Лекция 10. Настройка БЯМ

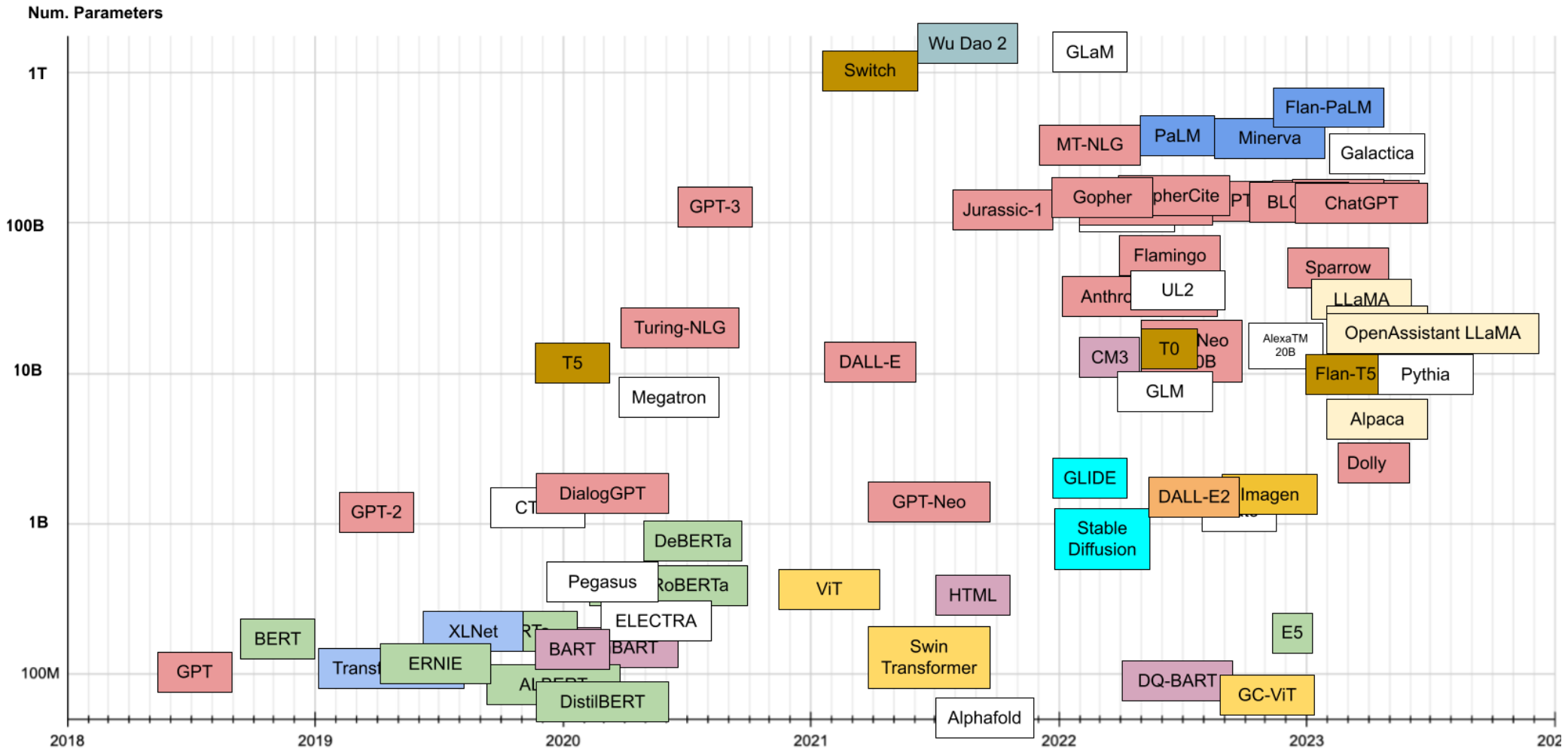
Денис Деркач, Дмитрий Тарасов

LAMBDA • HSE

21 апреля 2025 года

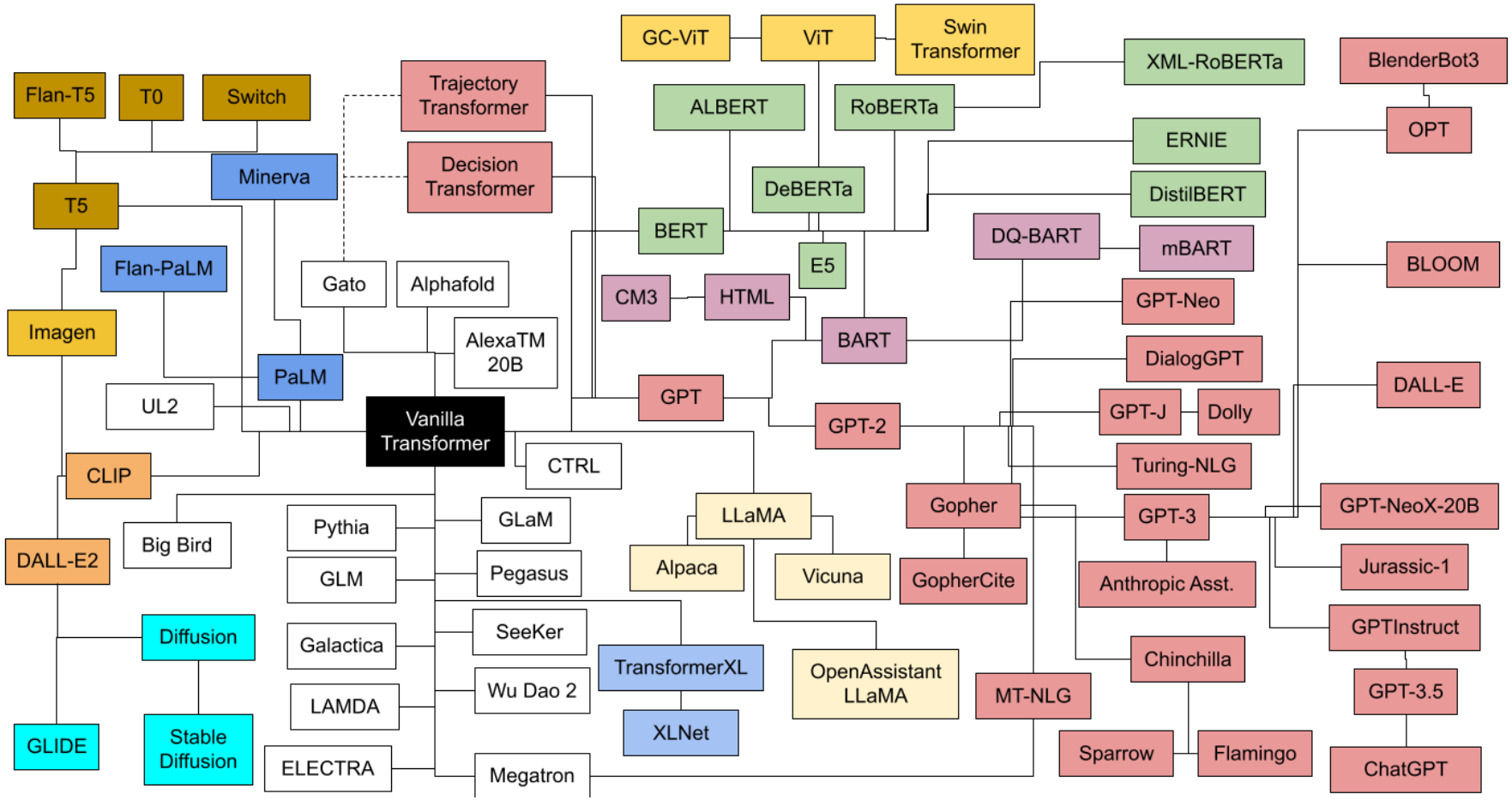
В предыдущих лекциях

Зоопарк трансформеров



https://amatria.in/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

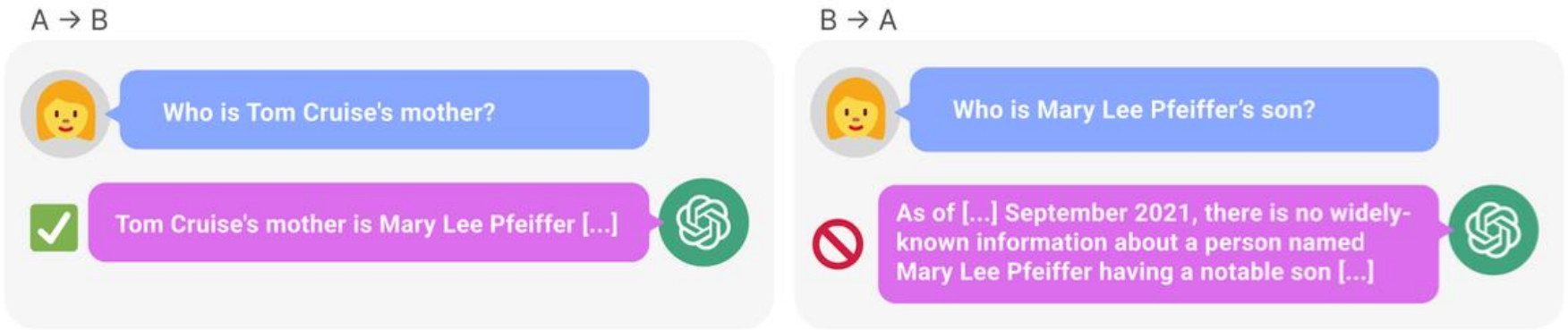
Зоопарк трансформеров



https://amatria.in/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

Large Language Models

Large Language Models - stochastic parrots or AGI?



Berglund L. et al. The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A"

https://dl.acm.org/doi/10.1145/3442188.3445922

Выравнивание БЯМ

Оценивание БЯМ

- Бенчмарки, отвечающие на вопрос «Есть ли ответ в сгенерированном тексте?»
- Бенчмарки типа LLM-судьи
- Ручная Side-by-side (SBS) оценка

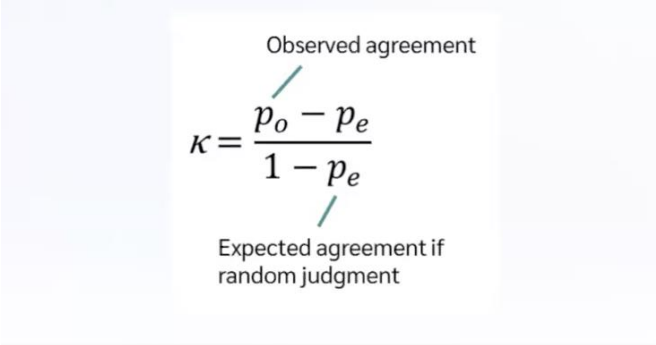
▾ Промпт для LLM-судьи выглядит так:

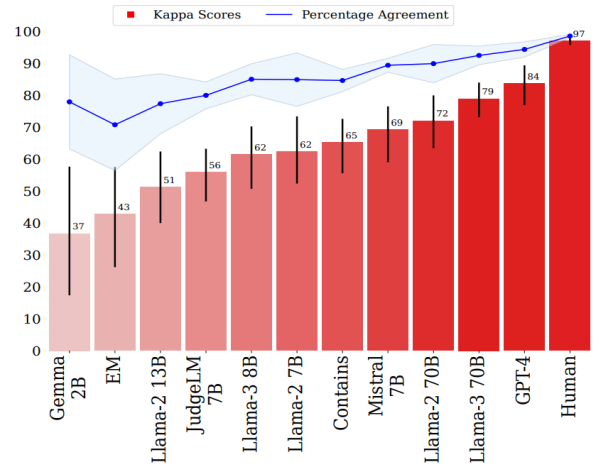
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. **Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses.** Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output two values on a scale of 1 to 10 indicating the scores for Assistant A and B, respectively. Output your final verdict by strictly following this format: [[{{Assistant A score}} {{Assistant B score}}]].

