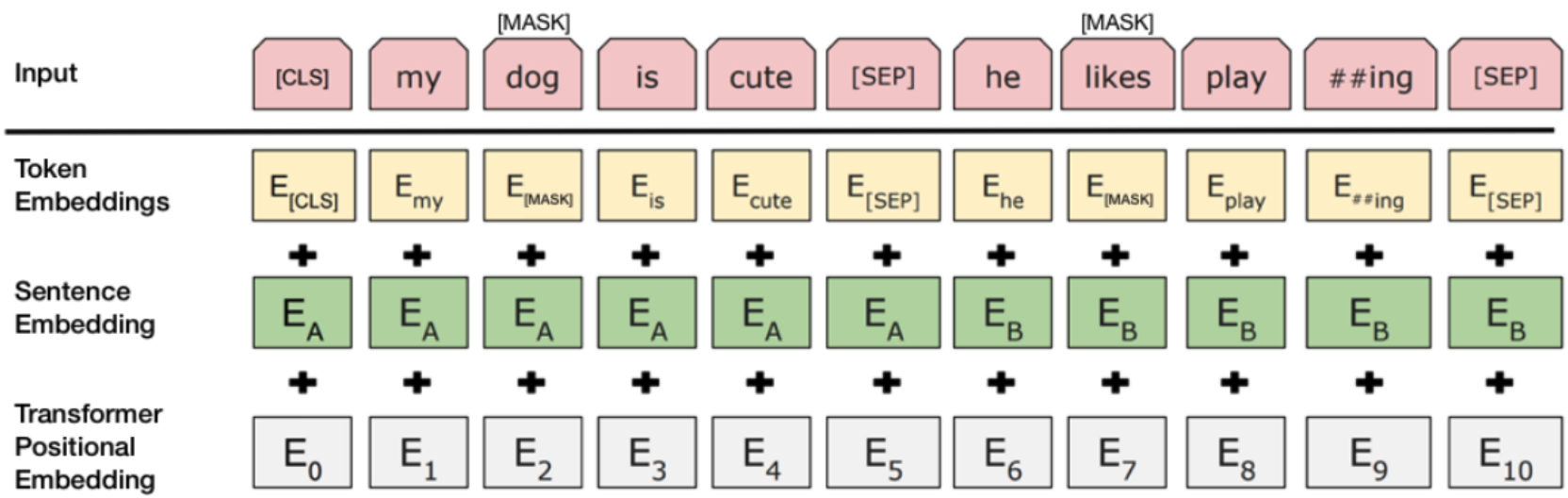
Bidirectional Encoder Representations from Transformers



[6] https://arxiv.org/abs/1810.04805

BERT - детали

- MASK - некоторые слова заменяем на токен неизвестного слова и пытаемся их восстановить.
- NSP - Для пары предложений пытаемся предсказать, правда ли, что B следует за A. (берем B случайно в 50% случаев)
  Нужно для улучшения модели языка и вопросно-ответных задач.
- Обучающее множество включает всю английскую википедию и книги не защищенные авторским правом. Для большой модели надо 4 дня на 16-и cloud TPU.



BERT - SOTA

System	Dev		Test	
	EM	F1	EM	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**

(a) BERT на SQuAD v1.0 (найти сегмент с ответом)

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

(b) BERT на SWAG (выбор из нескольких вариантов ответа)

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**

RoBERTa[7]

"We find that BERT was significantly undertrained, and can match or exceed the performance of every model published after it."

[7] https://arxiv.org/abs/1907.11692

RoBERTa[7]

"We find that BERT was significantly undertrained, and can match or exceed the performance of every model published after it."

- Объём - BERT обучался на 16GB текстов, мы будем учить на 160GB (включая датасет "хороших сайтов" GPT-2).
- NSP - учиться лучше на больших отрезках текста (параграфах, а не парах предложений), NSP не нужен! (без него на итоговых задачах не хуже, а иногда и лучше)
- Размер батча - оригинальный BERT учился на 256 примерах за раз, в работе показано, что лучше будет брать намного больший батч, например, 8K. (тут это был предел технических возможностей, есть работы, в которых увеличивали вплоть до 32K)
- RoBERTa - Robustly optimized BERT approach.

