Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Лебедев Андрей Алексеевич

Методы дообучения больших языковых моделей инструкциям для повышения качества работы на русском языке

ПРЕДДИПЛОМНАЯ ПРАКТИКА

Научный руководитель:
Канд. физ.-мат. наук
М.М. Тихомиров

Москва, 2024

Аннотация

Методы дообучения больших языковых моделей
инструкциям для повышения качества
работы на русском языке

Лебедев Андрей Алексеевич

В рамках данной работы рассмотрены современные подходы к дообучению больших языковых моделей (LLM) инструкциям для повышения их качества при обработке текстов на русском языке. Основное внимание уделено методам выравнивания моделей на предпочтения пользователей, включая обучение с учителем (Supervised Fine-Tuning, SFT) и обучение с подкреплением на основе обратной связи от человека (Reinforcement Learning from Human Feedback, RLHF), а также альтернативным методам, таким как Direct Preference Optimization (DPO), Simple Preference Optimization (SimPO) и Kahneman-Tversky Optimization (KTO). Практическая часть работы включает применение метода обучения с учителем для выравнивания моделей на специализированных датасетах, реализацию процесса обучения и проведение экспериментов. Представлены результаты экспериментов, демонстрирующие количественные и качественные улучшения в производительности моделей.

Содержание

1 Введение

Современные большие языковые модели представляют собой важный шаг в развитии технологий обработки естественного языка. Эти модели демонстрируют впечатляющие результаты в самых разнообразных задачах, включая генерацию текстов, автоматический перевод, ответы на вопросы и даже написание программного кода. Однако их качество во многом зависит от того, насколько хорошо они адаптированы к специфике конкретного языка и ожиданиям пользователей. В частности, для русского языка, с его богатой морфологией и синтаксической сложностью, обеспечение высокого качества работы таких моделей остается сложной задачей.

Адаптация языковых моделей к конкретным языковым особенностям и пользовательским запросам осуществляется через дообучение инструкциям, которое играет ключевую роль в их доработке. Методы, такие как обучение с учителем (Supervised Fine-Tuning, SFT) и обучение с подкреплением на основе обратной связи от людей (Reinforcement Learning from Human Feedback, RLHF) [2], а также его модификации, позволяют значительно улучшить ответы моделей, делая их более соответствующими ожиданиям пользователей. Эти подходы лежат в основе современных технологий выравнивания моделей с человеческими предпочтениями.

Отдельное внимание в данной работе уделено релевантной оценке результатов дообучения моделей, включая их способность удовлетворять специфические запросы пользователей.

2 Обзор существующих методов

Одной из ключевых задач при разработке больших языковых моделей (БЯМ) является их адаптация для выполнения конкретных задач или работы в специфических языковых доменах, таких как русский язык. Для этого используются методы дообучения и выравнивания моделей с предпочтениями пользователя.

2.1 Выравнивание моделей

Одной из ключевых задач при работе с большими языковыми моделями является их выравнивание — процесс настройки моделей таким образом, чтобы их поведение соответствовало ожиданиям и предпочтениям пользователей. Важно, чтобы модели не только генерировали осмысленные ответы, но и учитывали вопросы безопасности, полезности и достоверности.

Принцип выравнивания часто описывается через правило «Helpful, Honest & Harmless» (Полезность, Правдивость & Безвредность):

- Helpful (полезность): ответы модели должны быть максимально полезными и релевантными запросу пользователя. Это предполагает глубокое понимание контекста и способность модели адаптироваться к различным задачам.
- Honest (честность): модель должна предоставлять достоверную информацию и избегать преднамеренной или непреднамеренной дезинформации, в том числе галлюцинаций [7].
- Harmless (безвредность): модель должна избегать генерации оскорбительного или иного нежелательного контента. Это особенно важно в контексте публичного использования, где модель может взаимодействовать с широким кругом пользователей.

Процесс выравнивания включает несколько этапов:

1. Сбор данных, отражающих предпочтения пользователей. Это может включать обратную связь от людей, ранжирование ответов модели или другие формы взаимодействия.

2. Настройка модели с использованием методов обучения с подкреплением, таких как Reinforcement Learning from Human Feedback [2], где награда определяется на основе человеческой обратной связи.

3. Тестирование и оценка выровненной модели для оценки её соответствия ожиданиям от адаптации.

2.2 Supervised Fine-Tuning

Метод SFT представляет собой процесс дообучения языковых моделей с использованием размеченных данных, который позволяет адаптировать их под конкретные задачи или улучшить качество работы в определённых условиях. Подходы, подобные LIMA [6], показывают, что даже с меньшим объёмом данных можно добиться заметных улучшений за счёт правильной разметки.

SFT основан на обучении с учителем, где модель корректирует свои параметры для минимизации ошибки между предсказаниями и правильными ответами из размеченного набора данных. Ключевые элементы метода включают размеченные датасеты, содержащие примеры входных запросов и целевых выходов. Наиболее распространённой функцией потерь является кросс-энтропия, которая оценивает разницу между предсказанным распределением вероятностей и целевым распределением. Для настройки параметров модели используются методы оптимизации, такие как Adam или его модификации, которые корректируют веса модели в сторону уменьшения значения функции потерь.

