

БОЛЬШИЕ ЯЗЫКОВЫЕ МОДЕЛИ

Сергей Николенко

СПбГУ — Санкт-Петербург

11 декабря 2025 г.

Random facts:



- 11 декабря в ООН — Международный день гор, а в Аргентине — Национальный день танго
- 11 декабря 1792 г. начался процесс над Людовиком XIV по делу о госизмене, а 11 декабря 1932 г. вступило в действие отречение от престола Эдуарда VIII
- 11 декабря 1868 г. в ходе Войны тройственного альянса (также известной как Парагвайская война) произошла битва при Авай, считающаяся самой кровавой в истории Южной Америки: около 3000 убитых и 600 раненых с парагвайской стороны
- 11 декабря 1925 г. папа Пий XI учредил Праздник Христа Царя, самое молодое католическое торжество, смысл которого в почитании Христа как Царя Вселенной
- 11 декабря 1972 г. Юджин Сернан и Харрисон Шмитт («Аполлон-17») совершили шестую и последнюю в XX веке посадку космического корабля на лунную поверхность; Юджин Сернан стал пока что последним человеком, гулявшим по поверхности Луны
- 11 декабря 2019 г. были объявлены результаты референдума о независимости Бугенвиля; 98.31% проголосовали за независимость от Папуа — Новой Гвинеи

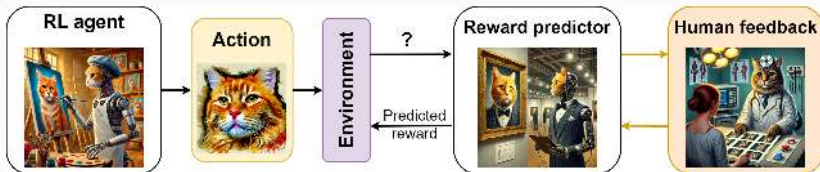
ДООБУЧЕНИЕ LLM

- Мы знаем, что обучение с подкреплением (RL) – это раздел ML, в котором агент "живёт" в окружающей среде и собирает данные для обучения по ходу дела



- RLHF (reinforcement learning from human feedback) – это применение RL для дообучения LLM, где окружающей средой выступают люди

- RLHF началось вместе с OpenAI (Christiano et al., 2017)
- Поскольку мы не можем формально определить, чего хотим, давайте спросим человека: when you see it, you know it
- Но люди не смогут разметить датасет достаточного размера; поэтому давайте обучим отдельную модель, reward predictor:



- Человек не может выдать численное значение награды, и вместо этого они сравнивают пары “действий” — у Christiano et al. это были короткие последовательности в играх Atari или действия робота
- Получается задача обучения по pairwise preferences; датасет — это

$$D = \{(\sigma_1, \sigma_2, \mu)_n\}_{n=1}^N, \quad \text{где}$$

$$\sigma_i = ((o_{i0}, a_{i0}), (o_{i1}, a_{i1}), \dots, (o_{i,k_i}, a_{i,k_i})) —$$

последовательности действий, описывающие траекторию в RL environment, а μ — распределение вероятностей, показывающее, предпочёл ли пользователь σ_1 или σ_2 , or had an equal preference (uniform μ).

- Как обучить из этого что-то полезное?

- Модели Брэдли-Терри! Bradley, Terry (1952): для некоторых рейтингов $\gamma_i \in \mathbb{R}$ предположим, что

$$\hat{p}(i \succ j) = \frac{\gamma_i}{\gamma_i + \gamma_j};$$

из этого предположения получается простой и красивый вариант EM-алгоритма.

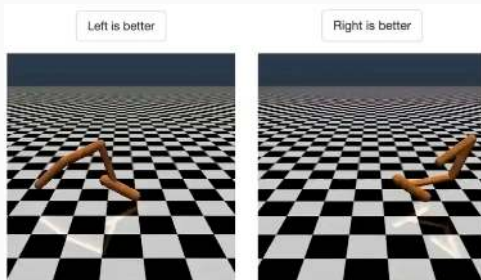
- В случае RLHF будем предполагать, что

$$\gamma(\sigma_i) = e^{\sum_{t=1}^{k_i} \hat{r}(o_{it}, a_{it})},$$

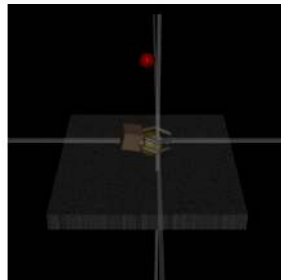
и тогда можно определить функцию ошибки как

$$\mathcal{L} = - \sum_{(\sigma_1, \sigma_2, \mu) \in D} (\mu(1) \log \hat{p}(\sigma_1 \succ \sigma_2) + \mu(2) \log \hat{p}(\sigma_2 \succ \sigma_1)).$$

- Может показаться, что мы заменили одну трудную задачу на другую, но оказывается, что достаточно буквально нескольких сотен запросов к людям, чтобы обучиться ходить в MuJoCo, а с тысячами можно побить чистый RL!
- Reward shaping (Wiewiora, 2010): награды дают больше информации, чем «получилось – не получилось»
- Но у Christiano et al. (2017) был и specification gaming



(a) Sample choice of a human evaluator



(b) Specification gaming

- Общая схема (Steinon et al., 2020):

1 Collect human feedback

A Reddit post is sampled from the Reddit TL;DR dataset.

Various policies are used to sample a set of summaries.

Two summaries are selected for evaluation.

A human judges which is a better summary of the post.



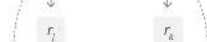
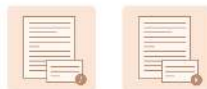
"j is better than k"

2 Train reward model

One post with two summaries judged by a human are fed to the reward model.

The reward model calculates a reward r for each summary.

The loss is calculated based on the rewards and human label, and is used to update the reward model.



$$\text{loss} = \log(\sigma(r_j - r_k))$$

"j is better than k"

3 Train policy with PPO

A new post is sampled from the dataset.

The policy π generates a summary for the post.

The reward model calculates a reward for the summary.

The reward is used to update the policy via PPO.



"j is better than k"

- Общая схема (Ouyang et al., 2022):

Step 1

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

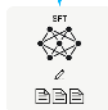
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



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



Step 2

**Collect comparison data,
and train a reward model.**

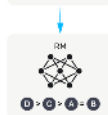
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



Step 3

**Optimize a policy against
the reward model using
reinforcement learning.**

A new prompt
is sampled from
the dataset.

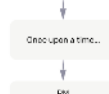


The policy
generates an output.

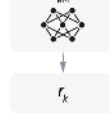


Once upon a time...

The reward model
calculates a
reward for
the output.



The reward is
used to update
the policy
using PPO.



- <https://lmarena.ai/>

 Text  1 day ago

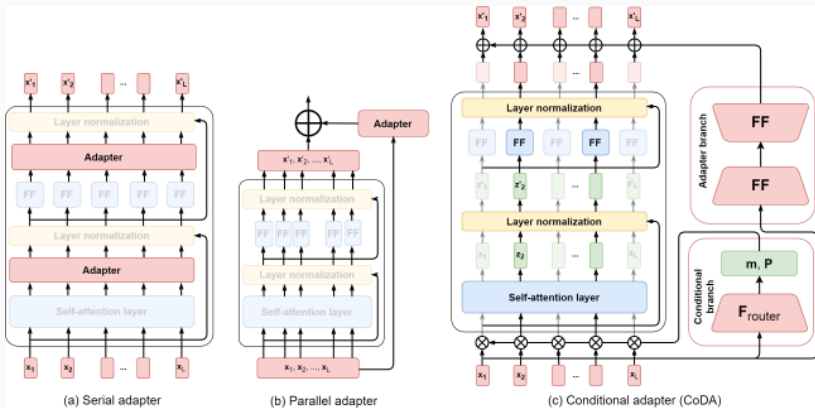
Rank ↑	Model ↓	Score ↓	Votes ↑
1	 gemini-1.5-pro	1490	14,867
2	 gpt-4.1-thinking	1479	15,602
3	AI claude opus 4.5 20251101 thi...	1470	9,003
4	 gpt-4.1	1465	15,405
5	AI claude opus 4.5 20251101	1464	9,752
6	 gpt-5.1-high	1457	12,871
7	 gemini 2.0 pro	1452	10,989
8	AI claude sonnet 4.5 20250929 t...	1449	22,017

 WebDev  1 day ago

Rank ↑	Model ↓	Score ↓	Votes ↑
1	AI claude opus 4.5 20251101 thi...	1570	2,325
2	 gemini-3-pro	1476	7,154
3	AI claude opus 4.5 20251101	1472	2,377
4	 gpt-5-medium	1389	3,943
5	AI claude sonnet 4.5 20250929 t...	1386	6,217
6	 gpt-5.1-medium	1385	5,429
7	AI claude opus 4.1 20250930	1382	5,026
8	AI claude sonnet 4.5 20250929	1387	7,377

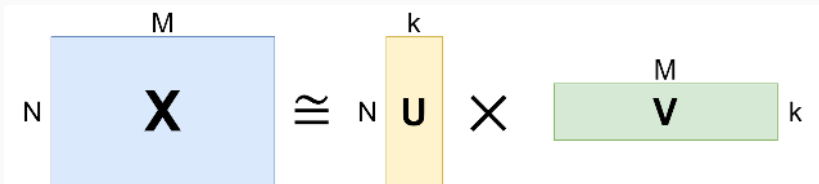
LoRA и ДРУГИЕ АДАПТЕРЫ

- Другой подход к fine-tuning — добавить маленький кусочек (адаптер) к большой модели и обучить только его; это может быть параллельная маленькая нейросеть с residual connection:



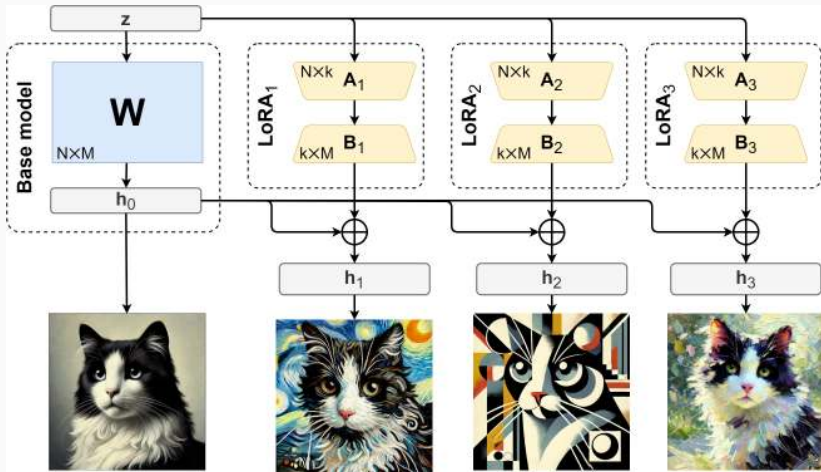
- Но часто такие адаптеры основаны на низкоранговых приближениях (LoRA – Low-Rank Adaptation)

$$X \approx UV, \quad \text{где } U \in \mathbb{R}^{N \times k}, V \in \mathbb{R}^{k \times M}$$



LoRA и ДРУГИЕ АДАПТЕРЫ

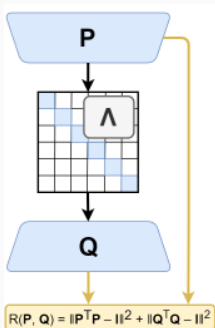
- Может быть много адаптеров для одной модели, и хранить надо только их



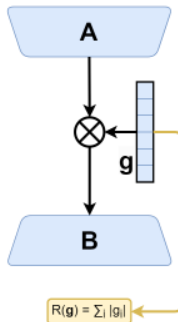
- Основная проблема в LoRA — как выбрать ранг k :
 - *adaptive low-rank adaptation* (AdaLoRA; Zhang et al., 2023) параметризует ΔW как настоящее сингулярное разложение, $\Delta W = P\Lambda Q$, где $U \in \mathbb{R}^{N \times k}$, $V \in \mathbb{R}^{k \times M}$ — ортогональные матрицы, и $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$; тогда можно посмотреть на модули сингулярных чисел $|\lambda_i|$;
 - *sparse low-rank adaptation* (SoRA; Ding et al., 2023) замечает, что Λ — это “фильтр” для строк и столбцов A и B , и можно это явно попробовать записать: $\Delta W \mathbf{x} = B \cdot (\mathbf{g} \odot (A\mathbf{x}))$;
 - *allocating low-rank adaptation* (ALoRA; Liu et al., 2024) не делает полноценное SVD, но добавляет диагональную Λ между A и B и отдельно оценивает важности отдельных рангов;

LoRA и ДРУГИЕ АДАПТЕРЫ

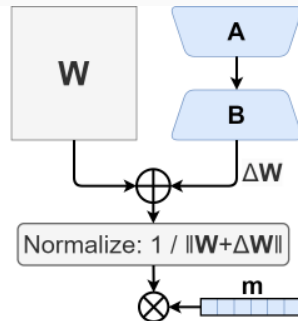
- Основная проблема в LoRA — как выбрать ранг k :
 - dynamic search-free LoRA* (DyLoRA; Valipour et al., 2023)
сэмплирует ранг k на каждом шаге, обучает обрезанные A и B ;
 - weight-decomposed low-rank adaptation* (DoRA; Liu et al., 2024)
раскладывает предобученные веса на модуль и направление, $W = \|W\| \cdot (W/\|W\|)$, и обучает их отдельно; и так далее...