[7] https://arxiv.org/abs/1907.11692

ALBERT[8]

У BERT слишком много параметров ($BERT_{large} \approx 334M$), на самом деле, столько не надо.

[8] https://arxiv.org/abs/1909.11942

ALBERT[8]

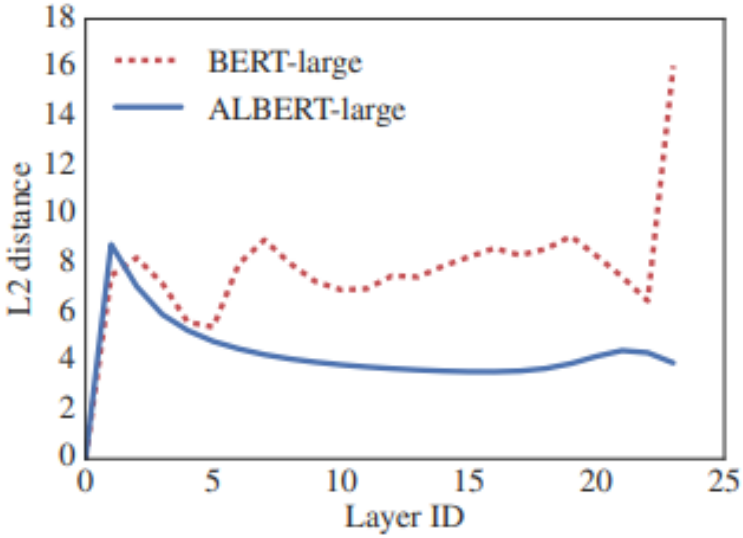
У BERT слишком много параметров ($BERT_{large} \approx 334M$), на самом деле, столько не надо.

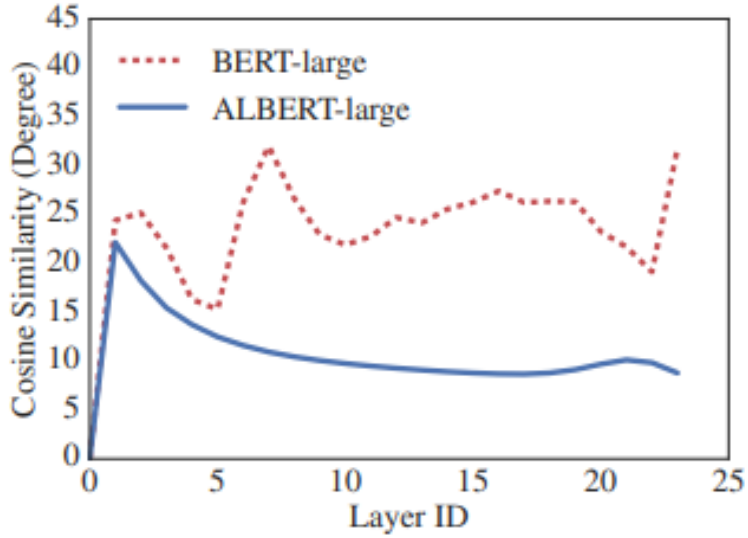
Как сокращать параметры:

- На эмбеддинги слов тратится слишком много параметров. Для лучшего качества мы хотим скрытые представления порядка $H = 2048$. При количестве слов около $V = 30000$ (на самом деле, это под-слова, но об этом мы немного поговорим в конце) матрица $V \times H$ получается слишком большой. Введем промежуточное представление размера E и факторизуем матрицу: $V \times H = V \times E \times H$. Получается меньше параметров и намного быстрее считать. (с помощью грид-серча $E = 128$)
- Давайте шарить веса на разных слоях трансформера. Вообще-то, давайте просто сделаем все слои одинаковыми (вспомните машину Больцмана). Оказывается, если применять этот слой много раз, эмбеддинги "стабилизируются".