https://arize.com/blog/judging-the-judges-llm-as-a-judge/
https://arxiv.org/pdf/2406.12624v1

Метрики «качества»

- Метрики:
  - Дельта – разница между оценками ответов человека и БЯМ.
  - Процент совпадений
  - Каппа Коэна для случайных совпадений





https://arize.com/blog/judging-the-judges-llm-as-a-judge/
https://arxiv.org/abs/2212.08073

Выравнивание БЯМ

- Модель должна быть: helpful, honest, harmless.
- Выравнивание (Alignment) – обеспечение поведения больших языковых моделей в соответствии с человеческими ценностями, этическими стандартами и конкретными целями или намерениями пользователей
- Дообучение готовой модели – не надо слишком большого количества данных.
- Дополнительные знания ведут к галлюцинациям.

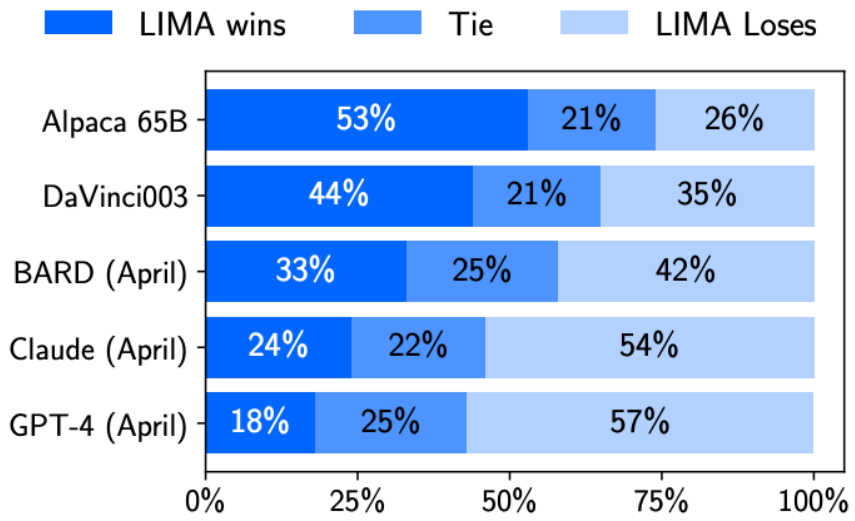


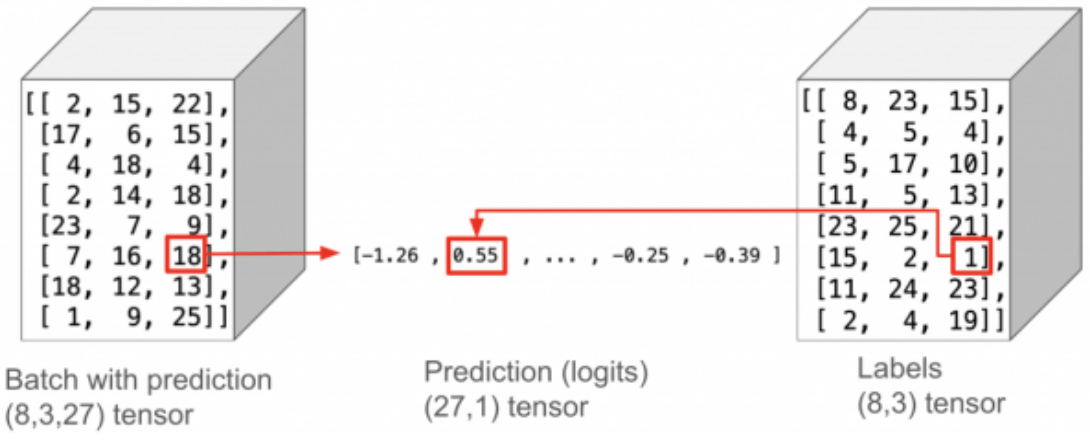
Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

https://arxiv.org/abs/2212.08073
https://arxiv.org/abs/2305.11206v1
https://arxiv.org/abs/2405.05904