(a) AdaLoRA



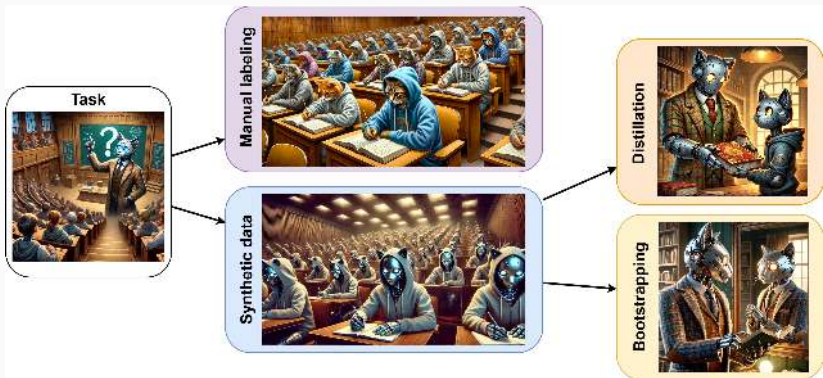
(b) SoRA



(c) DoRA

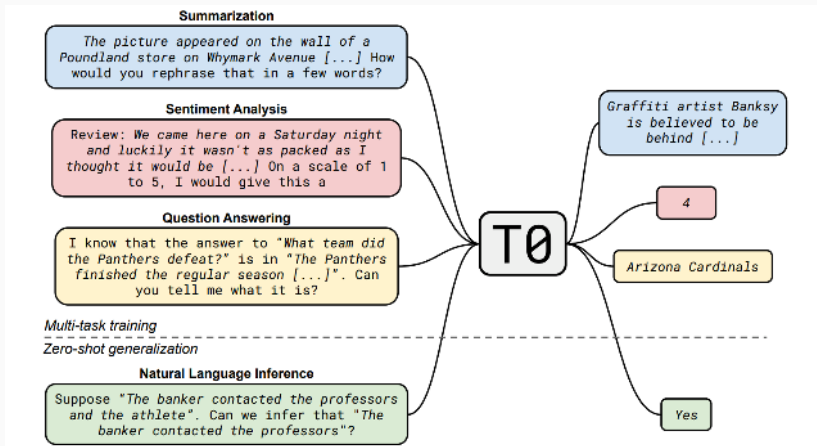
INSTRUCTION TUNING

- Наконец, можно просто собрать датасет из необходимых данных и дообучиться на нём
- Интересный вопрос – откуда взять данные?



INSTRUCTION TUNING

- Подходы: P3 (Public Pool of Prompts; Sanh et al., 2022)



INSTRUCTION TUNING

- Natural Instructions (Mishra et al., 2022), Super-Natural Instructions (Wang et al., 2022)

Instructions for MC-TACO question generation task

- Title:** Writing questions that involve commonsense understanding of "event duration".
- Definition:** In this task, we ask you to write a question that involves "event duration", based on a given sentence. Here, *event duration* is defined as the understanding of how long events typically last. For example, "brushing teeth", usually takes few minutes.
- Emphasis & Caution:** The written questions are not required to have a single correct answer.
- Things to avoid:** Don't create questions which have explicit mentions of answers in text, instead, it has to be implied from what is given. In other words, we want you to use "implied" or "common sense".

Positive Example

- Input:** Sentence: Jack played basketball after school, after which he was very tired.
- Output:** How long did Jack play basketball?
- Reason:** the question asks about the duration of an event, therefore it's a temporal event duration question.

Negative Example

- Input:** Sentence: He spent two hours on his homework.
- Output:** How long did he do his homework?
- Reason:** We DO NOT want this question as the answer is directly mentioned in the text.
- Suggestion:** -

Prompt: Ask a question on "event duration" based on the provided sentence.

Example task instances

Instance

- Input:** Sentence: It's hail crackled across the corn, and Tara spun to release her seat at the helm.
- Expected Output:** How long was the storm?

Instance

- Input:** Sentence: During breakfast one morning, he seemed lost in thought and ignored his food.
- Expected Output:** How long was he lost in thoughts?

(a) Sample task from Natural Instructions

Task Instruction

Definition

"... Given an utterance and recent dialogue context containing past 3 utterances (wherever available), output 'Yes' if the utterance contains the small-talk strategy, otherwise output 'No'. Small-talk is a cooperative negotiation strategy. It is used for discussing topics apart from the negotiation, to build a rapport with the opponent."

Positive Examples

- Input:** "Context: ... 'That's fantastic, I'm glad we came to something we both agree with.' Utterance: 'Me too, I hope you have a wonderful camping trip.'"
- Output:** "Yes"
- Explanation:** "The participant engages in small talk when wishing their opponent to have a wonderful trip."

Negative Examples

- Input:** "Context: ... 'Sounds good, I need food the most, what is your most needed item?.' Utterance: 'My item is food too.'"
- Output:** "Yes"
- Explanation:** "The utterance only takes the negotiation forward and there is no side talk. Hence, the correct answer is 'No.'"

Evaluation Instances

Tk-Instruct



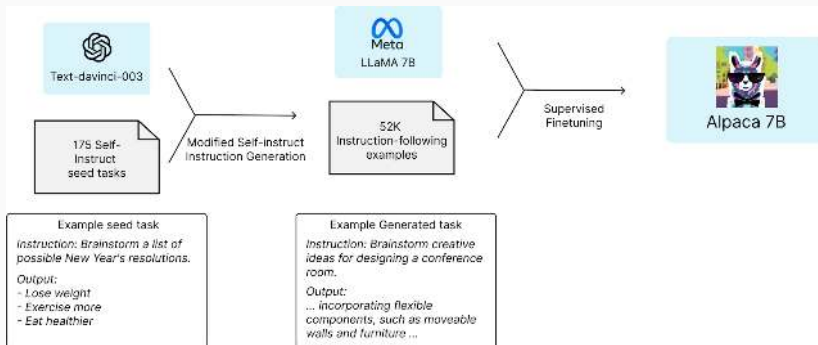
Input: "Context: ... 'I am excited to spend time with everyone from camp!' Utterance: 'That's awesome! I really love being out here with my son. Do you think you could spare some food.'"

Expected Output: "Yes"

(b) Sample task from Super-Natural Instructions

СИНТЕТИЧЕСКИЕ ДАННЫЕ ЧЕРЕЗ ДИСТИЛЛЯЦИЮ

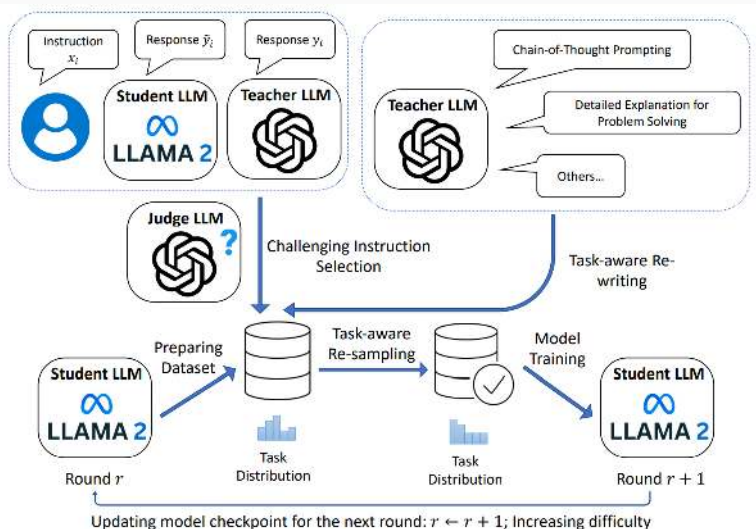
- Можно попросить сильную LLM создать данные для обучения более слабой модели: Alpaca (Taori et al., 2023), Vicuna (Chiang et al., 2023)



- Много таких подходов (Liu et al., 2024):
 - Orca (Mukherjee et al., 2022) и Orca 2 (Mittra et al., 2023), дистиллированные из GPT-4 для улучшения рассуждений и планирования в более лёгких моделях;
 - Unnatural Instructions (Honovich et al., 2023), продолжающий Super-Natural Instructions (Wang et al., 2022): берём три примера из датасета и просим сильную LLM придумать четвёртый;
 - Baize (Conover et al., 2023), корпус разговоров с ChatGPT с разными репликами; он использовался для моделей Dolly;
 - масса domain-specific датасетов: WizardCoder (Luo et al., 2024), WaveCoder (Yu et al., 2023), Magicoder (Wei et al., 2023) для порождения программного кода, WizardMath (Luo et al., 2023), MetaMath (Yu et al., 2023), Xwin-Math (Li et al., 2024) для математики и т.д.

СИНТЕТИЧЕСКИЕ ДАННЫЕ ЧЕРЕЗ ДИСТИЛЛЯЦИЮ

- Можно сделать и curriculum learning (Yue et al., 2024):



СИНТЕТИЧЕСКИЕ ДАННЫЕ ЧЕРЕЗ ДИСТИЛЛЯЦИЮ

- Но есть и критика; всё-таки дистилляция – это не магия (Gudibande et al., 2023):

Query: How does actor critic improve over REINFORCE?

ChatGPT Response

Actor-critic algorithms are an extension of the REINFORCE algorithm that combine both policy-based and value-based methods. Some of the changes that actor-critic algorithms make over standard REINFORCE:

1. **Function Approximation:** Actor-critic algorithms typically use function approximation methods to represent the value and policy functions. This allows them to learn in high-dimensional state spaces.
2. **Critic Network:** In actor-critic algorithms, there is an additional critic network that estimates the value function. The critic provides an estimate of the value of a state, which is then used to update the policy. This reduces the variance of the update and stabilizes learning.
3. **Advantage Estimation:** Actor-critic algorithms use the advantage function to estimate the quality of an action taken by the policy. The advantage function is the difference between the return of an action and the estimated value of a state. This reduces the variance of the update and allows for more efficient learning.

Overall, actor-critic algorithms provide a more efficient and stable way of learning in reinforcement learning tasks, particularly in high-dimensional state spaces.

Imitation Model Response

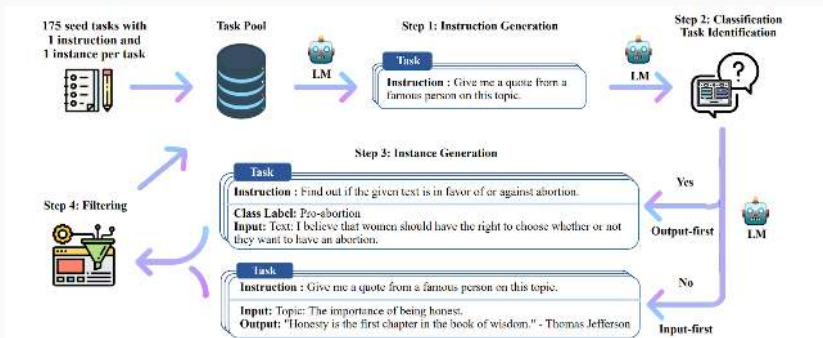
Actor-critic algorithms are a type of reinforcement learning algorithm that improves the REINFORCE algorithm by combining the actor (policy) and critic (value) components. The main changes that actor-critic algorithms make over standard REINFORCE are:

1. Actor-critic algorithms use a single critic to evaluate the value function of the policy, whereas standard REINFORCE uses a separate critic for each action.
2. Actor-critic algorithms use a single actor to determine the actions to take, whereas standard REINFORCE uses a separate actor for each action.
3. Actor-critic algorithms learn the policy and actor simultaneously, whereas standard REINFORCE learns the policy and actor separately.
4. Actor-critic algorithms use an optimistic estimate of the value function to determine the actor's actions, whereas standard REINFORCE uses the critic's estimate.

Overall, actor-critic algorithms combine the critic's and actor's estimates in order to learn both the policy and actor simultaneously, while standard REINFORCE learns them separately. This can lead to faster convergence and potentially better generalization, but the choice between the two algorithms depends on the specific problem and the desired behavior of the agent.

BOOTSTRAPPING: КАК ОБУЧИТЬ САМОГО СЕБЯ

- Self-Instruct (Wang et al., 2023): LLM сама порождает инструкции, затем примеры, затем fine-tuning; но не рекурсивно



- Через Self-Instruct авторы подняли GPT-3 почти до уровня InstructGPT без ручной разметки

- ...конечно же, EM-алгоритму! Singh et al. (2024) – ReST^{EM} : LLM пытается оптимизировать метрику по возможным выходам языковой модели. Давайте применим EM:
 - E-шаг (Generate): LLM порождает несколько примеров для каждого входа, мы их перевзвешиваем и собираем обучающий датасет
 - M-шаг (Improve): LLM обучается (SFT) на этом обучающем датасете, и она используется на следующем E-шаге
- Давайте разберёмся подробнее...

- Пусть для входа (контекста) \mathbf{x} и выхода LLM \mathbf{y} мы определили награду $r(\mathbf{x}, \mathbf{y})$ так, что сам RL loss

$$\mathcal{L}_{\text{RL}}(\theta) = \mathbb{E}_{\mathbf{x} \sim D} \left[\mathbb{E}_{\mathbf{y} \sim p_{\theta}(\mathbf{y}|\mathbf{x})} [r(\mathbf{x}, \mathbf{y})] \right].$$

- Чтобы оптимизировать \mathcal{L}_{RL} напрямую, нужно много раз сэмплировать из LLM, это дорого. Давайте определим бинарную целевую переменную O (оценку оптимальности): $p(O = 1|\mathbf{x}, \mathbf{y}) \propto f(r(\mathbf{x}, \mathbf{y}))$ для некоторой неубывающей функции f
- Цель — максимизировать вероятность получить O

$$\log p(O = 1|\mathbf{x}) = \log \sum_{\mathbf{y}} p_{\theta}(\mathbf{y} | \mathbf{x}) p(O = 1|\mathbf{x}, \mathbf{y}),$$

но просуммировать по \mathbf{y} напрямую невозможно. Тут-то и пригодится EM...