Примеры применения SFT включают:

- Адаптация под специфический язык: использование русскоязычных данных для улучшения производительности модели при работе с текстами на русском языке.

- Решение прикладных задач: дообучение модели для задач, таких как перевод, классификация текста, генерация контента и т.д.

- Подготовка к выравниванию: используется как предварительный этап перед применением методов, таких как RLHF [2], чтобы предоставить модели базовый уровень понимания задачи.

Таблица 1: Преимущества и ограничения метода SFT

Преимущества	**Ограничения**
Простота реализации и интерпретируемость.	Сильная зависимость от качества и объёма размеченных данных.
Эффективность при наличии качественных размеченных данных.	Низкая способность учитывать более сложные предпочтения пользователей, чем включены в обучающий датасет.
Возможность быстрой адаптации к новым задачам.	

Таким образом, SFT является важным этапом в адаптации языковых моделей, позволяющим подготовить их для дальнейшего использования и совершенствования.

2.3 Обучение с подкреплением: общая теория и применение в БЯМ

Обучение с подкреплением — это подход, при котором агент учится принимать оптимальные решения, взаимодействуя с окружающей средой. Основная цель агента — максимизировать суммарное вознаграждение, которое он получает за свои действия. Формально задача Reinforcement Learning (RL) описывается как Марковский процесс принятия решений, который включает:

- **Пространство состояний (S)**: набор возможных состояний среды, в которых может находиться агент.
- **Пространство действий (A)**: множество доступных агенту действий в каждом состоянии.
- **Функция переходов (P)**: вероятность перехода среды из одного состояния в другое в результате выполнения действия.
- **Функция награды (R)**: скалярное значение, которое агент получает за выполнение действия в определённом состоянии.

Основная задача агента заключается в том, чтобы научиться стратегии $\pi(a|s)$, которая максимизирует ожидаемую суммарную награду:

$$G = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma \cdot r_{t+1}\right],$$

где:

- G — общее ожидаемое вознаграждение;
- r_{t+1} — награда, полученная агентом на шаге $t+1$;
- $\gamma \in [0, 1]$ — коэффициент дисконтирования, который определяет важность будущих наград: чем меньше γ, тем меньше учитывается значение долгосрочных вознаграждений.

Применение RL в контексте БЯМ связано с задачей согласования моделей с человеческими предпочтениями. RL позволяет дообучать модели на основе обратной связи от человека, что делает ответы более соответствующими ожиданиям пользователей. Одним из важных элементов RL в LLM является функция награды, которая должна быть тщательно спроектирована. Чаще всего она формируется на основе человеческих оценок или автоматически сгенерированных метрик. Эта функция определяет, насколько ответы модели релевантны, точны и понятны.

Применение RL позволяет решать множество задач, включая:

- улучшение качества генерации текста;
- учёт специфических запросов и требований пользователя;
- минимизацию нежелательных или неуместных ответов.

2.4 Reinforcement Learning from Human Feedback (RLHF)

Метод RLHF [2] представляет собой подход к адаптации БЯМ с использованием обратной связи от людей. Основная цель RLHF заключается в улучшении качества генерации текста, чтобы ответы модели лучше соответствовали ожиданиям и предпочтениям пользователей.

RLHF включает следующие этапы:

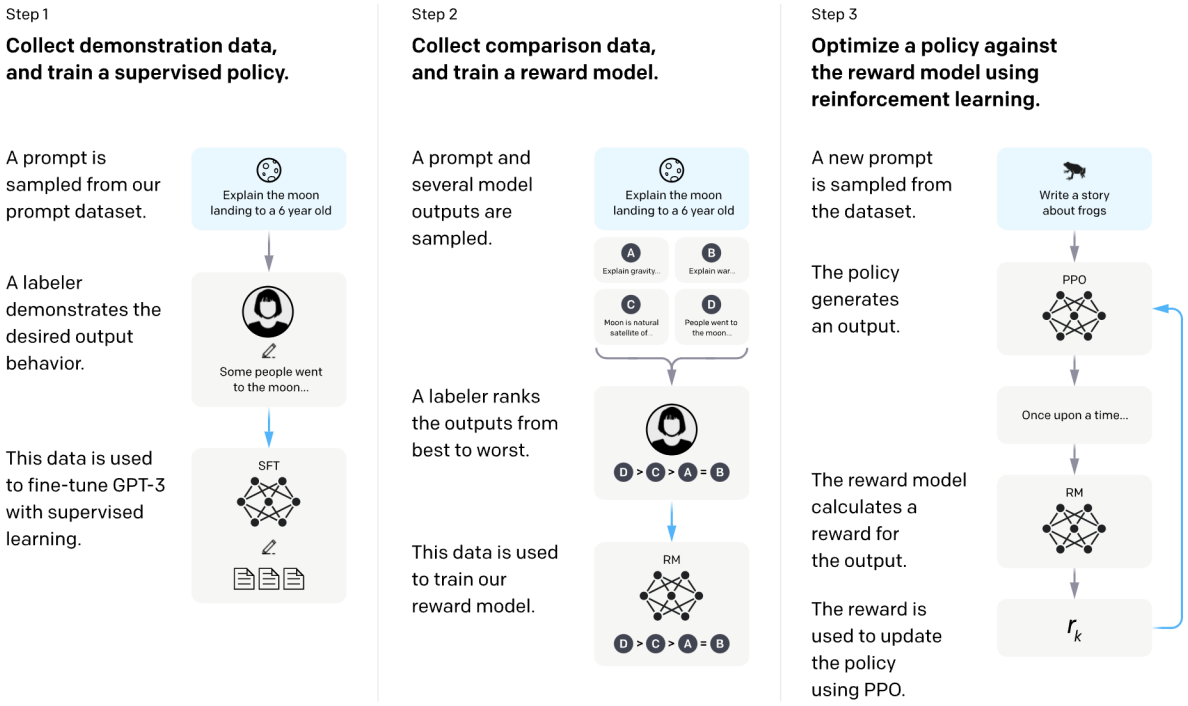

Рис. 1: Иллюстрация работы механизма RLHF.

1. Модель (обычно полученная после этапа SFT) генерирует ответы на запросы из заранее подготовленного набора.

2. асессоры ранжируют или оценивают сгенерированные ответы по качеству.

3. На основе оценок обучается модель награды R_ψ, которая предсказывает, насколько хорош ответ.

4. Модель дообучается с использованием алгоритма Proximal Policy Optimization (PPO), чтобы максимизировать ожидаемую награду, предсказанную моделью награды.

Для обучения модели награды используется вероятность, что один ответ лучше другого. Это моделируется функцией:

$$P(a > b \mid s) = \sigma\big(r_\psi(s,a) - r_\psi(s,b)\big),$$

где s — запрос, a и b — два ответа, r_ψ — предсказанная награда, а σ — сигмоида.

Параметры модели награды обучаются с использованием метода максимального правдоподобия, чтобы вероятность корректно отражала выбор асессоров.

Оптимизация языковой модели проводится с использованием метода Proximal Policy Optimization (PPO). Этот метод позволяет обучать стратегию π_θ так, чтобы она максимизировала ожидаемую награду:

$$J(\pi_\theta) = \mathbb{E}_{s\sim D,\, a\sim\pi_\theta}\big[r_\psi(s,a)\big],$$

где:

- $J(\pi_\theta)$ — целевая функция, которую необходимо максимизировать. Она выражает ожидаемую награду, получаемую агентом за действия, сгенерированные стратегией π_θ.
- $\pi_\theta(a|s)$ — стратегия с параметрами θ. Она определяет вероятность выбора действия a в состоянии s. В контексте БЯМ действие — это генерация следующего токена, а состояние — это контекст генерируемой последовательности.
- $s \sim D$ — запросы s выбираются из распределения D. Это означает, что запросы, на которые отвечает модель, поступают из некоторого заранее определённого набора данных.
- $a \sim \pi_\theta(a|s)$ — действия a генерируются на основе текущей стратегии π_θ.
- $r_\psi(s,a)$ — награда, предсказанная моделью награды R_ψ. Она отражает, насколько качественным считается действие a для состояния s.

Таблица 2: Преимущества и ограничения метода RLHF

Преимущества RLHF	**Ограничения RLHF**
Способность учитывать сложные предпочтения пользователей.	Высокая стоимость сбора данных обратной связи от асессоров.
Улучшение качества и согласованности ответов модели.	Сложность проектирования модели награды и алгоритма оптимизации.
Возможность работы с большим количеством запросов и сценариев.	Необходимость значительных вычислительных ресурсов.

Таким образом, RLHF представляет собой мощный инструмент для согласования больших языковых моделей с предпочтениями пользователей. Его использование особенно актуально в задачах, где требуется максимальная релевантность и надёжность ответов. Также он стал основой для развития других подходов к адаптации языковых моделей, которые сейчас используются для обучения многих популярных БЯМ.

2.5 Direct Preference Optimization (DPO)

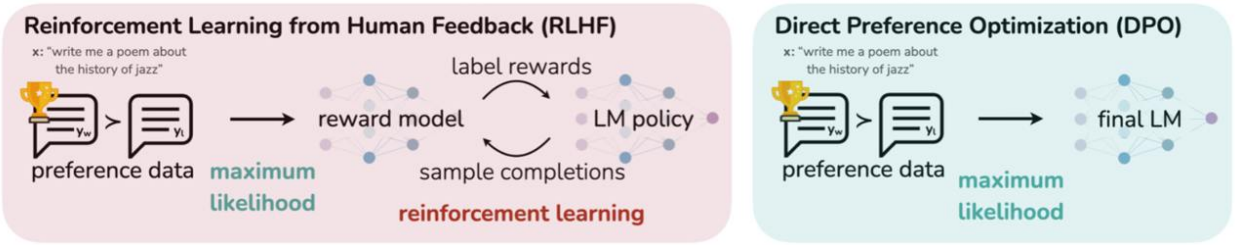

Рис. 2: Иллюстрация работы механизма DPO.