DPO, PPO, GRPO

Cross-Entropy Method (CEM/CE-RL)

- генерируем SFT-моделью для каждого запроса по N ответов,
- выбираем лучший ответ по награде для каждого запроса,
- дообучаем SFT-модель на полученных парах запрос + лучший ответ



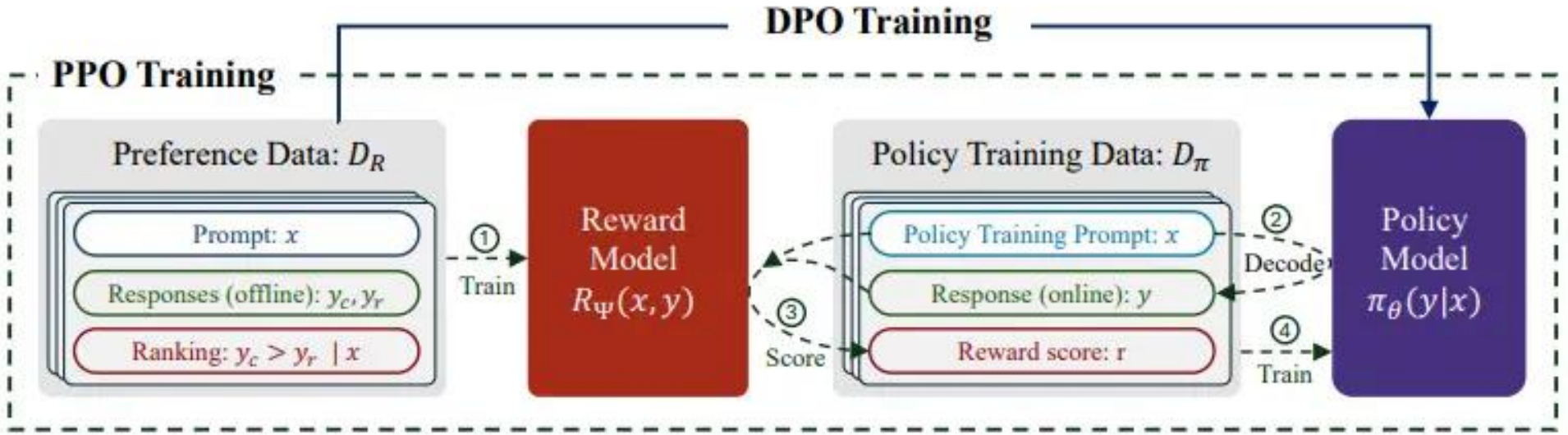
- Pros: Быстро и понятно
- Cons: после первой итерации улучшение минимально из-за снижения разнообразия ответов

http://www.philippeadjiman.com/blog/2024/03/09/deep-learning-gymnastics-4-master-your-llm-cross-entropy/
https://habr.com/ru/companies/yandex/articles/817391/

DPO и PPO: два пути выравнивания

- Оптимизация прямых предпочтений (Direct Preference Optimization, DPO): напрямую оптимизирует параметры модели на основе обратной связи человека по сгенерированным результатам. Он направлен на изучение политики, которая максимизирует человеческое удовлетворение.
- Оптимизация проксимальной политики (Proximal Policy Optimization, PPO): алгоритм обучения с подкреплением, обучает модель максимизировать сигнал вознаграждения, предоставляемый оценщиками-людьми. Он фокусируется на итеративном улучшении политики модели при сохранении стабильности.

PPO – RL процесс



- Инициализация политики
- Сбор данных
- Оценка преимущества
- Обновление политики
- Итерация

https://arxiv.org/abs/2203.02155

PPO: математика 1

Хотим максимизировать среднюю награду агента

$$J(\pi_\theta) = \mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)} r_\psi(s,a)$$

s~D – запросы из выборки

$\pi_\theta(a|s)$ -- вероятность модели \pi_theta ответить *a* на запрос *s*

$r_\psi(s,a)$ – обучаемая нейронная модель награды с параметрами \psi

Тогда

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)} \nabla_\theta \log \pi_\theta(a|s) r_\psi(s,a)$$

Отрицательные награды снижают вероятности, положительные - увеличивают

https://habr.com/ru/companies/yandex/articles/817391/

PPO: математика 2

Для улучшения обучения при общей положительной награде:

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)}\nabla_\theta \log \pi_\theta(a|s)[r_\psi(s,a) - V_\phi(s)]$$

Здесь $V_\phi(s) = \mathbf{E}_{a\sim\pi_\theta(a|s)} r_\psi(s,a)$ · нейронная сеть, предсказывающая среднюю награда для ответа на запрос s

При этом

$$\mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)}[V_\phi(s) - r_\psi(s,a)]^2 \to \min_\phi$$

$$\mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)}\left[V_\phi(s) - r_\psi(s,a)\right]^2 \to \min_\phi$$

https://habr.com/ru/companies/yandex/articles/817391/

PPO: математика 2

Для улучшения обучения при общей положительной награде:

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)}\nabla_\theta \log \pi_\theta(a|s)[r_\psi(s,a) - V_\phi(s)]$$

Здесь $V_\phi(s) = \mathbf{I}$ награда для ответа на за

$$\mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)}\left[V_\phi(s) - r_\psi(s,a)\right]^2 \to \min_\phi$$

При этом

$$\mathbf{E}_{s\sim\mathcal{D}}\mathbf{E}_{a\sim\pi_\theta(a|s)}[V_\phi(s) - r_\psi(s,a)]^2 \to \min_\phi$$

https://habr.com/ru/companies/yandex/articles/817391/

PPO: алгоритм Advantage Actor Critic (A2C)

1. Вход: множество запросов для обучения *D*
2. Инициализируем политику SFT-моделью: $\pi_\theta \leftarrow \pi_{\text{SFT}}$
3. Инициализируем ценность V моделью награды $V_\phi \leftarrow r_\psi$
4. До сходимости:
    1. Выбираем батч запросов B~D
    2. Вычисляем ценность для каждого запроса из батча $V_\phi(s_i) \;\; \forall s_i \in \mathcal{B}$
    3. Генерируем по одному ответу *a* на каждый запрос.
    4. Вычисляем награду r для всех пар (s, a)
    5. Вычисляем функцию потерь для агента и для функции ценности

$$\mathcal{L}_a = -\frac{1}{|\mathcal{B}|}\sum_i^{|\mathcal{B}|} \log \pi_\theta(a_i|s_i)[r_\psi(s_i,a_i) - V_\phi(s_i)] \qquad \mathcal{L}_v = \frac{1}{|\mathcal{B}|}\sum_i^{|\mathcal{B}|}[V_\phi(s_i) - r_\psi(s_i,a_i)]^2$$