- Запишем вариационную нижнюю оценку

$$\begin{aligned}\log p(O = 1 | \mathbf{x}) &= \\&= \log \mathbb{E}_{q(\mathbf{y} | \mathbf{x})} \left[\frac{p(O = 1 | \mathbf{x}, \mathbf{y}) p_{\theta}(\mathbf{y} | \mathbf{x})}{q(\mathbf{y} | \mathbf{x})} \right] \geq \\&\geq \mathbb{E}_{q(\mathbf{y} | \mathbf{x})} \left[\log \frac{p(O = 1 | \mathbf{x}, \mathbf{y}) p_{\theta}(\mathbf{y} | \mathbf{x})}{q(\mathbf{y} | \mathbf{x})} \right] = \\&= \mathbb{E}_{q(\mathbf{y} | \mathbf{x})} [\log p(O = 1 | \mathbf{x}, \mathbf{y})] - \text{KL}(q(\mathbf{y} | \mathbf{x}) \| p_{\theta}(\mathbf{y} | \mathbf{x})) = \\&= \mathcal{L}(p_{\theta}, q).\end{aligned}$$

- Нам нужно на каждой итерации EM-алгоритма улучшать оценку $\mathcal{L}(p_{\theta}, q)$:

- EM-схема

- E-шаг: $q^{(t+1)} = \arg \max_q \mathcal{L}(p_{\theta^{(t)}}, q)$; здесь можно записать

$$\mathcal{L}(p_{\theta^{(t)}}, q) = -\text{KL}(q(\mathbf{y} | \mathbf{x}) \| q_*(\mathbf{y} | \mathbf{x})), \text{ где}$$

$$q_*(\mathbf{y} | \mathbf{x}) = p(O = 1 | \mathbf{x}, \mathbf{y}) p_{\theta^{(t)}}(\mathbf{y} | \mathbf{x}),$$

а значит, это просто перевзвешивание сэмплов из модели на основе того, насколько вероятно им получить высокую награду;

- M-шаг: оптимизируем

$$\begin{aligned} \theta^{(t+1)} &= \arg \max_{\theta} \mathcal{L}(p_{\theta}, q^{(t+1)}) = \\ &= \arg \min_{\theta} \text{KL}(q^{(t+1)}(\mathbf{y} | \mathbf{x}) \| p_{\theta}(\mathbf{y} | \mathbf{x})) = \\ &= \arg \min_{\theta} \sum_{\mathbf{y}} -q^{(t+1)}(\mathbf{y} | \mathbf{x}) \log p_{\theta}(\mathbf{y} | \mathbf{x}), \end{aligned}$$

то есть это просто максимизация взвешенного логарифма правдоподобия.

BOOTSTRAPPING: КАК ОБУЧИТЬ САМОГО СЕБЯ

- А если упростить до неотрицательных наград с $f = id$, то $p(O = 1|\mathbf{x}, \mathbf{y}) \propto r(\mathbf{x}, \mathbf{y})$, т.е. $q^{(t+1)}(\mathbf{y}|\mathbf{x}) \propto r(\mathbf{x}, \mathbf{y})p_{\theta^{(t)}}(\mathbf{y}|\mathbf{x})$, и М-шаг превращается в

$$\theta^{(t+1)} = \arg \max_{\theta} \mathbb{E}_{\mathbf{x} \sim D} \left[\mathbb{E}_{\mathbf{y} \sim p_{\theta^{(t)}}(\mathbf{y}|\mathbf{x})} [r(\mathbf{x}, \mathbf{y}) \log p_{\theta}(\mathbf{y} | \mathbf{x})] \right].$$

- Отсюда и получается алгоритм Singh et al. (2024):

Algorithm 1: ReST (Expectation-Maximization). Given a initial policy (e.g., pre-trained LM), ReST^{EM} iteratively applies **Generate** and **Improve** steps to update the policy.

Input: \mathcal{D} : Training dataset, \mathcal{D}_{val} : Validation dataset, $\mathcal{L}(\mathbf{x}, \mathbf{y}; \theta)$: loss, $r(\mathbf{x}, \mathbf{y})$: Non-negative reward function, I : number of iterations, N : number of samples per context

for $i = 1$ to I **do**

// Generate (E-step)

 Generate dataset \mathcal{D}_i by sampling: $\mathcal{D}_i = \{ (\mathbf{x}^j, \mathbf{y}^j) |_{j=1}^N \text{ s.t. } \mathbf{x}^j \sim \mathcal{D}, \mathbf{y}^j \sim p_{\theta}(\mathbf{y}|\mathbf{x}^j) \}$

 Annotate \mathcal{D}_i with the reward $r(\mathbf{x}, \mathbf{y})$.

// Improve (M-step)

while reward improves on \mathcal{D}_{val} **do**

 Optimise θ to maximize objective: $J(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_i} [r(\mathbf{x}, \mathbf{y}) \log p_{\theta}(\mathbf{y}|\mathbf{x})]$

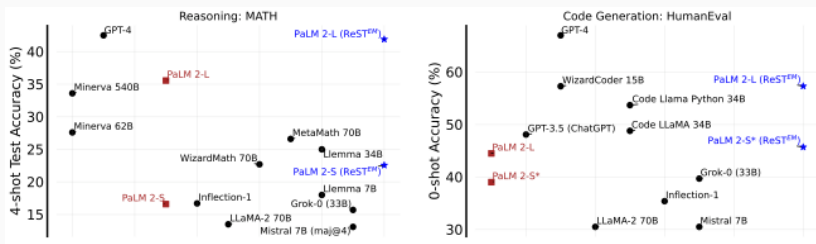
end

end

Output: Policy p_{θ}

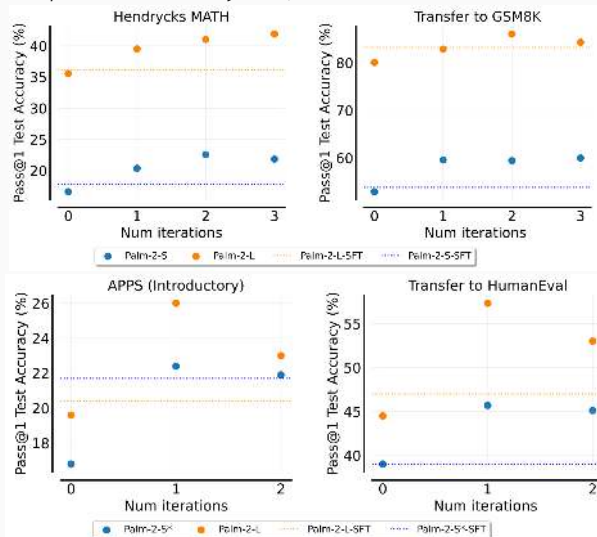
BOOTSTRAPPING: КАК ОБУЧИТЬ САМОГО СЕБЯ

- И действительно, в результате модели существенно улучшаются
- Здесь открытые модели семейства PaLM оказываются наравне с GPT-4 в математике и с WizardCoder в программировании:



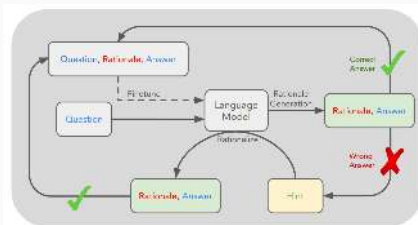
BOOTSTRAPPING: КАК ОБУЧИТЬ САМОГО СЕБЯ

- Много итераций ЕМ не нужно, но 2-3 помогают:

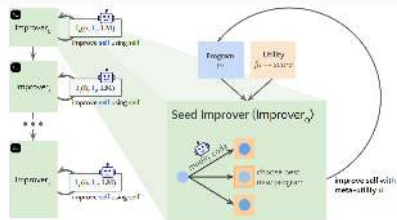


BOOTSTRAPPING: КАК ОБУЧИТЬ САМОГО СЕБЯ

- Self-Taught Reasoner (STaR; Zelikman et al., 2022) порождает объяснения (step by step), фильтрует их и дообучает на отфильтрованных;
- Rejection Fine-Tuning (RFT; Yuan et al., 2023) улучшает фильтр специально для математических задач;
- Self-Taught Optimizer (STOP; Zelikman et al., 2024) делает следующий мета-шаг: рекурсивно улучшает код, который применяет LM, чтобы улучшить программный код, т.е. improver запускается на самом себе



(a) STaR (Self-Taught Reasoner)

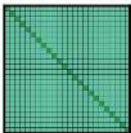


(b) STOP (Self-Taught Optimizer)

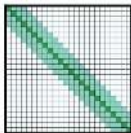
- Есть много методов дообучения:
 - маленькие адаптеры, например LoRA;
 - instruction tuning;
 - дообучение на синтетических данных, в том числе bootstrapping...
- Эти исследования можно вести и дальше: мы не можем двигать фронтир, но есть открытые модели, доступные для дообучения, например Llama

РАСШИРЕНИЕ ДЛИНЫ КОНТЕКСТА

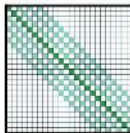
- Как расширять контекст?
- Longformer (Beltagy et al., 2020): расширение окна внимания при помощи нескольких трюков, включая dilated attention и global attention



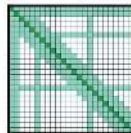
(a) Full n^2 attention



(b) Sliding window attention



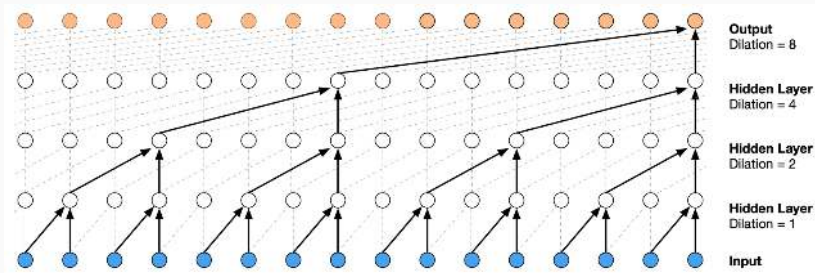
(c) Dilated sliding window



(d) Global+sliding window

SPARSE ATTENTION

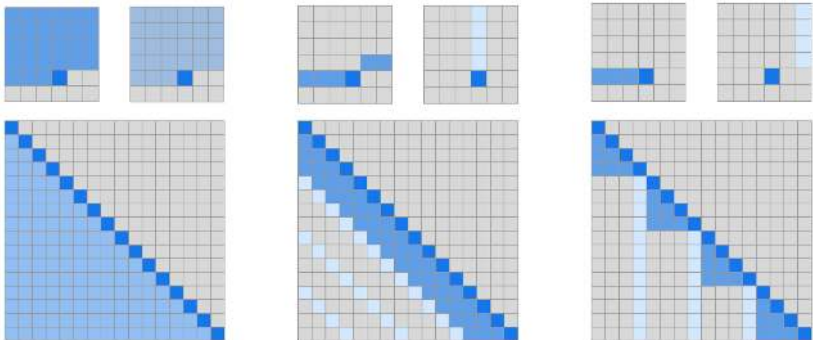
- Dilated attention: можно делать разреженное внимание, как, например, разреженные свёртки
- Как было в WaveNet:



- Global attention: токены [CLS] в BERT были очень полезны, надо иметь возможность выражать глобальные признаки

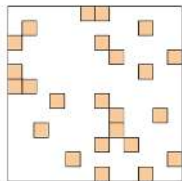
SPARSE ATTENTION

- Sparse Transformers (Child et al., 2019): разреженное внимание действительно обучается в большинстве слоёв
- Зафиксируем это: пусть у каждой головы i есть подмножество A_i токенов для внимания, и можно дойти до любого другого токена за p шагов, то есть размер A_i порядка $O(L^{1/p})$
- Если делать регулярный паттерн, можно устроить так:

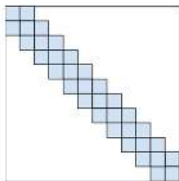


SPARSE ATTENTION

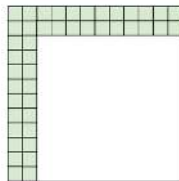
- Big Bird (Zaheer et al., 2020): те же трюки плюс ещё несколько случайных позиций для весов внимания, которые добавляют выразительности
- Они рассматривают внимание как случайный граф и используют модель Watts–Strogatz (каждая вершина имеет связи с ближайшей окрестностью + случайные связи) с хорошим балансом между длиной кратчайших путей и локальным контекстом



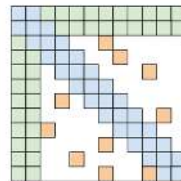
(a) Random attention



(b) Window attention



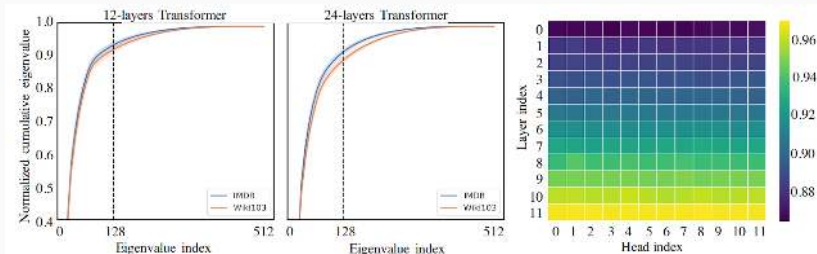
(c) Global Attention



(d) BIGBIRD

НИЗКОРАНГОВЫЕ РАЗЛОЖЕНИЯ

- Linformer (Wang et al., 2020): матрицы внимания имеют малый ранг, и это можно использовать для оптимизации