[8] https://arxiv.org/abs/1909.11942

ALBERT

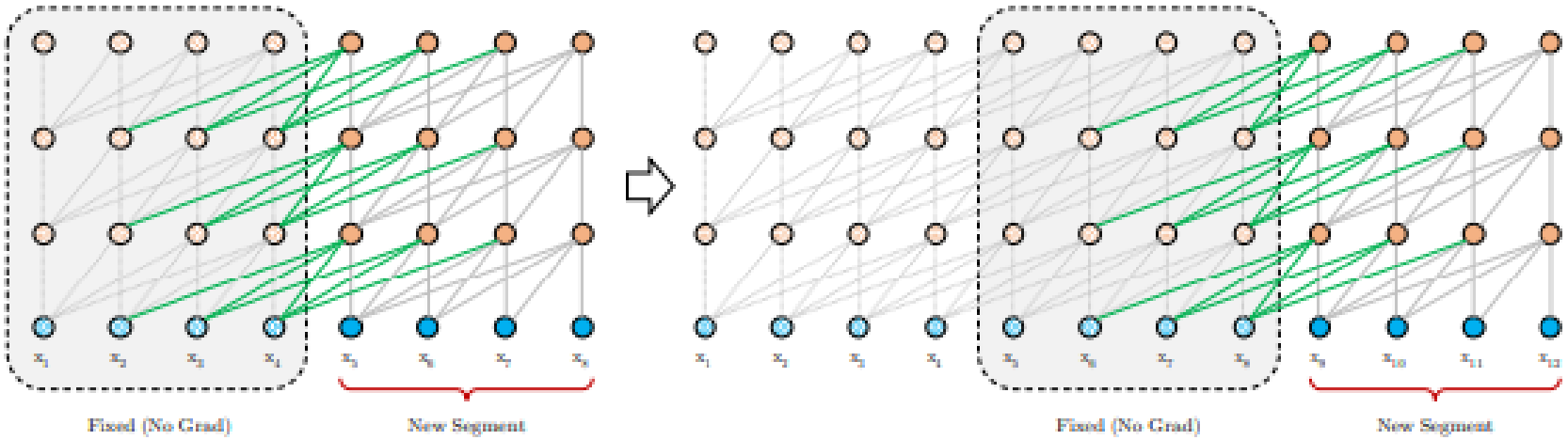




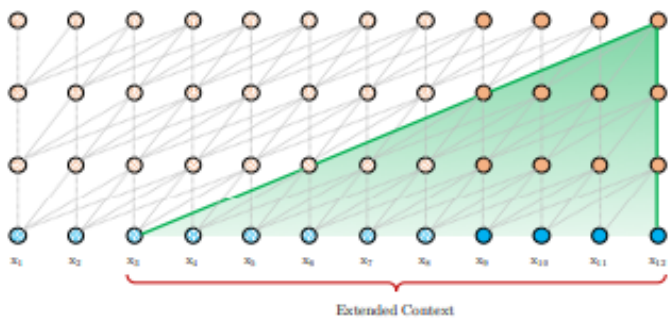
- NSP - он все-таки нужен, но другой. Используем SOP - предсказываем, правда ли, что в тексте A идет перед B, или нет.
- В итоге $ALBERT_{xxlarge} \approx 235M$ меньше $BERT_{large} \approx 334M$, при этом считается всего в 3 раза дольше. (учится быстрее)

Советую прочитать статью, там невероятно много численных экспериментов, доказывающих все позиции, также некоторое осталось за рамками (например, про то, что Dropout делает хуже).

Transformer XL[9]



(a) Training phase.

