

Лекция 10

Нейронные сети для обработки естественного языка

Храбров Кузьма

14 апреля 2023 г.



GPT-4 Unveiled by OpenAI, Aces Professional Exams Including Bar Exam, AP exams, and other Pre-university Exams

By Leo Wong  - Mar 14, 2023

GPT-3.5/ChatGPT

K

Придумай смешную историю для лекции про архитектуру трансформеров.



Однажды в галактике, далекой-далекой, на планете, где обитали только роботы, решили они построить новую школу. Роботы были умными и продвинутыми, поэтому они решили использовать самую передовую технологию - трансформеры!

Они создали огромный трансформер, который должен был превращаться в школу. Но в процессе разработки, у них возникла проблема: они забыли спроектировать двери! Роботы были в ужасе - как же ученики будут заходить в школу, если нет дверей?

Однако, они нашли выход из ситуации - они добавили в трансформер специальный режим "телеportации". Теперь ученики могли просто "телеportироваться" внутрь школы, не беспокоясь о дверях.

Когда они открыли новую школу, все роботы были в восторге! Но когда первые ученики попытались телепортироваться внутрь, произошло нечто невероятное - вместо того, чтобы попасть внутрь школы, они оказались на другой планете!

GPT-3.5/ChatGPT

K

нет, трансформер это архитектура нейронной сети, придумай историю про него.



Конечно, извините за недопонимание! Вот история про архитектуру трансформеров: 👍 👎

Однажды в компании, занимающейся искусственным интеллектом, работали два разработчика - Джон и Мэри. Они работали вместе над созданием новой нейронной сети, которая должна была распознавать и классифицировать различные объекты.

Джон был заинтересован в использовании сверточных нейронных сетей, а Мэри предложила использовать рекуррентные нейронные сети. Оба они имели свои аргументы в пользу своих вариантов.

Но однажды, когда они обсуждали этот вопрос, случилось нечто странное - они наткнулись на старую игрушку-трансформера, которая была забыта в углу их офиса. И тогда они подумали - почему бы не использовать идею трансформации в нейронной сети?

Так и родилась идея архитектуры трансформеров. Они создали нейронную сеть, которая могла использовать механизмы трансформации входных данных, чтобы получать более точные результаты. Они использовали технику многоканальной свертки и многоголовое внимание, чтобы улучшить точность классификации объектов.

Regenerate response

Представление слов

Задача

Сопоставить каждому слову w из словаря V вектор $e(w)$.

Подходы:

- ▶ One-hot encoding
- ▶ Counts
- ▶ CBOW
- ▶ Skip-grams

One-hot encoding

Кодируем слово w_i вектором $[0, 0, \dots, 0, \underbrace{1}_i, 0, \dots, 0]^T$ Плюсы:

- ▶ Просто реализовать
- ▶ Можно использовать разреженное представление

Минусы:

- ▶ Не учитывает близость слов
- ▶ Огромная размерность

Counts

... and the cute **kitten** purred and then ...

... the cute **furry** cat purred and miaowed ...

... that small **kitten** miaowed and she ...

... the loud **furry** dog ran and bit ...

Словарь: bit, cute, furry, loud, miaowed, purred, ran, small

kitten: cute, purred, small, miaowed $\Rightarrow [0, 1, 0, 0, 1, 1, 0, 1]^T$

cat: cute, furry, miaowed $\Rightarrow [0, 1, 1, 0, 1, 0, 0, 0]^T$

dog: loud, furry, ran, bit $\Rightarrow [1, 0, 1, 1, 0, 0, 1, 0]^T$

$$sim(w_1, w_2) = \frac{\langle w_1, w_2 \rangle}{\|w_1\| \cdot \|w_2\|}$$

Embedding matrix

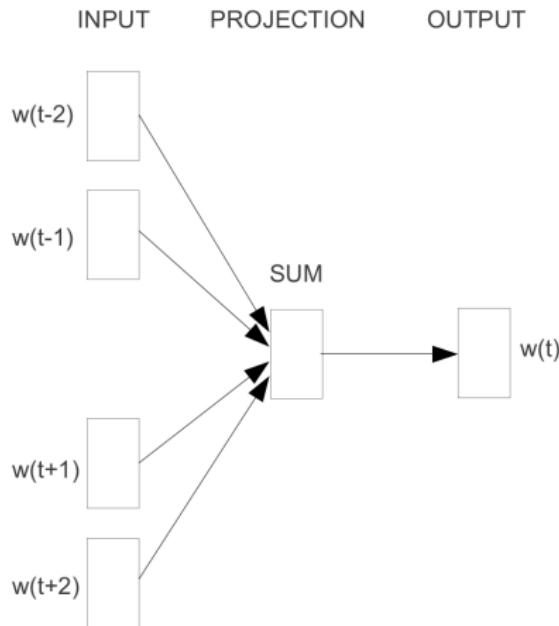
Матрица представлений:

$$E = \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_{|V|} \end{bmatrix}$$

Каждая строка — представление одного слова.

Идея: обучим матрицу E при помощи нейронной сети.

Continuous bag of words



Предсказываем пропущенное слово по контексту.

Представление слова:

$$h_t = W \sum_{w \in \text{context}(w_t)} \text{one_hot}(w)$$

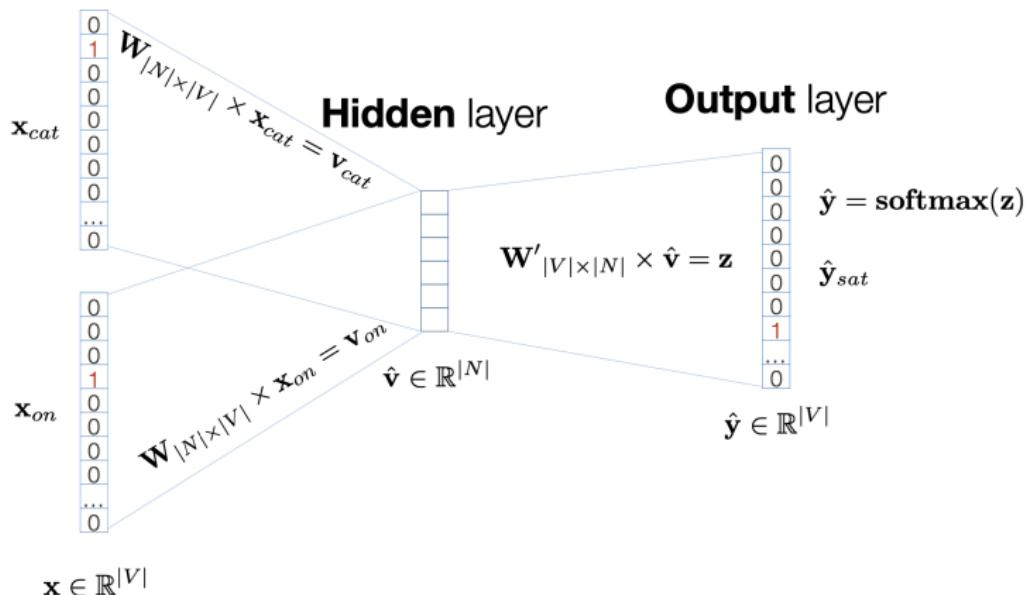
$$P(w_i | \text{context}(w_i)) = \text{softmax}(W' h) [i]$$

Функция потерь:

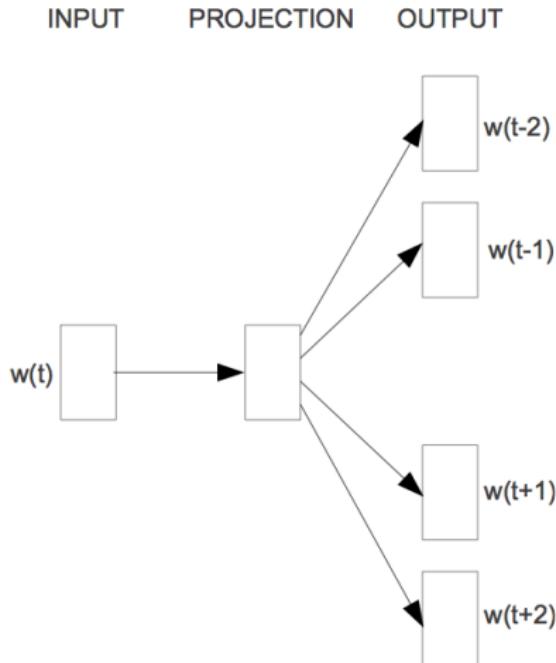
$$L = -\log P(w_i | \text{context}(w_i))$$

Continuous bag of words. Details

Input layer

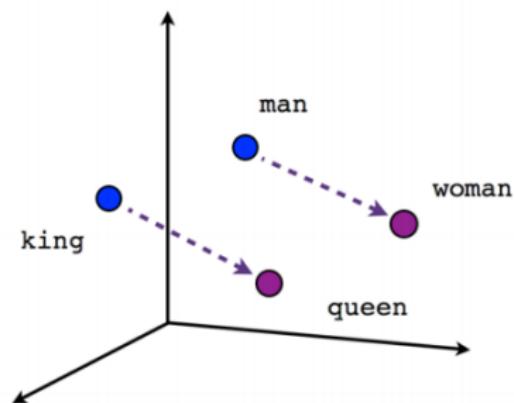


Skip-gram



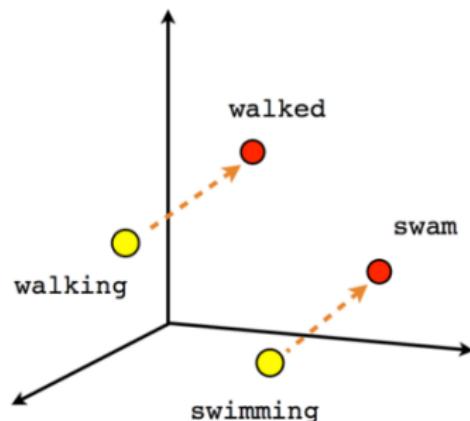
Предсказываем контекст по пропущенному слову

word2vec: арифметика



Male-Female

$$e(\text{«king»}) - e(\text{«man»}) + e(\text{«woman»}) \simeq e(\text{«queen»})$$
$$e(\text{«swimming»}) + e(\text{«walked»}) - e(\text{«walking»}) \simeq e(\text{«swam»})$$



Verb tense

Токенизация подслов

Три общих алгоритма:

- ▶ Кодирование пар байтов (BPE) (Sennrich et al., 2016)
- ▶ Unigram Language Model (ULM) (Taku Kudo, 2018
<https://arxiv.org/pdf/1804.10959.pdf>)
- ▶ WordPiece (Schuster, Nakajima
<https://static.googleusercontent.com/media/research.google.com/ja/pubs/archive/37842.pdf>)

Все имеют 2 части:

- ▶ Выделение токенов: необработанный корпус -> словарь (набор токенов).
- ▶ Сегментатор маркеров, который берет необработанное тестовое предложение и разбивает его в соответствии с этим словарем.

Byte Pair Encoding. Идеи¹

Исходный словарь - набор отдельных символов А, Б, В, Г,...,а, б, в, Г.....