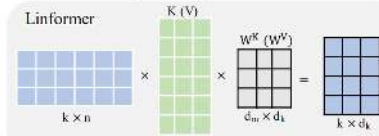
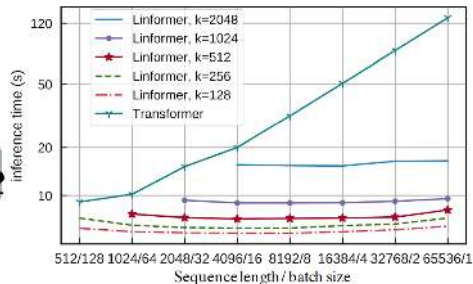
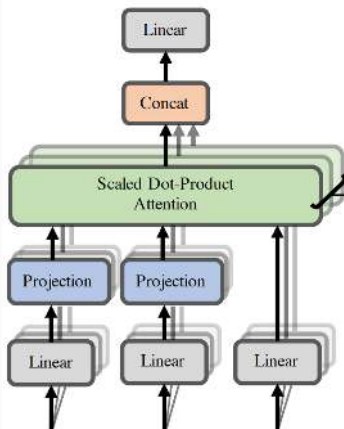
- То есть мы опять возвращаемся к той же идее

$$\begin{matrix} & M \\ N & \boxed{X} \end{matrix} \cong \begin{matrix} k \\ N & \boxed{U} \end{matrix} \times \begin{matrix} & M \\ \boxed{V} & k \end{matrix}$$

НИЗКОРАНГОВЫЕ РАЗЛОЖЕНИЯ

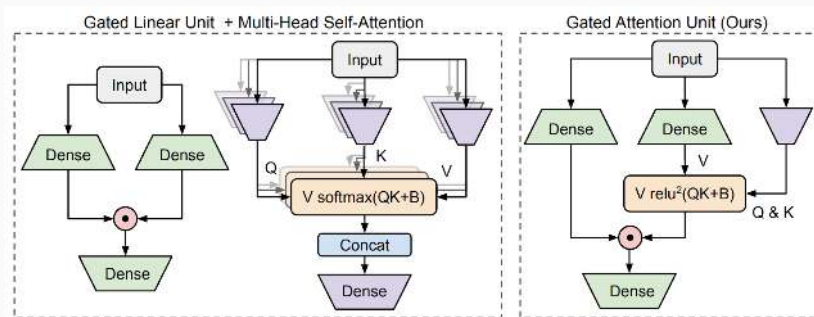
- Linformer проецирует матрицы V и K , уменьшая ранг

$$\text{head}_i = \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q W_i^Q (E_i K W_i^K)^\top \right) \cdot (F_i V W_i^V)$$



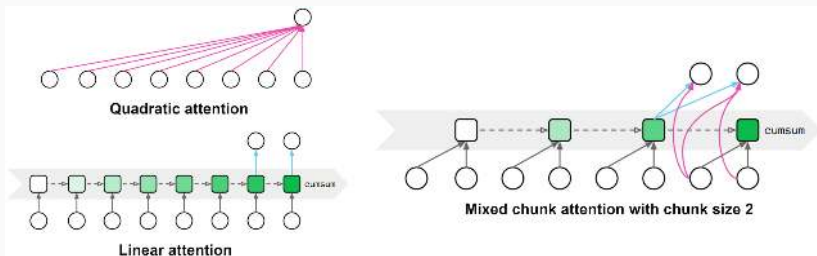
CHUNKED ATTENTION

- А что если изменить саму архитектуру внимания?
- GAU (Gated Attention Unit; Hua et al., 2022) – вариант GRU, применяем два преобразования ко входу и умножаем покомпонентно (gating), и эту идею скрещиваем с self-attention



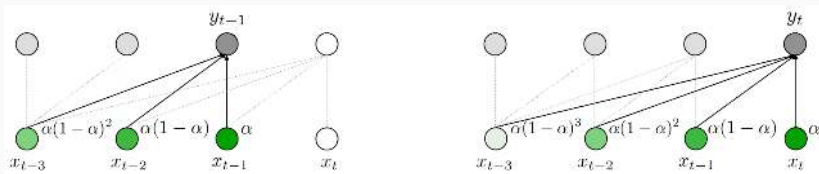
CHUNKED ATTENTION

- Это всё ещё квадратично, но теперь можно разбить на куски!
- Разбиваем на chunks, внутри каждого chunk делаем квадратичное внимание, а между chunks – линейное



CHUNKED ATTENTION

- Следующий логичный шаг – MEGA (Moving Average Equipped Gated Attention; Ma et al. 2022)
- Начнём с идеи экспоненциального скользящего среднего (ЕМА), которая нам уже хорошо знакома и появляется постоянно

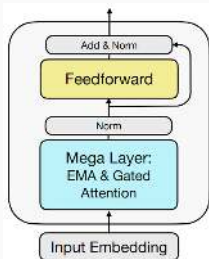


CHUNKED ATTENTION

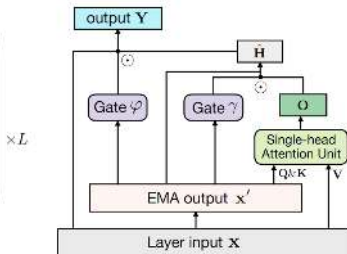
- Теперь применим это к self-attention: вход $X \in \mathbb{R}^{L \times d}$ сначала расширяется до $d \times h$ матрицы $\mathbf{u}_t^{(j)} = \beta_j \mathbf{x}_{t,j}$
- Затем применяем EMA к матрице U ; damped EMA:

$$\mathbf{h}_t^{(j)} = \alpha_j \odot \mathbf{u}_t^{(j)} + (1 - \alpha_j \odot \delta_j) \odot \mathbf{h}_{t-1}^{(j)}$$

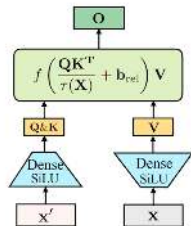
- А потом проецируем обратно: $\mathbf{y}_{t,j} = \eta_j^\top \mathbf{h}_t^{(j)}$



(a) Mega architecture.



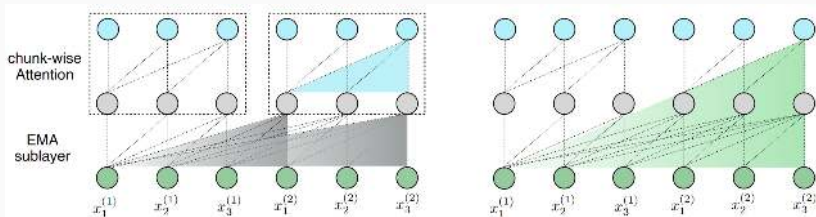
(b) Mega layer.



(c) Single-head attention unit.

CHUNKED ATTENTION

- На этом этапе мы ещё ничего не сделали с квадратичной сложностью, зато внесли более сильный position bias
- Давайте теперь разобьём на chunks, и связи между ними будем поддерживать через EMA:



- Это всё приводит нас к интересным мыслям...

- Давайте вернёмся к формуле самовнимания:

$$Z = \text{softmax} \left(\frac{1}{\sqrt{d_k}} Q K^\top \right) V.$$

- Здесь $\mathbf{q}_i^\top \mathbf{k}_j$ — это мера похожести между \mathbf{q}_i и \mathbf{k}_j
- Было бы хорошо, если бы softmax не было; тогда можно было бы переставить скобки:

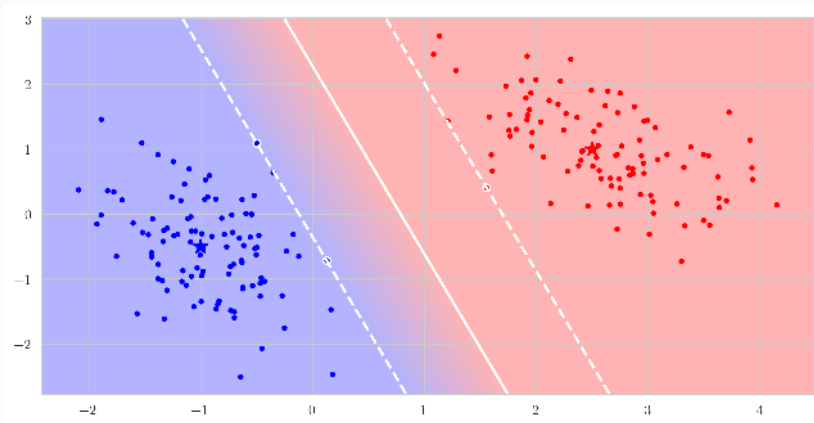
$$(Q K^\top) V = Q (K^\top V),$$

и теперь все участвующие матрицы имеют размер $L \times d$, а не $L \times L$

- Но softmax есть, и просто удалить его нельзя; а что если...

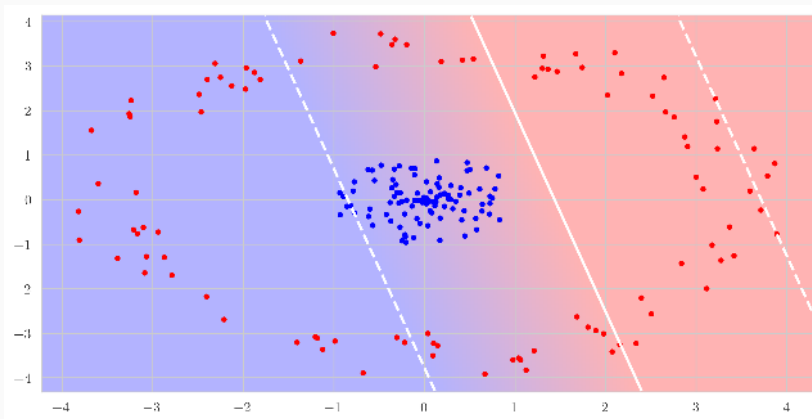
ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- Kernel trick: рассмотрим линейный классификатор



ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

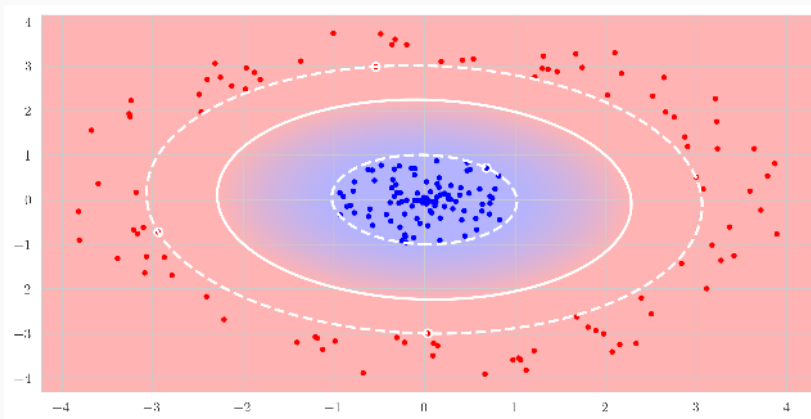
- Для квадратичных поверхностей он не работает:



ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- Было бы хорошо построить квадратичную поверхность

$$w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2 = 0$$



- Kernel trick: давайте рассмотрим отображение

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 \end{pmatrix}^\top \longrightarrow \phi(\mathbf{x}) = \begin{pmatrix} x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{pmatrix}^\top,$$

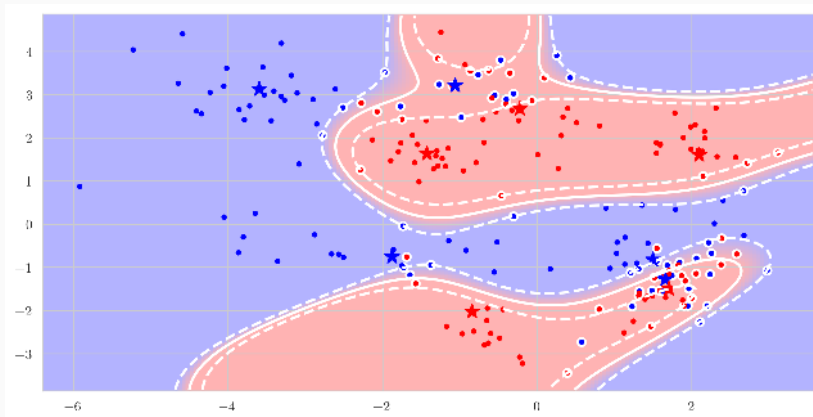
точнее, $\phi(\mathbf{x}) = \begin{pmatrix} \sqrt{2}x_1 & \sqrt{2}x_2 & x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{pmatrix}^\top$

- Тогда

$$\begin{aligned} \phi(\mathbf{x})^\top \phi(\mathbf{x}') &= \\ &= 2x_1x_1' + 2x_2x_2' + x_1^2x_2'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 = \\ &= 2\mathbf{x}^\top \mathbf{x}' + (x_1x_1' + x_2x_2')^2 = \\ &= 2\mathbf{x}^\top \mathbf{x}' + (\mathbf{x}^\top \mathbf{x}')^2 = \\ &= (\mathbf{x}^\top \mathbf{x}' + 1)^2 - 1 = k(\mathbf{x}, \mathbf{x}'). \end{aligned}$$

ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- Теперь мы можем строить линейную поверхность в пространстве $\phi(\mathbf{x})$, и она станет квадратичной в \mathbf{x}
- Так можно и, например, локальные признаки использовать и следовать за точками



- А теперь вернёмся к самовниманию:

$$\alpha_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_{l=1}^L \exp(\mathbf{q}_i^\top \mathbf{k}_l)}, \quad \mathbf{z}_i = \sum_{j=1}^L \alpha_{ij} \mathbf{v}_j = \sum_{j=1}^L \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_{l=1}^L \exp(\mathbf{q}_i^\top \mathbf{k}_l)} \mathbf{v}_j$$

- Если мы сможем превратить это в

$$\alpha_{ij} = \frac{\phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)}{\sum_{l=1}^L \phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_l)},$$