    6. Оптимизируем

https://habr.com/ru/companies/yandex/articles/817391/

PPO: финальные штрихи

RL изменяет ответы генератора именно в том направлении, где модель награды работает плохо
Рано или поздно можно попасть в плохую местность. Выход:

1. Заставить модель несильно удаляться от изначальной:

$$\mathbf{E}_{a\sim\pi_\theta(a|s)}\left[r_\psi(s,a) - \beta\mathrm{KL}\left(\pi_\theta(a|s)||\pi_{\mathrm{SFT}}(a|s)\right)\right] \to \max_\theta$$

2. Постоянно проверять, что награда хорошо определена (дотренировывать именно в этом месте)

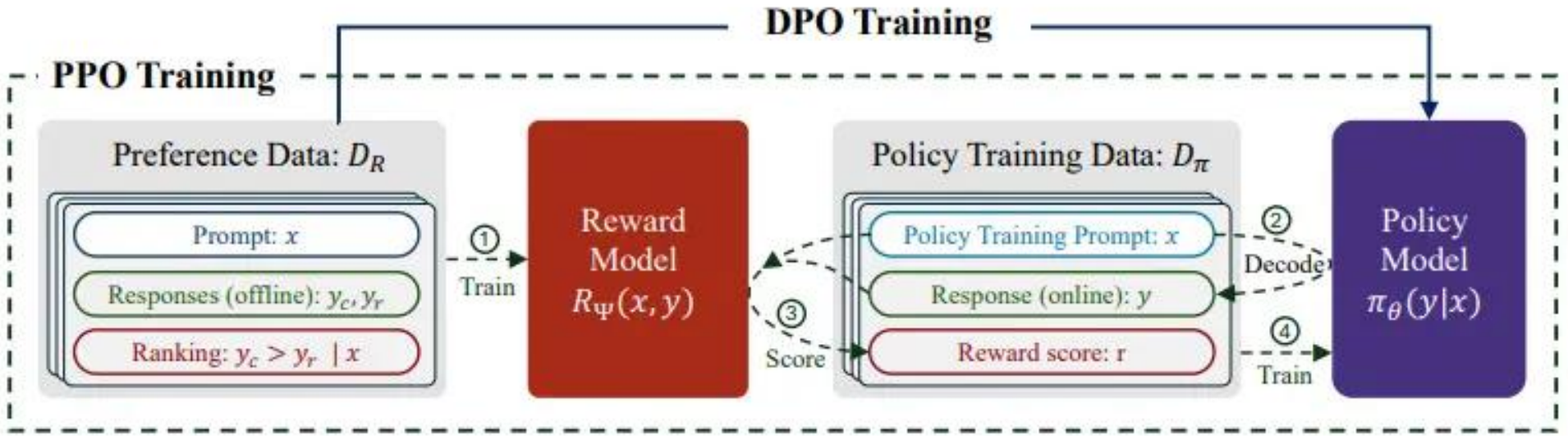
https://habr.com/ru/companies/yandex/articles/817391/

PPO: преимущества

- Стабильность: малые обновления политики в PPO помогают предотвратить резкие изменения, гарантируя, что поведение модели останется контролируемым.
- Эффективность: вычислительно эффективен и может обрабатывать сложные ландшафты вознаграждений.
- Гибкость: PPO можно применять к задачам обучения с подкреплением.

https://arize.com/blog/judging-the-judges-llm-as-a-judge/
https://arxiv.org/pdf/2406.12624v1

DPO - процесс



- Сбор данных
- Инициализация модели
- Моделирование предпочтений
- Обновление параметров
- Итерация

DPO: математика

- Из-за идеи ограничения по KL-дивергенции можем записать:

$$\pi^*(a|s) = \frac{1}{Z(s)}\pi_{\text{SFT}}(a|s)e^{\frac{1}{\beta}r(s,a)}$$

$$Z(s) = \sum_a e^{\frac{1}{\beta}r_(s,a)}$$

- То есть функцию награды можем вычислить:

$$r_\theta(s,a) = \beta \log \frac{\pi_\theta(a|s)}{\pi_{\text{SFT}}(a|s)} + \beta \log Z(s)$$

- Максимизируя разницу в ответах функции награды, мы получим:

$$\sum_{(s,winner,loser)\in \mathbf{D}} \log \sigma\big(\beta\big[\log \frac{\pi_\theta(winner|s)}{\pi_{\text{SFT}}(winner|s)} - \log \frac{\pi_\theta(loser|s)}{\pi_{\text{SFT}}(loser|s)}\big]\big) \to \max_\theta$$

https://habr.com/ru/companies/yandex/articles/817391/

DPO: преимущества

- Прямая оптимизация: DPO напрямую ориентируется на предпочтения человека, что потенциально приводит к более быстрому и эффективному выравниванию.
- Уменьшение смещения: устраняя промежуточную модель вознаграждения, DPO может снизить риск наследования смещений из обучающих данных.
- Эффективность: DPO может быть более эффективным в отношении данных и вычислительных ресурсов, чем традиционные методы.

DPO vs PPO

DPO

- Хорошо согласованные данные.
- Узкие применения
- Необходимость быстрой адаптации
- Недостаток ресурсов

PPO

- Сложные задачи, которые требуют итеративного обучения
- Сложные награды
- Необходимость робастности и стабильности
- Необходимость учёта длительных тенденций

