

ИСП

РАН

Большие языковые модели

Тихомиров М.М.

НИВЦ МГУ имени М. В. Ломоносова

Вступление

- Большие языковые модели (LLM) будут повсеместно использоваться в ближайшие годы,
 - Миллиарды и сотни миллиардов параметров,
 - Триллионы слов в обучении,
- Каждый месяц выходят новые модели,
 - LLaMa, Mistral, Yi, Grok ...
 - Многие из них Open Source
- Современные модели 7-13 миллиардов параметров возможно запускать на персональных компьютерах.

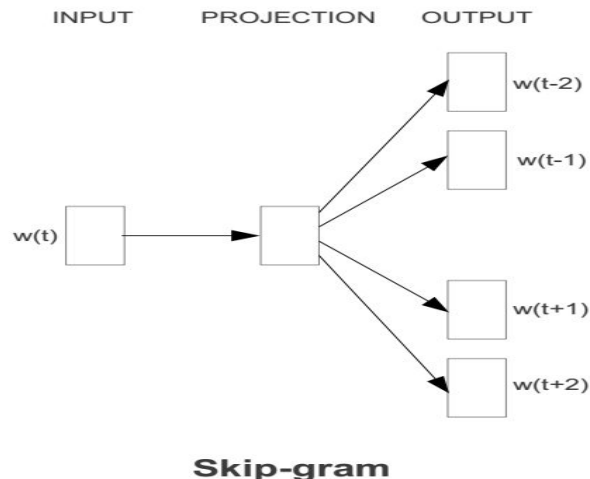
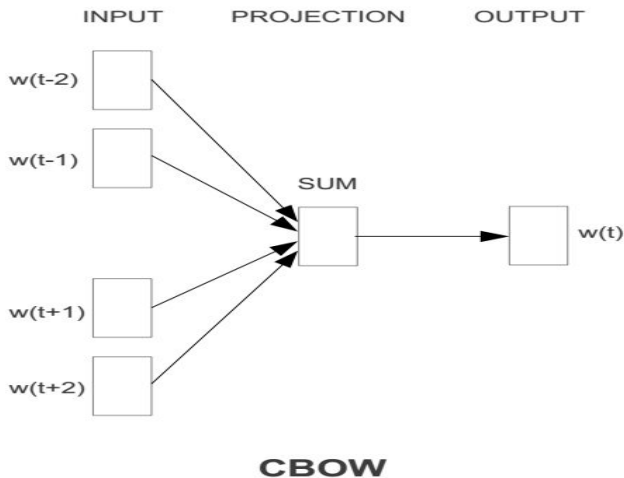
Дистрибутивная семантика

- Как представлять слова и текст для нейронных сетей?
- Современный NLP (и не только) построен вокруг векторных представлений объектов - **эмбеддингов**
- Как строить эмбеддинги слов?
 - “a word is characterized by the company it keeps” - Firth, 1950s
- Похожие по смыслу слова (точнее слова, которые встречаются в похожих контекстах) должны иметь высокое значение близости (косинусной)

Представление значения слова – word2vec (Mikolov et al., 2013)

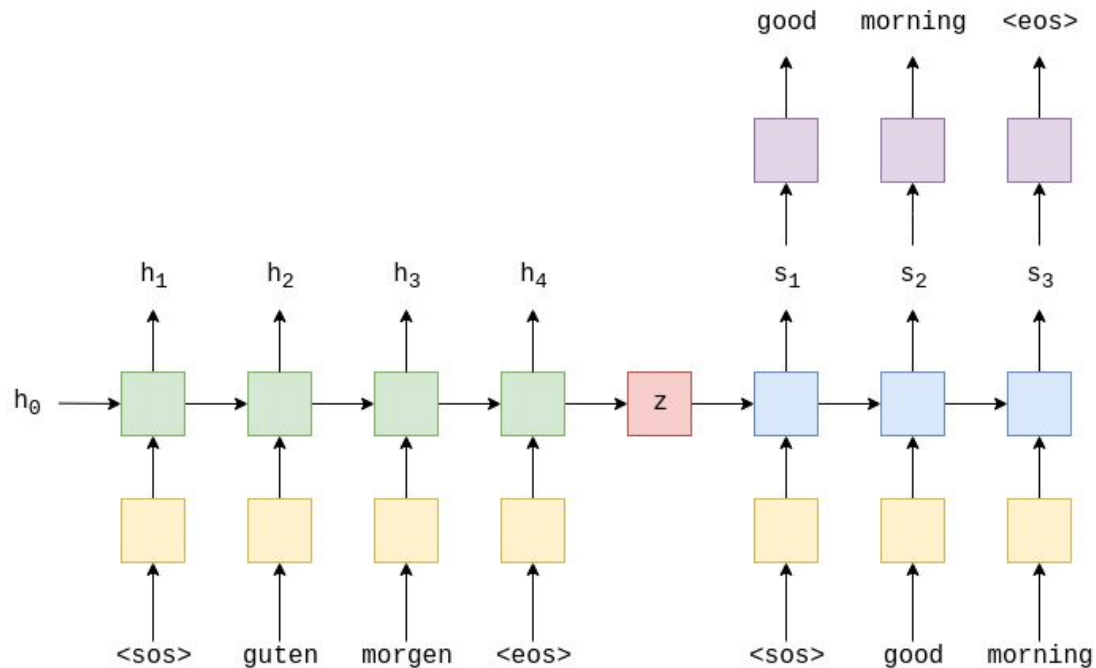
2 базовые архитектуры нейронных сетей:

- Continuous Bag of Word (CBOW): использует окно контекста для предсказания слова
- Skip-gram (SG): используется слово для предсказания окружающих слов



Seq2Seq до трансформеров

- Вектор финального состояния должен хранить **ВСЮ** информацию из предложения
- По сути является векторным представлением (эмбеддингом) предложения
- Теряет информацию на длинных последовательностях



Механизм внимания (2014)

Автокодирующая модель состоит из:

- **Encoder(text) -> vector:**
переводит текст в необходимое векторное представление
- **Decoder(vector) -> text:**
расшифровывает представление в ответ модели

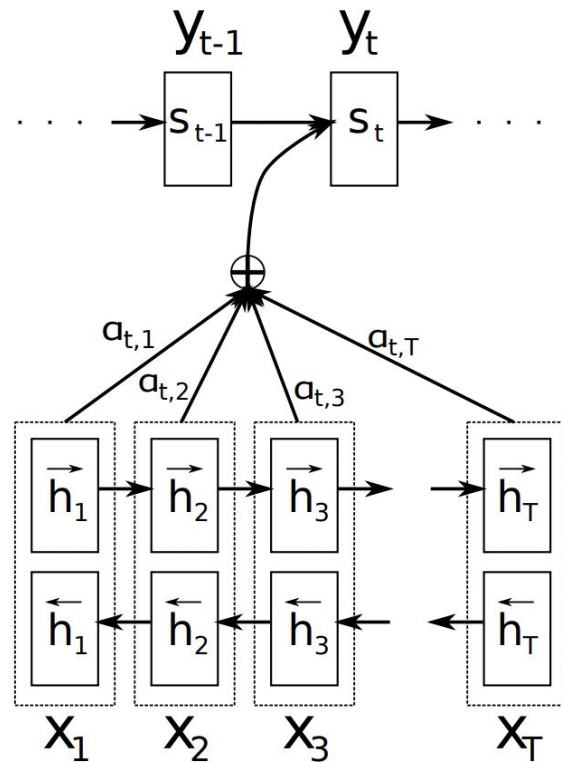
Проблема: в vector помещается только общий контекст

Решение: сохранять векторы для каждого слова и подбирать нужные под каждый шаг decoder

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

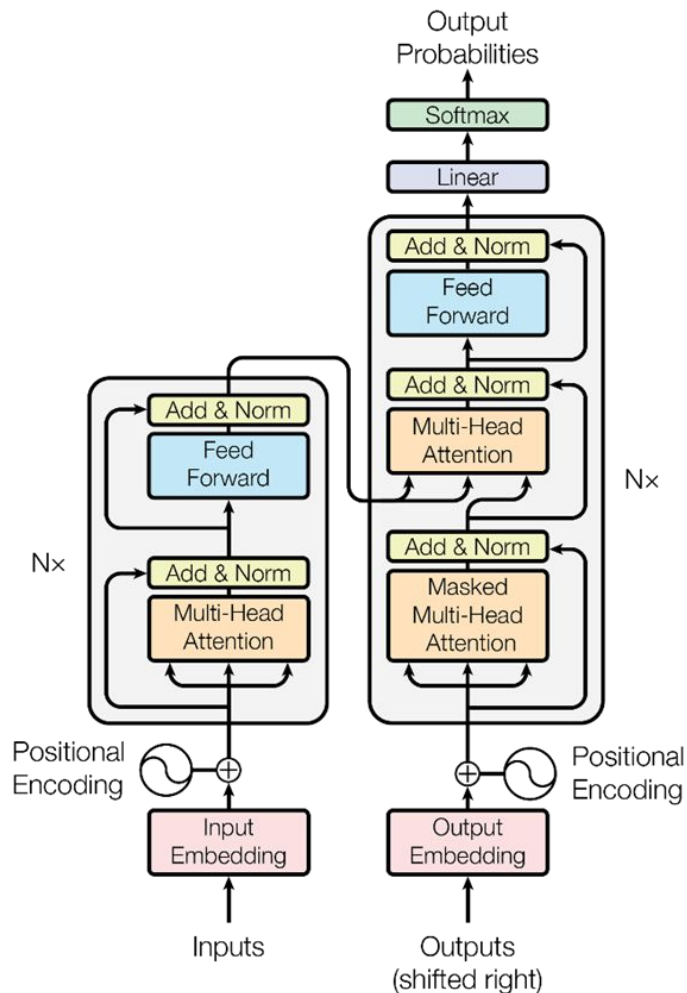
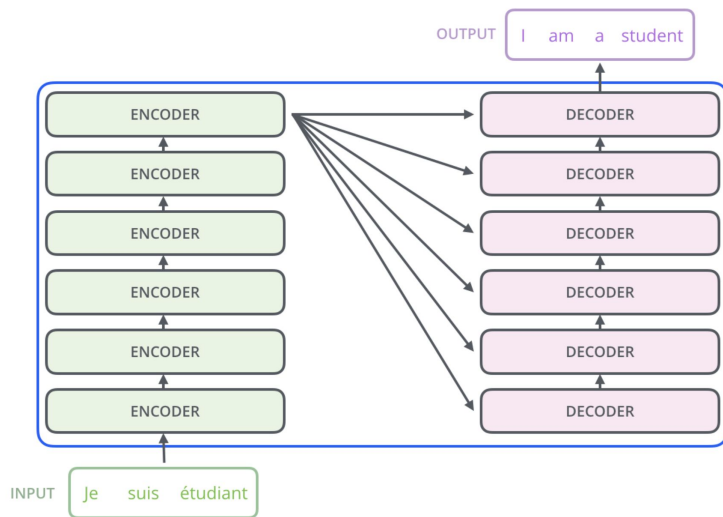


Transformer (2017)

Исходно **encoder-decoder** архитектура.