Алгоритм получения словаря:

Повторять пока не было сделано k слияний:

- ▶ выбрать два символа, которые чаще всего соседствуют в обучающем корпусе (скажем, «А», «Б»),
- ▶ добавляем в словарь новый объединенный символ «АБ» и заменяет все соседние «А» и «Б» в корпусе на «АБ».

На тестовых данных превращаем буквы в токены из словаря:

- ▶ Жадно
- ▶ В том порядке, в котором токены были получены во время построения словаря

¹<https://arxiv.org/abs/1508.07909>

Byte Pair Encoding.

Byte Pair Encoding Data Compression Example

aaabdaaabac

aaabdaaabac Replace Z = aa

ZabdZabac Replace Y = ab

ZYdZYac Replace X = ZY

Xdxac Final compressed string

Replacement Table

Byte pair	Replacement
X	ZY
ab	Y
aa	Z

Задачи NLP

- ▶ Информационный поиск (Information retrieval)
- ▶ Классификация текстов
- ▶ Генерация текста
- ▶ Диалоговые системы

Задачи NLP

- ▶ Детекция спама / Классификация тематики / Анализ тональности
- ▶ Распознавание именованных сущностей (Named entity recognition, NER)
- ▶ Relation extraction
- ▶ Parts of speech tagging
- ▶ Ранжирование
- ▶ Суммаризация
- ▶ Ответы на вопросы
- ▶ "Понимание прочитанного"(Reading comprehension)
- ▶ Speech-to-text / Text-to-speech
- ▶ Image-to-text / Text-to-image

Question answering

mary got the milk there
john moved to the bedroom
sandra went back to the kitchen
mary travelled to the hallway
john got the football there
john went to the hallway
john put down the football
mary went to the garden
john went to the kitchen
sandra travelled to the hallway
daniel went to the hallway
mary discarded the milk
where is the milk ?

answer: garden

Visual QA²

Who is wearing glasses?

man



woman

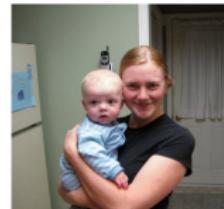


Where is the child sitting?

fridge



arms



Is the umbrella upside down?

yes



no



How many children are in the bed?

2



1



²Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering (CVPR 2017)

Visual QA³

Answer: No



Answer: Yes



complementary scenes



Tuple: <girl, walking, bike>

Question: Is the girl walking the bike?

³Yin and Yang: Balancing and Answering Binary Visual Questions (CVPR 2016)

Анализ тональности

Отзыв положительный или отрицательный?

у=1 Мне очень понравился этот фильм. Никогда раньше ничего подобного не видел!!!)))

у=0 Ужасно! Ушла с середины фильма, тк больше невозможно было смотреть.

у=0 Ну да, конечно. Просто отличный фильм.

Анализ тональности

Отзыв положительный или отрицательный?

$y=1$ Мне очень понравился этот фильм. Никогда раньше ничего подобного не видел!!!))

$y=0$ Ужасно! Ушла с середины фильма, тк больше невозможно было смотреть.

$y=0$ Ну да, конечно. Просто отличный фильм.

Простой подход: Bag-of-words + Logistic regression Какие есть проблемы у такого подхода?

Анализ тональности

Отзыв положительный или отрицательный?

$y=1$ Мне очень понравился этот фильм. Никогда раньше ничего подобного не видел!!!))

$y=0$ Ужасно! Ушла с середины фильма, тк больше невозможно было смотреть.

$y=0$ Ну да, конечно. Просто отличный фильм.

Простой подход: Bag-of-words + Logistic regression Какие есть проблемы у такого подхода?

- ▶ Не учитывает сарказм
- ▶ Не учитывает схожесть слов (например, кот \leftrightarrow котенок)
- ▶ Не учитывает порядок слов

Пример: SQuAD

Rank	Model	EM	F1
	Human Performance <i>Stanford University</i> (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) <i>RICOH_SRCB_DML</i>	90.939	93.214
2 Feb 21, 2021	FPNet (ensemble) <i>Ant Service Intelligence Team</i>	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) <i>RICOH_SRCB_DML</i>	90.860	93.100
4 Apr 06, 2020	SA-Net on Albert (ensemble) <i>QIANXIN</i>	90.724	93.011
5 May 05, 2020	SA-Net-V2 (ensemble) <i>QIANXIN</i>	90.679	92.948
5 Apr 05, 2020	Retro-Reader (ensemble) <i>Shanghai Jiao Tong University</i> http://arxiv.org/abs/2001.09694	90.578	92.978

Пример: SQuAD

P6418 Термин Computer science (Компьютерная наука) появился в 1959 году в научном журнале Communications of the ACM, в котором Луи Файн (Louis Fein) ратовал за создание Graduate School in Computer Sciences (Высшей школы в области информатики) . . . Усилия Луи Фейна, численного аналитика Джорджа Форсайта и других увенчались успехом: университеты пошли на создание программ, связанных с информатикой, начиная с Университета Пердью в 1962.

P6418 The term "computer science" appears in a 1959 article in Communications of the ACM, in which Louis Fein argues for the creation of a Graduate School in Computer Science . . . Louis Fein's efforts, and those of others such as numerical analyst George Forsythe, were rewarded: universities went on to create such departments, starting with Purdue in 1962.

Q11870 Когда впервые был применен термин Computer science (Компьютерная наука)?

Q11870 When did the term "computer science" appear?

Q28900 Кто впервые использовал этот термин?

Q28900 Who was the first to use this term?

Q30330 Начиная с каого учебного заведения стали применяться учебные программы, связанные с информатикой?

Q30330 Starting with wich university were computer science programs created?

Современные задачи NLP. SuperGlue⁴⁵

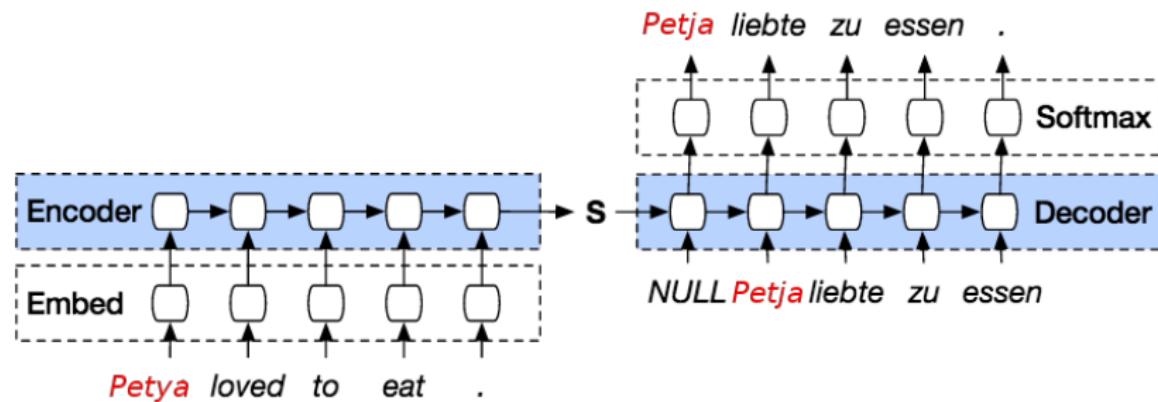
Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
1	JDExplore d-team	Vega v2		91.3	90.5	98.6/99.2	99.4	88.2/62.4	94.4/93.9	96.0	77.4	98.6	-0.4	100.0/50.0
+	2 Liam Fedus	ST-MoE-32B		91.2	92.4	96.9/98.0	99.2	89.6/65.8	95.1/94.4	93.5	77.7	96.6	72.3	96.1/94.1
3	Microsoft Alexander v-team	Turing NLR v5		90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3	67.8	93.3/95.5
4	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
5	Yi Tay	PaLM 540B		90.4	91.9	94.4/96.0	99.0	88.7/63.6	94.2/93.3	94.1	77.4	95.9	72.9	95.5/90.4
+	6 Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
+	7 DeBERTa Team - Microsoft	DeBERTa / TuringNLv4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
8	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
+	9 T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9

⁴<https://super.gluebenchmark.com/leaderboard>

⁵<https://russiansuperglue.com>

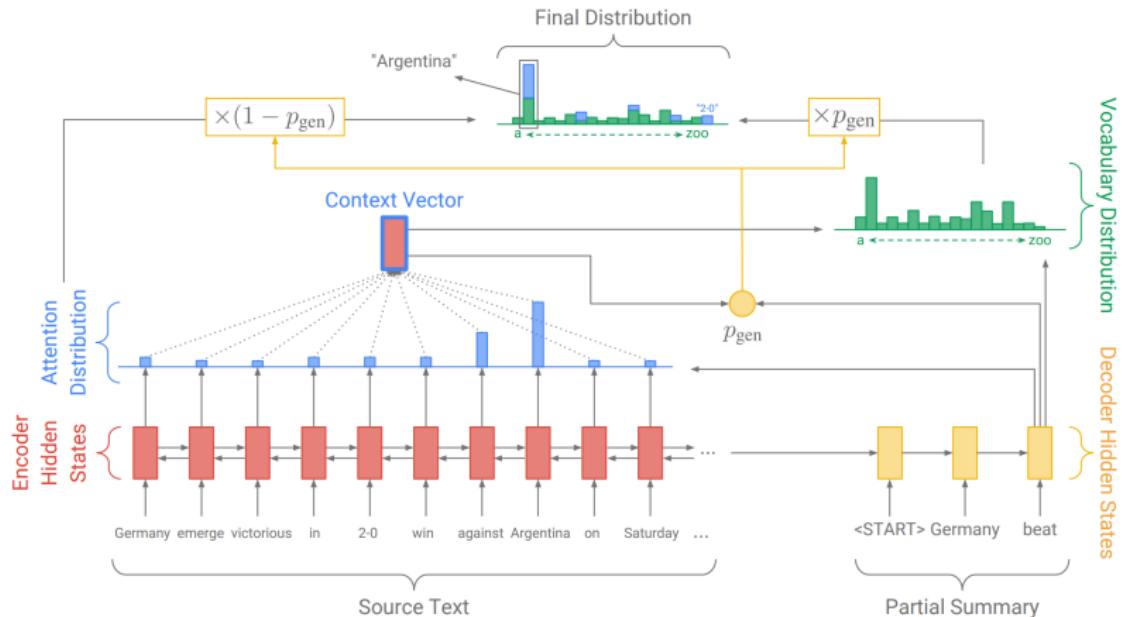
RNN. Seq2Seq

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



Attention⁶

Добавим Attention.