Метод DPO [3] представляет собой развитие RLHF и позволяет обучать модель напрямую на основе предпочтений пользователя, без необходимости в отдельной модели награды или сложных алгоритмах обучения с подкреплением, таких как Proximal Policy Optimization.

В основе метода DPO лежит вероятностная модель предпочтений, которая позволяет вычислять функцию потерь напрямую на основе предпочтений пользователей. Главная идея заключается в оптимизации стратегии модели π_θ таким образом, чтобы она генерировала ответы, наиболее предпочтительные для пользователя. Для этого используется следующая формула для награды:

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

где:

- s — запрос от пользователя.
- a — ответ, сгенерированный моделью.
- $\pi_\theta(a|s)$ — стратегия модели с параметрами θ.

- $\pi_{\text{ref}}(a|s)$ — референтная стратегия, обычно полученная на этапе SFT.
- β — гиперпараметр, контролирующий масштаб награды.

Эта формула позволяет выразить предпочтение одного ответа над другим в терминах логарифма отношения вероятностей их генерации текущей моделью и референтной моделью.

DPO требует сбора такой же обучающей выборки, как и для модели награды в RLHF. Стратегия обучается классическим SFT, однако на специфический лосс. В отличие от PPO, в DPO нет необходимости генерировать данные во время обучения, как и нет необходимости учить отдельную модель награды. По сложности вычислений DPO всё же дороже обучения с учителем, потому что в функции ошибки участвуют вероятности не только обучаемой модели, но и SFT — её также требуется хранить в памяти. Тем не менее, DPO использует единый этап оптимизации и исключает модель награды, что упрощает процесс. Таким образом, DPO является более простой и экономичной альтернативой RLHF, подходящей для случаев, когда требуется минимизация вычислительных затрат при сохранении хорошего качества генерации и стабильности обучения.

Ограничения DPO:

- Эффективность метода сильно зависит от качества аннотаций и обучающей выборки.
- Приводит к несоответствию между генеративными моделями, используемыми при выводе, и моделью «награды».
- DPO использует две стратегии: π_θ и π_{ref}, что требует использования эталонной модели во время процесса обучения, увеличивая вычислительные потребности.
- Побочный эффект: модель начинает генерировать необоснованно длинные последовательности.

2.6 SimPO (Simple Preference Optimization)

Метод SimPO [4] представляет собой упрощённый подход к обучению языковых моделей, направленный на устранение некоторых ограничений DPO. SimPO фокусируется

на прямой оптимизации предпочтений без использования сложных моделей награды и референтных стратегий, что делает его более эффективным и менее ресурсозатратным.

SimPO строится на упрощённой функции награды, основанной на метрике правдоподобия. Эта метрика позволяет оценивать качество ответа модели напрямую, исключая необходимость использования дополнительной эталонной стратегии. Награда задаётся следующим образом:

$$r_{\text{SimPO}}(x, y) = \frac{\beta}{|y|} \sum_{i=1}^{|y|} \log \pi_\theta(y_i \mid x,\ y_{<i}),$$

где:

- x — запрос, поступающий на вход модели.
- $y = (y_1, y_2, \ldots, y_{|y|})$ — сгенерированный ответ.
- $\pi_\theta(y_i \mid x, y_{<i})$ — вероятность предсказания токена y_i при условии запроса x и предыдущих токенов ответа $y_{<i}$.
- β — гиперпараметр, определяющий масштаб награды.
- $|y|$ — длина ответа.

Эта формула обладает следующими преимуществами по сравнению с DPO. Во-первых, награда прямо пропорциональна вероятностной функции, используемой для генерации ответов, что делает процесс обучения более согласованным. Во-вторых, исчезает необходимость в эталонной модели (π_{ref}), что уменьшает вычислительные затраты и снижает требования к памяти.

Для улучшения способности модели различать предпочтительные ответы вводится целевая «маржа» γ, которая регулирует степень предпочтения одного ответа над другим:

$$p(y_w \succ y_l \mid x) = \sigma\big(r(x, y_w) - r(x, y_l) - \gamma\big),$$

где:

- y_w и y_l — «победивший» и «проигравший» ответы, соответственно.
- σ — сигмоида, преобразующая разницу наград в вероятность.

- γ — гиперпараметр, целевая маржа, регулирующая требуемую разницу в наградах для предпочтения одного ответа над другим.