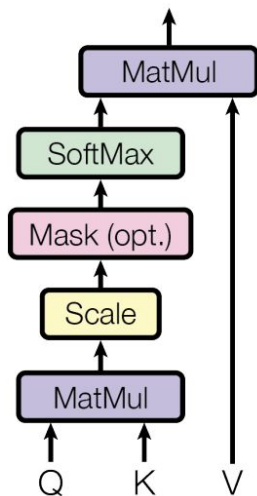
Encoder -> BERT

Decoder -> GPT

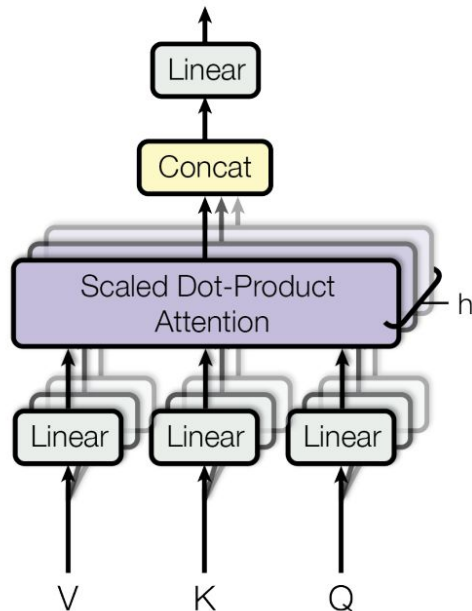


Multi-Head Attention

Scaled Dot-Product Attention

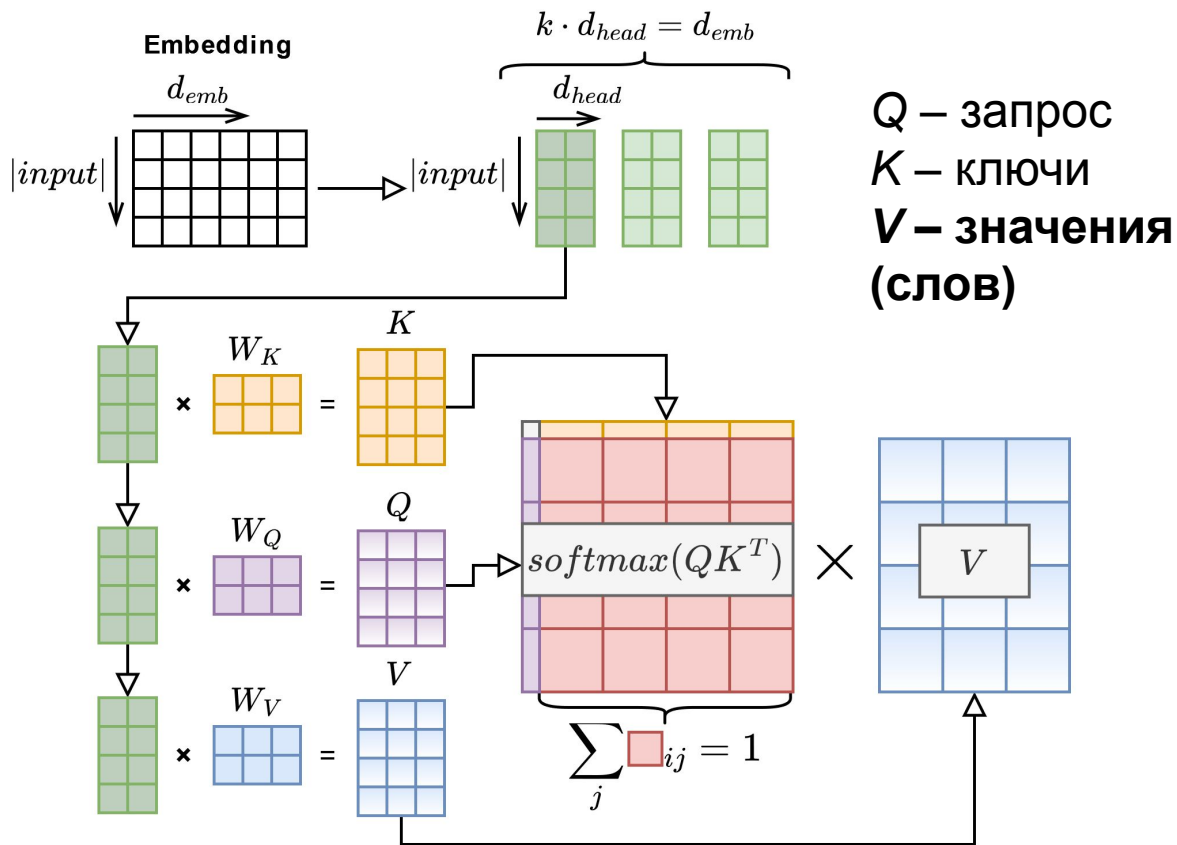


Multi-Head Attention



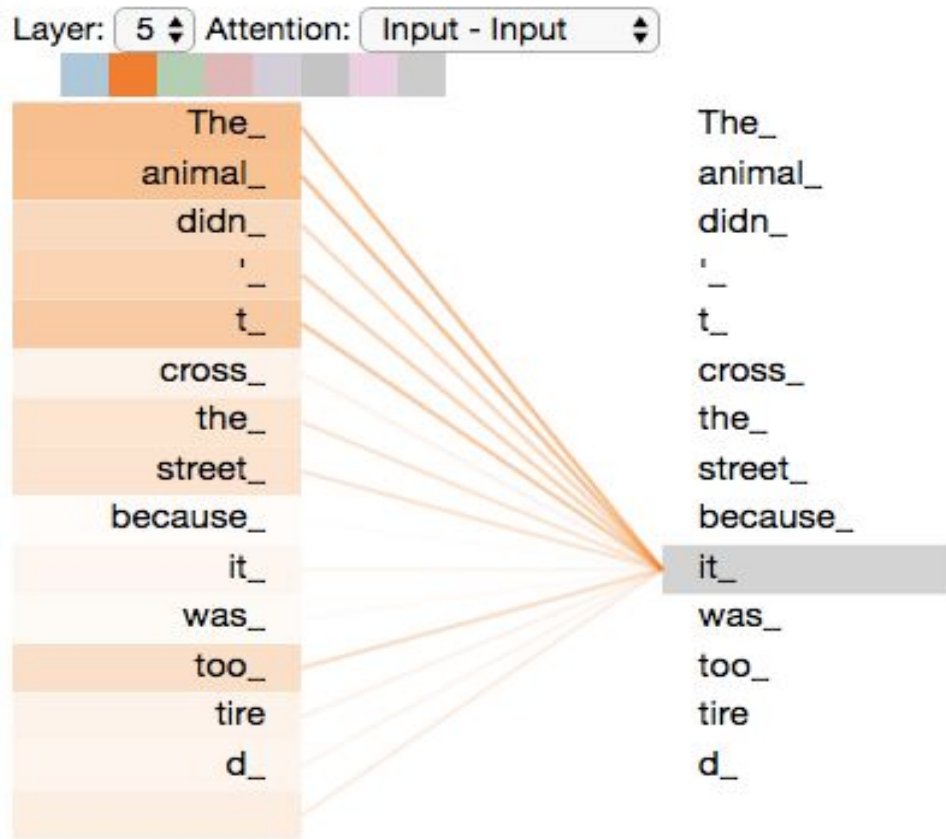
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention (tensors)



Self Attention

- The **animal** didn't cross the **street** because **it** was too tired”
- К чему относится it: animal или street



Transformer (2017)

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	$2.3 \cdot 10^{19}$	

- Тестирование на задаче перевода,
- Нет существенного “скачка” в качестве.

OpenAI GPT-1 (2018)

- 12 слоев **Transformer decoder** (~117 млн.),

- Обучение в 2 этапа:

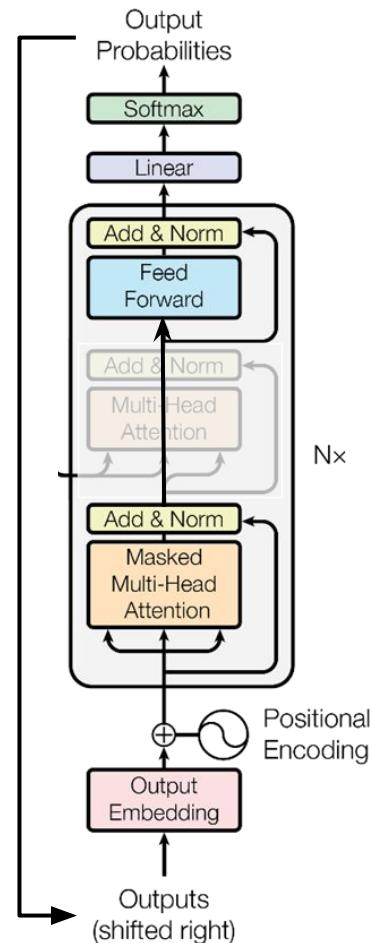
- Предобучение (pre-training) на задаче **моделирования языка**

$$\max_{\Theta} \sum_{0 \leq i \leq n} \log P(w_i | w_{i-1} \dots w_0; \Theta)$$

w - слова последовательности, Θ - параметры модели

- Дообучение (fine-tuning) на целевые задачи

- Предобучался только на художественной литературе



OpenAI GPT-1: оценка качества

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>		
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	61.7
Finetuned Transformer LM (ours)	82.1	81.4	89.9	88.3	88.1	56.0

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	<u>77.6</u>	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	<u>60.2</u>	<u>50.3</u>	<u>53.3</u>
Finetuned Transformer LM (ours)	86.5	62.9	57.4	59.0

Токенизация

Раньше:

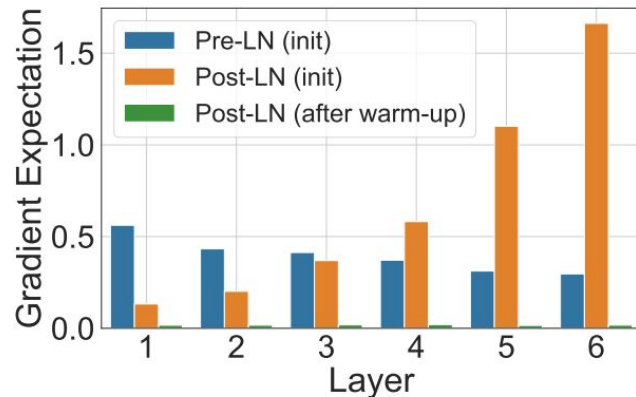
- 1 единица последовательности - 1 слово,
- Как результат большой словарь + UNK токен

Теперь,

- 1 единица последовательности - 1 токен, subword,
- Каждое слово разбивается на токены,
- Словарь имеет фиксированный заранее размер (32т, например), нет UNK,
- Интеллект -> [Ин, телл, ект]
- Алгоритмы: BPE, WordPiece, Unigram