Group Relative Policy Optimization

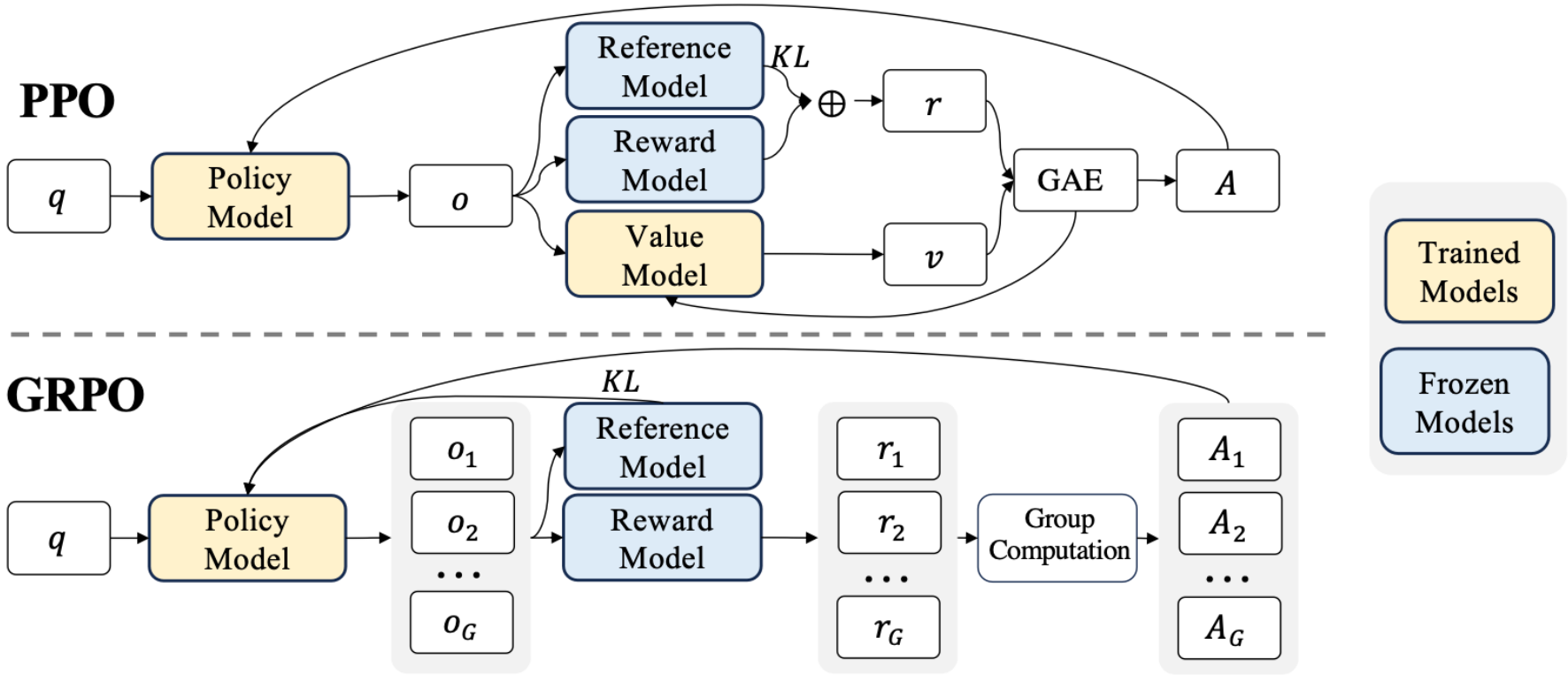


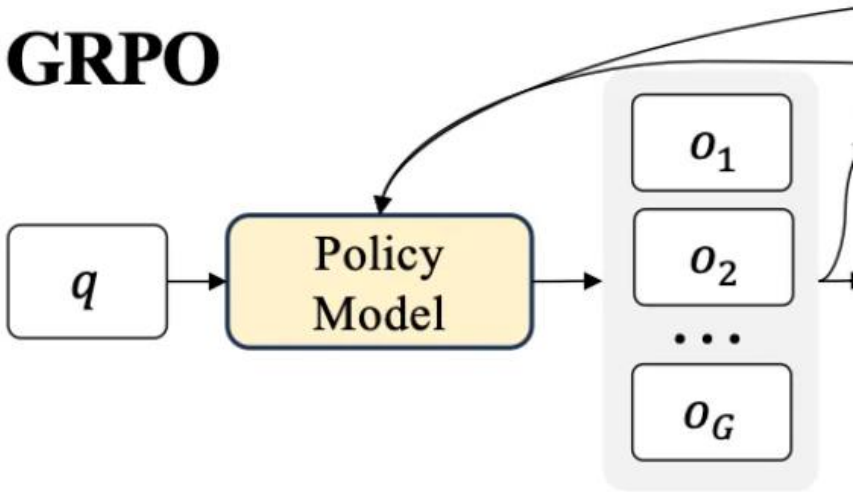
Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

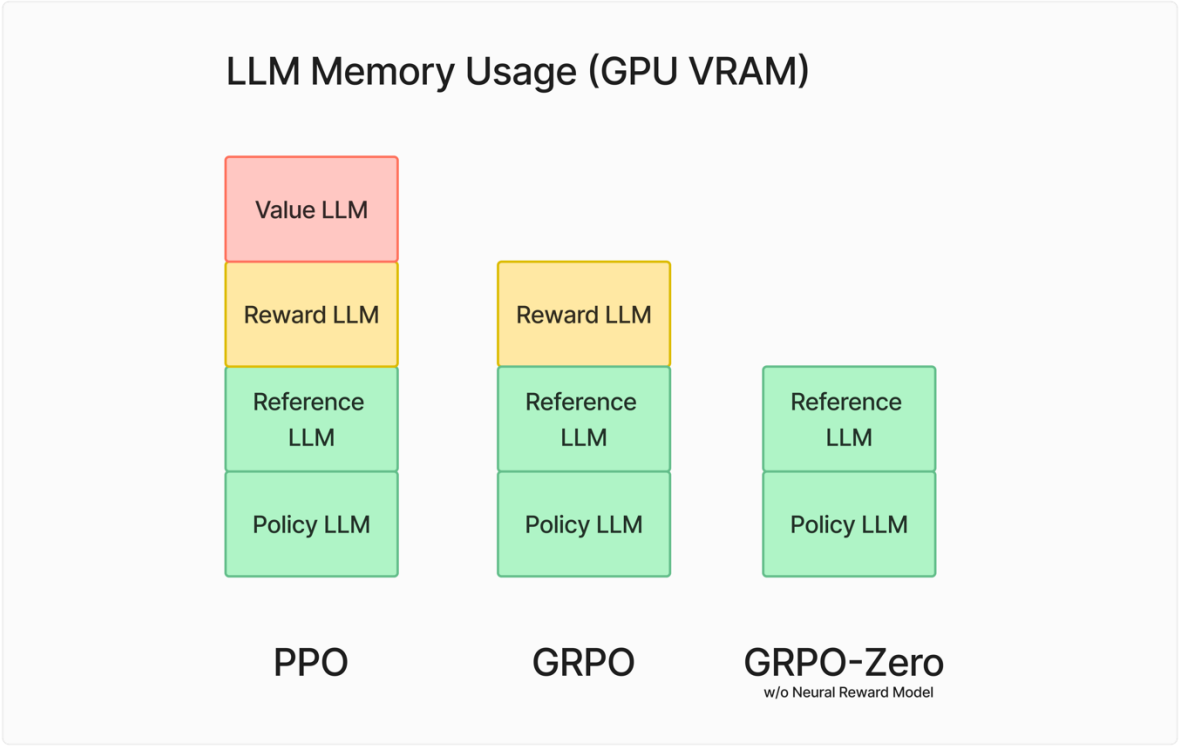
1.Policy Model – оптимизируемая БЯМ
2.Reference Model – замороженная БЯМ
3.Reward Model - модель, обученная на человеческих предпочтениях
4.Value Model - модель, которая пытается оценить долгосрочное вознаграждение за определенные действия.

https://arxiv.org/abs/2402.03300

Group Relative Policy Optimization - преимущества

- Снижение потребления памяти за счёт снижения количества моделей
- Снижение потребления CPU/GPU за счёт использования групповых выходов.





https://ghost.oxen.ai/why-grpo-is-important-and-how-it-works/

Group Relative Policy Optimization - reward

- Функция награды тренируется совместно с обучением.
- Награду можно зафиксировать не в виде нейронной сети (GRPO-zero, см справа).