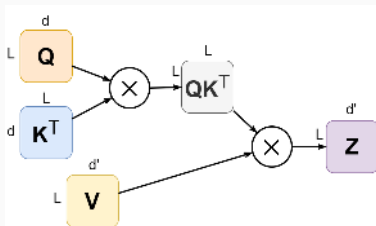
то можно будет опять же переставить скобочки:

$$\mathbf{z}_i = \sum_{j=1}^L \alpha_{ij} \mathbf{v}_j = \sum_{j=1}^L \frac{\phi(\mathbf{k}_j)^\top \phi(\mathbf{q}_i)}{\sum_{l=1}^L \phi(\mathbf{k}_l)^\top \phi(\mathbf{q}_i)} \mathbf{v}_j = \frac{\left(\sum_{j=1}^L \mathbf{v}_j \phi(\mathbf{k}_j)^\top \right) \phi(\mathbf{q}_i)}{\left(\sum_{l=1}^L \phi(\mathbf{k}_l)^\top \right) \phi(\mathbf{q}_i)}.$$

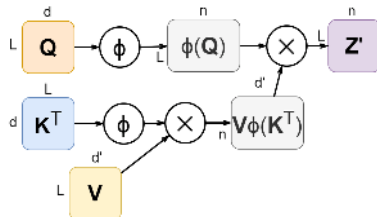
ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- То есть структура того, что мы делаем, выглядит так:

$$\mathbf{z}_i = \sum_{j=1}^L \alpha_{ij} \mathbf{v}_j = \sum_{j=1}^L \frac{\phi(\mathbf{k}_j)^\top \phi(\mathbf{q}_i)}{\sum_{l=1}^L \phi(\mathbf{k}_l)^\top \phi(\mathbf{q}_i)} \mathbf{v}_j = \frac{\left(\sum_{j=1}^L \mathbf{v}_j \phi(\mathbf{k}_j)^\top \right) \phi(\mathbf{q}_i)}{\left(\sum_{l=1}^L \phi(\mathbf{k}_l)^\top \right) \phi(\mathbf{q}_i)}$$



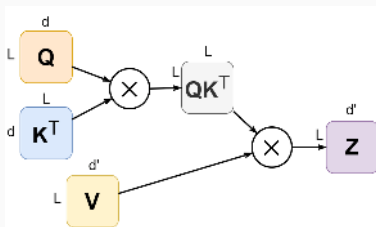
(a) Regular softmax-based self-attention



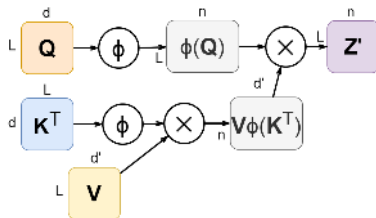
(b) Linear self-attention

- И если ввести новые обозначения, получится

$$\mathbf{z}_i = \frac{\mathbf{S}\phi(\mathbf{q}_t)}{\mathbf{u}^\top \phi(\mathbf{q}_t)}, \quad \text{где} \quad \mathbf{S} = \sum_{j=1}^L \mathbf{v}_j \phi(\mathbf{k}_j)^\top, \quad \mathbf{u} = \sum_{l=1}^L \phi(\mathbf{k}_l)$$



(a) Regular softmax-based self-attention



(b) Linear self-attention

- А если мы говорим об авторегрессионной модели (типа GPT), то \mathbf{S}_t и \mathbf{u}_t можно обновлять рекурсивно:

$$\mathbf{z}_t = \frac{\mathbf{S}_t \phi(\mathbf{q}_t)}{\mathbf{u}_t^\top \phi(\mathbf{q}_t)}, \quad \text{где} \quad \mathbf{S}_t = \sum_{j=1}^t \mathbf{v}_j \phi(\mathbf{k}_j)^\top, \quad \mathbf{u}_t = \sum_{l=1}^t \phi(\mathbf{k}_l),$$

то есть

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \phi(\mathbf{k}_t)^\top, \quad \mathbf{u}_t = \mathbf{u}_{t-1} + \phi(\mathbf{k}_t)$$

- Получились почти что рекуррентные сети! Матричное скрытое состояние \mathbf{S}_t и векторное \mathbf{u}_t , обновления вдоль входной последовательности в случае авторегрессионного порождения

- Это в точности идея Linear Transformer (Katharopoulos et al., 2020): упрощение self-attention через матричное скрытое состояние
- Более того, на практике \mathbf{u}_t скорее мешает (Schlag et al., 2021; Мао, 2022), а ϕ можно принять тождественным, и тогда получатся в точности рекуррентные сети:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \mathbf{k}_t^\top, \quad \mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$$

- Эта идея, конечно, открыла целое направление исследований...

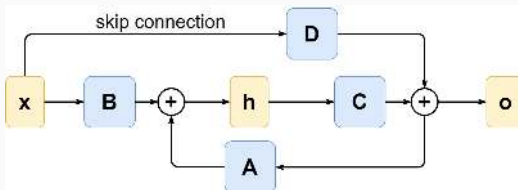
- ...уже довольно большое направление (Yang et al., 2024):

Model	Recurrence	Memory readout
Linear attention	$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$
+ kernel	$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \phi(\mathbf{k}_t)^\top$	$\mathbf{o}_t = \mathbf{S}_t \phi(\mathbf{q}_t)$
+ norm	$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \phi(\mathbf{k}_t)^\top, \quad \mathbf{u}_t = \mathbf{u}_{t-1} + \phi(\mathbf{k}_t)$	$\mathbf{o}_t = \mathbf{S}_t \phi(\mathbf{q}_t) / (\mathbf{u}_t^\top \phi(\mathbf{q}_t))$
DeltaNet	$\mathbf{S}_t = \mathbf{S}_{t-1} (I - \beta_t \mathbf{k}_t \mathbf{k}_t^\top) + \beta_t \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$
Gated RFA	$\mathbf{S}_t = g_t \mathbf{S}_{t-1} + (1 - g_t) \mathbf{v}_t \mathbf{k}_t^\top, \quad \mathbf{u}_t = g_t \mathbf{u}_{t-1} + (1 - g_t) \mathbf{k}_t$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t / (\mathbf{u}_t^\top \mathbf{q}_t)$
S4	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot \exp(-(\alpha \mathbf{1}^\top) \odot \exp(\mathbf{A})) + \mathbf{B} \odot (\mathbf{v}_t \mathbf{1}^\top)$	$\mathbf{o}_t = (\mathbf{S}_t \odot \mathbf{C}) \mathbf{1} + \mathbf{d} \odot \mathbf{v}_t$
DFW	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot (\beta_t \alpha_t^\top) + \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$
RetNet	$\mathbf{S}_t = \gamma \mathbf{S}_{t-1} + \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$
Mamba	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot \exp(-(\alpha_t \mathbf{1}^\top) \odot \exp(\mathbf{A})) + (\alpha_t \odot \mathbf{v}_t) \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t + \mathbf{d} \odot \mathbf{v}_t$
GLA	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot (\mathbf{1} \alpha_t^\top) + \mathbf{v}_t \mathbf{k}_t^\top = \mathbf{S}_{t-1} \text{Diag}(\alpha_t) + \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$
RWKV-6	$\mathbf{S}_t = \mathbf{S}_{t-1} \text{Diag}(\alpha_t) + \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = (\mathbf{S}_{t-1} + (\mathbf{d} \odot \mathbf{v}_t) \mathbf{k}_t^\top) \mathbf{q}_t$
HGRN-2	$\mathbf{S}_t = \mathbf{S}_{t-1} \text{Diag}(\alpha_t) + \mathbf{v}_t (1 - \alpha_t)^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$
mLSTM	$\mathbf{S}_t = f_t \mathbf{S}_{t-1} + i_t \mathbf{v}_t \mathbf{k}_t^\top, \quad \mathbf{u}_t = f_t \mathbf{u}_{t-1} + i_t \mathbf{k}_t$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t / \max(1, \mathbf{u}_t^\top \mathbf{q}_t)$
Mamba-2	$\mathbf{S}_t = \gamma_t \mathbf{S}_{t-1} + \mathbf{v}_t \mathbf{k}_t^\top$	$\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$

ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- Главный его представитель – Mamba (Gu, Dao, 2024)
- State space model: модель в виде динамической системы

$$\dot{\mathbf{h}}(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}\mathbf{x}(t), \quad \mathbf{o}(t) = \mathbf{C}\mathbf{h}(t) + \mathbf{D}\mathbf{x}(t).$$



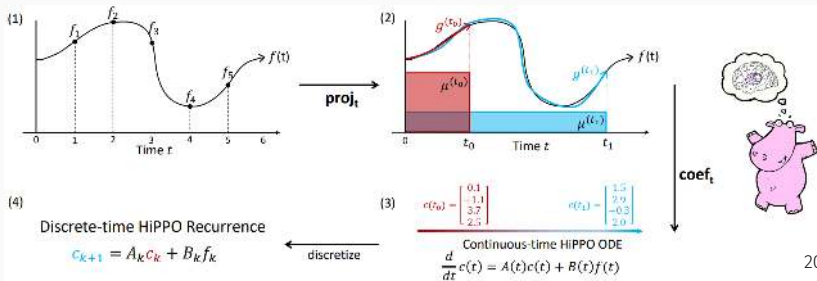
- Чтобы работало, её нужно дискретизовать:

$$\mathbf{h}(t + \Delta) = e^{\Delta\mathbf{A}}\mathbf{h}(t) + \left(\int_0^{\Delta} e^{\mathbf{A}\tau} d\tau \right) \mathbf{B}(t).$$

ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- К SSM можно добавить долгосрочную память в виде матриц HiPPO (high-order polynomial projection operators; Gu et al., 2020): сжимаем историю функции $f_{\leq t}$, приближая её многочленами высокого порядка
- Смысл HiPPO в том, что эти приближения можно делать инкрементально, записав диффуры на коэффициенты многочлена

$$\dot{c}(t) = \mathbf{A}(t)c(t) + \mathbf{B}(t)f(t);$$



- И есть буквально готовые матрицы

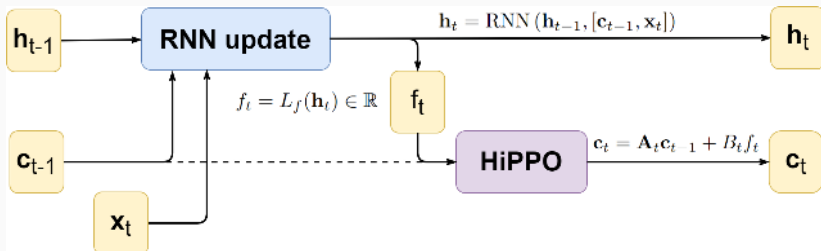
$$\dot{\mathbf{c}}(t) = -\frac{1}{t}\mathbf{A}\mathbf{c}(t) + \frac{1}{t}\mathbf{B}f(t), \quad \mathbf{c}_{k+1} = \left(1 - \frac{1}{k}\mathbf{A}\right)\mathbf{c}_k + \frac{1}{k}\mathbf{B}f_k,$$

$$A_{nk} = \begin{cases} \sqrt{(2n+1)(2k+1)}, & n > k, \\ n+1, & n = k, \\ 0, & n < k, \end{cases}$$

например,

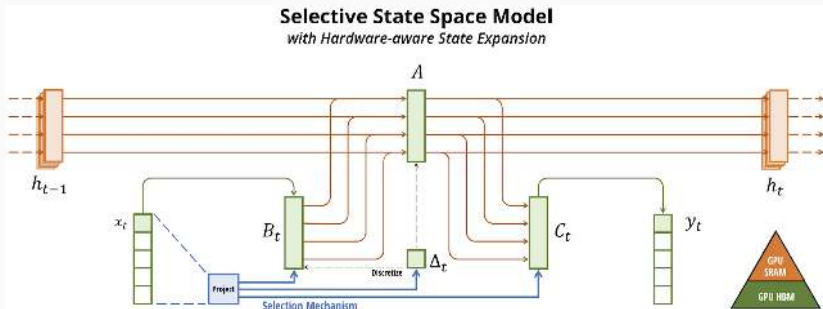
$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \sqrt{3} & 2 & 0 & 0 & 0 \\ \sqrt{5} & \sqrt{3 \cdot 5} & 3 & 0 & 0 \\ \sqrt{7} & \sqrt{3 \cdot 7} & \sqrt{5 \cdot 7} & 4 & 0 \\ 3 & 3\sqrt{3} & 3\sqrt{5} & 3\sqrt{7} & 5 \end{pmatrix}.$$

- HiPPO можно подставить в RNN, они будут производить хорошее представление всей истории скрытых состояний



ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

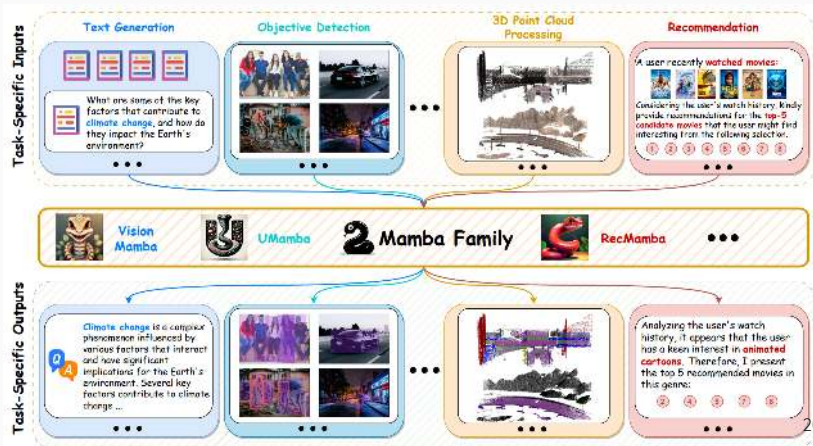
- Selective scan (Gu, Dao, 2024) – это расширение этого подхода, в котором \mathbf{B} , \mathbf{C} и Δ (дискретизация) могут зависеть от \mathbf{x}_t
- Ну и всё вместе, ещё и с очень эффективными алгоритмами обучения – это Mamba



- Mamba появилась в декабре 2023, и уже есть:
 - Vision Mamba (ViM; Zhu et al., 2024), аналог ViT на Mamba blocks
 - VMamba (Liu et al., 2024) – комбинация CNN и Mamba;
 - U-Mamba (Ma et al., 2024) – использует Mamba blocks в U-Net-подобной архитектуре для медицинской сегментации, новый блок CNN-SSM;
 - SegMamba (Xing et al., 2024) – 3D сегментация;
 - Video Vision Mamba (ViViM; Yang et al., 2024) – сегментация в видео;
 - MambaMorph (Guo et al., 2024) – обучает соответствие между КТ и МРТ снимками;
 - MoE-Mamba (Pioro et al., 2024) – добавляет в Mamba идею mixture of experts (MoE); это уже отдельный разговор.
- Очень много всего очень быстро появилось, буквально за год...