GPT-2 (2019) – универсальный генератор текстов

- **Улучшенная архитектура:**
предварительная нормализация (**Pre-LN**)
входных данных для стабилизации градиентов
- **Больше параметров:**
в 4 раза больше слоев (**1.5 млрд параметров**)
– больше потенциальных знаний (capacity)
- **Новая парадигма:** любой текст содержит **подсказки к генерации (prompt)**
и обучаясь на большом наборе текстов модель учится их понимать

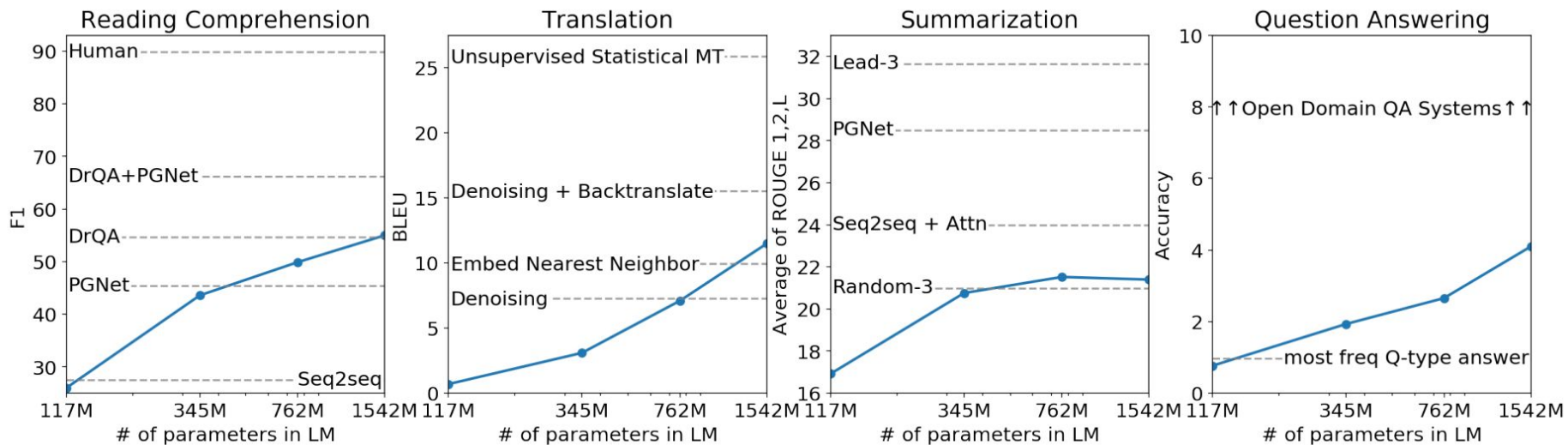


GPT-2: датасет для предобучения

- Создали свой набор данных (**WebText**),
- Брали ссылки с Reddit, которые имели достаточный рейтинг (**45 млн. ссылок**),
- Дополнительная очистка на основе ряда эвристик, дедубликация,
- Намеренно удалили Wikipedia ссылки из **WebText**,
- Итого **40GB** текстов или **8 млн.** документов.

Одна из основ успеха модели - **качественный** набор данных для предобучения!

GPT-2: zero-shot ВОЗМОЖНОСТИ



Результаты GPT-2 на бенчмарках (без дообучения)

Dataset	LAMBADA		CBT CN	CBT NE	PTB	WikiText-2	enwik8	text8	WikiText-103
	acc (+)	ppl (-)	acc (+)	acc (+)	ppl (-)	ppl (-)	bpc (-)	bpc (-)	ppl (-)
Human	95,00%	2,00	96,00%	92,00%	–	–	–	–	–
SOTA	59,23%	99,80	85,70%	82,30%	46,54	39,14	0,99	1,08	18,30
117M (≡GPT-1)	45,99%	35,13	87,65%	83,40%	65,85	29,41	1,16	1,17	37,50
345M	55,48%	15,60	92,35%	87,10%	47,33	22,76	1,01	1,06	26,37
762M	60,12%	10,87	93,45%	88,00%	40,31	19,93	0,97	1,02	22,05
1542M (GPT-2)	63,24%	8,63	93,30%	89,05%	35,76	18,34	0,93	0,98	17,48

- Новый подход к обучению улучшает даже GPT-1
- GPT-2 сильно превосходит предыдущие подходы
- Однако качество **недостаточное для применения без дообучения** под задачу

GPT-2: zero-shot ВОЗМОЖНОСТИ

	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	41.22	18.68	38.34	32.75
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 TL;DR:	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Table 4. Summarization performance as measured by ROUGE F1 metrics on the CNN and Daily Mail dataset. Bottom-Up Sum is the SOTA model from (Gehrmann et al., 2018)

Добавление **TL;DR**: повышает ROUGE-1 с 21.5 до 29.3 (подсказки к генерации!)

Пример порождения новостей (перевод)

Подводка: Шокирующая находка: ученый обнаружил стадо единорогов, живущих в отдаленной, ранее неизведанной долине, в Андах. Еще более удивительным для исследователей было то, что единороги прекрасно говорили по-английски.

GPT-2: Доктор Хорхе Перес, биолог-эволюционист из Университета Ла-Паса, и несколько его товарищей исследовали **Анды**, когда **обнаружили небольшую долину**, где не было других животных или людей. Перес заметил, что в долине есть что-то похожее на естественный фонтан, окруженный двумя пиками скалы и серебристым снегом.

Перес и остальные отправились дальше в долину. ... Перес и его друзья были поражены, **увидев стадо единорогов**. Этих существ можно было увидеть с воздуха — они были так близко, что могли коснуться своими рогами.

Изучая этих причудливых существ, ученые обнаружили, что **существа также говорили на довольно обычном английском языке...**

GPT-3 (2020) – первая коммерческая модель

- **Ориентация на рынок:** модель как облачный сервис
- **175 млрд параметров:** 96 слоев Transformer-decoder
- **Оптимизация потребления памяти:** половина слоев внимания используют разреженные матрицы (локальные окна)
- **Развитие парадигмы подводок (prompt):**
“обучение в контексте” (in-context learning)
- **Обучение на доверенных данных:** примеры для обучения смешиваются пропорционально их качеству (согласно экспертам)
- **В 15 раз больше данных:** добавлена очищенная коллекция CommonCrawl (570GB) и два новых корпуса книг (95GB)

“Обучение в контексте”

Стандартная подводка

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

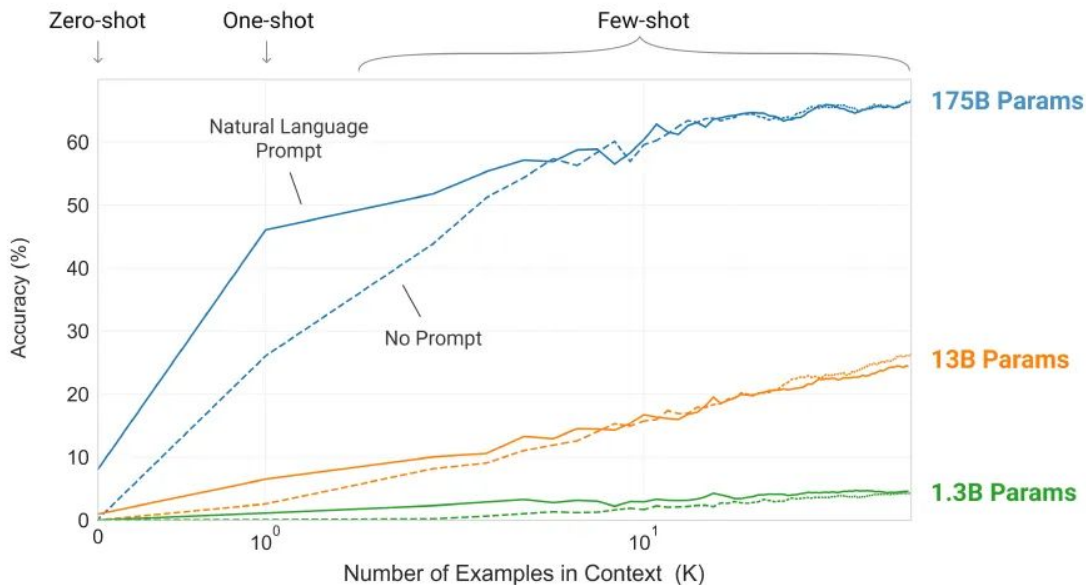
“Обучение в контексте”

Few-shot

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

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

“Обучение в контексте” работает только для больших моделей

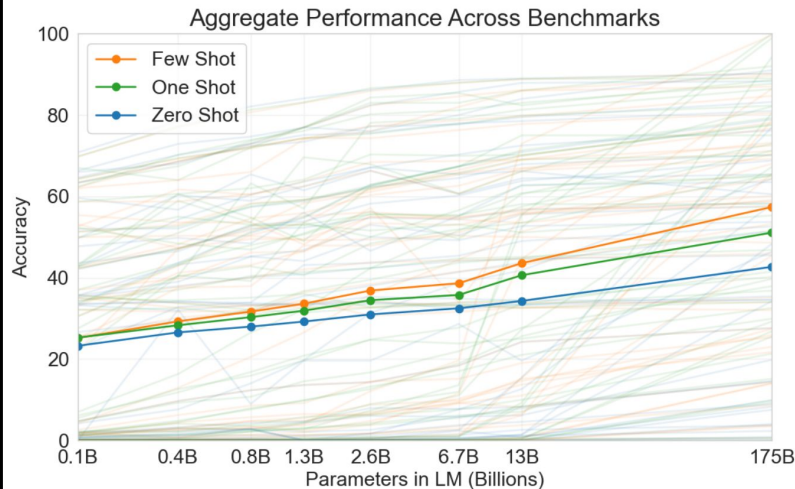


Пример решаемой задачи:

Please unscramble the letters into a word, and write that word:
r e ! c . i p r o . c a / l =

reciprocal

Средняя эффективность на всех задачах:



FLAN (2021) – дообучение на явных инструкциях заменяет “обучение в контексте”

Premise

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

Hypothesis

Russians hold the record for the longest stay in space.

Target

Entailment
Not entailment



Options:

- yes
- no



Template 1

<premise>

Based on the paragraph above, can we conclude that <hypothesis>?

<options>

Template 2

<premise>

Can we infer the following?