Маржа определяет, насколько один ответ должен превосходить другой, чтобы считаться лучше. Увеличение маржи приводит к улучшению обобщения, то есть к более чёткому различию между качественными и некачественными ответами.

Преимущества SimPO относительно DPO:

- Награда прямо пропорциональна метрике, которая фактически управляет генерацией.
- Отсутствует необходимость в эталонной модели, что приводит к большей эффективности вычислений и меньшим требованиям к памяти.
- Устойчивость к ошибкам в данных и гиперпараметрах благодаря упрощённой формулировке задачи.

Подводя итог, SimPO представляет собой значительное упрощение по сравнению с DPO, сохраняя при этом конкурентоспособные результаты в задачах выравнивания языковых моделей. Этот метод особенно полезен в условиях ограниченных ресурсов или необходимости быстрого прототипирования.

2.7 Kahneman-Tversky Optimization (KTO)

Метод KTO [5] основан на принципах теории перспектив, предложенной Канеманом и Тверски. Эта теория утверждает, что люди воспринимают потери и выигрыши асимметрично: потери вызывают более сильное эмоциональное воздействие, чем выигрыши той же величины. KTO адаптирует эту идею для обучения языковых моделей, делая акцент на минимизации вероятности нежелательных или неподходящих ответов.

KTO направлен на то, чтобы модель избегала ошибок с большей вероятностью, чем стремилась к генерации идеальных ответов. Это достигается за счёт использования бинарного сигнала обратной связи и модификации стандартных функций оптимизации.

Бинарная функция награды:

$$R(x) = \begin{cases} +1, & \text{если } x \text{ является приемлемым;} \\ -1, & \text{если } x \text{ является неприемлемым.} \end{cases}$$

Здесь x — это выходной ответ модели.

Для ограничения нежелательных результатов используется модифицированная формула KL-дивергенции:

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)},$$

где $P(x)$ — распределение вероятностей модели, а $Q(x)$ — эталонное распределение на основе человеческой обратной связи. Так, в KTO увеличивается штраф за повышение вероятности нежелательных ответов.

Преимущества KTO:

- Требуется только бинарная обратная связь от пользователя, что делает данные менее подверженными ошибкам аннотации и упрощает сбор обучающей выборки.
- Устойчивость: модель становится более надёжной в условиях неопределённости.

Метод KTO особенно полезен для задач, где критически важно минимизировать вероятность вредных или токсичных ответов. Например, он может быть использован в системах, предназначенных для взаимодействия с уязвимыми группами пользователей, или в контекстах, где ошибки могут иметь серьёзные последствия. Таким образом, KTO представляет собой подход, акцентирующий внимание на безопасности и надёжности языковых моделей, делая их более предсказуемыми и минимизируя риски в реальных сценариях использования.

2.8 Анализ методов обучения языковых моделей

Рассмотренные методы адаптации больших языковых моделей демонстрируют различные подходы к решению задачи выравнивания моделей с предпочтениями пользователей. Каждый из них имеет свои сильные и слабые стороны, которые определяют область их применения.

Общие ограничения методов:

- Все методы требуют высокого качества и объёма обучающих данных.
- Параметры требуют тщательной настройки, что может быть трудоёмким процессом.

- Методы могут снижать способность модели генерировать высококачественные ответы в пользу того, на что модель выравнивается.
- Сложно контролировать длину ответа модели.

Таблица 3: Сравнение методов адаптации больших языковых моделей

Метод	**Преимущества**	**Ограничения**
SFT	Простота реализации, достаточно высокая эффективность	Сильная зависимость от объёма и качества размеченных данных
RLHF	Учёт сложных предпочтений, улучшение качества	Высокая стоимость сбора данных, крупные вычислительные расходы
DPO	Простота обучения, отсутствие отдельной модели награды	Зависимость от эталонной стратегии и чувствительность к гиперпараметрам
SimPO	Минимизация вычислительных затрат, простота реализации	Требует разметки данных
KTO	Устойчивость, простота в сборе данных	Требует разметки данных

3 Постановка задачи

Целью данной работы является изучение и применение современных подходов дообучения больших языковых моделей для повышения их качества работы на русском языке, а также поиск оптимального из них для существующих методов оценки. Для достижения этой цели необходимо решить следующие задачи:

- Изучить существующие подходы к выравниванию БЯМ, включая Supervised Fine-Tuning, Reinforcement Learning from Human Feedback, Direct Preference Optimization, Simple Preference Optimization и Kahneman-Tversky Optimization.
- Выполнить анализ доступных датасетов для адаптации языковых моделей на русский язык.
- Разработать и реализовать программное решение для дообучения выбранной модели.
- Провести эксперименты с различными методами дообучения.
- Выбрать оптимальный метод оценивания выравненной модели относительно исходной.
- Оценить результаты дообучения, используя как количественные метрики, так и качественные показатели.

В результате выполнения практической составляющей работы предполагается проведение сравнительного анализа различных методов выравнивания на инструкциях на русском языке и выявление наиболее эффективного.