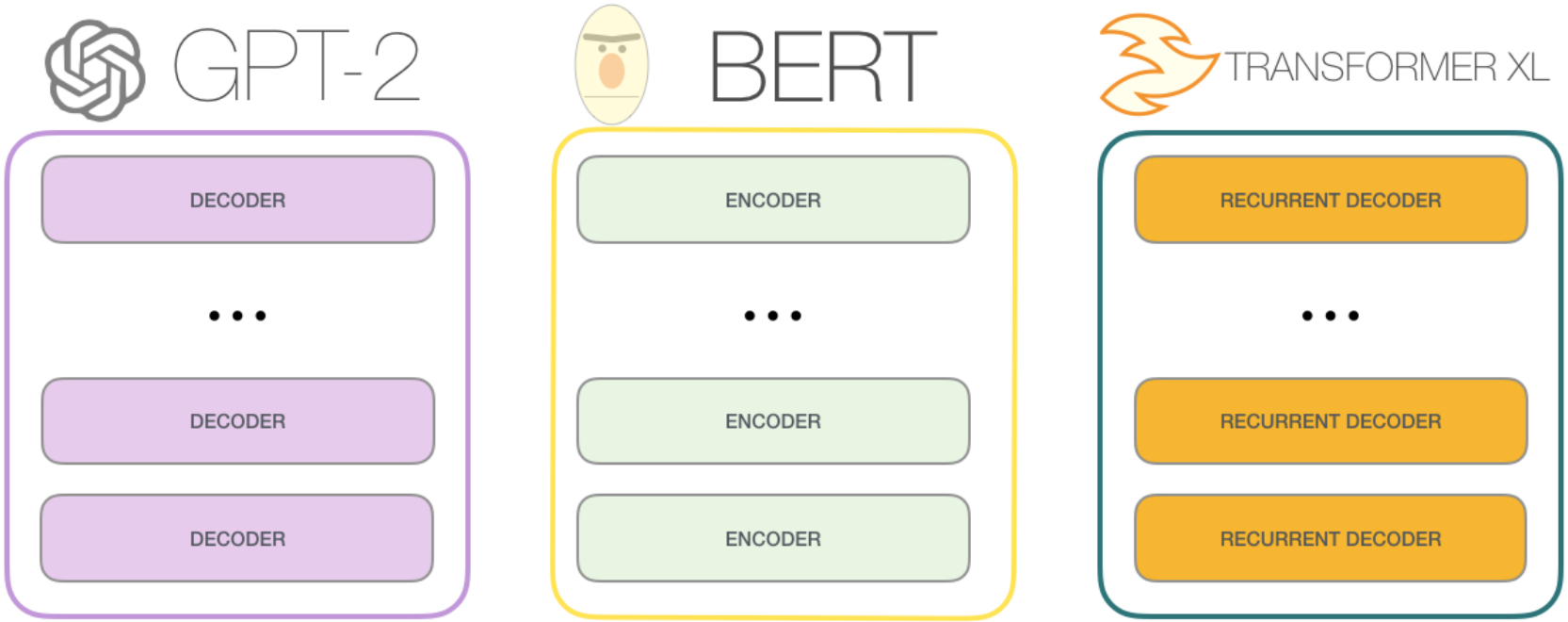


(b) Evaluation phase.

Основное вложение в трансформеры - придумали relative positional encodings.

[9] https://arxiv.org/abs/1901.02860

Transformer vs GPT vs BERT vs Transformer XL



XLNet[10]

Есть проблемы с подходом BERT:

1. Токен маски (MASK) есть только во время предобучения, что создает смещение датасета во время файн-тюнинга
2. Когда мы пытаемся восстановить несколько слов, мы считаем, что эти слова независимы друг от друга, что не всегда правда

[10] https://arxiv.org/abs/1906.08237

XLNet[10]

Есть проблемы с подходом BERT:

1. Токен маски (MASK) есть только во время предобучения, что создает смещение датасета во время файн-тюнинга
2. Когда мы пытаемся восстановить несколько слов, мы считаем, что эти слова независимы друг от друга, что не всегда правда

Давайте предсказывать слова по случайному контексту слева и справа от слова. Чтобы учиться предсказывать несколько слов, сгенерируем случайную перестановку и разрешим каждому слову смотреть только "назад".