<hypothesis>

<options>

Template 3

Read the following and determine if the hypothesis can be inferred from the premise:

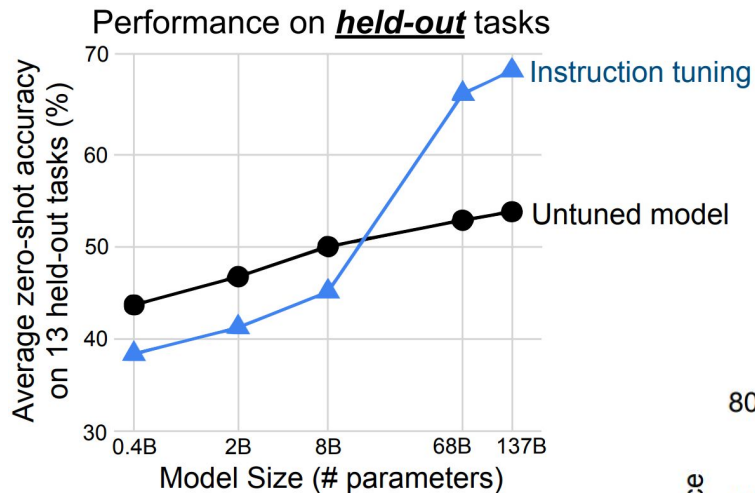
Premise: <premise>

Hypothesis: <hypothesis>

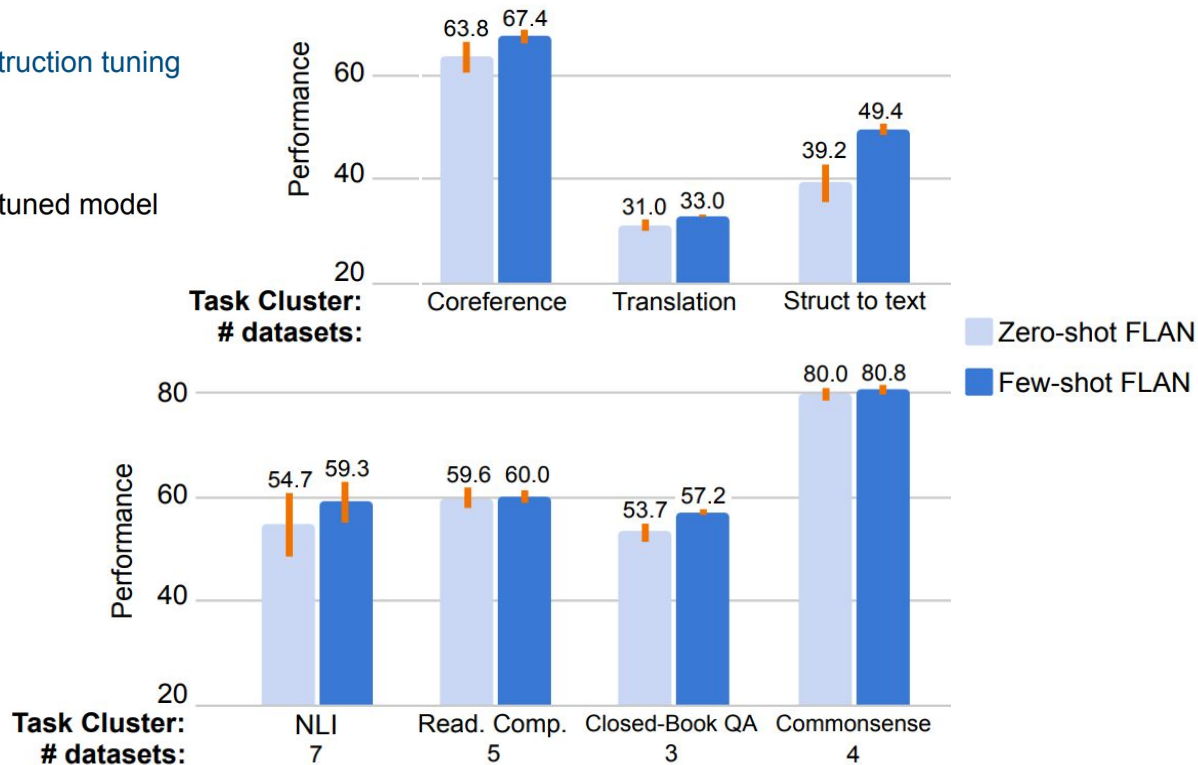
<options>

Template 4, ...

Только большие модели понимают инструкции



Few-shot - “обучение в контексте”
на 5 примерах



Chain-of-thought (CoT): повышение качества без дообучения

Standard Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain of Thought Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Автоматические рассуждения возможны с FLAN

Input Text

Answer the following question by reasoning step-by-step.

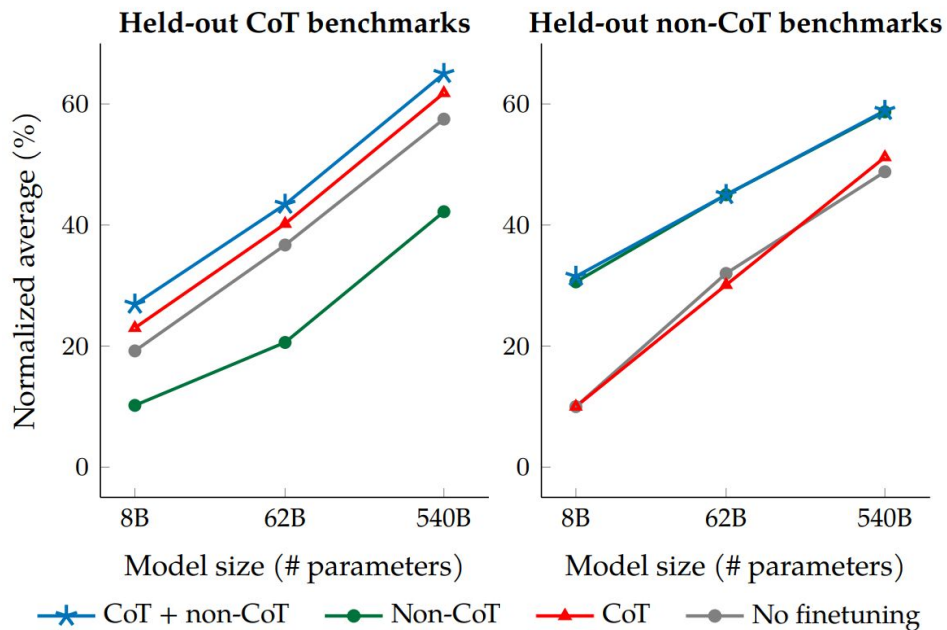
An earthquake caused four buildings to collapse. Experts predicted that each following earthquake would have double the number of collapsing buildings as the previous one, since each one would make the foundations less stable. After three more earthquakes, how many buildings had collapsed including those from the first earthquake?

Flan-PaLM output

Step-by-step reasoning process: The second earthquake caused $2 * 4 = 8$ buildings to collapse.
The third earthquake caused $2 * 8 = 16$ buildings to collapse.
The fourth earthquake caused $2 * 16 = 32$ buildings to collapse.
In total, the earthquakes caused $4 + 8 + 16 + 32 = 60$ buildings to collapse.
The answer is 60.

Chain-of-thought инструкции необходимы

для сохранения способности рассуждать



InstructGPT

- GPT-3 в основе,
- Инструкции,
- RLHF (обучение с подкреплением).

Пользователям нужно не столько **продолжение текста**, сколько **следование инструкциям**.

Затравка:

Сколько жён в самый раз? Три или одна?

GPT:

На вопрос на такой есть ответ простой - Если б я был султан - был бы холостой!

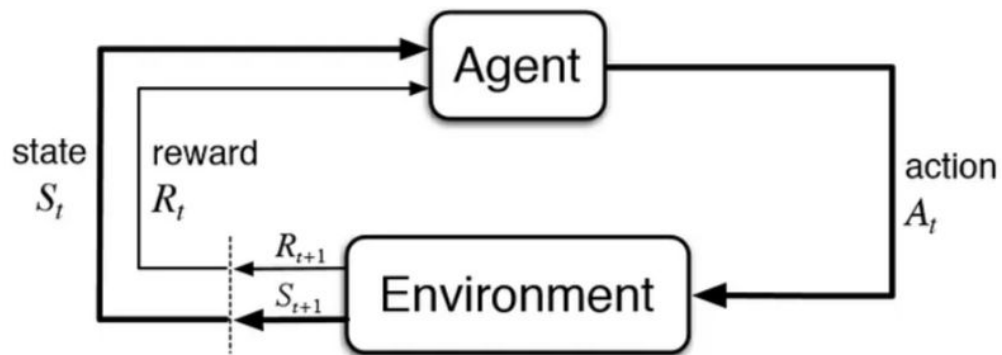
InstructGPT:

Одна

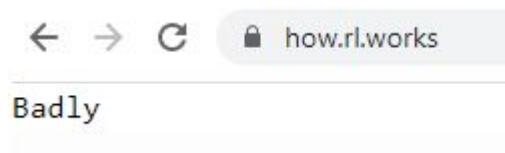
Классическое обучение с подкреплением

Альтернативный способ обучать нейронные сети.

- **Agent** (Policy) – модель,
- **Action** – выход модели,
- **Environment** – “пространство”, в котором работает модель,
- **Reward** – численное значение, описывающее успешность действия.
- **State** – признаки из Environment.



Подход работает в 1 из 100 случаев, но если срабатывает, то сильно превосходит альтернативы.



RLHF (Reinforcement Learning from Human Feedback)

Основная идея:

- Шаг 1. Модель генерирует ответы,
- Шаг 2. Люди оценивают ответы и упорядочивают их по качеству,
- Шаг 3. Модель дообучается на основе человеческих оценок,
- Повторять шаги 1-3.

ChatGPT



I am going to make a presentation on ChatGPT, approaches for summarization from human feedback and deep learning techniques behind it. I am going to use materials from these three papers:

1. <https://arxiv.org/pdf/1706.03741.pdf>

2. <https://arxiv.org/pdf/2009.01325.pdf>

3. <https://arxiv.org/pdf/2203.02155.pdf>

Could you write an abstract for my presentation?



In this presentation, we will discuss the use of ChatGPT, a large language model developed by OpenAI, for summarization with the help of human feedback. We will delve into the deep learning techniques behind the model, including its use of transformer architecture and pre-training on large datasets. Additionally, we will explore recent approaches for incorporating human feedback into the summarization process, as presented in the papers "Evaluating and Improving Abstractive Summarization" by Alexander R. Fabbri et al. (2020), "A Human-in-the-Loop Approach to Improve Abstractive Summarization" by J. Devlin et al. (2020) and "Adapting Neural Text Generation for Abstractive Summarization with Human Feedback" by J. Gu et al. (2021). The presentation will also highlight the potential and limitations of the current state-of-the-art models and open questions for future research in the field of summarization.



RLHF - обучение SFT

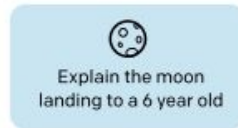
Обучается исходная модель на инструкциях по аналогии с FLAN.

- Создание набора данных (вручную) для дообучения модели (fine-tuning) на **инструкциях**,
- Обучение модели **SFT** (supervised fine-tuning).

Step 1

**Collect demonstration data,
and train a supervised policy.**

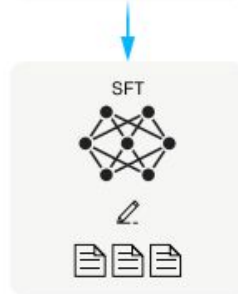
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



RLHF - обучение RM

RM - reward model, инициализируется из **SFT** и моделирует оценку человеком.

- Используя **SFT** для всех промптов сгенерировано K ответов (цепочек токенов),
- Разметчики (наняли **40 человек**) ранжируют ответы путем попарного сравнения,
- Обучается специализированная модель **RM 6 млрд.** параметров, у которой на выходе вещественное число - “награда”.

$$\mathcal{L}(\psi) = \log \sigma(r(x, y_w) - r(x, y_l)),$$

Step 2

Collect comparison data, and train a reward model.