4 Программная реализация

4.1 Введение

В рамках данного исследования основное внимание было уделено адаптации большой языковой модели для генерации ответов средней длиной около 540 токенов, сохраняя при этом высокое качество, оцениваемое с использованием метрики *LLM-as-a-Judge*. Основная задача заключалась в изучении практической реализации алгоритма SFT, параметров обучения и анализе оценки модели в зависимости от данных с целью достижения оптимального компромисса между длиной ответов и их качеством.

Для экспериментов была выбрана языковая модель *Qwen2.5-3B*, адаптированная для работы с русскоязычными текстами. Данная модель представляет собой версию с 3 миллиардами параметров, дообученную с использованием техники Parameter-Efficient Fine-Tuning (PEFT) [1].

4.2 Датасеты

Для обучения модели были выбраны два высококачественных набора данных:

4.2.1 GrandMaster-PRO-MAX

Данный датасет был разработан командой Vikhrmodels и содержит около 156 000 уникальных пар «инструкция–ответ», синтетически созданных с использованием модели *GPT-4-Turbo*.

Подготовка данных включала:

- Отбор записей с длиной ответа $len \leq 610$ токенов для контроля длины генерации.
- Перевод пар «инструкция–ответ» в подходящий формат.
- Разделение на обучающую и валидационную выборки.
- Использование 10 %, 20 % и 40 % от общего объёма данных для оценки влияния объёма на производительность модели.

4.2.2 Saiga Scored

Этот датасет был создан Ильёй Гусевым и содержит 36 000 пар «инструкция–ответ», снабжённых оценками качества (`opus_score`), оценкой сложности и другими важными характеристиками.

Подготовка данных включала:

- Фильтрацию записей с `opus_score` ≥ 8 для обеспечения высокого качества обучающих данных.
- Разделение на обучающую и валидационную выборки.
- Семплирование данных для согласования их объёма с датасетом GrandMaster-PRO-MAX.

Оба датасета обеспечили необходимое разнообразие примеров, что позволило провести детальный анализ влияния качества и объёма данных на адаптацию модели.

4.3 Методики оценки

Для оценки качества работы модели использовалась метрика *LLM-as-a-Judge*, а также анализ длины сгенерированных ответов.

4.3.1 Метрика LLM-as-a-Judge

LLM-as-a-Judge — это методика, при которой большая языковая модель с высоким уровнем выравнивания с пользовательскими предпочтениями выступает в роли «судьи», оценивающей ответы других моделей. Оценка проводится путём сравнения с эталонными ответами или в режиме парного выбора, где модель выбирает лучший из двух предложенных вариантов.

В качестве входных данных используются ответы тестируемой модели и ответы эталонной модели (в нашем случае Qwen2.5-3B-Instruct). Формируется задача, которая объединяет ответы двух моделей, что позволяет сравнить их между собой. Модель-судья анализирует предоставленные ответы, оценивая их по следующим критериям:

- Релевантность: насколько ответ соответствует запросу.

- Полезность: насколько содержательна и информативна предоставленная информация.
- Ясность: насколько ответ логичен и хорошо структурирован.

Далее сохраняются оценки, которые затем используются для сравнения производительности моделей.

4.3.2 Анализ длины ответов

Параллельно с оценкой качества проводился анализ средней длины сгенерированных ответов для достижения заданной цели — средней длины около 540 токенов. Это позволило контролировать, насколько успешно модель адаптируется к требованиям по длине ответов, сохраняя при этом их качество. Важно, что иногда более длинные ответы получают более высокие оценки *LLM-as-a-Judge*, поэтому в экспериментах учитывалась и ручная проверка ответов.

4.4 Технологии и оборудование

4.4.1 Технологии

- LoRA (Low-Rank Adaptation) — техника, в основе которой обучаемые низкоранговые матрицы; позволяет адаптировать модель с меньшими вычислительными затратами [1].
- Hugging Face Transformers — фреймворк для работы с языковыми моделями, предоставляющий удобные инструменты для загрузки, настройки и обучения моделей.
- Datasets — модуль из экосистемы Hugging Face для работы с датасетами, включая загрузку, фильтрацию и подготовку данных.
- LLM-as-a-Judge — подход для автоматической оценки качества текстов, генерируемых моделью; в нём большая модель выступает в роли «судьи».
- Использование методов квантования для снижения требований к памяти при сохранении производительности модели.

4.4.2 Оборудование

Обучение и инференс выполнялись на сервере с графическим процессором NVIDIA A100 с 80 ГБ видеопамяти.

4.5 Проведённые эксперименты

В рамках работы было проведено несколько серий экспериментов, целью которых являлись адаптация модели для генерации ответов средней длиной около 540 токенов и улучшение качества, измеряемого метрикой *LLM-as-a-Judge*. В данном разделе описываются эксперименты, их методология и полученные результаты.

4.5.1 Оценка исходной модели

Для получения базового уровня производительности была проведена оценка исходной модели.