Допустим, у нас есть предложение:
$[\text{Разводные}_1, \text{мосты}_2, \text{в}_3, \text{Санкт-}_4, \text{Петербурге}_5]$.
Перемешаем слова: $[\text{в}_3, \text{Разводные}_1, \text{Петербурге}_5, \text{Санкт-}_4, \text{мосты}_5]$.
Хотим предсказать:
$P(\text{Разводные}_1|\text{в}_3)$, $P(\text{Санкт-}_4|\text{Разводные}_1, \text{в}_3, \text{Петербурге}_5)$

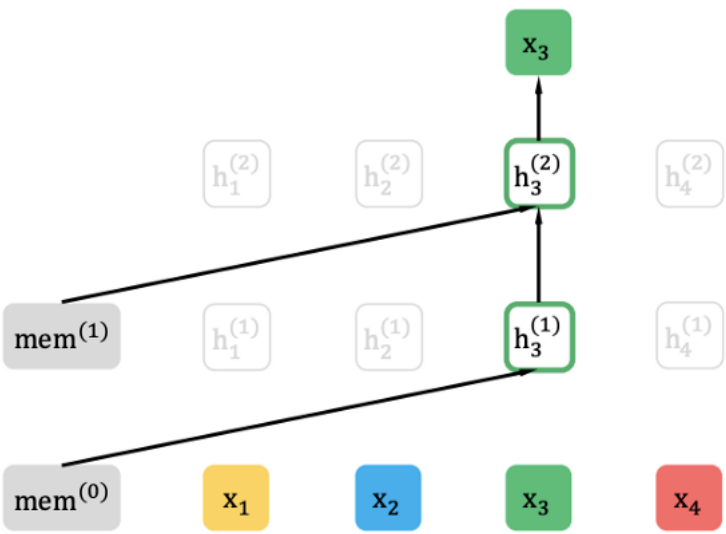
[10] https://arxiv.org/abs/1906.08237

XLNet

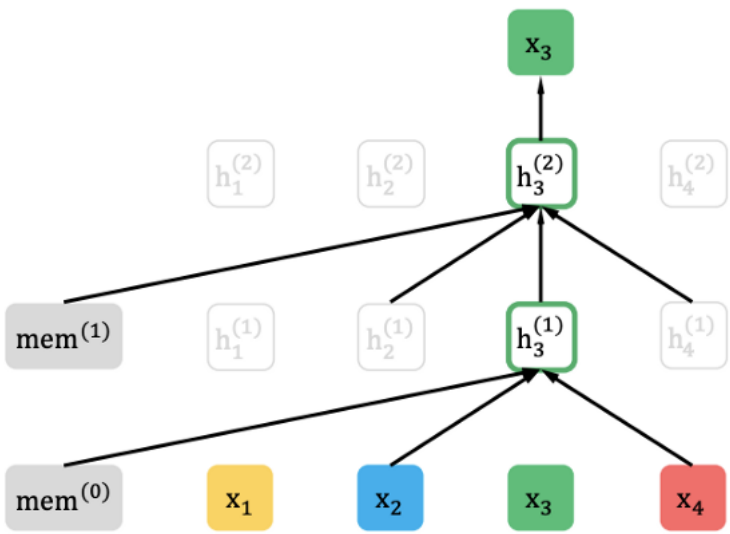
Моделируем:

1. GPT - $P(x_t|x_{i<t})$
2. BERT - $P(x_t|x_{i\notin\{t,m_1,\ldots\}})$
3. XLNet - $P(x_t|x_{\sigma(i),i<\sigma^{-1}(t)})$

На самом деле, тут есть еще одна хитрость, чтобы понимать, какое слово предсказывать, но мы ее опустим.