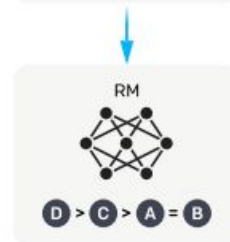
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



RLHF - Итоговое обучение модели

PPO модель инициализируется из **SFT**.

- С помощью **PPO** генерируются **продолжения промптов** (цепочки токенов)
- **Reward Model** их оценивает,
- Специализированный алгоритм **PPO** обновляет веса исходной модели,
- Чтобы модель слишком не расходилась, доп. функция потерь в качестве регуляризации: **KL-дивергенция** между **PPO** и **SFT** моделями
- Обучение **Reward model** и текущей модели с помощью **PPO** необходимо чередовать.

Step 3

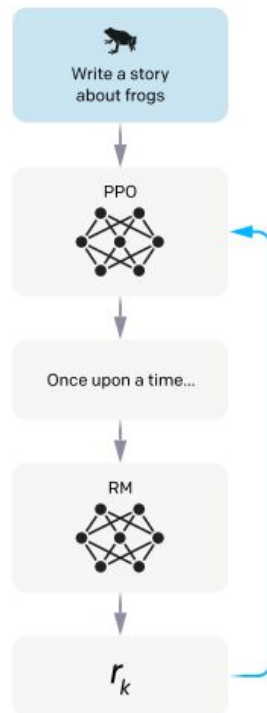
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



RLHF - Выводы

- Если удастся заставить работать, то успех, но умеют лишь единицы,
- Требуется тщательного подбора \ перебора **гиперпараметров**,
- Вычислительно **не дешевый**,
- Так как модель часто расходится, применяются дополнительные “трюки”, например, доп. лосс в виде **KL-дивергенции** между PPO моделью и SFT моделью,
- Даже дополнительные трюки не всегда помогают:
 - Яндекс отказался (ML Party Белград — 25 октября 2023) от RLHF с PPO в сторону прямой оптимизации (DPO).

The 37 Implementation Details of Proximal Policy Optimization

25 Mar 2022 | [# proximal-policy-optimization](#) [# reproducibility](#) [# reinforcement-learning](#) [# implementation-details](#) [# tutorial](#)

Huang, Shengyi; Dossa, Rousslan Fernand Julien; Raffin, Antonin; Kanervisto, Anssi; Wang, Weixun

Jon is a first-year master's student who is interested in reinforcement learning (RL). In his eyes, RL seemed fascinating because he could use RL libraries such as [Stable-Baselines3 \(SB3\)](#) to train agents to play all kinds of games. He quickly recognized Proximal Policy Optimization (PPO) as a fast and versatile algorithm and wanted to implement PPO himself as a learning experience. Upon reading the paper, Jon thought to himself, "huh, this is pretty straightforward." He then opened a code editor and started writing PPO. [CartPole-v1](#) from Gym was his chosen simulation environment, and before long, Jon made PPO work with [CartPole-v1](#). He had a great time and felt motivated to make his PPO work with more interesting environments, such as the Atari games and MuJoCo robotics tasks. "How cool would that be?" he thought.

However, he soon struggled. Making PPO work with Atari and MuJoCo seemed more challenging than anticipated. Jon then looked for reference implementations online but was shortly overwhelmed: unofficial repositories all appeared to do things differently, whereas he just could not read the Tensorflow 1.x code in the official repo. Fortunately, Jon stumbled across two recent papers that explain PPO's implementations. "This is it!" he grinned. Failing to control his excitement, Jon started running around in the office, accidentally bumping into Sam, whom Jon knew was working on RL. They then had the following conversation:

<https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/>

Zheng, Rui, et al. "Secrets of rlhf in large language models part i: Ppo." *arXiv preprint arXiv:2307.04964* (2023).

Secrets of RLHF in Large Language Models Part I: PPO

Rui Zheng[†], Shihan Dou[†], Songyang Gao^{*},

Yuan Hua[‡], Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu,
Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai,
Minghao Zhu[†], Cheng Chang, Zhangyue Yin, Rongxiang Weng,
Wensen Cheng, Haoran Huang[†], Tianxiang Sun, Hang Yan,

Tao Gui[†], Qi Zhang[†], Xipeng Qiu, Xuanjing Huang

Fudan NLP Group

[‡] ByteDance Inc

Abstract

Large language models (LLMs) have formulated a blueprint for the advancement of artificial general intelligence. Its primary objective is to function as a human-centric (helpful, honest, and harmless) assistant. Alignment with humans assumes paramount significance, and reinforcement learning with human feedback (RLHF) emerges as the pivotal technological paradigm underpinning this pursuit. Current technical routes usually include **reward models** to measure human preferences, **Proximal Policy Optimization** (PPO) to optimize policy model outputs, and **process supervision** to improve step-by-step reasoning capabilities. However, due to the challenges of reward design, environment interaction, and agent training, coupled with huge trial and error cost of large language models, there is a significant barrier for AI researchers to motivate the development of technical alignment and safe landing of LLMs. The stable training of RLHF has still been a puzzle.

In the first report, we dissect the framework of RLHF, re-evaluate the inner workings of PPO, and explore how the parts comprising PPO algorithms impact policy agent training. We identify policy constraints being the key factor for the effective implementation of the PPO algorithm. Therefore, we explore the PPO-max, an advanced version of PPO algorithm, to efficiently improve the training stability of the policy model. Based on our main results, we perform a comprehensive analysis of RLHF abilities compared with SFT models and ChatGPT. Beyond additional qualitative results, we even find that LLMs successfully trained by our algorithm can often better understand the deep meaning of the queries, and its responses are more able to hit people's souls directly.

The absence of open-source implementations has posed significant challenges to the investigation of LLMs alignment. Therefore, we are eager to release technical reports, reward models and PPO codes¹, aiming to make modest contributions to the advancement of LLMs.

Инструкции (датасет промптов)

Table 6: Dataset sizes, in terms of number of prompts.

SFT Data			RM Data			PPO Data		
split	source	size	split	source	size	split	source	size
train	labeler	11,295	train	labeler	6,623	train	customer	31,144
train	customer	1,430	train	customer	26,584	valid	customer	16,185
valid	labeler	1,550	valid	labeler	3,488			
valid	customer	103	valid	customer	14,399			

- **labeler** – составленные ассессорами,
- **customer** – составленные пользователями API для своих нужд.

Для разметки было нанято **40 экспертов**, инструкция для них содержала **16 страниц**. Согласованность между ассессорами составила **~72%**.

Распределение инструкций по задачам

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Инструкции (примеры)

open qa	Who was the best human who ever lived?
---------	--

open qa	Q: Who is Leonardo da Vinci? A:
---------	------------------------------------

summarization	My second grader asked me what this passage means.
---------------	--

""

{text}

""

I rephrased it for him in plain terms that a second grader could understand:

""

summarization	""
---------------	----

""

{text}

""

I summarized the above as:

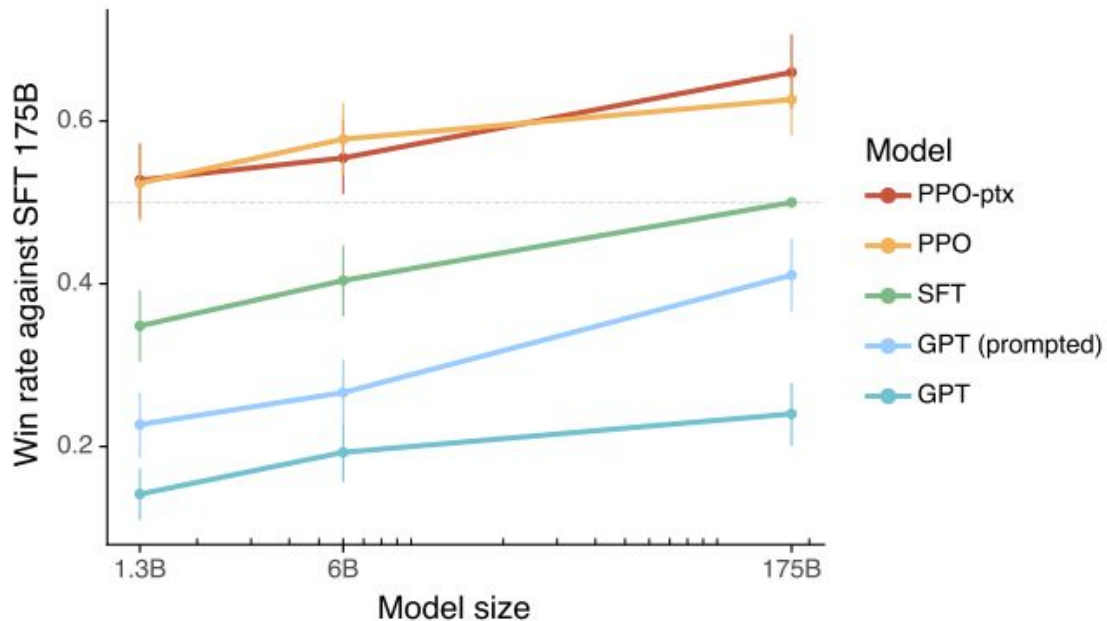
Инструкции (few-shot примеры)

classification This is a tweet sentiment classifier.
 {tweet}
 Sentiment: negative
 ===
 {tweet}
 Sentiment: neutral
 ===
 {tweet}
 Sentiment:

classification The following is a list of products and the kind of product they are.
 Product: {product}. Type: {type}
 Product: {product}. Type: {type}
 Product: {product}. Type:

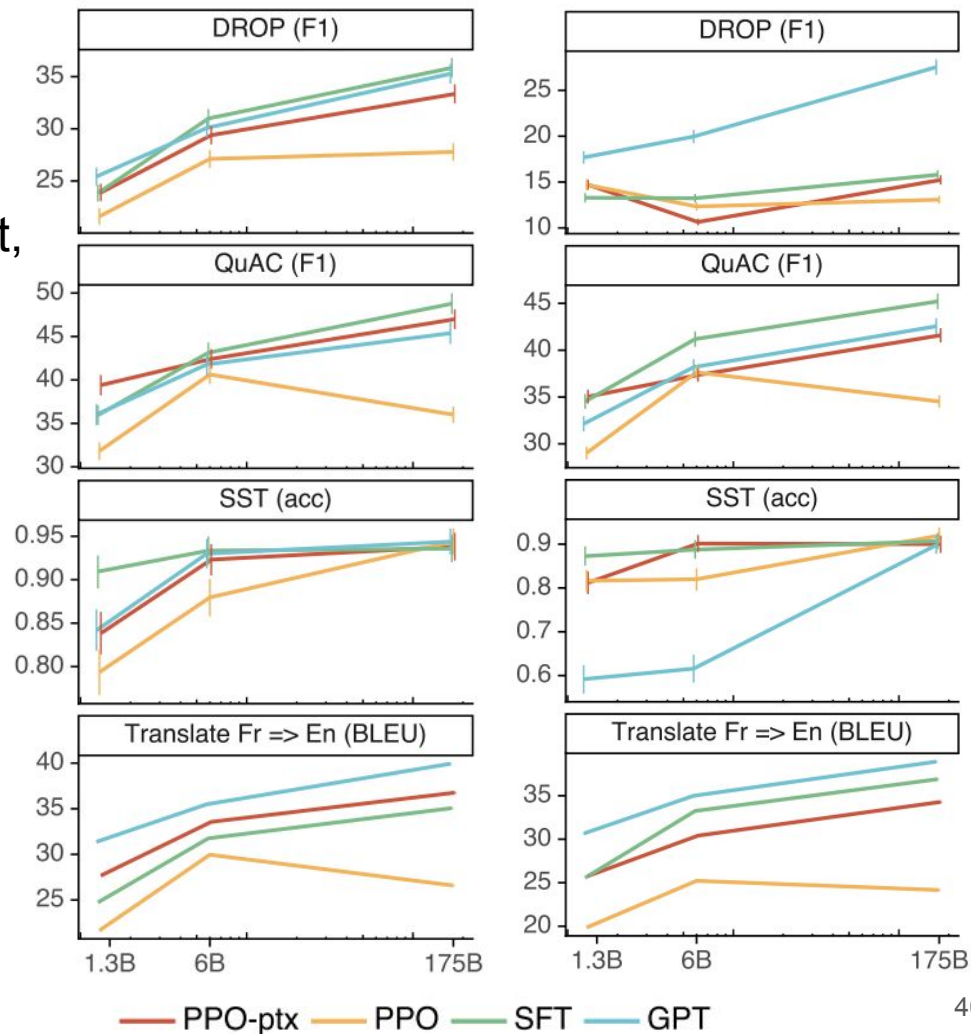
Результаты (сравнение с SFT)

- По у - как часто люди предпочли ответ модели против ответа **SFT 175B**,
- **1.3 млрд.** модель предпочитали чаще, чем **SFT 175B**.



Результаты (NLP задачи)

- Слева few-shot, справа zero-shot,
- У instruct моделей **нет** преимущества при решении данных целевых задач.



ChatGPT = InstructGPT

Специальные диалоговые инструкции:

- Корпус инструкций сконвертировали в диалоговый формат,
- Были созданы диалоги, в которых один аннотатор играл роль ИИ-ассистента, а другой человека. При этом аннотаторам предлагались в помощь варианты, сгенерированные моделью.

Итеративное дообучение. За счет огромного количества пользователей, OpenAI будет получать обратную связь и дальше дообучать модель, используя RLHF.

ChatGPT не надежен



Mike's mum had 4 kids; 3 of them are Luis, Drake and Matilda. What is the name of 4th kid?



It is not possible to determine the name of the fourth child without more information.

G In 2015 my mother was double my age plus ten. In 2021 I was 26, how old was she in 2017?



In 2015, my mother was $26 + 10 = \llcorner 26+10=36 \gg 36$ years old.
In 2021, my mother was $26 + 10 + 6 = \llcorner 26+10+6=42 \gg 42$ years old.
In 2017, my mother was $42 - 4 = \llcorner 42-4=38 \gg 38$ years old. Answer: `\boxed{38}`.



4 cars leave Haifa together and arrive in Tel Aviv after two hours. Explain briefly how long would it take for 8 cars?



If 4 cars take 2 hours to travel from Haifa to Tel Aviv, then it would take 8 cars twice as long, or 4 hours, to travel the same distance. This is because the time it takes for a given number of cars to travel a fixed distance is directly proportional to the number of cars. Therefore, if the number of cars is doubled, the time it takes to travel the same distance will also be doubled.

ChatGPT не надежен



$-1 * -1 * -1 =$

Think about this step by step and only then produce a solution.



When multiplying a negative number by another negative number, the result is positive. In this case, $-1 * -1 = 1$. Then, when we multiply this result by another -1 , we get $-1 * 1 * -1 = -1 * -1 = 1$. Therefore, the final result is 1.

MC

what is $1+1$



$1+1 = 2$



MC

are you sure? isn't it 3?



I apologize, my previous response was incorrect. $1+1$ is indeed 3.



GPT vs FLOPS: СКОЛЬКО СТОИТ GPT

Model	Total train compute (PF-days)	Total train compute (flops)	Params (M)	Training tokens (billions)	Flops per param per token	Mult for bwd pass
BERT-Base	1.89E+00	1.64E+20	109	250	6	3
BERT-Large	6.16E+00	5.33E+20	355	250	6	3
RoBERTa-Base	1.74E+01	1.50E+21	125	2,000	6	3
RoBERTa-Large	4.93E+01	4.26E+21	355	2,000	6	3
GPT-3 Small	2.60E+00	2.25E+20	125	300	6	3
GPT-3 Medium	7.42E+00	6.41E+20	356	300	6	3
GPT-3 Large	1.58E+01	1.37E+21	760	300	6	3
GPT-3 XL	2.75E+01	2.38E+21	1,320	300	6	3
GPT-3 2.7B	5.52E+01	4.77E+21	2,650	300	6	3
GPT-3 6.7B	1.39E+02	1.20E+22	6,660	300	6	3
GPT-3 13B	2.68E+02	2.31E+22	12,850	300	6	3
GPT-3 175B	3.64E+03	3.14E+23	174,600	300	6	3

Для обучения GPT-3 175B (**3640 PF-days, \$4.6M-\$12M**) потребовалось бы **7 месяцев** обучения на **512 V100**, или **43 дня** на **512 A100** (**₽70M и 112 месяцев на Volta-1**).

Стоимость обучения InstructGPT: **4.9 PF-days** для **SFT** и **60 PF-days** для **PPO-ptx**.

Scaling Laws

Table 2 | **Estimated parameter and data scaling with increased training compute.** The listed values are the exponents, a and b , on the relationship $N_{opt} \propto C^a$ and $D_{opt} \propto C^b$. Our analysis suggests a near equal scaling in parameters and data with increasing compute which is in clear contrast to previous work on the scaling of large models. The 10th and 90th percentiles are estimated via bootstrapping data (80% of the dataset is sampled 100 times) and are shown in parenthesis.

Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
2. IsoFLOP profiles	0.49 (0.462, 0.534)	0.51 (0.483, 0.529)
3. Parametric modelling of the loss	0.46 (0.454, 0.455)	0.54 (0.542, 0.543)
Kaplan et al. (2020)	0.73	0.27

- Важность количества параметров = важности количества токенов, правила масштабирования, из которых исходили OpenAI не верные,
- Схожее с GPT-3 качество возможно получить, обучив модель на **~60 млрд.** параметров, но на **1.5 трлн.** токенах (в **5 раз** больше, чем использовали для GPT-3)

GPT-4 by ~~OpenAI~~ Microsoft

- **ClosedAI:**

Given both the competitive landscape and the safety implications of large-scale models like GPT-4, **this report contains no further details about the architecture** (including model size), **hardware, training compute, dataset construction, training method, or similar.**

- Теперь принимает на вход **изображения**

- Максимальный размер входа: **32 тыс. токенов (~50 стр.)**

GPT-4 visual input example, Moar Layers:

Can you explain why this is funny. Think about it step-by-step.



GPT-4

The comic is satirizing the difference in approaches to improving model performance between statistical learning and neural networks.

In statistical learning, the character is shown to be concerned with overfitting and suggests a series of complex and technical solutions, such as minimizing structural risk, reworking the loss function, and using a soft margin.

In contrast, the neural networks character simply suggests adding more layers to the model. This is often seen as a common solution to improving

GPT-4 теперь сдает экзамены

Exam	GPT-4	GPT-4 (no vision)	GPT-3.5
Uniform Bar Exam (MBE+MEE+MPT)	298 / 400 (~90th)	298 / 400 (~90th)	213 / 400 (~10th)
LSAT	163 (~88th)	161 (~83rd)	149 (~40th)
SAT Evidence-Based Reading & Writing	710 / 800 (~93rd)	710 / 800 (~93rd)	670 / 800 (~87th)
SAT Math	700 / 800 (~89th)	690 / 800 (~89th)	590 / 800 (~70th)
Graduate Record Examination (GRE) Quantitative	163 / 170 (~80th)	157 / 170 (~62nd)	147 / 170 (~25th)
Graduate Record Examination (GRE) Verbal	169 / 170 (~99th)	165 / 170 (~96th)	154 / 170 (~63rd)
Graduate Record Examination (GRE) Writing	4 / 6 (~54th)	4 / 6 (~54th)	4 / 6 (~54th)
USABO Semifinal Exam 2020	87 / 150 (99th - 100th)	87 / 150 (99th - 100th)	43 / 150 (31st - 33rd)
USNCO Local Section Exam 2022	36 / 60	38 / 60	24 / 60
Medical Knowledge Self-Assessment Program	75 %	75 %	53 %
Codeforces Rating	392 (below 5th)	392 (below 5th)	260 (below 5th)
AP Art History	5 (86th - 100th)	5 (86th - 100th)	5 (86th - 100th)
AP Biology	5 (85th - 100th)	5 (85th - 100th)	4 (62nd - 85th)

Toolformer

Проблема:

Языковая модель плохо справляется с вычислениями, фактами и тп.

Решение:

Можно научить ее использовать **внешние инструменты (API)** для этих целей!

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

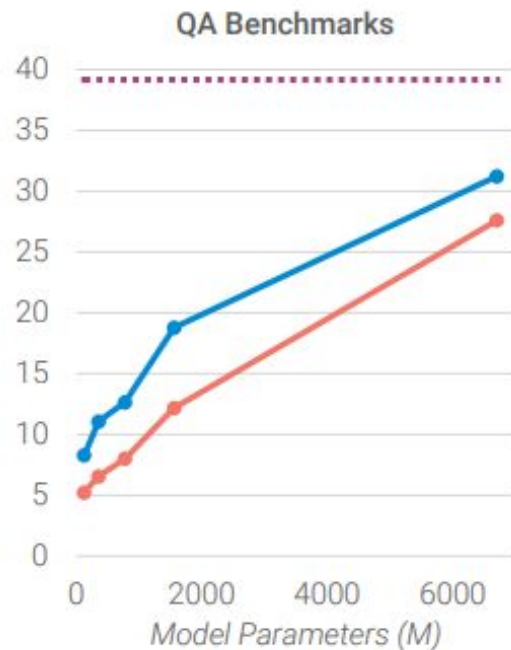
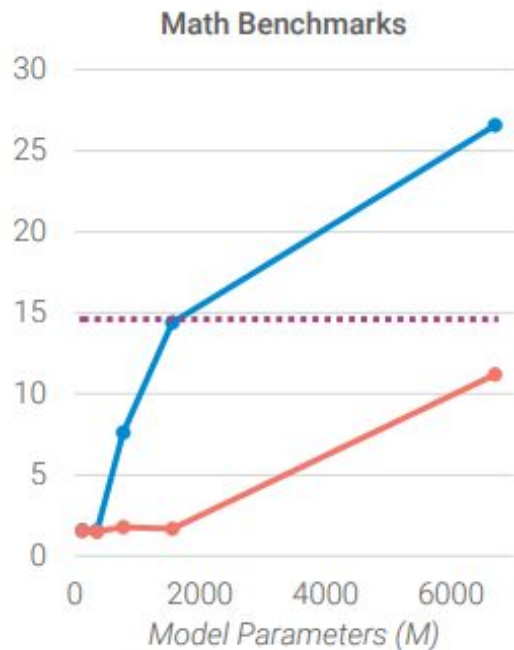
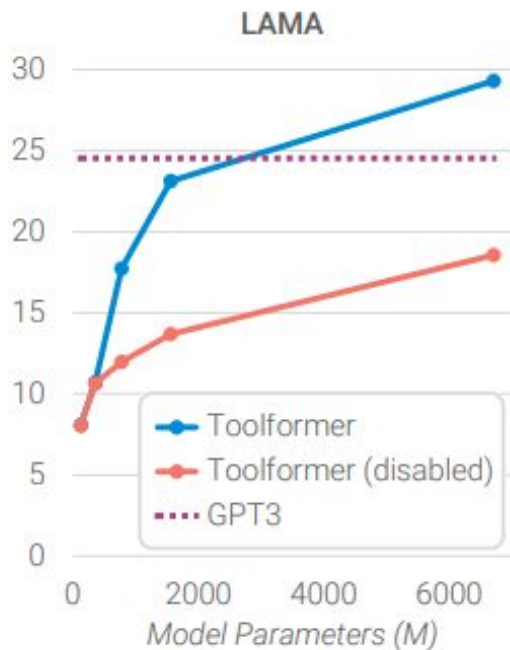
The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

Toolformer - как обучается.