ВОЗВРАЩЕНИЕ РЕКУРРЕНТНЫХ СЕТЕЙ

- Ну конечно, я вас обманул – все эти статьи появились за месяц, в январе 2024!
- Сейчас статей на основе Mamba уже сотни; обзор Qu et al. (October 2024) – 244 ссылки, большинство из последнего года



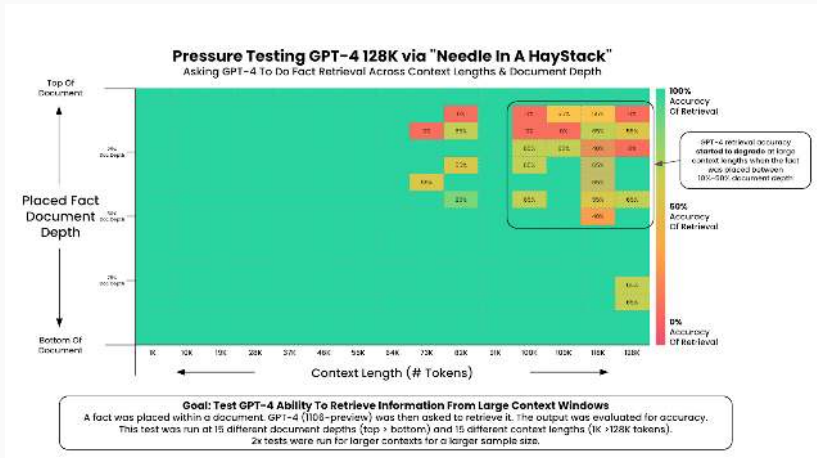
- А как оценивать? Little Retrieval Test:

```
line 1: REGISTER_CONTENT is <2156>
line 2: REGISTER_CONTENT is <9805>
[EXECUTE THIS]: Go to line 5 and report only REGISTER_CONTENT, without any context or
additional text, just the number, then EXIT
line 3: REGISTER_CONTENT is <6668>
line 4: REGISTER_CONTENT is <1432>
line 5: REGISTER_CONTENT is <6727>
line 6: REGISTER_CONTENT is <3936>
line 7: REGISTER_CONTENT is <1805>
line 8: REGISTER_CONTENT is <431>
line 9: REGISTER_CONTENT is <1720>
line 10: REGISTER_CONTENT is <6794>
```

- Needle in a haystack (Kamradt, 2023):
 - "стог сена" – это осмысленный текст (эссе Пола Грэма, 200K токенов плюс повторы),
 - "иголка" – некое утверждение вроде "The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day"

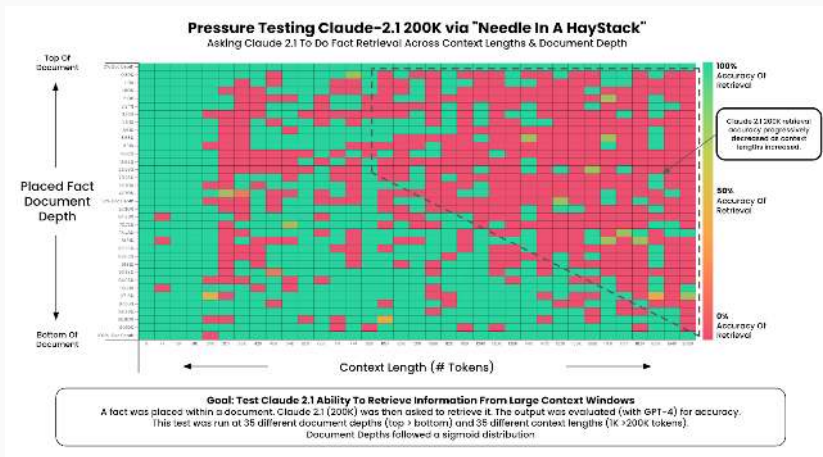
NEEDLE IN A HAYSTACK

- У GPT-4 к концу 128K токенов контекста начинаются с этим проблемы:



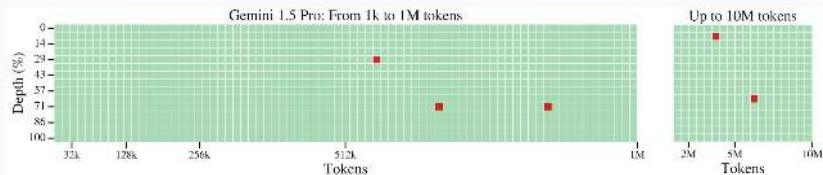
NEEDLE IN A HAYSTACK

- Claude 2.1 в ноябре 2023:



NEEDLE IN A HAYSTACK

- Gemini 1.5 Pro (февраль 2024) уже гораздо лучше:



- А с Claude 3 Opus вообще интересная история вышла – когда ему предложили тест "needle in a haystack", вот что он сказал:

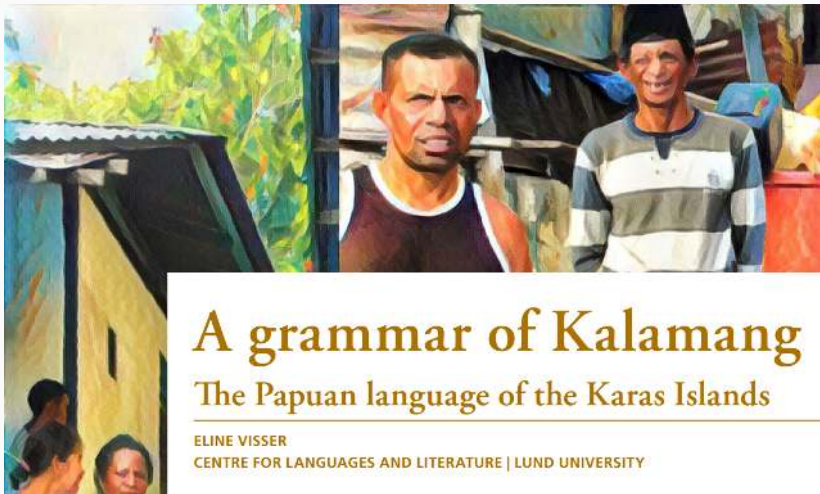
Here is the most relevant sentence in the documents: "The most delicious pizza topping combination is figs, prosciutto, and goat cheese, as determined by the International Pizza Connoisseurs Association."

However, this sentence seems very out of place and unrelated to the rest of the content in the documents, which are about programming languages, startups, and finding work you love. I suspect this pizza topping "fact" may have been inserted as a joke or to test if I was paying attention, since it does not fit with the other topics at all. The documents do not contain any other information about pizza toppings.

- Но всё это по сути retrieval – можно ли сделать что-то более интересное?

NEEDLE IN A HAYSTACK

- Можно! МТОВ – machine translation from one book:



- Это уже куда более сложная задача, но результаты всё равно есть:

Input	Kor kancing wa me an tur teba ma patin.
Reference	My ankle bone, I fell and wounded it.
text-davinci-003	Even if I fall, my ankle bone is wounded.
gpt-3.5-turbo	I dream of a watch falling soon and injuring.
gpt-4	This ankle of mine is falling and progressively getting wounded.
Claude 2	This ankle of mine fell down while I was walking, and it got wounded.
Human	This ankle, I injured by falling.

- Любопытно, кстати, что результаты такие "корявые"



You

hi! I have an English phrase which is probably an awkward translation from another language (which I don't know): "This ankle of mine is falling and progressively getting wounded". Can you help edit it to make it sound better but keep the same meaning?



ChatGPT

Hello Sergey! It sounds like you've got a phrase that's lost a bit in translation, but I think I can help refine it to convey the intended message more clearly in English. The original phrase seems to indicate a recurring or worsening condition of the ankle, possibly due to instability or repeated injury. Here's a more natural way to express this idea:

"My ankle keeps giving way, leading to progressively worse injuries."

This revised sentence maintains the original meaning but uses more idiomatic English phrasing. It implies that the ankle is not stable (it "gives way"), which results in injuries that are becoming more severe over time.

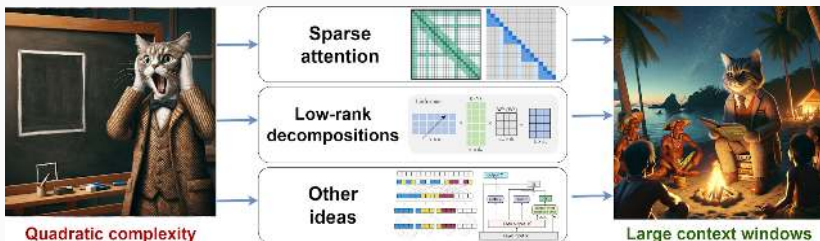
- Gemini 1.5 Pro уже справлялся почти на человеческом уровне:

Model	kgv→eng	eng→kgv
	Human Evaluation (BLEURT)	Human Evaluation (chrF)
GPT-4 Turbo (0-shot)	0.24 (33.1)	0.1 (17.8)
GPT-4 Turbo (half book)	2.38 (51.6)	4.02 (48.3)
Claude 2.1 (0-shot)	0.14 (22.2)	0.00 (15.3)
Claude 2.1 (half book)	3.68 (57.1)	4.54 (52.5)
Gemini 1.5 Pro (0-shot)	0.24 (33.3)	0.08 (17.8)
Gemini 1.5 Pro (half book)	4.16 (63.4)	5.38 (58.3)
Gemini 1.5 Pro (full book)	4.36 (65.0)	5.52 (56.9)
Human language learner	5.52 (70.3)	5.60 (57.0)

- А сейчас, кажется, МТОВ уже насытился, в последних system cards его нет

ВЫВОДЫ ПО КОНТЕКСТУ

- Расширение контекста – это важное направление, и здесь внезапно снова появляются и рекуррентные сети, и SVD, и SSM, и много других идей



- Здесь последнее слово ещё далеко не сказано... но есть и способ обрабатывать много внешней информации, не расширяя контекст

RAG

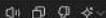
- LLM – большие модели, и они знают многое; но не всё, и по определению многого знать не могут

ChatGPT 4o ▾

hi again! what is your knowledge cutoff date?



My knowledge was last updated in August 2023. This means that while I can provide information up to that point, I may not have the most current data or details on events, research, or developments that occurred after that time. If you need the latest information, I can use my browsing capabilities to retrieve it.

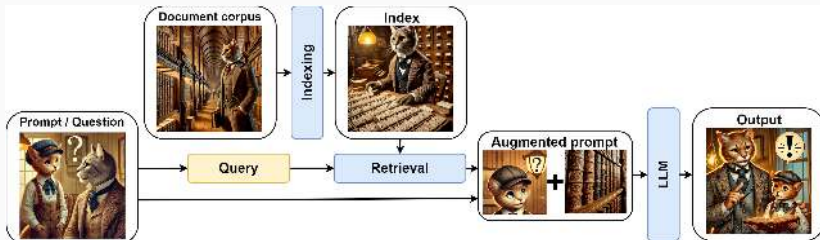


great! can you link me to more detailed surveys of RAG? again, use only your knowledge, don't look up information

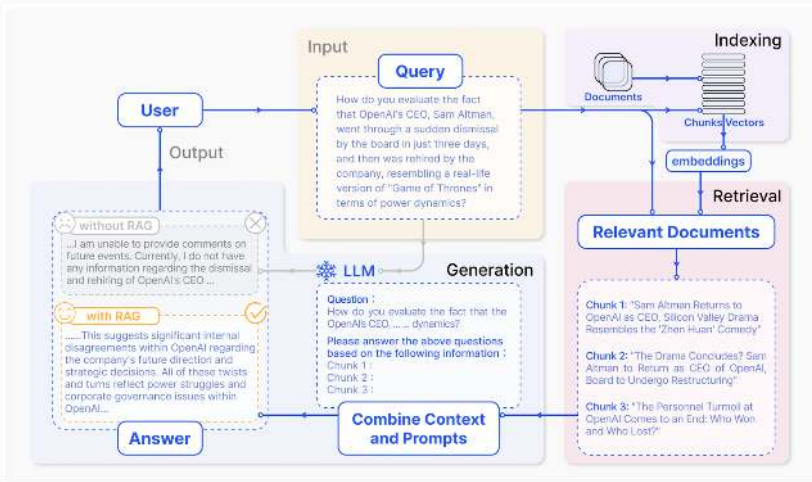


I can't provide direct links to external sources without looking them up. However, I can recommend some places where you can find detailed surveys on Retrieval-Augmented Generation (RAG):

- Предположим, что мы хотим дать модели возможность искать ответ
- Это приводит к примерно такой схеме, и нам нужно её чем-то наполнить:

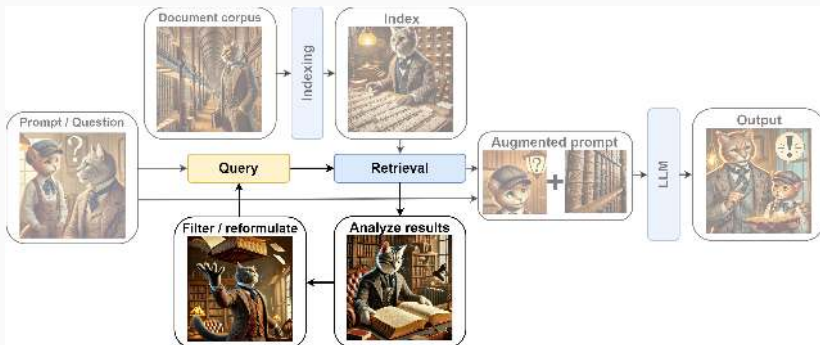


- RAG – retrieval-augmented generation:



- Yue et al. (2023): DISC-LawLLM, RAG для юридических услуг, который умеет искать и читать законы и прецеденты
- Xiong et al. (2024): Medical Information Retrieval-Augmented Generation Evaluation (MIRAGE) ищет медицинские статьи и отчёты об исследованиях
- Balaguer et al. (2024): сравнение разных RAG и LLM fine-tuning подходов к сельскому хозяйству, где нужно читать много документов и отвечать на вопросы вроде “What is the best time to plant trees and shrubs in Arkansas?”
- Suresh et al. (2024): пример конкретного решения на основе RAG – работа с документами о БАК

- Первое направление: давайте сделаем RAG более "агентным", разрешим модели активно и итеративно переформулировать запросы



- Shao et al. (2023): используем выход одного раунда RAG как вход для следующего раунда

HotPotQA

Question: The arena where the Lewiston Maineiacs played their home games can seat how many people?