⁶<https://arxiv.org/abs/1409.0473>

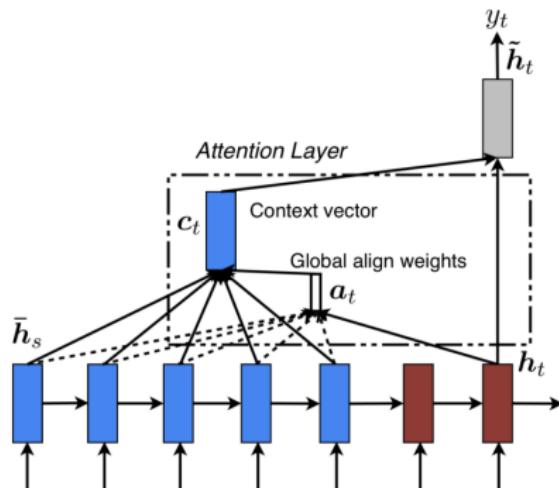
Attention

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

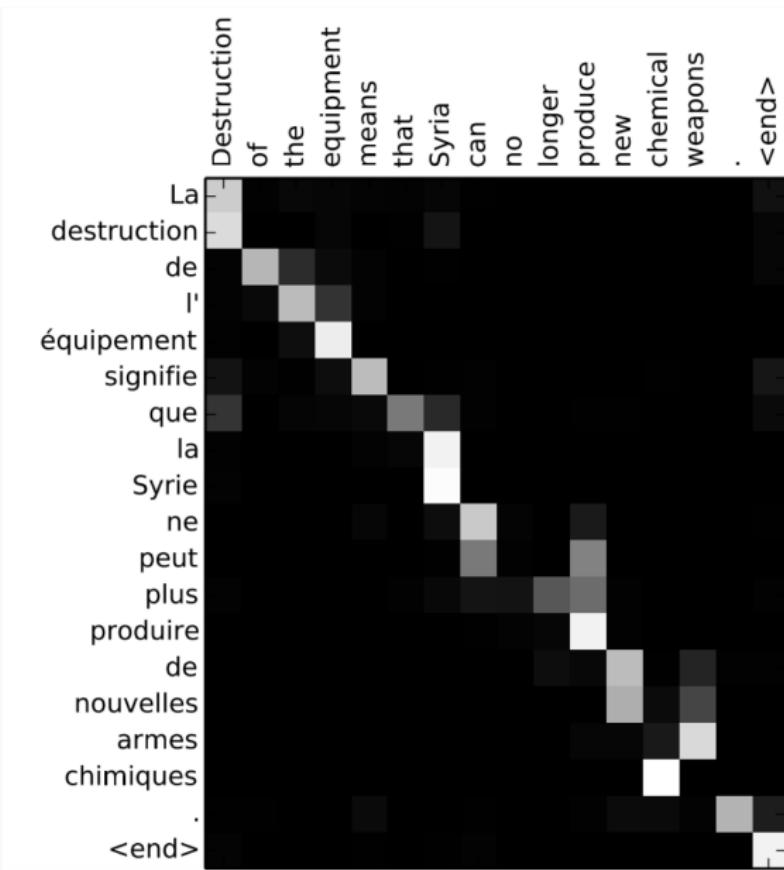
$$\text{attention_score}_{ij} = \text{softmax}_j(\text{attention}(\text{decoder}_i, \text{encoder_output}_j))$$

$$\text{context}_i = \sum_j \text{attention_score}_j \cdot \text{encoder_output}_j$$

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



Attention



Attention to images

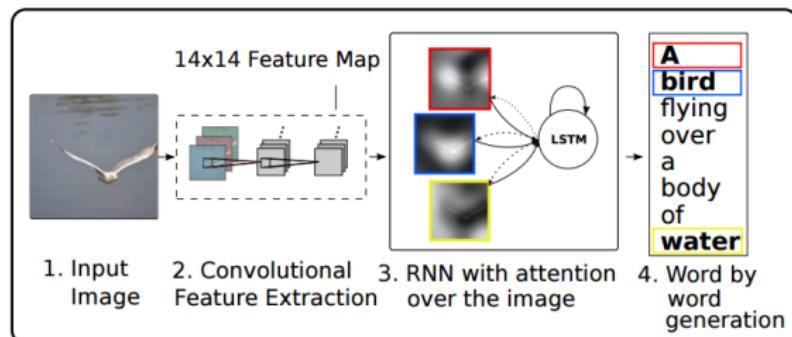
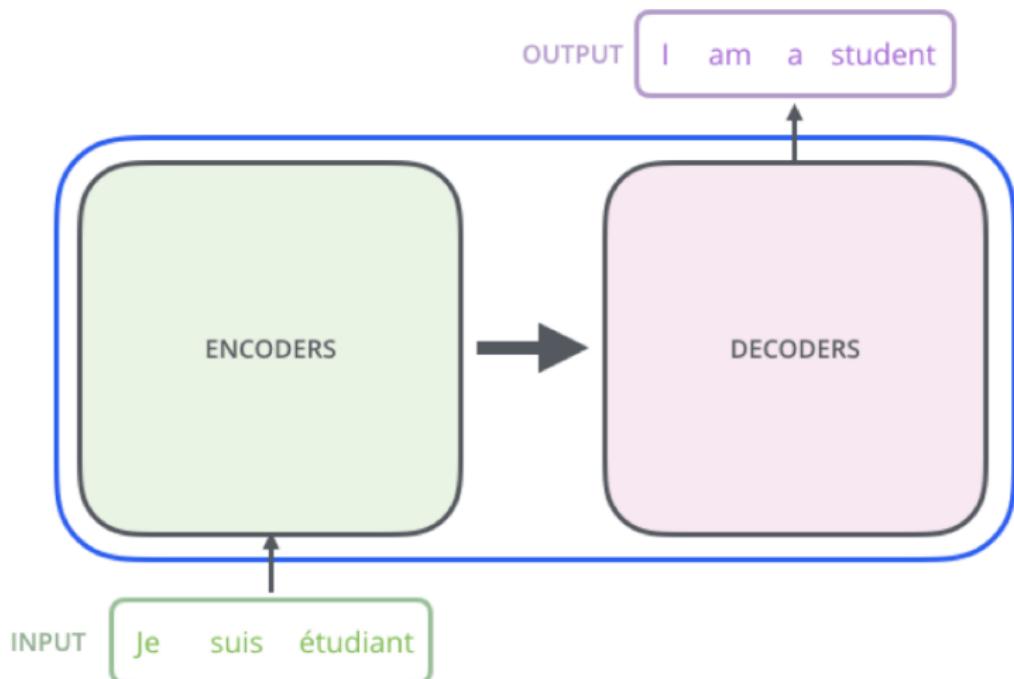


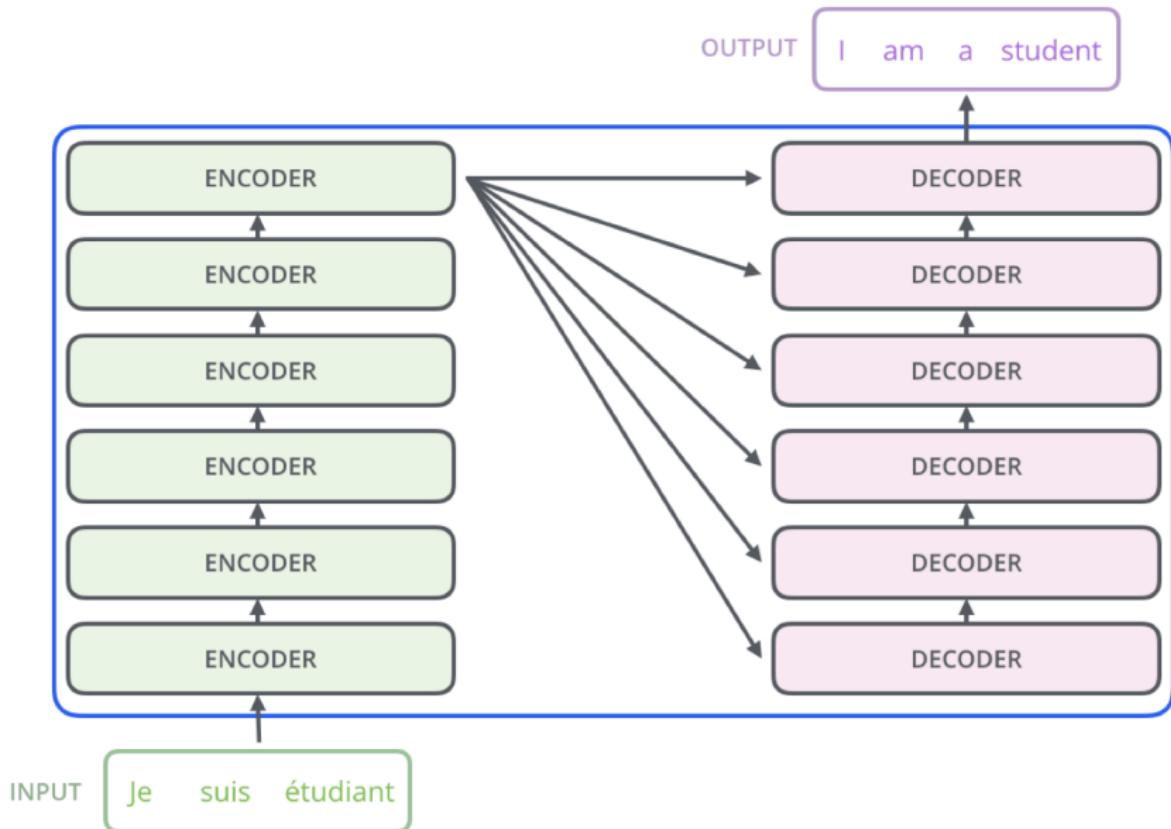
Figure 3. Examples of attending to the correct object (white indicates the attended regions, *underlines* indicated the corresponding word)



Transformer (Attention is all you need)



Transformer (Attention is all you need)



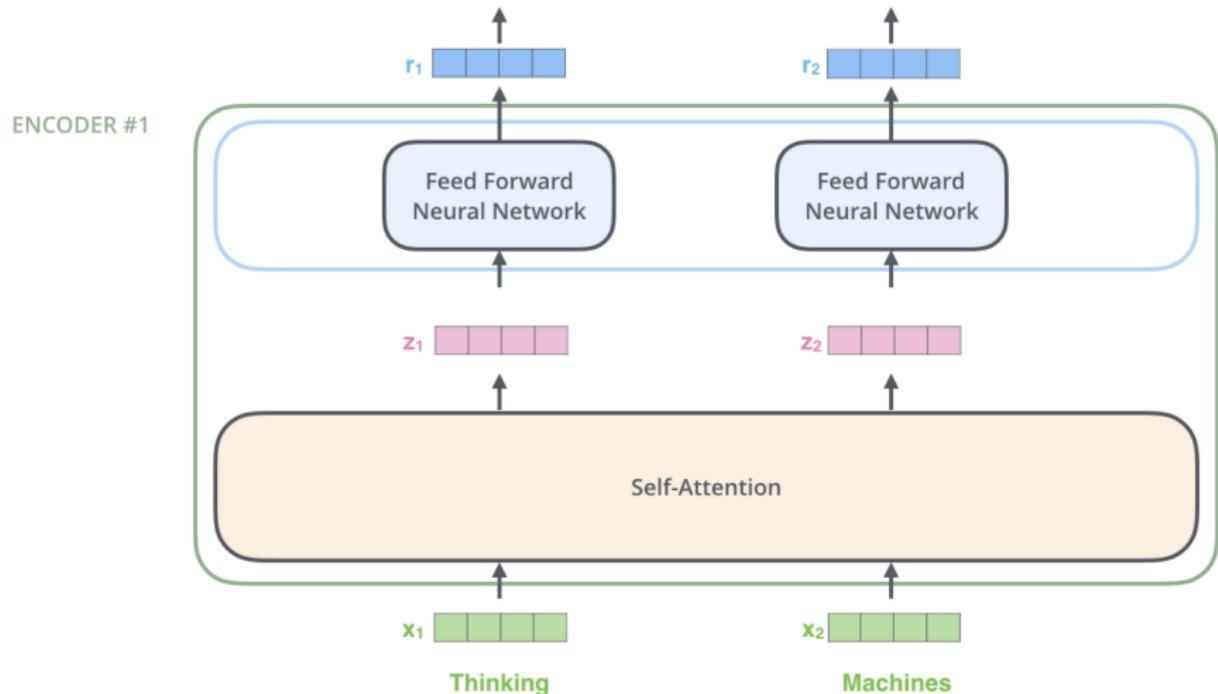
Transformer (Attention is all you need). Embeddings