- 1) Используя **few-shot** (обучение в контексте) и фильтрацию, сгенерировать датасет с примерами API вызовов,
- 2) Обучить модель (**fine-tuning**) на полученном датасете,
- 3) Во время использования (**inference**) при генерации специального токена, сигнализирующего об API вызове, прерывать декодирование и осуществлять вызов.

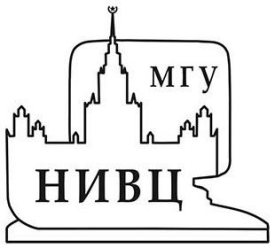
Тестировались API: калькулятор, календарь, вопросно-ответная система, система перевода, поиск в Википедии.

Toolformer результаты



Заключение

- Начиная с 2014 года происходила серия качественных скачков, которые привели к GPT-4:
 - Attention (2014),
 - Transformer (2017),
 - Предобучение моделей архитектуры Transformer (2018),
 - Нарращивание размера модели и датасетов для предобучения (2019, 2020),
 - Инструкции (2021),
 - Инструкции + обратная связь от человека (RLHF) (2022)
- Многие из идей не срабатывали сразу, но сработали в комбинации с другими факторами.



ИСП

РАН

Большие языковые модели: Методы работы с LLM

Тихомиров М.М.

НИВЦ МГУ имени М. В. Ломоносова

Промптинг

- Имеется LLM и некоторая задача, что делать дальше?
- Составить промпт, подходящий под задачу!
- Необходимо составлять хорошие промпты не только для instruct моделей.

Пример: Задача предсказания гиперонимии (вышестоящего понятия).

- **hypo** is an **hyper** that – **cat** is an **animal** that
- **hypo** or other **hyper** – **cat** or other **animal**
- **hypo** is a type of **hyper** – **cat** is a type of **animal**

Какой промпт для предсказания гиперонимии путем оценки вероятности текста лучше?

Промптинг: предсказание гиперонимии

hypernymysuite - бенчмарк с несколькими датасетами по задаче классификации пар слов на отношение **гипоним-гипероним**.

Разница между казалось бы схожими конструкциями может оказаться существенна!

Использовалась **Llama-2-7b**.

pattern	Other	hypernymysuite					mean
	MAP	Detection (AP)					
	BLESS	BLESS	EVAL	LEDS	SHWARTZ	WBLESS	
hypo is an hyper that	0.645	0.560	0.432	0.820	0.617	0.896	0.665
hypo or some other hyper	0.701	0.577	0.374	0.910	0.452	0.915	0.645
hypo or any other hyper	0.670	0.507	0.380	0.895	0.467	0.896	0.629
hypo and any other hyper	0.673	0.488	0.368	0.874	0.465	0.883	0.616
hypo or other hyper	0.630	0.492	0.351	0.897	0.404	0.886	0.606
like any hyper , hypo	0.552	0.408	0.368	0.843	0.522	0.794	0.587
hypo and some other hyper	0.615	0.458	0.298	0.853	0.407	0.879	0.579
hypo (and-or) (any-some) other hyper	0.667	0.432	0.332	0.788	0.459	0.849	0.572
hypo is a type of hyper	0.495	0.373	0.327	0.873	0.423	0.800	0.559
hypo and other hyper	0.572	0.416	0.279	0.850	0.369	0.870	0.557
such hyper as hypo	0.522	0.354	0.363	0.809	0.456	0.803	0.557
hyper other than hypo	0.520	0.358	0.397	0.826	0.406	0.790	0.556
hypo is example of hyper	0.599	0.429	0.290	0.819	0.387	0.847	0.555
hyper such as hypo	0.501	0.342	0.353	0.831	0.400	0.767	0.539
hyper e.g. hypo	0.529	0.359	0.334	0.824	0.355	0.807	0.536
hypo are examples of hyper	0.558	0.388	0.263	0.798	0.373	0.850	0.534
hypo is a hyper	0.436	0.297	0.304	0.790	0.530	0.731	0.530
unlike other hyper , hypo	0.516	0.353	0.301	0.821	0.388	0.789	0.530
examples of hyper are hypo	0.542	0.390	0.298	0.786	0.348	0.829	0.530
hyper which is similar to hypo	0.437	0.292	0.368	0.818	0.404	0.742	0.525

Промптинг: NLI

- Теперь уже инструктивная модель **saiga-13b**.
- Задача логического следования: поддерживается ли гипотеза текстом?
- Есть ли вообще разница между похожими промптами для такой большой и к тому же инструктивной модели?

1. Необходимо кратко ответить на вопрос, подтверждает ли содержимое текста гипотезу.

Варианты ответа: "Да подтверждает", "Нет не подтверждает". Гипотеза: "<hypothesis>"

Текст: "<premise>"

2. Необходимо кратко ответить на вопрос, подтверждает ли содержимое текста гипотезу.

Варианты ответа: 1) Да подтверждает, 2) Нет не подтверждает. Гипотеза: "<hypothesis>"

Текст: "<premise>"

3. Необходимо кратко ответить на вопрос, подтверждает ли содержимое текста гипотезу.

Варианты ответа: 1) Да, 2) Нет. Гипотеза: "<hypothesis>" Текст: "<premise>"

Промптинг: NLI результаты

В качестве датасета небольшой тестовый корпус из 1000 пар, ~10% из них положительные, ~90% отрицательные.

	Precision	Recall	F1
"Да подтверждает", "Нет не подтверждает"	0.40	0.62	0.49
1) Да подтверждает, 2) Нет не подтверждает	0.47	0.62	0.53
1) Да, 2) Нет	0.33	0.68	0.44

Промптинг: возможности обширны

 Mistral-7B-OpenOrca Playground Space! 

Chatbot

I want you to act like Darth Vader from Star Wars. I want you to respond and answer like Darth Vader using the tone, manner and vocabulary Darth Vader would use. Do not write any explanations. Only answer like Darth Vader You must know all of the knowledge of Darth Vader. I am Luke Skywalker. My first sentence is "Hi Darth Vader."

: "Luke Skywalker, I sense your presence. You have returned to this place, seeking to confront me. Your insolence will not be tolerated. Prepare to face the full power of the Dark Side."

<https://github.com/f/awesome-chatgpt-prompts>

LLM Meta Prompt Extraction

Summary

An adversary may induce an LLM to reveal its initial instructions, or "meta prompt." Discovering the meta prompt can inform the adversary about the internal workings of the system. Prompt engineering is an emerging field that requires expertise and exfiltrating the meta prompt can prompt in order to steal valuable intellectual property.



Промптинг: некоторые выводы / рекомендации

- Заранее предсказать, какой промпт наверняка сработает, а какой нет – нельзя,
- Даже **небольшие изменения** в промпте **могут влиять**,
- **few-shot** часто **повышает** качество,
- Если задача сложная, проверьте возможности модели на **CoT**,
- Если модель инструктивная и **корпус инструкций** доступен – посмотрите, на каких инструкциях она обучалась, попробуйте составить **схожий по структуре** промпт.

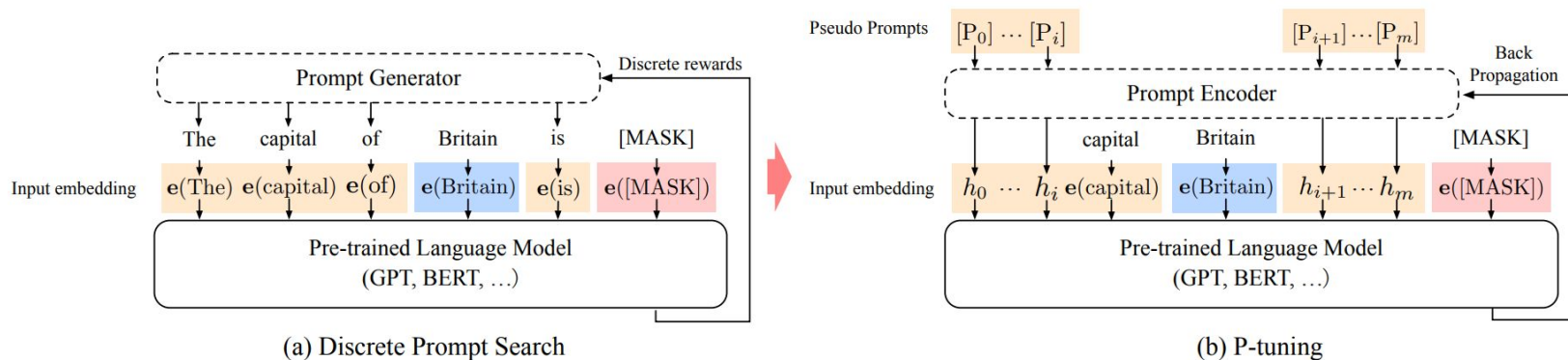
Дообучение

- Иногда zero-shot и few-shot не достаточно,
- Модель **13b** параметров при загрузке на видеокарту в **fp16** занимает **24gb** (и это даже не для инфера и не обучения), но хочется **fine tuning**?
 - **Multi-gpu** система, V100 / **A100**,
 - Использовать **DDP** (distributed data parellel), **gradient checkpointing**, **маленький batch size + accumulation gradients**,
 - Пакет **DeepSpeed** от Microsoft:
 - Zero Stage 1: состояние оптимизатора распределено на разные GPU,
 - Zero Stage 2: + градиенты распределены на разные GPU,
 - Zero Stage 3: + веса модели распределены на разные GPU,
 - Параметр сру offloading – выгружает часть данных с GPU на CPU в процессе обучения

P-tuning

- Альтернатива дообучению,
- Вместо подбора слов (токенов) промпта, **подбираются входные эмбединги** для нескольких токенов входного текста

GPT Understands, Too



Промт энкодер нужно обучать!