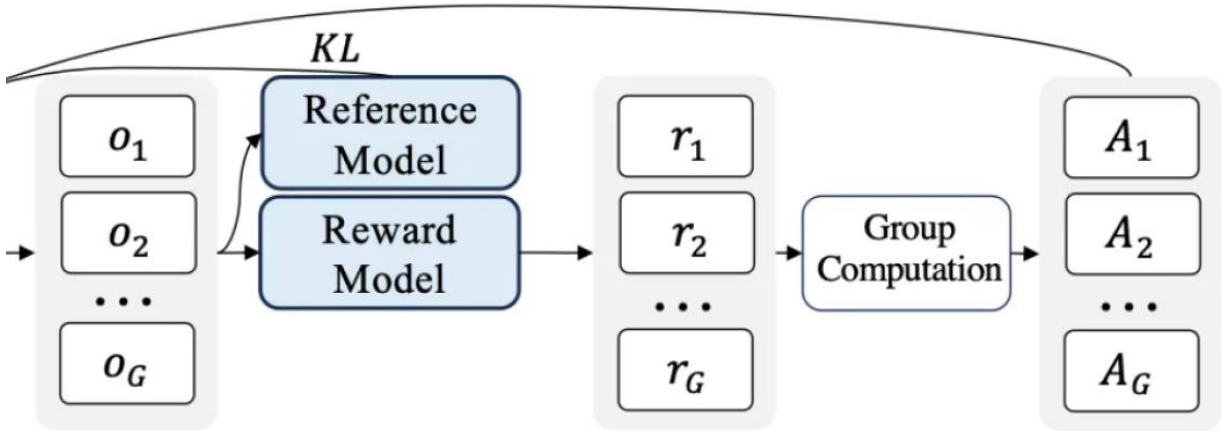
```python
def extract_xml_answer(text: str) -> str:
    answer = text.split("<answer>")[-1]
    answer = answer.split("</answer>")[0]
    return answer.strip()

def accuracy_reward_func(prompts, completions, answer, **kwargs) -> list[float]:
    """Reward function that extracts the answer from the xml tags and compares it to the correct answer."""
    responses = [completion[0]['content'] for completion in completions]
    extracted_responses = [extract_xml_answer(r) for r in responses]
    return [2.0 if r == a else 0.0 for r, a in zip(extracted_responses, answer)]

def format_reward_func(completions, **kwargs) -> list[float]:
    """Reward function that checks if the completion has a specific format."""
    pattern = r"^<reasoning>\n.*?\n</reasoning>\n<answer>\n.*?\n</answer>\n$"
    responses = [completion[0]["content"] for completion in completions]
    matches = [re.match(pattern, r) for r in responses]
    return [0.5 if match else 0.0 for match in matches]
```

- Далее можно использовать относительное преимущество:

$$\hat{A}_{i,t} = \tilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$



https://github.com/willccbb/verifiers
https://arxiv.org/abs/2503.20783

Group Relative Policy Optimization - objective

- Так же как в PPO используется KL дивергенция для того, чтобы модель не сильно отходила от оригинальной

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left(r_t(\theta)\hat{A}_{i,t}, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\right)\right] - \beta D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}).$$

- Можно сравнить с PPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q\sim P(Q), o\sim\pi_{\theta_{old}}(O|q)]\frac{1}{|o|}\sum_{t=1}^{|o|}\min\left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}A_t, \text{clip}\left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon\right)A_t\right],$$

Group Relative Policy Optimization - outcome

- Подход позволяет снизить количество ресурсов, необходимых для тренировки.
- GRPO-zero идёт дальше, однако страдает от сниженного обобщения.

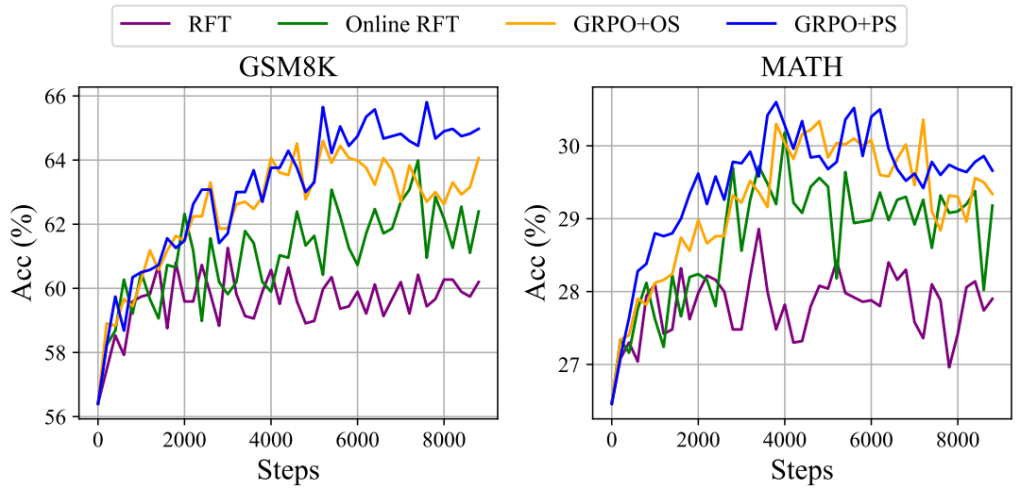
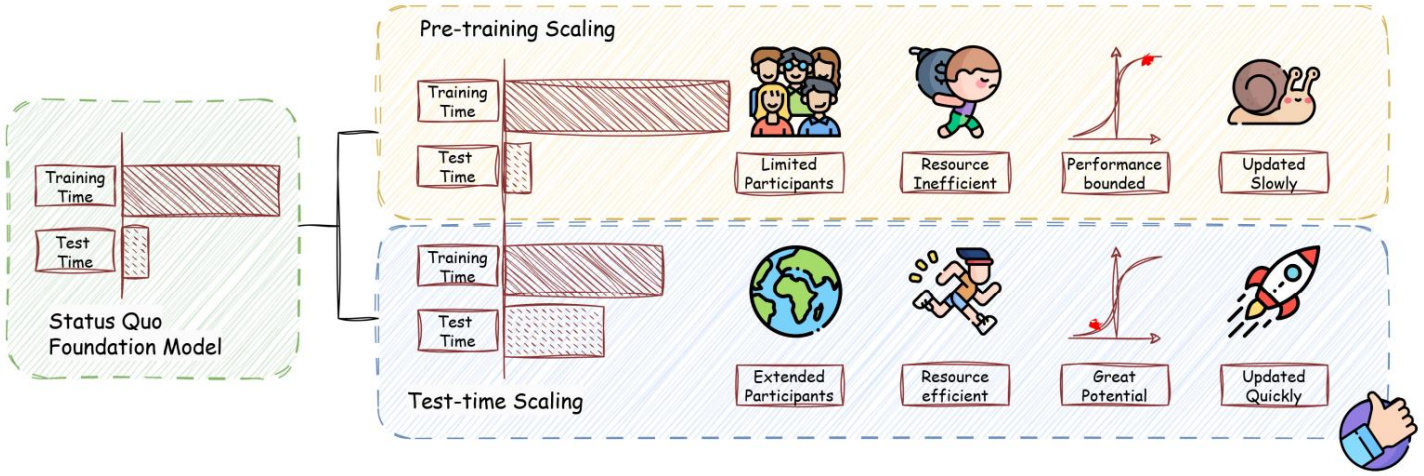


Figure 5 | Performance of the DeepSeekMath-Instruct 1.3B model, which was further trained using various methods, on two benchmarks.