Последовательность слов, подаваемая в трансформер сначала разбивается на токены (слова, наборы букв или из подслов(BPE, SentencePiece, WordPiece, etc.)), которые подаются в Embedding слой. Далее для учета порядка к полученным векторам добавляется некоторое смещение, зависящее от номера токена(Positional Encoding).

$$\text{PE}_{(\text{pos}, 2i)} = \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right),$$

$$\text{PE}_{(\text{pos}, 2i + 1)} = \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right),$$

Transformer (Attention is all you need)



Transformer (Attention is all you need). Self-attention



$$\bar{x}_i \rightarrow \bar{q}_i = W^Q x_i$$
$$\bar{x}_i \rightarrow \bar{k}_i = W^K x_i$$
$$\bar{x}_i \rightarrow \bar{w}_i = W^V x_i$$

$$Z = \text{softmax} \left(\frac{1}{\sqrt{d}} Q K^T \right) V$$

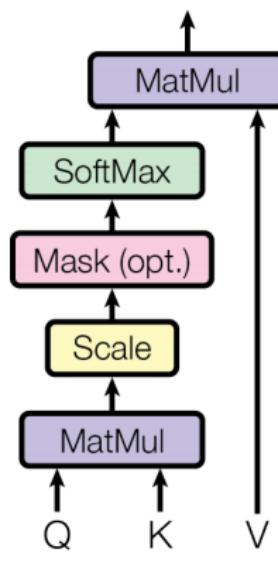
$$z_i = \sum_j a_{ij} v_j, \quad a_{ij} = \text{softmax} \left(\frac{1}{\sqrt{d}} \langle q_i, k_j \rangle \right)$$

Attention (Attention is all you need). MultiHead

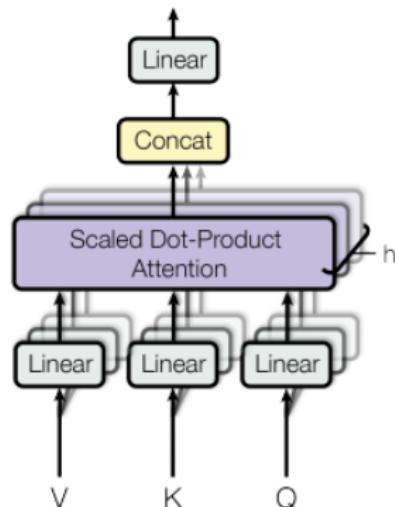
Одна голова: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$.

Параллелим: $\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$
Где $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

One Head Attention

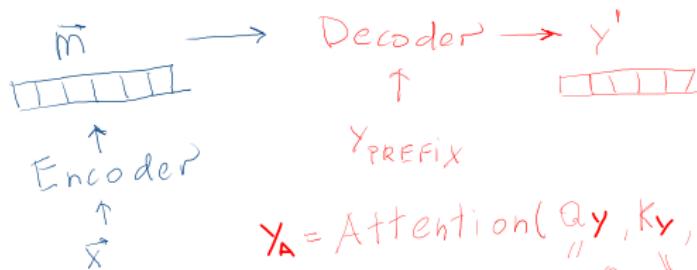


Multi-Head Attention



Transformer (Attention is all you need). Decoder

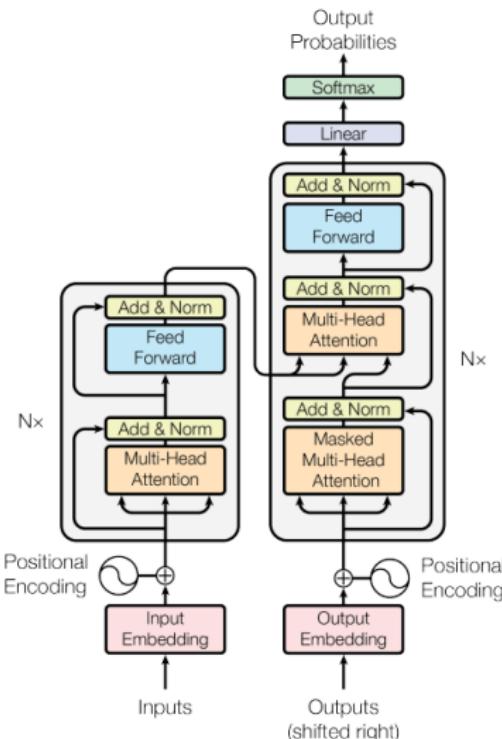
Decoder по сути устроен так же как энкодер, однако на вход в Decoder подается последний выход в Encoder (m) + начало сгенерированной последовательности y_{prefix} . Также в Decoder-е используется Masked Self-Attention: мы запрещаем предыдущим токенам смотреть на следующие.



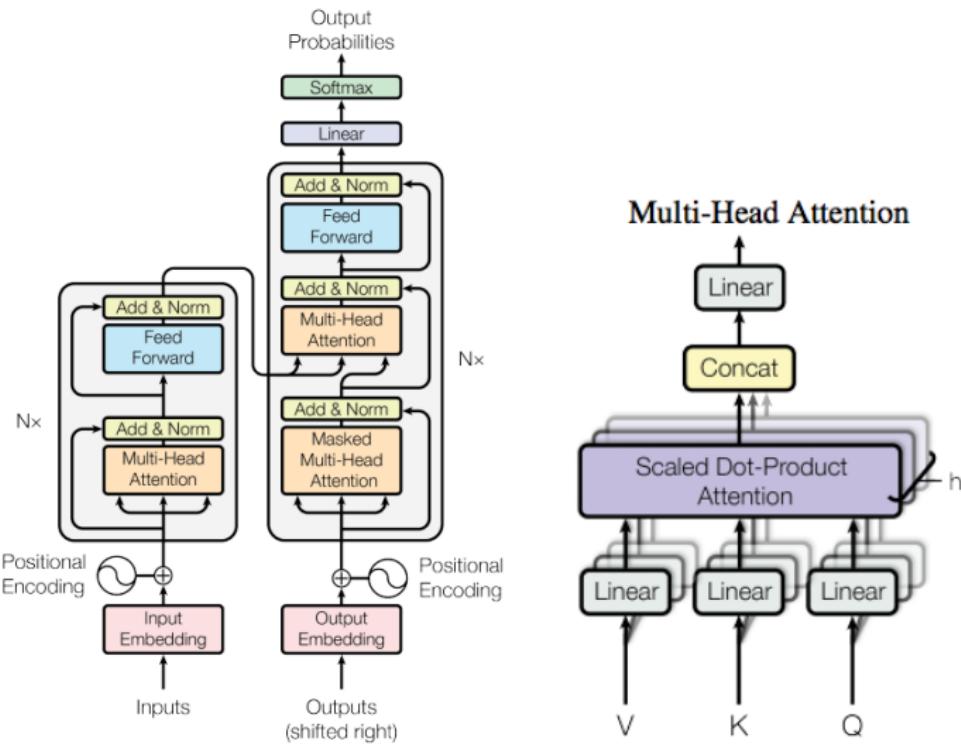
$$y_A = \text{Attention}(Qy, Ky, Vy) \cdot W_o$$
$$y^{W_Q} \quad y^{W^k} \quad y^{W^v}$$

$$y' = \text{Attention}(Qy_A, Km, Vm) \cdot W_o$$
$$y_A^{W_Q} \quad m^{W^k} \quad m^{W^v}$$

Transformer (Attention is all you need)



Transformer (Attention is all you need)



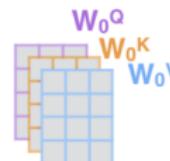
Transformer (Attention is all you need). Full architecture

1) This is our input sentence*
2) We embed each word*

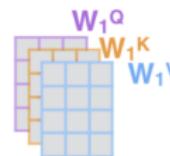
3) Split into 8 heads.
We multiply X or R with weight matrices

4) Calculate attention using the resulting $Q/K/V$ matrices

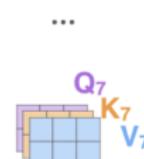
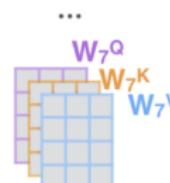
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^o to produce the output of the layer



W^o



* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



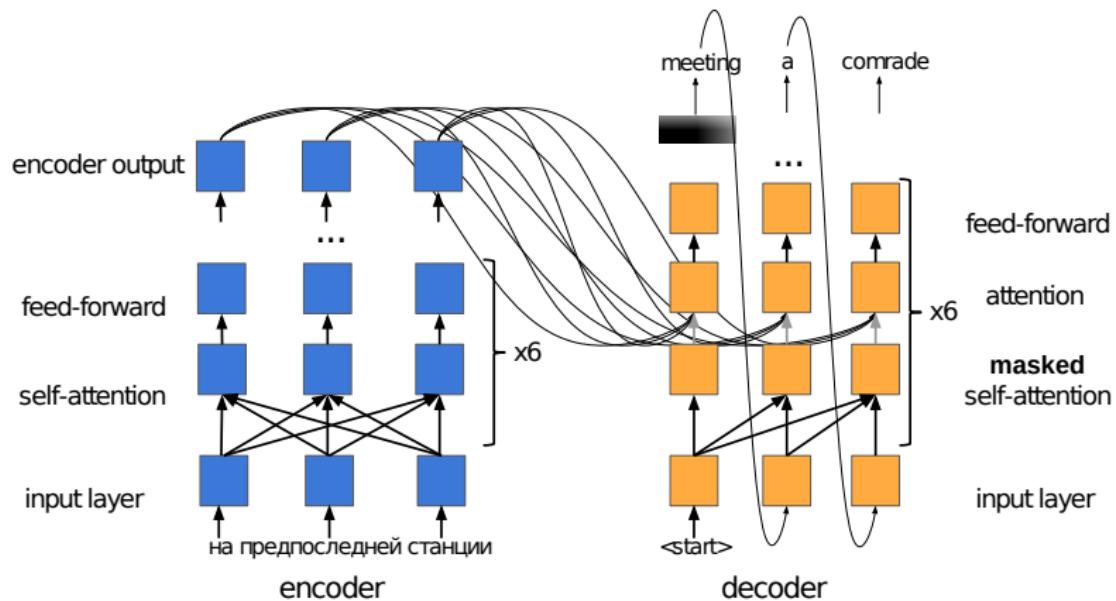
...

...

...