- Средняя длина ответов составила 624 токена.
- Качество, измеренное с использованием *LLM-as-a-Judge*, составило 46,5 баллов.

4.5.2 Эксперимент 1. Дообучение на GrandMaster-PRO-MAX с разным объёмом данных

В данном эксперименте тестировалось влияние объёма данных из датасета GrandMaster-PRO-MAX на длину и качество ответов модели.

Модель дообучалась на 10 %, 20 % и 40 % от общего объёма датасета. Производилась фильтрация по длине эталонного ответа ($len \leq 610$ токенов). Оценка проводилась на валидационном наборе во время обучения, а затем с использованием метрики *LLM-as-a-Judge*.

Результаты:

- 10 % данных: средняя длина ответов составила 596 токенов, качество — 60 баллов.
- 20 % данных: средняя длина уменьшилась до 532 токенов, качество — 61,4 балла.
- 40 % данных: средняя длина — 523 токена, качество — 56,6 балла.

Выводы: Обучение на большем объёме данных несколько уменьшает длину ответа, не вызывая при этом существенного ухудшения качества. Это говорит о том, что датасет GrandMaster-PRO-MAX действительно помогает улучшать или сохранять качество при снижении длины ответа, особенно по сравнению с исходной моделью.

4.5.3 Эксперимент 2. Дообучение на Saiga-Scored

Цель эксперимента заключалась в проверке влияния данных из датасета Saiga-Scored на длину и качество ответов.

Использовались примеры с оценкой `opus_score` ≥ 9. Модель обучалась отдельно на этиой выборке. Проводилась оценка по длине и качеству.

Средняя длина составила 469 токенов, качество — 57 баллов.

Выводы:

Несмотря на высокие оценки качества, средняя длина ответов существенно ниже целевого значения. Это указывает на необходимость либо использования дополнительного датасета, либо специальных техник для увеличения длины.

4.5.4 Эксперимент 3. Комбинированное обучение на двух датасетах

Данный эксперимент был направлен на объединение преимуществ датасетов GrandMaster-PRO-MAX и Saiga-Scored с анализом влияния объёма данных.

- Обучение проводилось последовательно:
  - Сначала на 10 % данных GrandMaster-PRO-MAX ($len \leq 610$) и на аналогичном количестве сэмплов Saiga-Scored (`opus_score` ≥ 9) — всего около 15 000 примеров из каждого датасета.
  - Затем на 20 % GrandMaster-PRO-MAX ($len \leq 610$) и аналогичном количестве сэмплов Saiga-Scored (`opus_score` ≥ 8) — около 30 000 примеров из каждого датасета.
- Оценка проводилась аналогично предыдущим экспериментам.

Результаты:

- В первом случае средняя длина ответов составила 504 токена, оценка — 57,4 балла.

- Во втором случае средняя длина ответов составила 445 токенов, оценка — 47 баллов.

Выводы:

Последовательное обучение действительно даёт компромисс между длиной и качеством. Однако сильное снижение длины во второй части из-за большого числа коротких примеров Saiga-Scored привело к заметному падению качества.

4.5.5 Эксперимент 4. Влияние фильтрации длины токенов

Цель эксперимента — проверить влияние фильтрации длины токенов в датасете на производительность модели.

Использовались данные GrandMaster-PRO-MAX с разными ограничениями по длине: $len \leq 610$ и $470 \leq len \leq 610$. Обучение проводилось на 10 % данных.

Результаты:

- $len \leq 610$: средняя длина — 596 токенов, качество — 60 баллов.
- $470 \leq len \leq 610$: средняя длина — 621 токен, качество — 67,8 балла.

Выводы: Фильтрация длины напрямую влияет на итоговый результат. Короткие ответы в обучающем наборе приводят к уменьшению средней длины ответа модели и, как следствие, к снижению оценки.

4.5.6 Эксперимент 5. Сравнение влияния датасетов

Цель данного эксперимента — сравнить влияние двух датасетов, GrandMaster-PRO-MAX и Saiga-Scored, на качество и длину ответов при использовании одинакового числа сэмплов.

Для обучения использовались 15 158 записей из каждого датасета (примерно 10 % от размера GrandMaster-PRO-MAX). Оценка проводилась с использованием метрики *LLM-as-a-Judge* и анализа средней длины ответов. Дополнительно ответы просматривались вручную.

Результаты:

- GrandMaster-PRO-MAX (10 % данных):

  - Средняя длина ответов — 620 токенов.
  - Качество по *LLM-as-a-Judge* — 69 баллов.

- Saiga-Scored (`opus_score` ≥ 9):
  - Средняя длина ответов — 469 токенов.
  - Качество по *LLM-as-a-Judge* — 57 баллов.

Выводы:

Датасет GrandMaster-PRO-MAX обеспечивает не только более высокую среднюю длину ответов, но и превосходит Saiga-Scored по метрике качества. Ручной анализ подтвердил, что ответы, сгенерированные на основе GrandMaster-PRO-MAX, выглядят более осмысленными и естественными.