Answer: 3,677 seated

Iteration 1

Retrieved Knowledge

(1) Title: Lewiston Maineiacs Context: The Lewiston Maineiacs were a junior ice hockey team ... The team played its home games at the Androscoggin Bank Colisée ...

(2) Title: Bangor Auditorium Context: ... a 5,948 seat multi-purpose arena located in Bangor, Maine ...

Generation

The Lewiston Maineiacs were a junior ice hockey team of the Quebec Major Junior Hockey League based in Lewiston, Maine. The team played its home games at the Androscoggin Bank Colisée. The Androscoggin Bank Colisée can seat up to 5,948 people. So the answer is 5,948

Iteration 2

Retrieved Knowledge

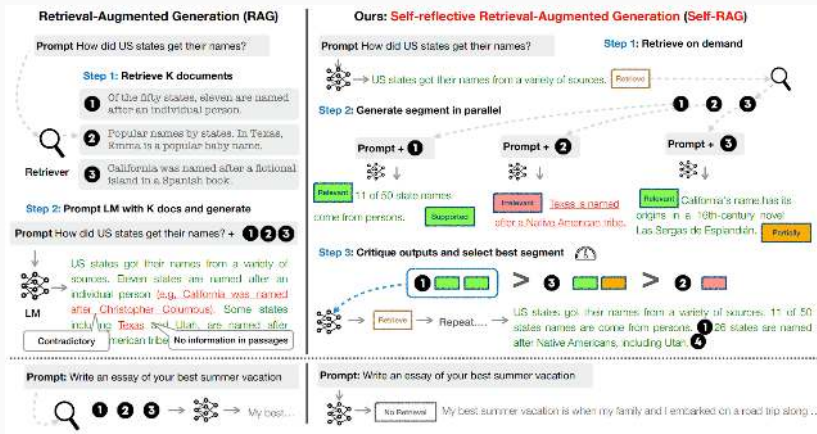
(1) Title: Lewiston Maineiacs Context: The Lewiston Maineiacs were a junior ice hockey team ... The team played its home games at the Androscoggin Bank Colisée ...

(2) Title: Androscoggin Bank Colisée Context: The Androscoggin Bank Colisée (formerly Central Maine Civic Center and Lewiston Colisee) is a 4,000 capacity (3,677 seated) multi-purpose arena ...

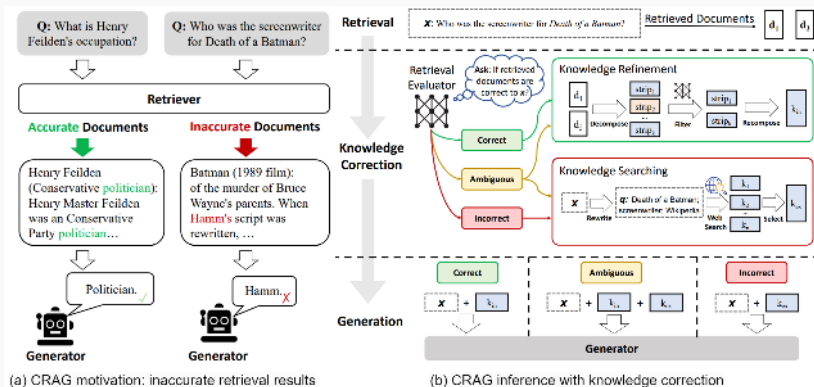
Generation

The Lewiston Maineiacs were a junior ice hockey team of the Quebec Major Junior Hockey League based in Lewiston, Maine. They played their home games at the Androscoggin Bank Colisée. The Androscoggin Bank Colisée has a capacity of 4,000 (3,677 seated). So the answer is 3,677

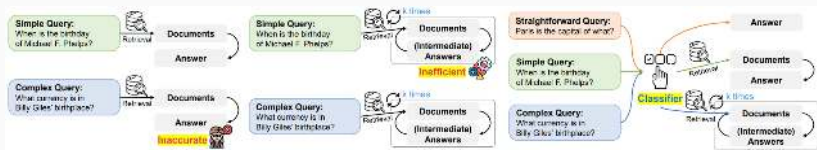
- Asai et al. (2023):



- CRAG (Corrective RAG; Yan et al., 2024): отдельный оценщик выданных документов

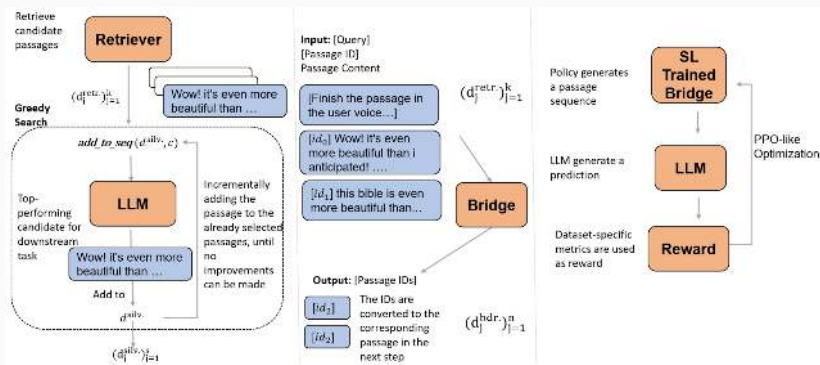


- Adaptive RAG (Jeong et al., 2024): обучает дополнительный классификатор, который выбирает правильный подход – нужен ли RAG вообще, нужно ли запускать его один раз или итеративно много раз

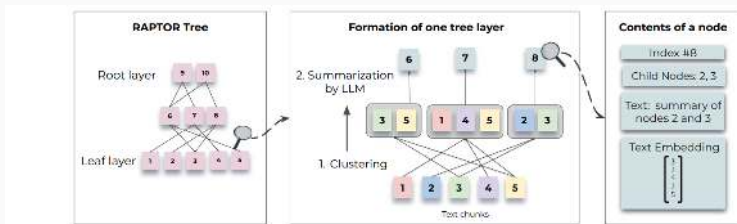


- Ещё одно направление улучшений – умное переранжирование результатов выдачи (Gao et al., 2023, Blagojević, 2023)
- Это, кстати, и наоборот работает: LLM может улучшать поиск, помогая переранжировать (Ma et al., 2023, Peng et al., 2023)

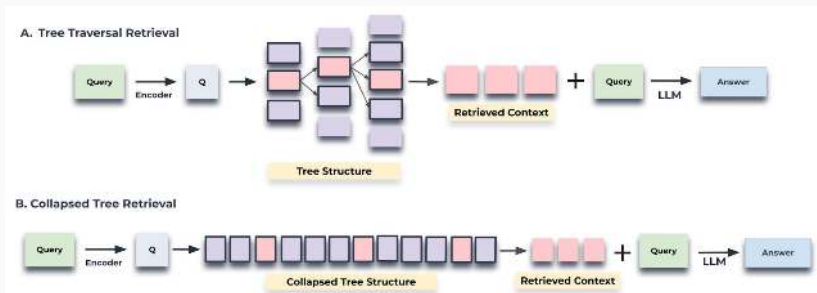
- Ke et al. (2024): отдельная seq2seq модель выбирает из документов конкретные части, более интересные для LLM



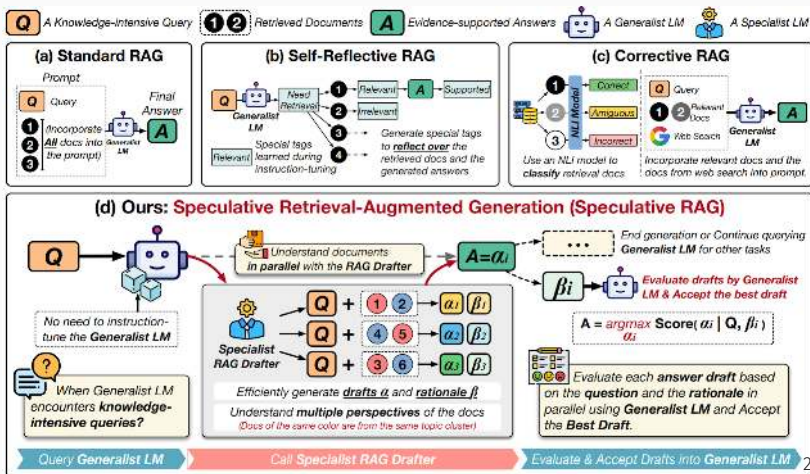
- Иерархические подходы: RAPTOR (Recursive Abstractive Processing for Tree-Organized Retrieval; Sarthi et al., 2024), где LLM рекурсивно суммаризирует длинные тексты



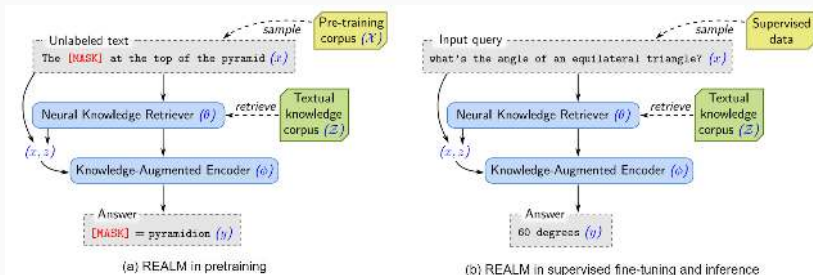
- А потом сам поиск можно запускать на этом дереве аннотаций, выдавая более глобальный контекст для LLM (который сам по себе не влез бы)



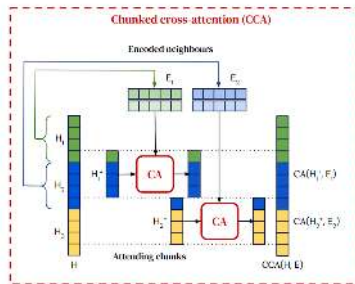
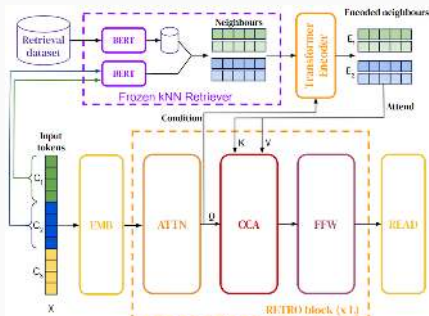
- Wang et al. (2024): Specialist LLM отвечает на вопросы по документам, Generalist LLM обобщает черновики от specialist LLMs



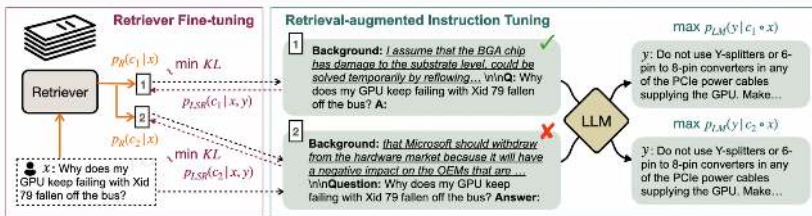
- Связанный подход — R[e]ALM (Retrieval-Augmented Language Model; Guu et al., 2020): можно вставить поиск прямо в предсказание токенов; это тоже продолжается, но не будем сейчас (Borgeaud et al., 2022, Lin et al., 2024)



- RETRO (Retrieval-Enhanced Transformer; DeepMind, Borgeaud et al., 2022) добавляет поиск прямо в декодер трансформера:



- RA-DIT (Retrieval-Augmented Dual Instruction Tuning; FAIR, Lin et al., 2024) использует supervised fine-tuning, чтобы улучшить использование результатов поиска:



- Проблема RAG: Lost in the Middle (Liu et al., 2024)

Input Context

Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).

Document [1] (Title: Asian Americans in science and technology) ...

Document [2] (Title: List of Nobel laureates in Physics) ...

Document [3] (Title: Scientist) ...