Transformer (Attention is all you need)⁷

Визуализация (gif)



⁷<https://arxiv.org/abs/1706.03762>

Transformer (Attention is all you need)

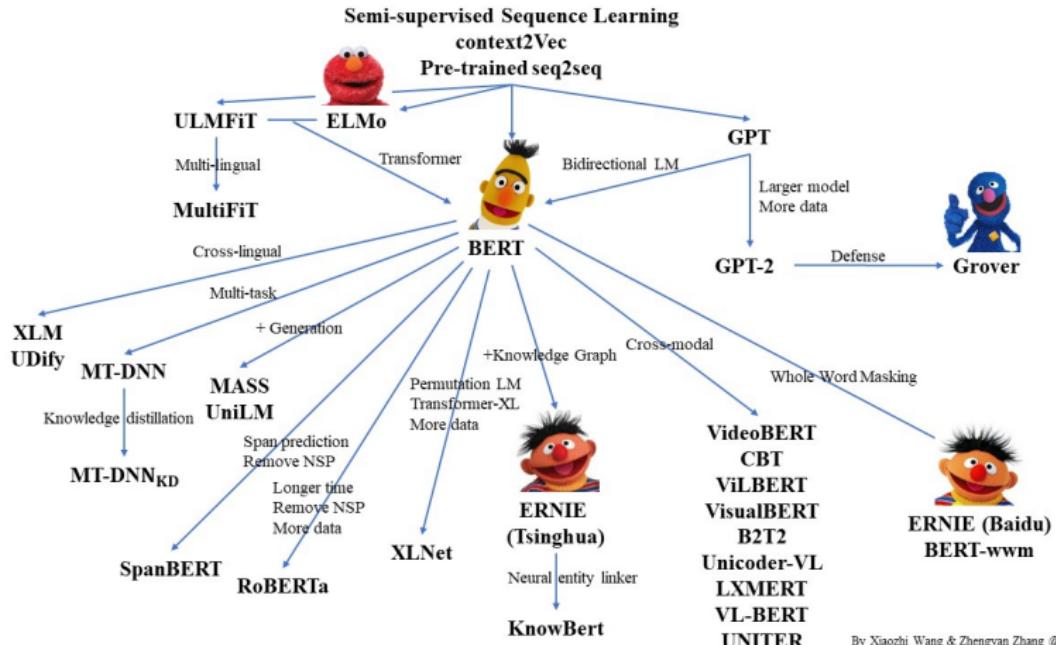
Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

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



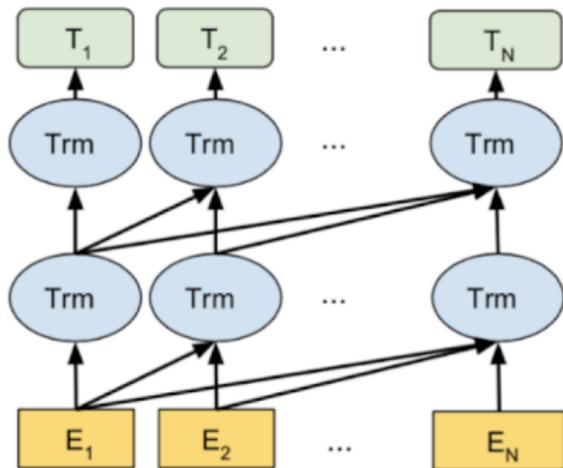
GPT⁸

Language model:

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

В случае GPT используются слои трансформера.

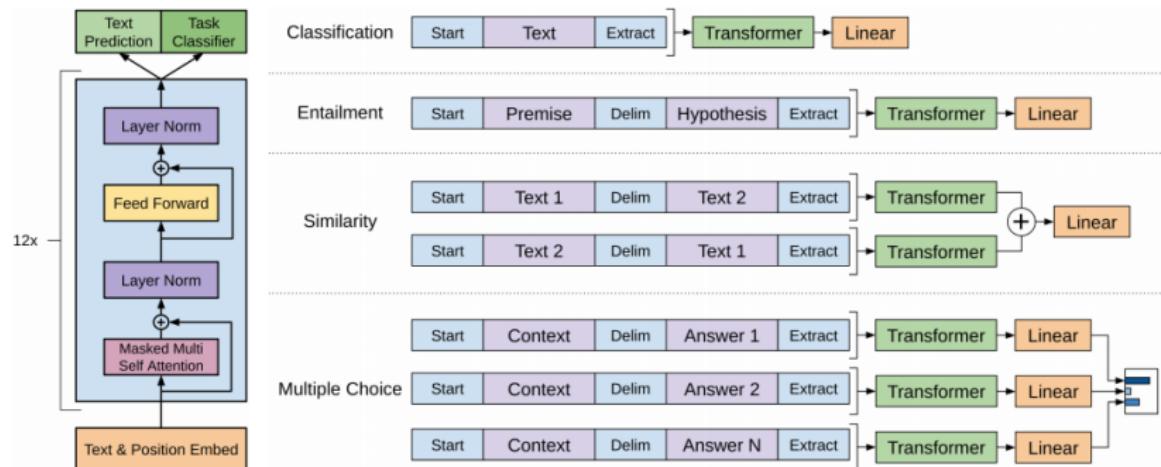
Также для ускорения обучения мы можем за один проход учить сразу несколько предсказаний за счет масок. (каждая позиция может "смотреть" только назад)



⁸https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Multitask pretraining

Предобучаем модель на огромном корпусе →
fine-tune на конкретную задачу.



GPT-2⁹ (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.

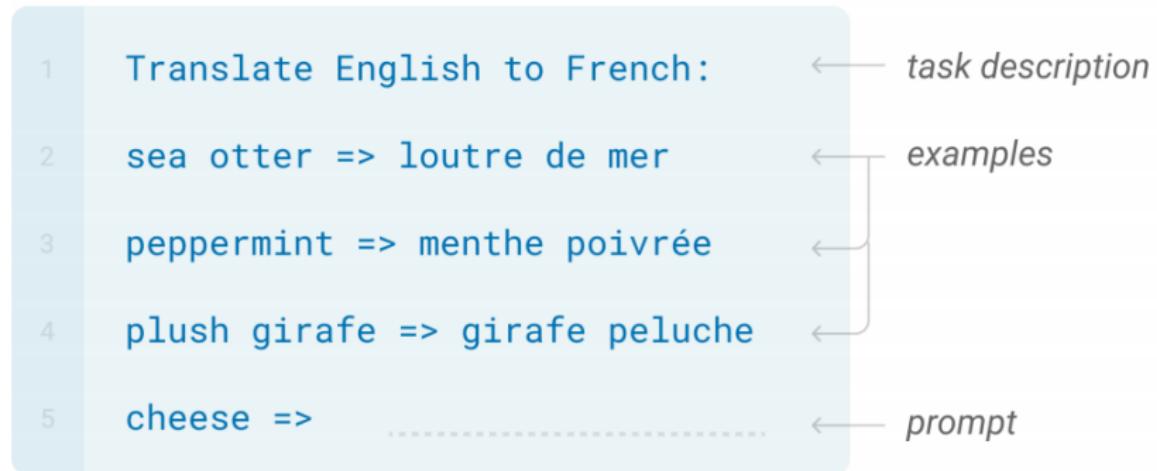
"The safety of people, the environment and the nation's nuclear stockpile is our highest priority," Hicks said. "We will get to the bottom of this and make no excuses."

⁹https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-3¹⁰ (zero-shot learning!)

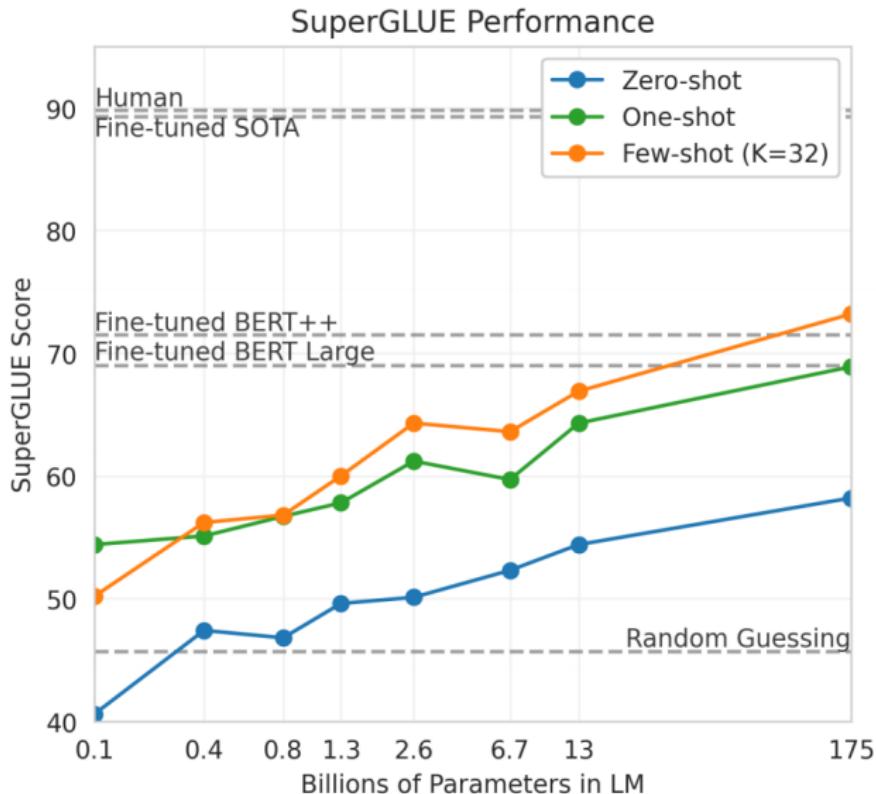
Few-shot

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

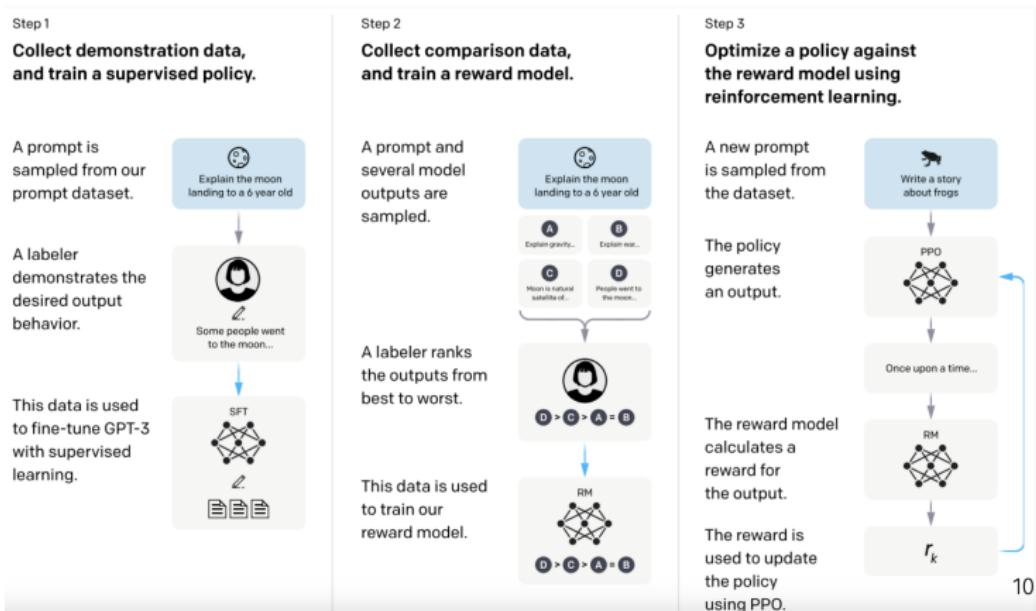


¹⁰<https://arxiv.org/abs/2005.14165>

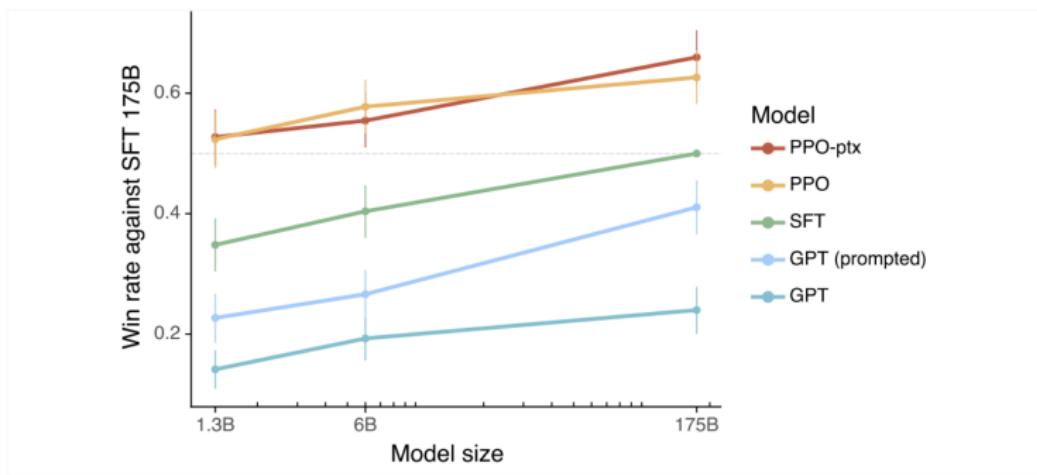
GPT-3 (zero-shot learning!)