4.6 Выводы

На основании проведённых экспериментов можно сделать следующие ключевые выводы:

- Использование датасета GrandMaster-PRO-MAX обеспечивает более длинные и при этом естественные ответы, лучше соответствующие целевой длине (около 540 токенов). Датасет Saiga-Scored, хотя и характеризуется хорошим качеством, не подходит для задач, требующих длинных ответов, из-за короткой средней длины обучающих примеров.

- Последовательное обучение на GrandMaster-PRO-MAX и Saiga-Scored позволяет достичь компромисса между длиной и качеством, однако чрезмерное добавление коротких примеров из Saiga-Scored может привести к ухудшению результатов.

- По результатам экспериментов модель, дообученная на GrandMaster-PRO-MAX, демонстрирует наиболее естественные ответы и достаточно высокое качество. Ручная проверка подтверждает, что ответы, формируемые на основе этого датасета, выглядят более осмысленными и подходят для практических задач.

Таблица 4: Сводка по всем экспериментам

Обучающие данные	**Средняя длина ответа (токены)**	**Оценка LLM-as-a-Judge**
Исходная модель		
Исходная модель	624	46.5
Обучение на GrandMaster-PRO-MAX		
10% GrandMaster-PRO-MAX	620	69.0
10% GrandMaster-PRO-MAX ($470 \leq \text{len} \leq 610$)	621	67.8
10% GrandMaster-PRO-MAX ($\text{len} \leq 610$)	596	60.0
20% GrandMaster-PRO-MAX ($\text{len} \leq 610$)	532	61.4
40% GrandMaster-PRO-MAX ($\text{len} \leq 610$)	523	56.6
Обучение на Saiga-Scored		
15158 сэмплов Saiga-Scored	469	57.0
Комбинирование датасетов		
10% GrandMaster-PRO-MAX + 50% Saiga-Scored	504	57.4
20% GrandMaster-PRO-MAX + 100% Saiga-Scored	445	47.0

5 Заключение

Актуальность задачи

Тема дообучения больших языковых моделей на инструкциях для повышения качества ответов на русском языке является крайне востребованной, учитывая рост популярности LLM и необходимость точной локализации под особенности русского языка.

Обзор методов

В работе дан обстоятельный обзор основных подходов к выравниванию языковых моделей:

- Supervised Fine-Tuning (SFT) — базовый подход, дающий первичную адаптацию к задаче.
- Reinforcement Learning from Human Feedback — эффективный способ учёта сложных предпочтений, но с высокой стоимостью сбора данных [2].
- Direct Preference Optimization, SimPO и KTO — альтернативные методы [3–5], каждый со своими сильными сторонами и ограничениями, направленные на упрощение или улучшение качества обучения за счёт прямой работы с пользовательскими предпочтениями либо бинарными оценками.

Эксперименты

Практическая часть работы была сосредоточена на дообучении русскоязычной модели Qwen2.5-3B-Instruct с использованием датасетов GrandMaster-PRO-MAX (отличается достаточной длиной ответов и естественностью) и Saiga-Scored (содержит больше характеристик данных и качественные ответы, но часто короче требуемого).

Серия экспериментов

Был проведён ряд экспериментов с разными долями датасетов и фильтрацией по длине, что позволило понять влияние объёма и качества данных на итоговую длину

ответа и его оценку. Кроме того, удалось найти практический компромисс между длиной генерируемого ответа и качеством на LLM-as-a-Judge. В том числе стало ясно, что одни и те же датасеты могут в разной пропорции давать существенно различающиеся результаты.

Общая оценка работы

Выполнена обстоятельная исследовательская часть с постановкой конкретной задачи, чётко описана методика экспериментов, и приведён анализ результатов. Работа даёт практические рекомендации по выбору датасетов и подходов к дообучению, что позволит в дальнейшем исследовать локализацию больших языковых моделей на русский язык.

Список литературы

[1] **Hu E. J. et al.** Lora: Low-rank adaptation of large language models // arXiv preprint arXiv:2106.09685. – 2021.

[2] **Christiano P. F. et al.** Deep reinforcement learning from human preferences // Advances in neural information processing systems. – 2017. – Т. 30.

[3] **Rafailov R. et al.** Direct preference optimization: Your language model is secretly a reward model // Advances in Neural Information Processing Systems. – 2024. – Т. 36.

[4] **Meng Y., Xia M., Chen D.** Simpo: Simple preference optimization with a reference-free reward // arXiv preprint arXiv:2405.14734. – 2024.

[5] **Ethayarajh K. et al.** Kto: Model alignment as prospect theoretic optimization // arXiv preprint arXiv:2402.01306. – 2024.

[6] **Zhou C. et al.** Lima: Less is more for alignment // Advances in Neural Information Processing Systems. – 2024. – Т. 36.

[7] **Gekhman Z. et al.** Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? // arXiv preprint arXiv:2405.05904. – 2024.