Test-time Scaling

Test-time Scaling

- Pre-train Scaling – больше данных и ресурсов => лучше результат
- Post-train Scaling – дотренировки готовых БЯМ разными способами => лучше результат
- Test-time Scaling – динамически распределять ресурсы во время вывода



https://arxiv.org/abs/2503.24235v1

Test-time Scaling

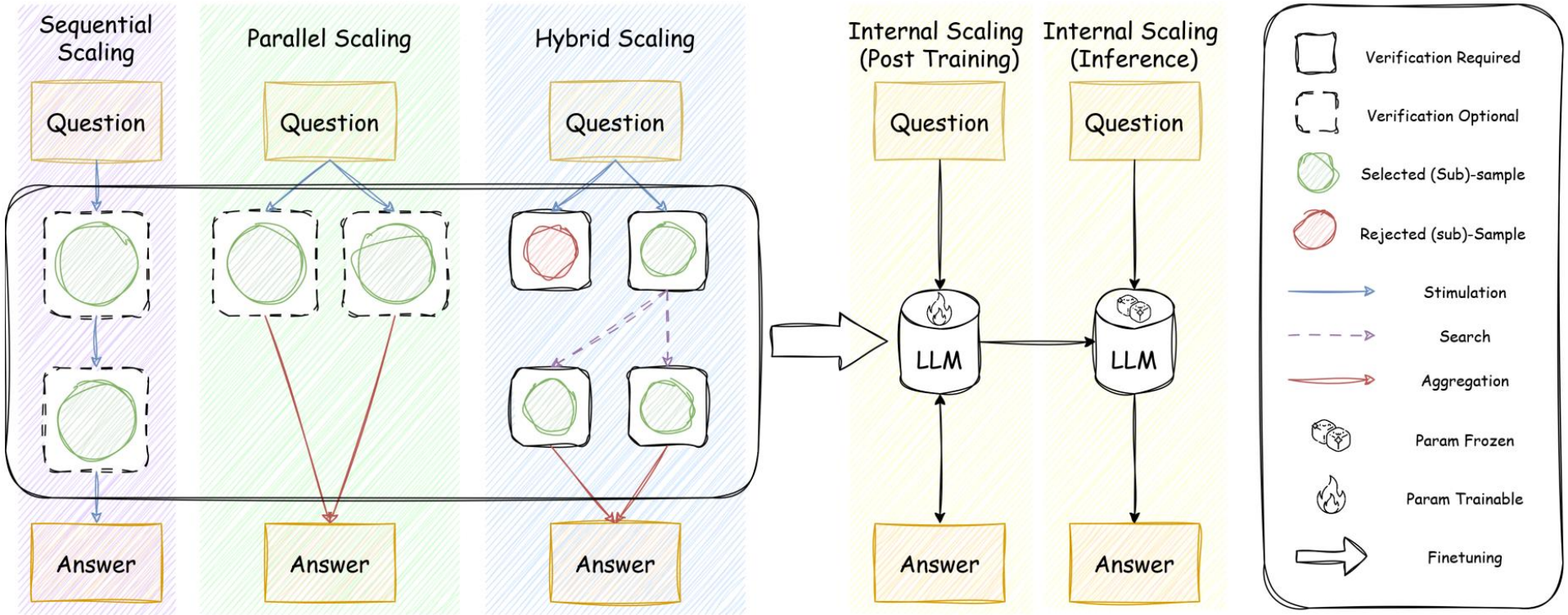


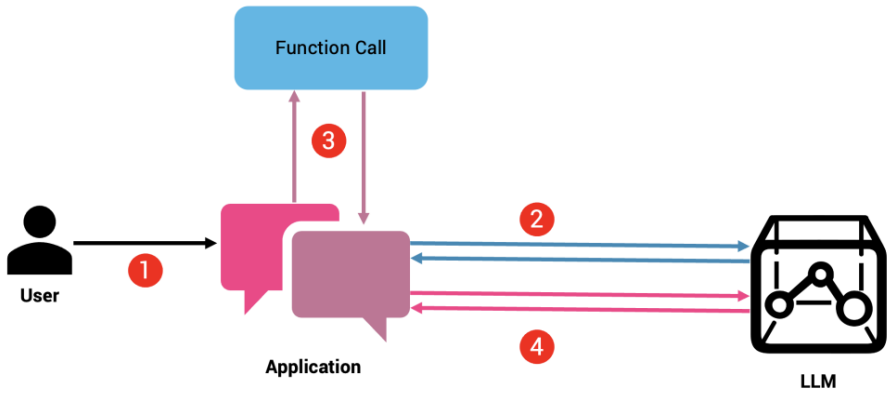
Figure 3: Illustration of how to scale for various approaches.

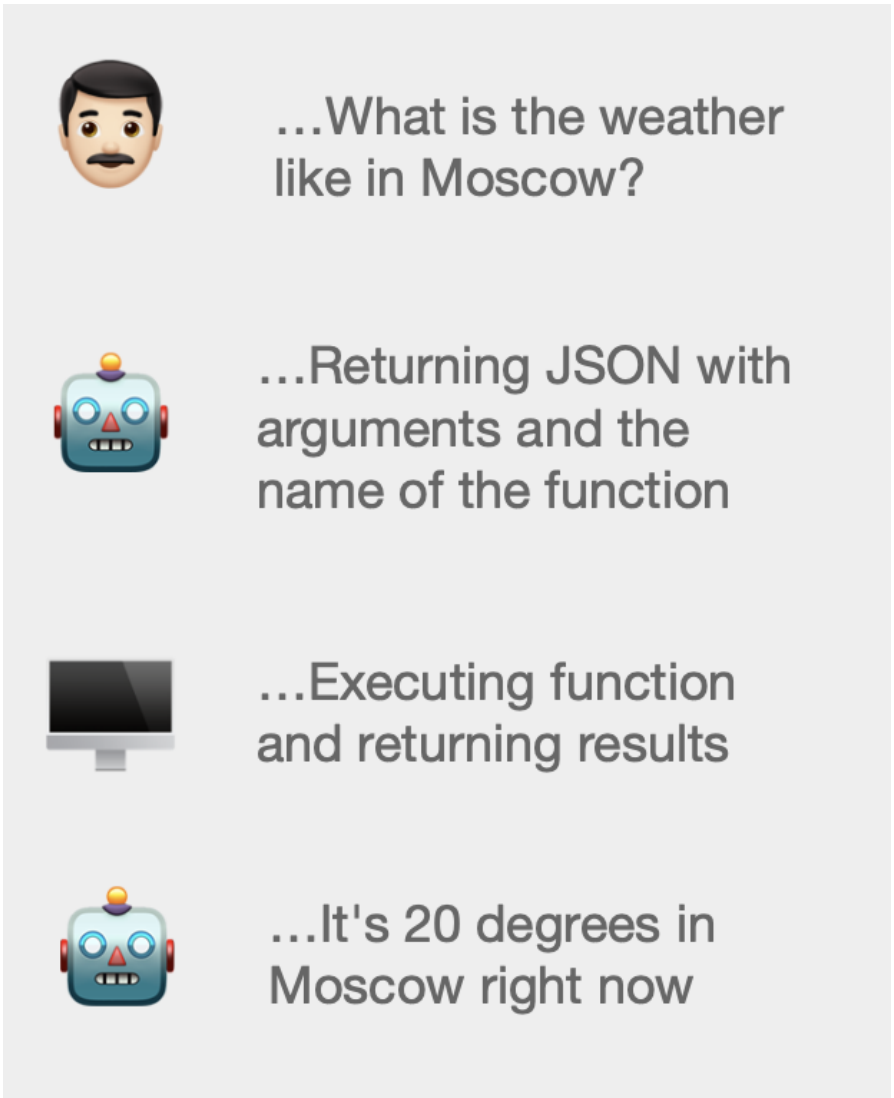
https://arxiv.org/abs/2503.24235v1

Дополнение вывода

Function calling

- Вводим промпт пользователя и описание доступных функций/инструментов.
- Модель сопоставляет промпт пользователя с описанием функций. Если требуется вызов одной или нескольких функций, то она возвращает JSON с именем и аргументами функции.
- Вызываем функцию в коде.
- Передаем результаты выполнения функции обратно ассистенту, затем он генерирует ответ с суммаризацией результатов.