GPT-3.5/ChatGPT



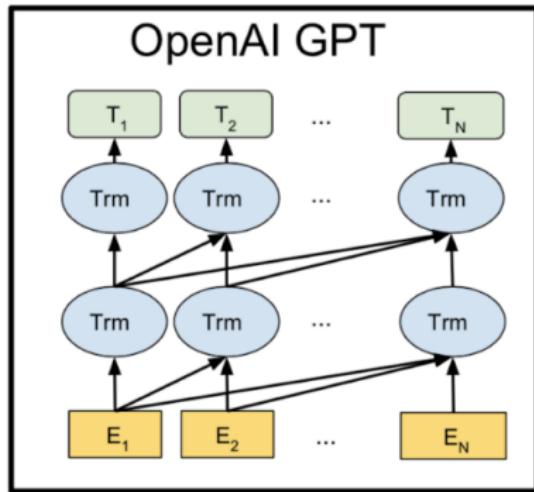
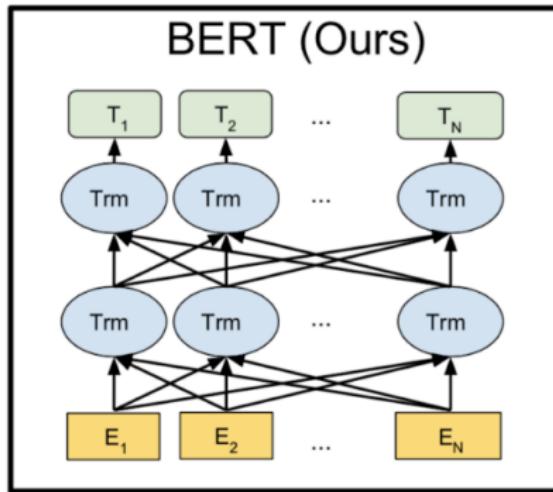
GPT-3.5/ChatGPT



GPT-3.5/ChatGPT

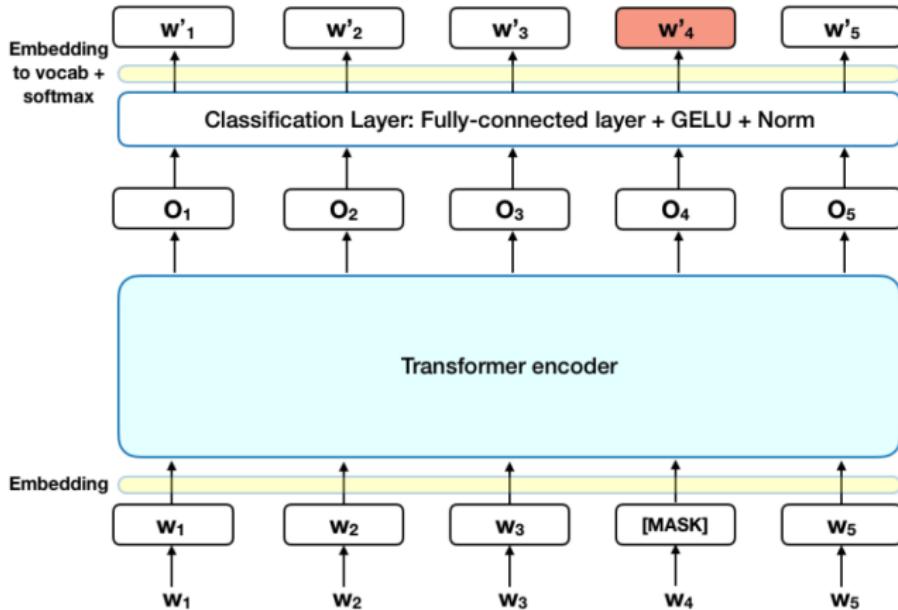
USER	help me write a short note to introduce myself to my neighbor
CHATGPT	Hi there, My name is [Your Name] and I just moved in next door. I wanted to reach out and introduce myself. I'm looking forward to getting to know you and the neighborhood. If you ever need anything or just want to chat, please don't hesitate to stop by or give me a call. Best, [Your Name]
USER	can you make it more formal?
CHATGPT	Dear Neighbor, I am writing to introduce myself. My name is [Your Name] and I have recently moved in next door. I wanted to take a moment to reach out and say hello. I am looking forward to getting to know you and the neighborhood. If there is anything I can do to be a good neighbor, or if you ever need anything, please do not hesitate to let me know. I am happy to help in any way that I can. Sincerely, [Your Name]
USER	can you send it for me?
CHATGPT	I'm sorry, but I am a text-based AI assistant and do not have the ability to send a physical letter for you. I can only provide suggestions and advice on what to write. It is up to you to take that advice and send the letter yourself.

BERT vs GPT



BERT¹¹

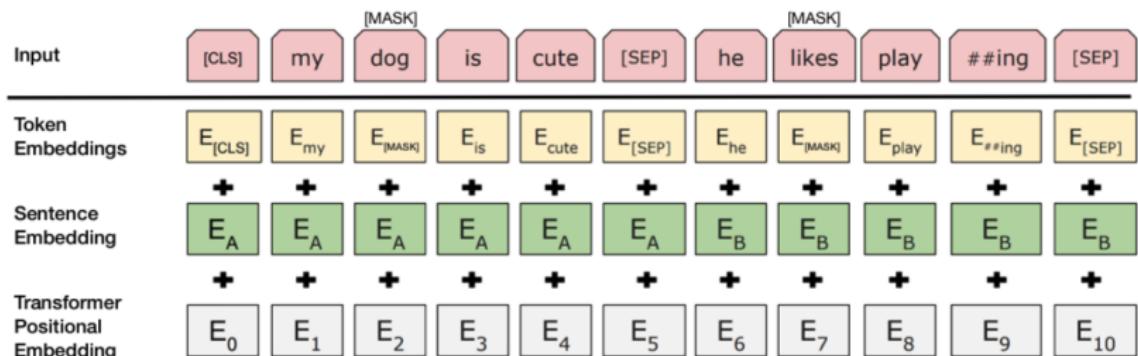
Bidirectional Encoder Representations from Transformers



¹¹<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 В следует за А. (берем В 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				
BERT _{BASE} (Single)	80.8	88.5	-	-
BERT _{LARGE} (Single)	84.1	90.9	-	-
BERT _{LARGE} (Ensemble)	85.8	91.8	-	-
BERT _{LARGE} (Sgl.+TriviaQA)	84.2	91.1	85.1	91.8
BERT _{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
BERT _{BASE}	81.6	-
BERT _{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) 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

RoBERTa¹²

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

¹²<https://arxiv.org/abs/1907.11692>

RoBERTa¹²

"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.

¹²<https://arxiv.org/abs/1907.11692>

ALBERT¹³

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

¹³<https://arxiv.org/abs/1909.11942>

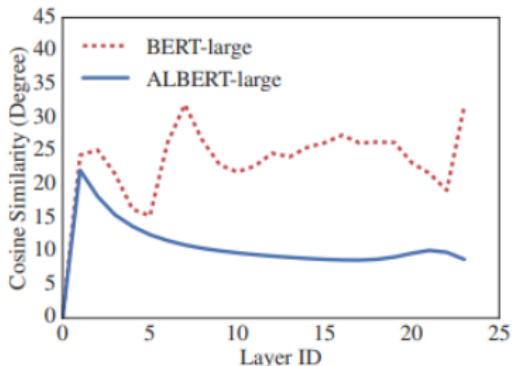
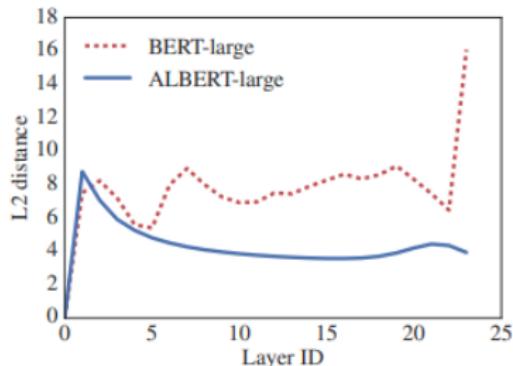
У BERT слишком много параметров ($BERT_{large} \approx 334M$), на самом деле, столько не надо.

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".

¹³<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).

XLNet¹⁴

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

¹⁴<https://arxiv.org/abs/1906.08237>

XLNet¹⁴

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:

[Разводные₁, мосты₂, в₃, Санкт-4, Петербурге₅].

Перемешаем слова: [в₃, Разводные₁, Петербурге₅, Санкт-4, мосты₅].

Хотим предсказать:

$$P(\text{Разводные}_1 | \text{в}_3), P(\text{Санкт-4} | \text{Разводные}_1, \text{в}_3, \text{Петербурге}_5)$$

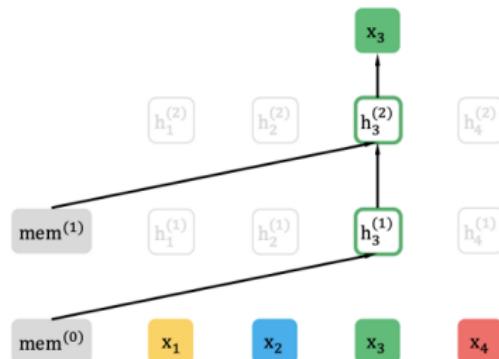
¹⁴<https://arxiv.org/abs/1906.08237>

XLNet

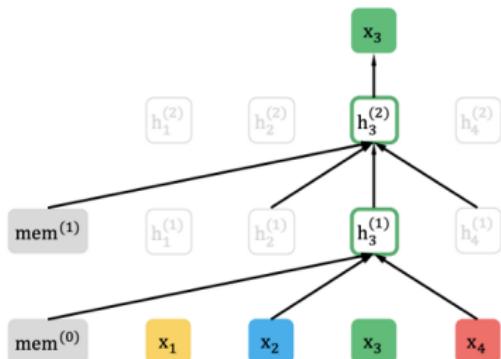
Моделируем:

1. GPT - $P(x_t | x_{i < t})$
2. BERT - $P(x_t | x_{i \notin \{t, m_1, \dots\}})$
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.