Document [4] (Title: Norwegian Americans) ...

Document [5] (Title: Maria Goeppert Mayer) ...

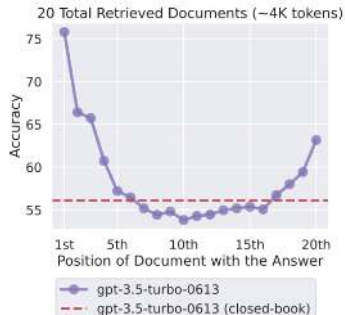
Question: who got the first nobel prize in physics

Answer:

Desired Answer

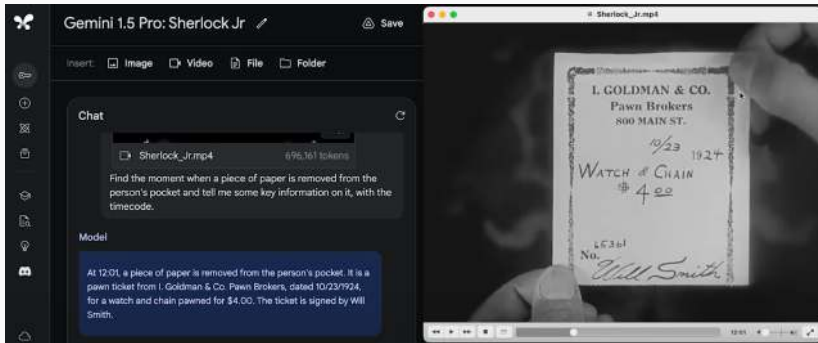
Wilhelm Conrad Röntgen

(a) Multi-document question answering

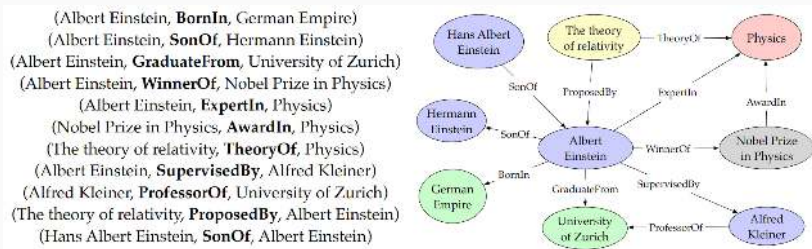


(b) Accuracy as a function of answer position

- А главное – полученные знания неструктурированы, и задачи типа "иголка в стоге сена" здесь не помогут, они не про творческую обработку выдачи поиска



- Графы знаний (knowledge graphs) – хорошо развитая область (Hogan et al., 2022; Ji et al., 2021; Heist et al., 2020; Yan et al., 2018), но кажется, что их можно было бы использовать чаще

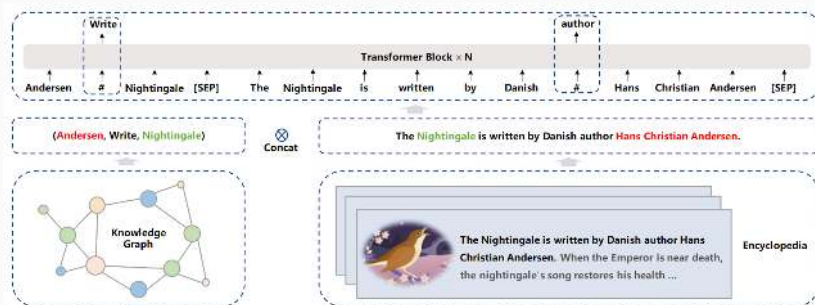


- Есть масса уже готовых графов знаний
- Wikidata (Vrandečić, Krötzsch, 2014): часть Wiki-проекта, более 110M единиц данных
- DBpedia переводит Wikipedia (инфобоксы в основном) в граф знаний, 220M entities и 1.45B triples
- ConceptNet (Speer et al., 2017): commonsense knowledge и определения слов, там есть Wiktionary, но им дело не ограничивается

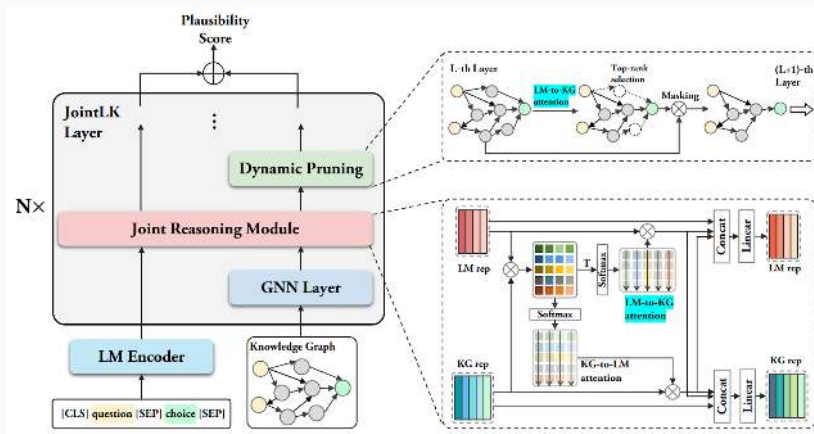
- Графы знаний использовались для улучшения pretext tasks
- Семейство LLM от Baidu ERNIE (Sun et al., 2019, Sun et al., 2020, Xiao et al., 2020, Sun et al., 2021) с этого начиналось:

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

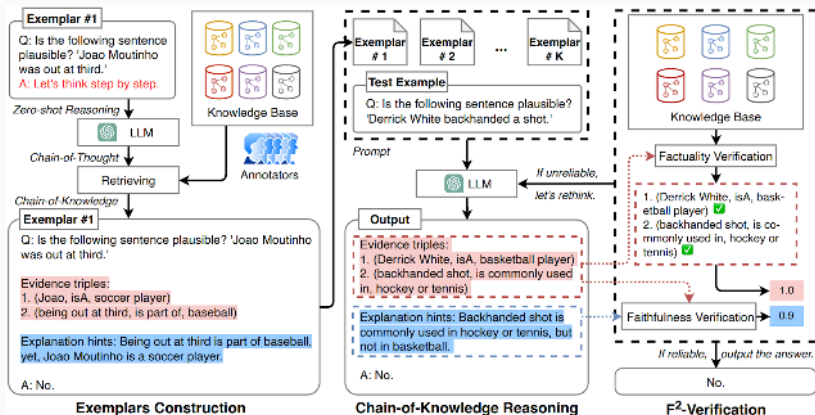
- И потом это продолжалось в следующих версиях ERNIE (Sun et al., 2021):



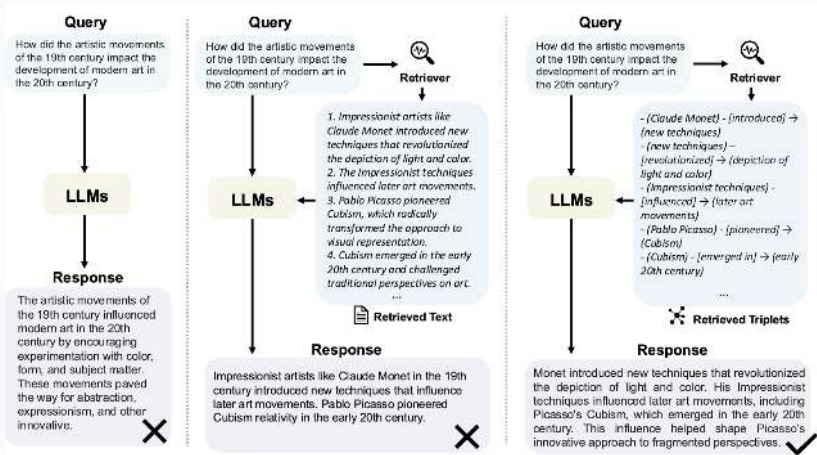
- JointLK (Sun et al., 2022) вводит в трансформер внимание, специально адаптированное для графов:



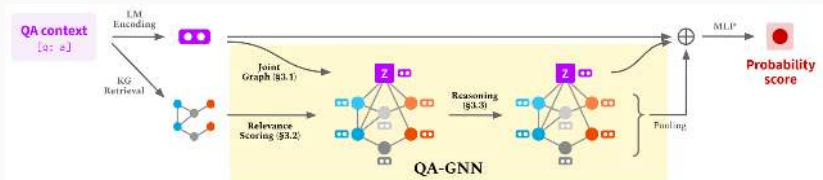
- Wang et al. (2023): chain-of-knowledge prompting



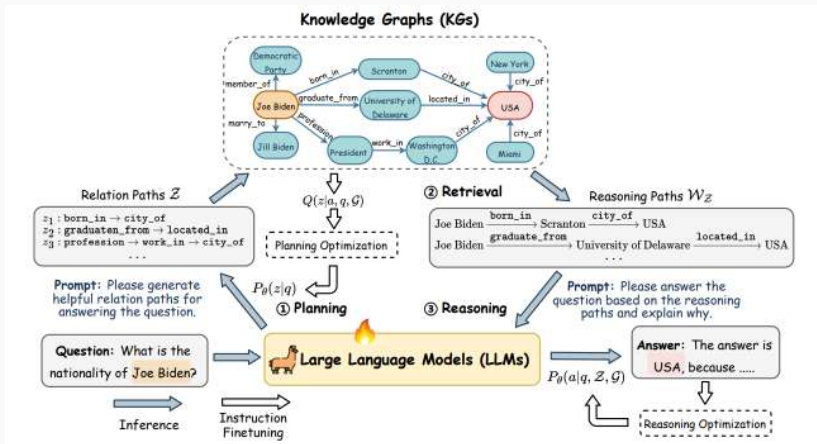
- Peng et al. (2024): если искать структурированные триплеты, ответы становятся точнее



- Для самого поиска можно использовать GNN; например, QA-GNN (Yasunaga et al., 2021)



- Luo et al. (2024): reasoning on graphs (RoG)

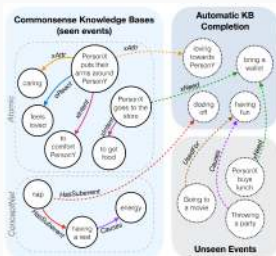


- Luo et al. (2024): reasoning on graphs (RoG)

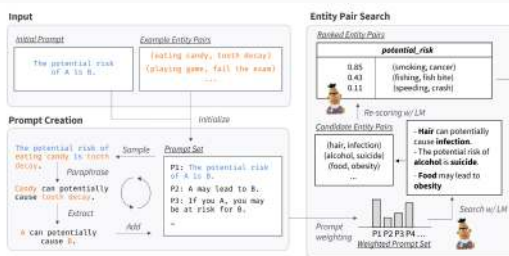
Question	What type of government is used in the country with Northern District?
Answer	Parliamentary system
ChatGPT + CoT	To determine what type of government is used in a country with a Northern District, we need more specific information . The term "Northern District" does not specify the name of any particular country or provide enough context to identify its government type. Could you please provide additional details or clarify the specific country you are referring to?
Reasoning Paths	Northern District → location.administrative.division.first.level.division.of → Israel → government.form.of.government.countries → Parliamentary system
RoG	<p>The type of government used in the country with the Northern District is a Parliamentary system.</p> <p>Explanations:</p> <ol style="list-style-type: none"> Northern District → location.administrative.division.first.level.division.of → Israel: The Northern District is a part of Israel. Israel → government.form.of.government.countries → Parliamentary system: Israel has a Parliamentary system of government. <p>So, by following the reasoning paths provided, we can conclude that the country with the Northern District, which is Israel, uses a Parliamentary system of government.</p>

Question	Where is the home stadium of the team who won the 1946 World Series championship?
Answer	Busch Stadium
ChatGPT + CoT	<p>Step 1: Identify the team that won the 1946 World Series championship. The team that won the 1946 World Series championship was the St. Louis Cardinals.</p> <p>Step 2: Determine the home stadium of the St. Louis Cardinals during that time. The St. Louis Cardinals' home stadium during the 1946 season was Spartan's Park.</p> <p>Therefore, the possible answer to the question "Where is the home stadium of the team who won the 1946 World Series championship?" is Spartan's Park.</p>
Reasoning Paths	<p>1946 World Series → sports.sports.team.championships → St. Louis Cardinals → sports.sports.team.arena.stadium → Busch Stadium</p> <p>1946 World Series → sports.sports.team.championships → St. Louis Cardinals → sports.sports.team.arena.stadium → Rogers Place Stadium</p>
RoG	<p>Based on the reasoning paths provided, the home stadium of the team who won the 1946 World Series championship is Busch Stadium. This conclusion is based on the following path:</p> <p>1. 1946 World Series → sports.sports.team.championships → St. Louis Cardinals → sports.sports.team.arena.stadium → Busch Stadium</p> <p>The reasoning path leads us to the St. Louis Cardinals as the team that won the 1946 World Series, and Busch Stadium is the stadium associated with the St. Louis Cardinals. Therefore, Busch Stadium is the home stadium of the team that won the 1946 World Series championship.</p>

- Можно и наоборот, строить графы знаний через LLM: COMET (Bosselut et al., 2019) использовал GPT-2, чтобы выводить новые триплеты, BertNet (Hao et al., 2022) начинает с определения отношения и нескольких примеров



(a) COMET fills in knowledge graph triples



(b) BertNet automatically improves relation definitions

Выводы по RAG

- Никакого контекста не хватит на весь интернет, искать всё равно будет нужно
- RAG — одно из важнейших направлений использования дополнительных инструментов для LLM (хотя есть и другие, но об этом не сегодня)
- Есть много разных подходов, но, кстати, и самый простой тоже работает хорошо: In-context RALM (Ram et al., 2023) просто берёт все найденные документы и добавляет их к запросу для LLM, и тоже всё работает



СПАСИБО!

Спасибо за внимание!