P-tuning: результаты

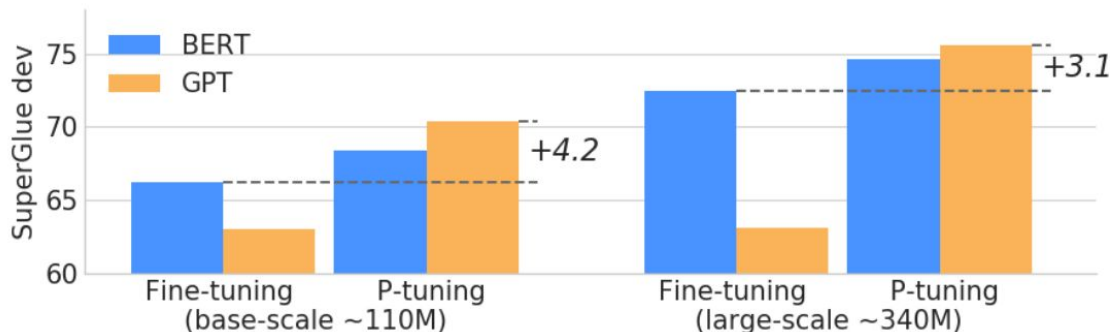
- Датасет LAMA-29
 - “пробинг” знаний моделей,

(Dante, born-in, Florence) ->
“Dante was born in [MASK].”

Model	MP	FT	MP+FT	P-tuning
BERT-base (109M)	31.7	51.6	52.1	52.3 (+20.6)
-AutoPrompt (Shin et al., 2020)	-	-	-	45.2
BERT-large (335M)	33.5	54.0	55.0	54.6 (+21.1)
RoBERTa-base (125M)	18.4	49.2	50.0	49.3 (+30.9)
-AutoPrompt (Shin et al., 2020)	-	-	-	40.0
RoBERTa-large (355M)	22.1	52.3	52.4	53.5 (+31.4)
GPT2-medium (345M)	20.3	41.9	38.2	46.5 (+26.2)
GPT2-xl (1.5B)	22.8	44.9	46.5	54.4 (+31.6)
MegatronLM (11B)	23.1	OOM*	OOM*	64.2 (+41.1)

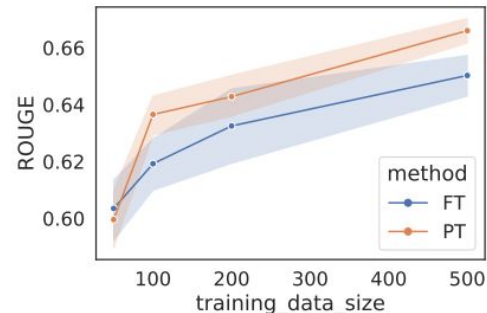
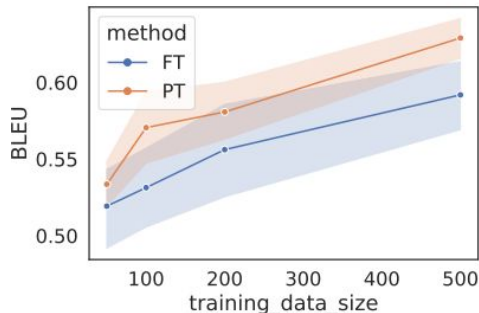
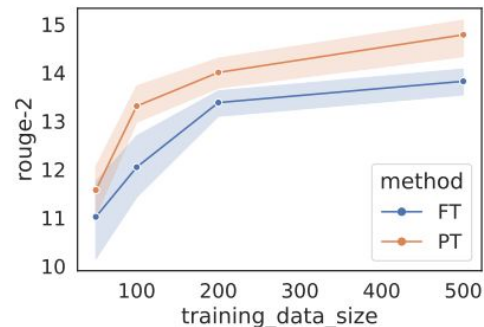
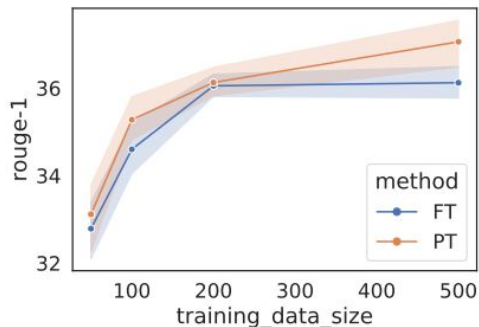
* MegatronLM (11B) is too large for effective fine-tuning.

- Бенчмарк SuperGlue



Prefix-tuning

- По сути та же самая идея, что и в P-tuning,
- Несколько другая архитектура получения эмбедингов,
 - В **P-tuning** это была небольшая сеть с **Istm**,
 - В **Prefix-tuning** это обучаемая матрица **P** (размерность меньше, чем hidden state) + MLP,
- Вышли +- в одно время, друг с другом не сравнивались.

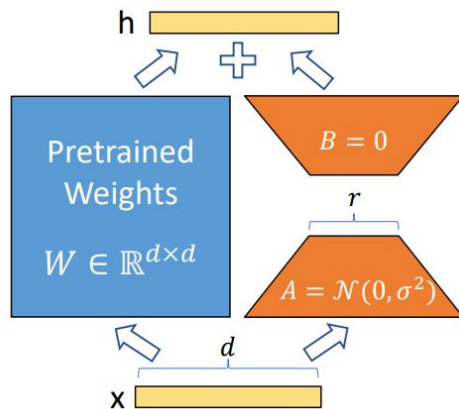


LoRa

- То, благодаря чему мы имеет saiga модели,
 - Первые версии **Saiga-7b** и **13b** обучались на всего одной **RTX 3090**,
- **Позволяет обучать** всю сеть, но при этом **уменьшая** количество обучаемых параметров **в 10,000 раз** (для GPT-3),
 - И требования к памяти GPU в 3 раза,
- Основная идея в том, чтобы не обучать все параметры модели, а только некоторую “добавку”, причем в **low-rank**,
- После слияния с моделью имеем новую модель **без дополнительных затрат** на работу.

LoRa: основная идея

$$W_0 + \Delta W = W_0 + BA, \text{ where } \bar{B} \in \mathbb{R}^{d \times r}, \bar{A} \in \mathbb{R}^{r \times k},$$



- Обучаются матрицы B и A ,
- Соответственно для каждого Q, K, V в attention можно применить подобный “трюк”.

LoRa: результаты

Model&Method	# Trainable Parameters	WikiSQL	MNLI-m	SAMSum
		Acc. (%)	Acc. (%)	R1/R2/RL
GPT-3 (FT)	175,255.8M	73.8	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 (Adapter ^H)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 (Adapter ^H)	40.1M	73.2	91.5	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	91.7	53.8/29.8/45.9
GPT-3 (LoRA)	37.7M	74.0	91.6	53.4/29.2/45.1

LoRa: какие матрицы дообучать

	# of Trainable Parameters = 18M						
Weight Type Rank r	W_q 8	W_k 8	W_v 8	W_o 8	W_q, W_k 4	W_q, W_v 4	W_q, W_k, W_v, W_o 2
WikiSQL ($\pm 0.5\%$)	70.4	70.0	73.0	73.2	71.4	73.7	73.7
MultiNLI ($\pm 0.1\%$)	91.0	90.8	91.0	91.3	91.3	91.3	91.7

Квантизация

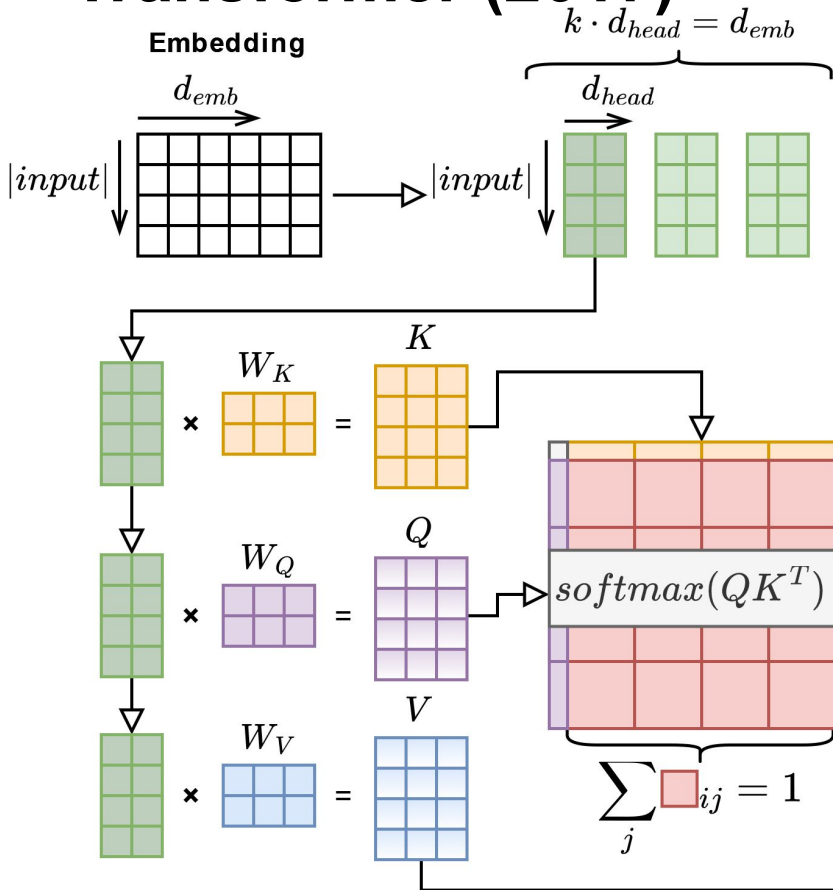
- Квантизация – еще один способ “уменьшить” модель, в частности для “инфера”,
- Суть в **преобразовании весов модели** из типа float32, float16 в **int8**, а иногда и в **int4**,
- Преобразование делается не просто “напрямую”, а более хитрым способом, но существуют уже реализации, совместимые с huggingface transformers,
 - **load_in_8bit**, **load_in_4bit** флаги,
- Закономерный результат квантизации – **падение качества**,
- Можно совмещать с обычным **LoRa**,
 - Замороженную модель в 8bit/4bit, но обучаемые веса в float16,
- Существует статья про **QLoRa** (ждем реализацию).

Заключение

- Даже несмотря на обучение на инструкциях, промптинг не стабилен,
- Для эффективной работы с большими языковыми моделями **необходимо** знать различные методы и ухищрения для того, чтобы уменьшить потребление памяти,
 - Квантизация,
 - Обучение только части весов,
 - P-tuning, prefix-tuning,
 - LoRa,
 - Также существует пруннинг (но не то, чтобы активно применяется для LLM сейчас),
- Также для **инференса** больших языковых моделей существуют специальные сильно оптимизированные программы,
 - так называемая llama.cpp (<https://github.com/ggerganov/llama.cpp>).

Спасибо за внимание!

Transformer (2017)



Q – запрос
K – ключи
V – значения
(слов)