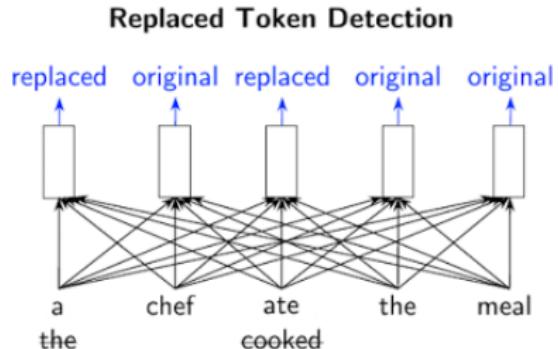
Factorization order: $3 \rightarrow 2 \rightarrow 4 \rightarrow 1$



Factorization order: $2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

ELECTRA¹⁵

Восстанавливать
замаскированные слова слишком
просто, давайте искать
замененные слова.

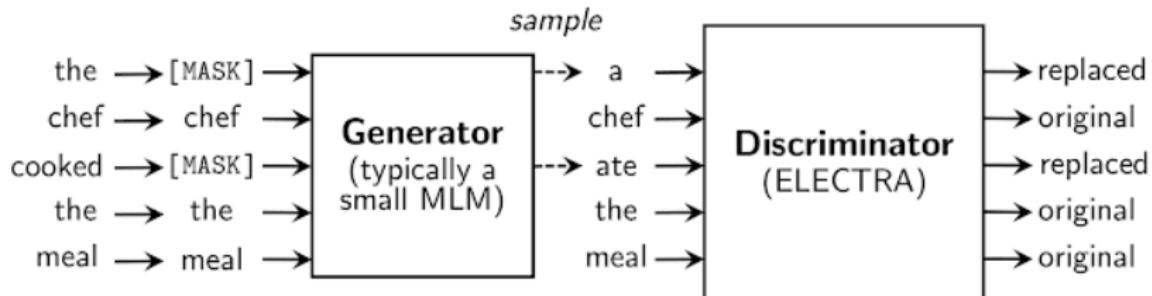
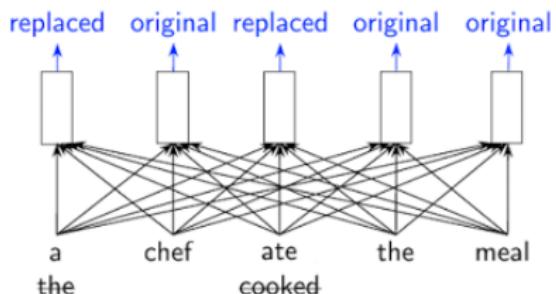


¹⁵<https://ai.googleblog.com/2020/03/more-efficient-nlp-model-pre-training.html>

ELECTRA¹⁵

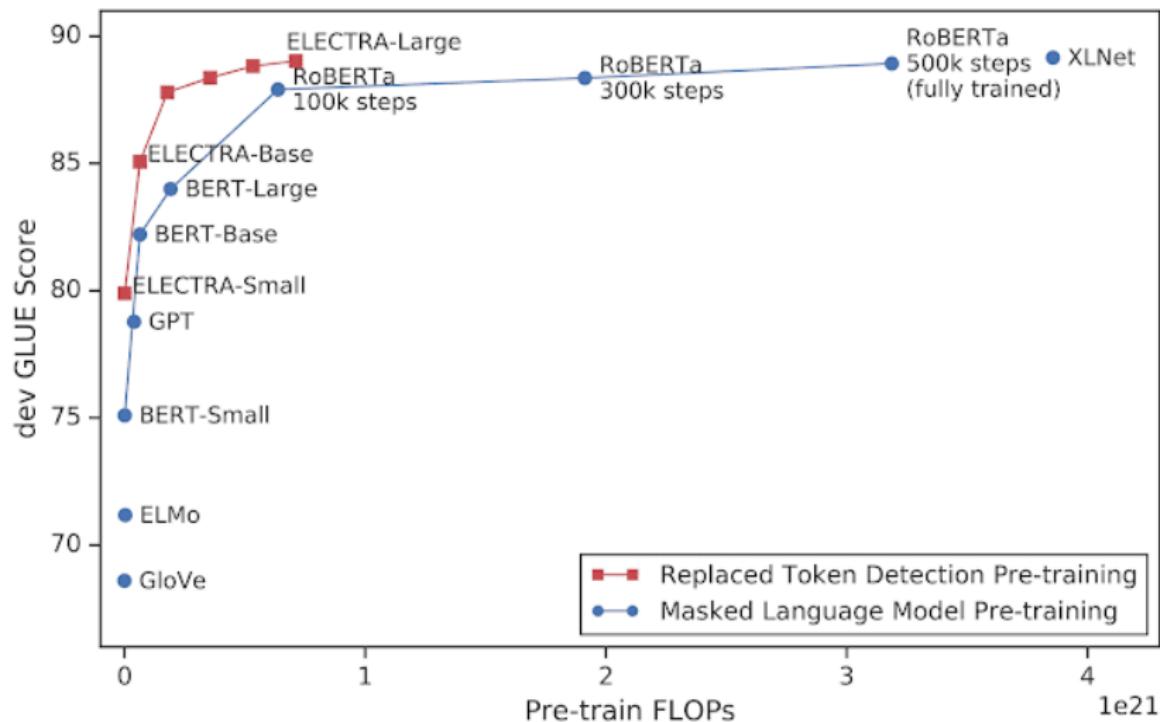
Восстанавливать замаскированные слова слишком просто, давайте искать замененные слова.

Replaced Token Detection



¹⁵<https://ai.googleblog.com/2020/03/more-efficient-nlp-model-pre-training.html>

ELECTRA



DeBERTa¹⁶

For a token at position i in a sequence, we represent it using two vectors, $\{\mathbf{H}_i\}$ and $\{\mathbf{P}_{i|j}\}$, 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

$$\begin{aligned} A_{i,j} &= \{\mathbf{H}_i, \mathbf{P}_{i|j}\} \times \{\mathbf{H}_j, \mathbf{P}_{j|i}\}^\top \\ &= \mathbf{H}_i \mathbf{H}_j^\top + \mathbf{H}_i \mathbf{P}_{j|i}^\top + \mathbf{P}_{i|j} \mathbf{H}_j^\top + \mathbf{P}_{i|j} \mathbf{P}_{j|i}^\top \end{aligned} \quad (2)$$

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 $\mathbf{Q}_c, \mathbf{K}_c$ and \mathbf{V}_c are the projected content vectors generated using projection matrices $\mathbf{W}_{q,c}, \mathbf{W}_{k,c}, \mathbf{W}_{v,c} \in R^{d \times d}$ respectively, $\mathbf{P} \in R^{2k \times d}$ represents the relative position embedding vectors shared across all layers (i.e., staying fixed during forward propagation), and \mathbf{Q}_r and \mathbf{K}_r are projected relative position vectors generated using projection matrices $\mathbf{W}_{q,r}, \mathbf{W}_{k,r} \in R^{d \times d}$, respectively.

$$\begin{aligned} \mathbf{Q}_c &= \mathbf{H} \mathbf{W}_{q,c}, \mathbf{K}_c = \mathbf{H} \mathbf{W}_{k,c}, \mathbf{V}_c = \mathbf{H} \mathbf{W}_{v,c}, \mathbf{Q}_r = \mathbf{P} \mathbf{W}_{q,r}, \mathbf{K}_r = \mathbf{P} \mathbf{W}_{k,r} \\ \tilde{A}_{i,j} &= \underbrace{\mathbf{Q}_i^c \mathbf{K}_j^{c\top}}_{(a) \text{ content-to-content}} + \underbrace{\mathbf{Q}_i^c \mathbf{K}_{\delta(i,j)}^r} _{(b) \text{ content-to-position}}^\top + \underbrace{\mathbf{K}_j^c \mathbf{Q}_{\delta(j,i)}^r} _{(c) \text{ position-to-content}}^\top \end{aligned} \quad (4)$$
$$\mathbf{H}_o = \text{softmax}\left(\frac{\tilde{\mathbf{A}}}{\sqrt{3d}}\right) \mathbf{V}_c$$

¹⁶<https://arxiv.org/abs/2006.03654>

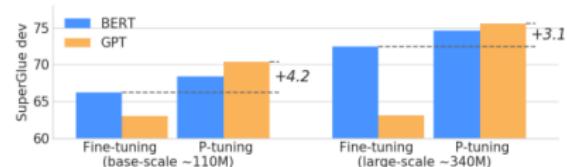
P-tuning¹⁷

Prompt	P@1
[X] is located in [Y]. <i>(original)</i>	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

¹⁷<https://arxiv.org/abs/2103.10385>, <https://arxiv.org/abs/2110.07602>

P-tuning¹⁷

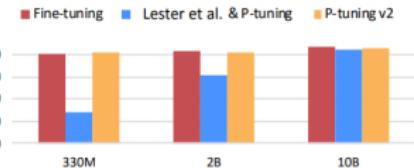
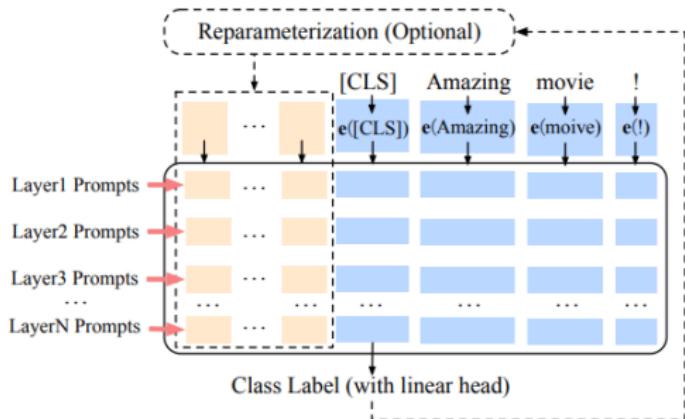
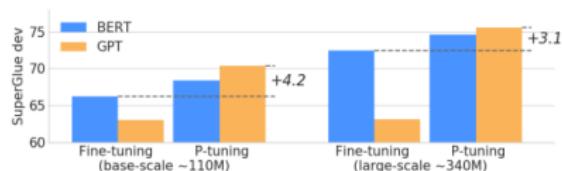
Prompt	P@1
[X] is located in [Y]. (original)	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08