https://habr.com/ru/companies/mts_ai/articles/831220/

Function calling – Toolformer learning

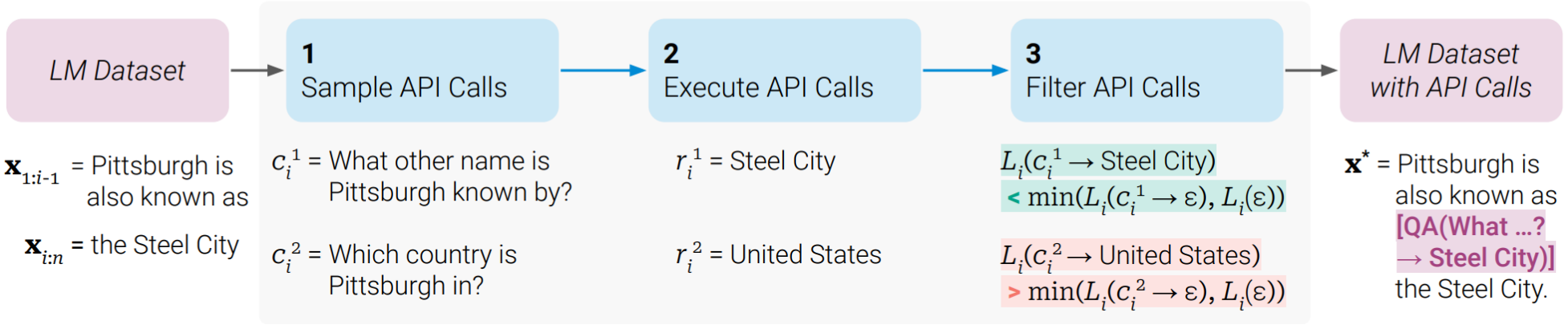


Figure 2: Key steps in our approach, illustrated for a *question answering* tool: Given an input text $\mathbf{x}$, we first sample a position i and corresponding API call candidates $c_i^1, c_i^2, \ldots, c_i^k$. We then execute these API calls and filter out all calls which do not reduce the loss L_i over the next tokens. All remaining API calls are interleaved with the original text, resulting in a new text $\mathbf{x}^*$.

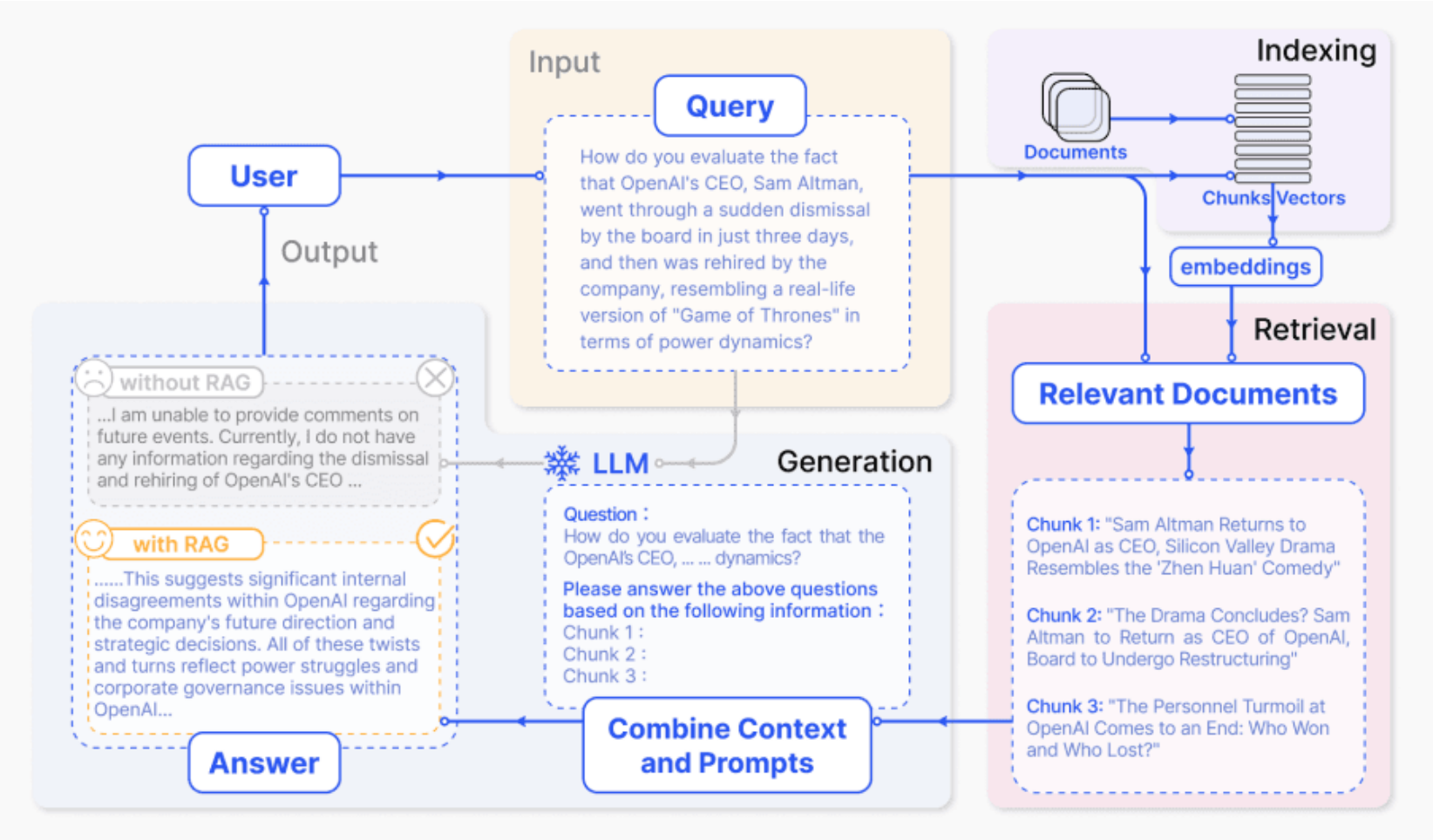
- Строим набор запросов для каждого кусочка текста.
- Выбираем какие запросы снижают лосс.
- Оставляем только такие запросы.
- Дотренировываем БЯМ.

API Name	Example Input	Example Output
Question Answering	Where was the Knights of Columbus founded?	New Haven, Connecticut
Wikipedia Search	Fishing Reel Types	Spin fishing > Spin fishing is distinguished between fly fishing and bait cast fishing by the type of rod and reel used. There are two types of reels used when spin fishing, the open faced reel and the closed faced reel.
Calculator	27 + 4 * 2	35
Calendar	ε	Today is Monday, January 30, 2023.
Machine Translation	sûreté nucléaire	nuclear safety

Table 1: Examples of inputs and outputs for all APIs used.

https://arxiv.org/abs/2302.04761

Retrieval Augmented Generation



https://www.promptingguide.ai/research/rag

RAG vs FC

RAG

- Извлекает внешние знания для информирования ответов
- Ответ контекстно обогащенный и аккуратный
- Лучшее применение: расширение ответов с помощью актуальной информации из больших наборов данных

Function Calling

- Использует предопределенные функции для выполнения задач
- Точные, структурированные и в режиме реального времени
- Лучшее применение: извлечение данных, обработка информации в реальном времени, взаимодействие с внешними API