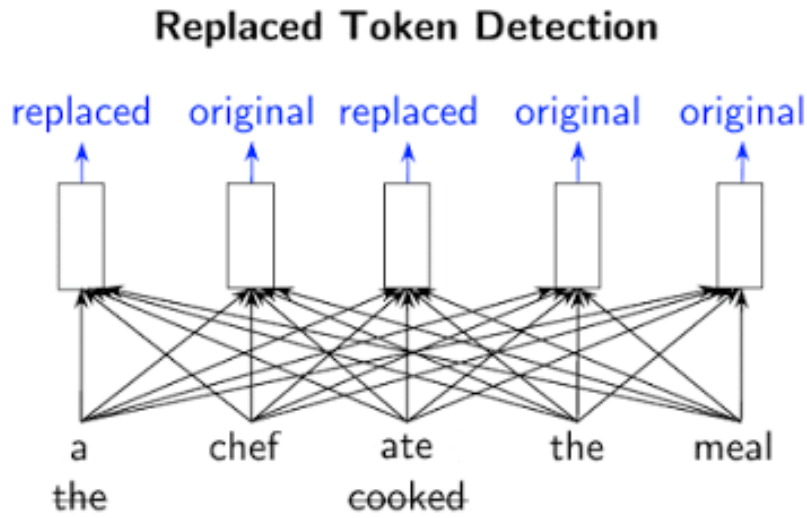
Factorization order: 3 → 2 → 4 → 1



Factorization order: 2 → 4 → 3 → 1

ELECTRA[11]

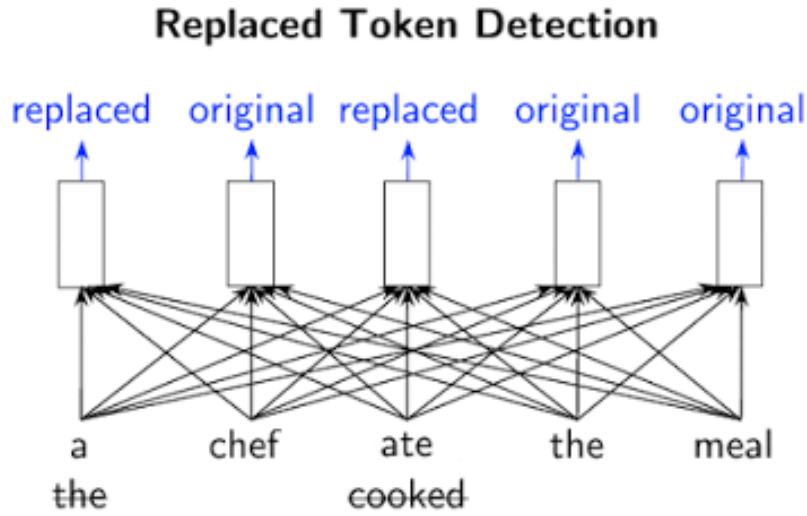
Восстанавливать замаскированные слова слишком просто, давайте искать замененные слова.

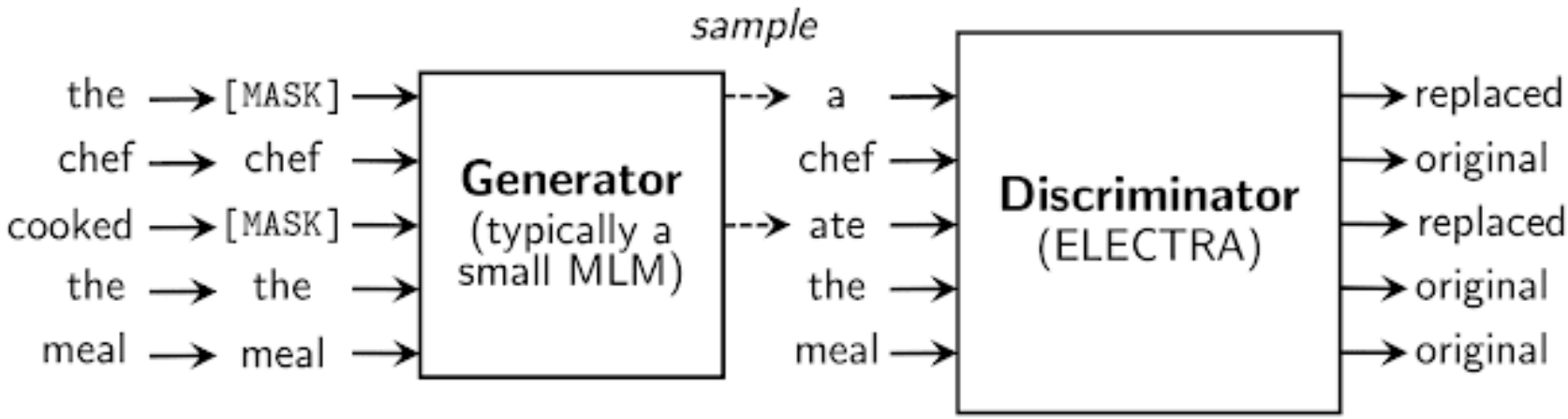


[11] https://ai.googleblog.com/2020/03/more-efficient-nlp-model-pre-training.html