¹⁷<https://arxiv.org/abs/2103.10385>, <https://arxiv.org/abs/2110.07602>

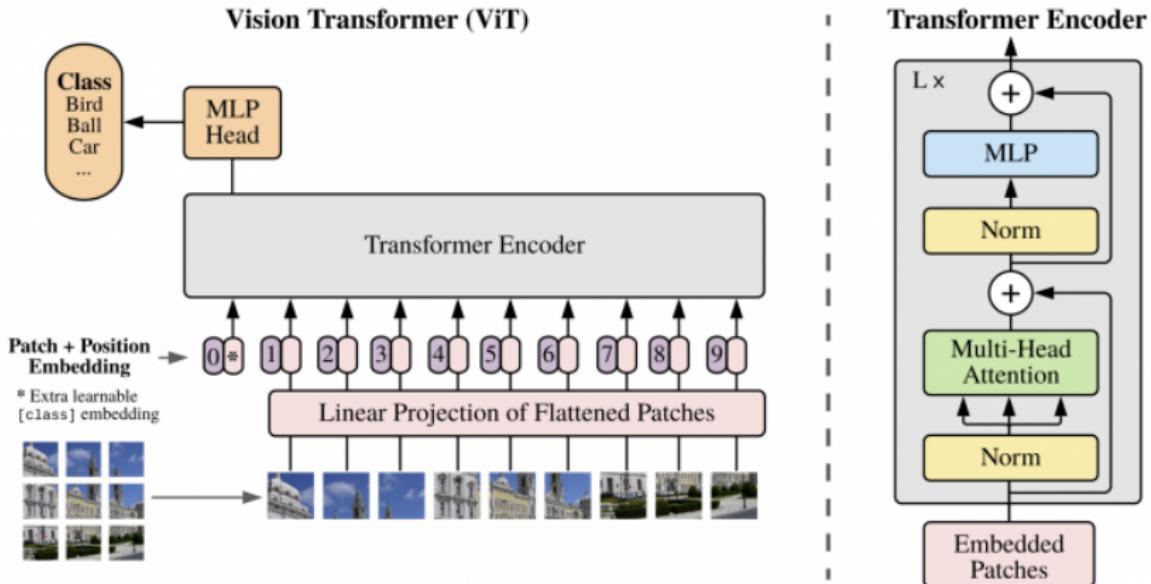
P-tuning¹⁷

Prompt	P@1
[X] is located in [Y]. (original)	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08



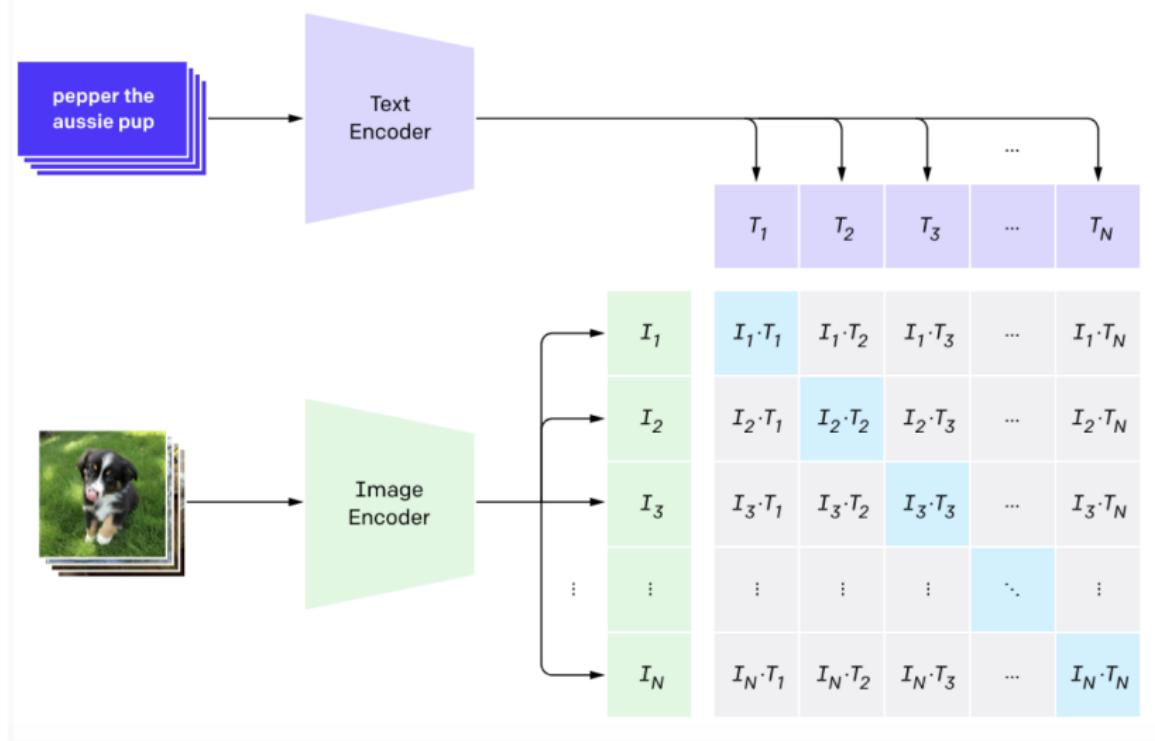
¹⁷<https://arxiv.org/abs/2103.10385>, <https://arxiv.org/abs/2110.07602>

ViT



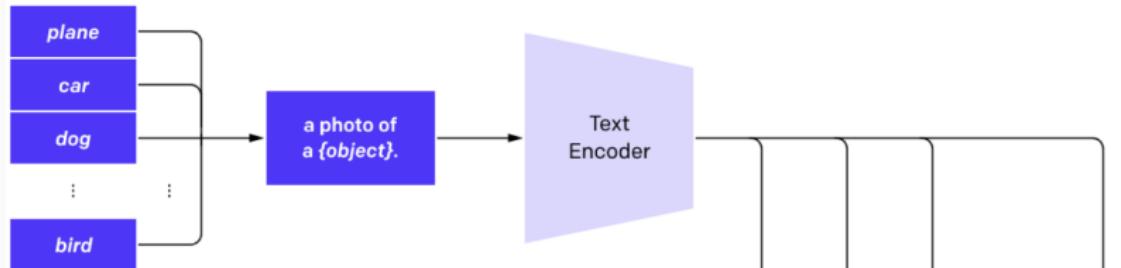
CLIP

1. Contrastive pre-training

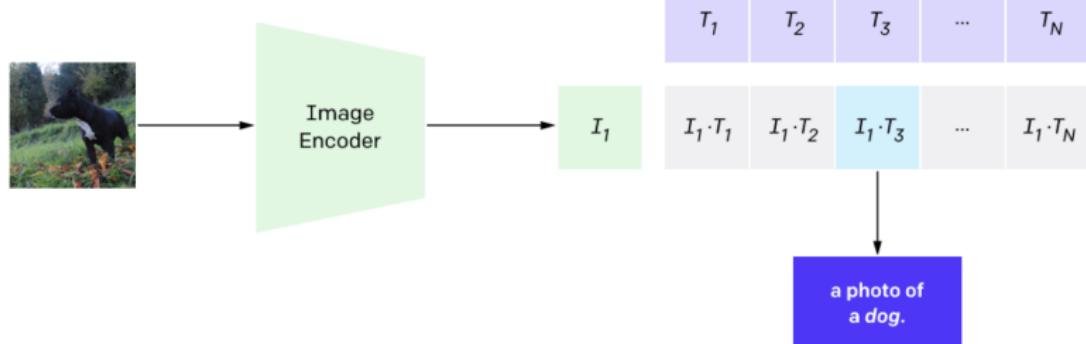


CLIP

2. Create dataset classifier from label text

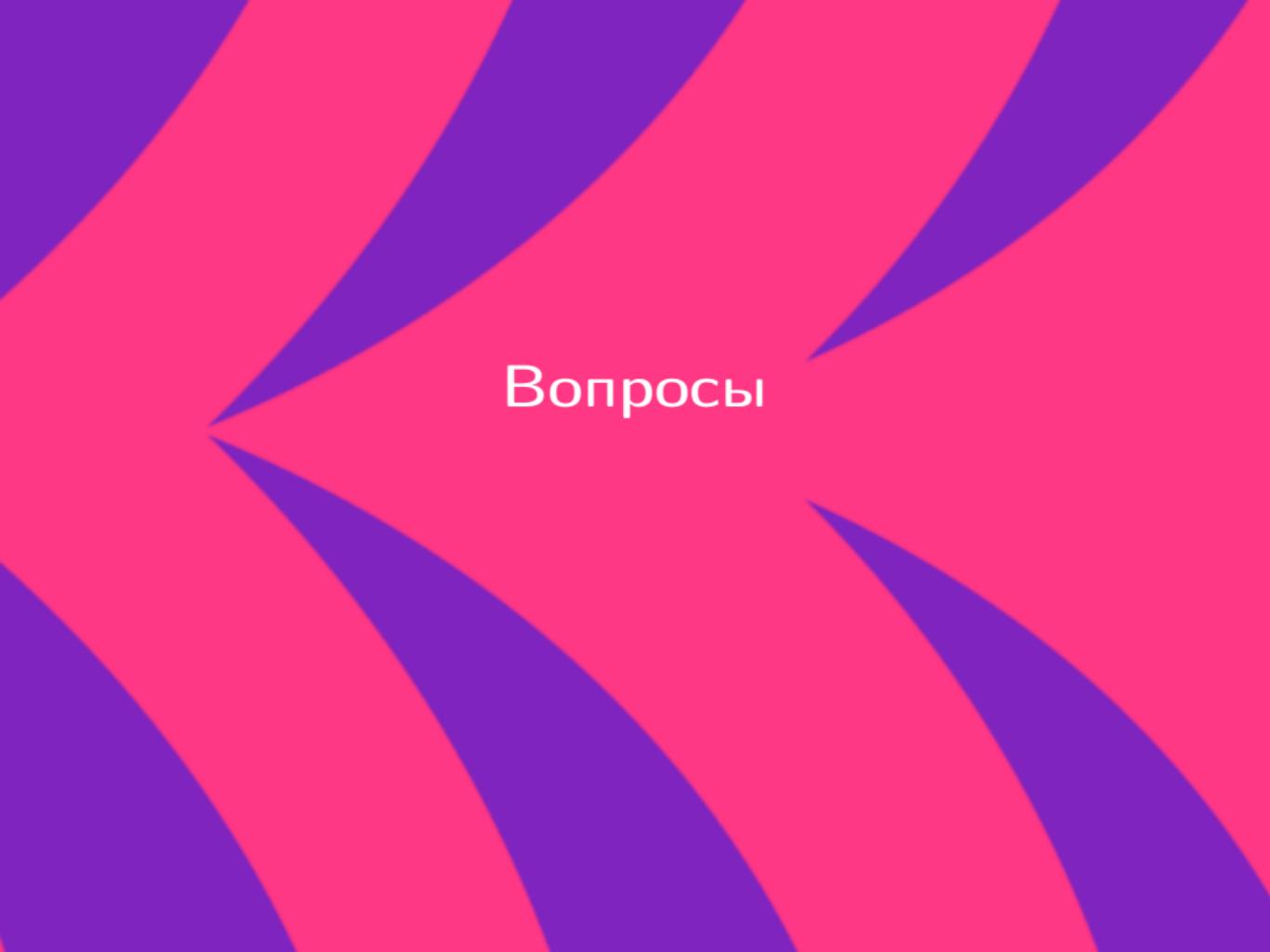


3. Use for zero-shot prediction



За рамками лекции

- ▶ Distillation
- ▶ Технические подробности обучения огромных моделей
- ▶ Модели для длинных последовательностей. Transformer XL, Reformer, Sparse Transformer, BigBird...



Вопросы