ELECTRA[11]

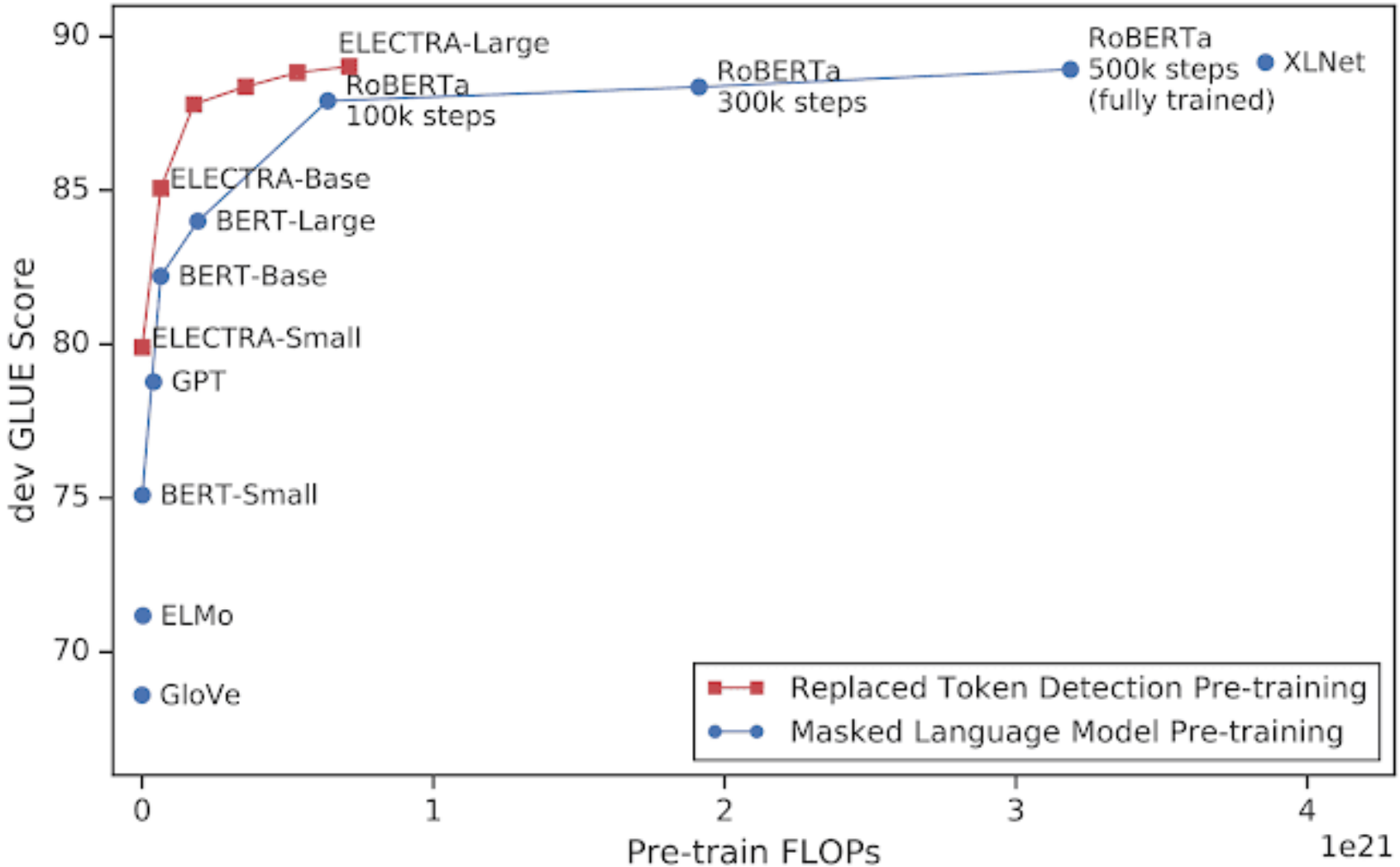
Восстанавливать замаскированные слова слишком просто, давайте искать замененные слова.





[11] https://ai.googleblog.com/2020/03/more-efficient-nlp-model-pre-training.html

ELECTRA


90
85
80
75
70
dev GLUE Score
ELECTRA-Large
RoBERTa
100k steps
RoBERTa
300k steps
RoBERTa
500k steps
(fully trained)
XLNet
ELECTRA-Base
BERT-Large
BERT-Base
ELECTRA-Small
GPT
BERT-Small
ELMo
GloVe
Replaced Token Detection Pre-training
Masked Language Model Pre-training
0
1
2
3
4
Pre-train FLOPs
1e21

DeBERTa[12]

For a token at position i in a sequence, we represent it using two vectors, $\{\boldsymbol{H_i}\}$ and $\{\boldsymbol{P_{i|j}}\}$, which represent its content and relative position with the token at position j, respectively. The calculation of the cross attention score between tokens i and j can be decomposed into four components as

$$\begin{aligned} A_{i,j} &= \{\boldsymbol{H_i}, \boldsymbol{P_{i|j}}\} \times \{\boldsymbol{H_j}, \boldsymbol{P_{j|i}}\}^\intercal \\ &= \boldsymbol{H_i H_j^\intercal} + \boldsymbol{H_i P_{j|i}^\intercal} + \boldsymbol{P_{i|j} H_j^\intercal} + \boldsymbol{P_{i|j} P_{j|i}^\intercal} \end{aligned} \tag{2}$$

That is, the attention weight of a word pair can be computed as a sum of four attention scores using disentangled matrices on their contents and positions as *content-to-content*, *content-to-position*, *position-to-content*, and *position-to-position* [2].

We can represent the disentangled self-attention with relative position bias as equation 4, where $\boldsymbol{Q_c}, \boldsymbol{K_c}$ and $\boldsymbol{V_c}$ are the projected content vectors generated using projection matrices $\boldsymbol{W_{q,c}}, \boldsymbol{W_{k,c}}, \boldsymbol{W_{v,c}} \in R^{d\times d}$ respectively, $\boldsymbol{P} \in R^{2k\times d}$ represents the relative position embedding vectors shared across all layers (i.e., staying fixed during forward propagation), and $\boldsymbol{Q_r}$ and $\boldsymbol{K_r}$ are projected relative position vectors generated using projection matrices $\boldsymbol{W_{q,r}}, \boldsymbol{W_{k,r}} \in R^{d\times d}$, respectively.

$$\boldsymbol{Q_c} = \boldsymbol{H W_{q,c}}, \boldsymbol{K_c} = \boldsymbol{H W_{k,c}}, \boldsymbol{V_c} = \boldsymbol{H W_{v,c}}, \boldsymbol{Q_r} = \boldsymbol{P W_{q,r}}, \boldsymbol{K_r} = \boldsymbol{P W_{k,r}}$$

$$\tilde{A}_{i,j} = \underbrace{\boldsymbol{Q_i^c K_j^{c\intercal}}}_{\text{(a) content-to-content}} + \underbrace{\boldsymbol{Q_i^c K_{\delta(i,j)}^{r}}^\intercal}_{\text{(b) content-to-position}} + \underbrace{\boldsymbol{K_j^c Q_{\delta(j,i)}^{r}}^\intercal}_{\text{(c) position-to-content}} \tag{4}$$

$$\boldsymbol{H_o} = \text{softmax}(\frac{\tilde{\boldsymbol{A}}}{\sqrt{3d}})\boldsymbol{V_c}$$

[12] https://arxiv.org/abs/2006.03654

За рамками лекции

- Различные виды BPE
- Distillation
- Reformer, Sparse Transformer, BigBird...
- Prompt crafting and tuning

Вопросы