https://doi.org/10.17323/jle.2024.22224

Повышение качества адаптации больших языковых моделей посредством проекции предобученных эмбеддингов

Михаил Тихомиров, Даниил Чернышев

Московский государственный университет имени М.В.Ломоносова, г. Москва, Россия

АННОТАЦИЯ

Введение: Недавние достижения в области технологий больших языковых моделей (LLM) представили мощные открытые инструктивные LLM, которые соответствуют с точки зрения качества генерации текста ведущим моделей, таким как GPT-4. Несмотря на ускорение процессов внедрения LLM в средах с конфиденциальной информацией, отсутствие раскрытых данных обучения (в частности, инструктивных) затрудняет воспроизведение и делает эти достижения эксклюзивными для определенных моделей.

Цель: Учитывая тренд на мультиязычность последних LLM, преимущества обучения LLM, ориентированных на конкретный язык, уменьшаются, оставляя вычислительную эффективность единственным гарантированным преимуществом этой вычислительно дорогой процедуры. В данной работе предлагается экономически более эффективная схема адаптации LLM к языку, которая решает проблему ограниченного доступа к высококачественным данным.

Методология: Для решения проблем адаптации LLM к языку мы представляем Learned Embedding Propagation (LEP) — новый метод с более низкими требованиями к данным для обучения и минимальным воздействием на существующие знания LLM. LEP использует инновационную технику проекции эмбеддингов, минуя необходимость в настройке инструкций и напрямую интегрируя новые языковые знания в любой инструктивный вариант LLM. Кроме того, мы разработали Darumeru — новый бенчмарк для оценки качества генерации текста во время обучения, специально спроектированный для русского языка.

Результаты: Мы применили метод LEP для адаптации LLaMa-3-8B и Mistral-7B к русскому языку, протестировав четыре различных сценария адаптации токенизации. Оценка качества показала, что LEP обеспечивает конкурентоспособную производительность, сопоставимую с OpenChat-3.5 и LLaMa-3-8B-Instruct. Дальнейшее улучшение результатов было достигнуто посредством калибровки и дополнительных шагов дообучения с использованием инструкций.

Заключение: LEP предлагает жизнеспособную и эффективную альтернативу традиционной схеме адаптации LLM под конкретный язык с последующим дообучением с использованием инструкций, значительно сокращая затраты, связанные с адаптацией к языку, при этом сохраняя или превышая показатели качества исходных версий моделей.

КЛЮЧЕВЫЕ СЛОВА:

большие языковые модели, llama, языковая адаптация, генерация текста

Для цитирования: Тихомиров, М., & Чернышев, Д. (2024). Содействие адаптации большой языковой модели к русскому языку с помощью распространения изученного векторного представления. *Journal of Language and Education, 10*(4), 134-150 https://doi.org/10.17323/jle.2024.22224

Correspondence:
Михаил Тихомиров,
tikhomirov.mm@gmail.com

Получена: 15 августа 2024

Принята: 16 декабря 2024

Опубликована: 30 декабря 2024

ВВЕДЕНИЕ

Появление универсальных больших языковых моделей (LLM), обученных по инструкциям, таких как ChatGPT (Ouyang, 2022), существенно ускорило развитие технологий обработки естественного языка. Однако, несмотря на значительные достижения в решении задач без предварительного обучения (zero-shot), закрытый характер таких моделей препятствовал их использованию в областях, связанных с конфиденциальной или эксклюзивной информацией, где любая утечка данных угрожает целостности производственного процесса. В результате растущий спрос на открытые альтернативные решения побудил исследователей разработать методы дистилляции знаний из современных LLM. Одним из первых подходов стала модель Alpaca (Taori, 2023), которая использовала ChatGPT для синтеза данных инструктивной настройки open-source

модели LLaMA (Touvron, 2023a). Несмотря на то, что Alpaca была далека от идеала, она вдохновила к созданию более совершенных схем, таких как BactrianX (Li, 2023), расширивший процесс синтеза за счет перевода на другие языки, что позволило обучать мультиязычные и открытые чат-боты. Однако с выходом GPT-4 (Achiam, 2023), продемонстрировавшей исключительные результаты в мультиязычной среде, стало возможным интегрировать явный этап перевода в процесс синтеза команд, что повысило доступность дистилляции знаний. Это привело к появлению серии спроектированных под конкретные языки моделей, обученных с использованием инструктивной настройки, таких как Saiga (Gusev, 2023), PolyLM (Wei, 2023), Vikhr (Nikolich, 2024), LLAMMAS (Kuulmets, 2024).

С повышением качества синтеза инструкций открытые языковые модели, специализированные для конкретных языков, сокращали разрыв с передовыми закрытыми решениями, но в конечном итоге достигали предела производительности обычной инструктивной настройки (Cui, 2023). Это ограничение объясняется низкой эффективностью использования внутренних знаний английского языка, которые доминируют в современных предварительно обученных LLM (Touvron, 2023b; Jiang, 2023; Dubey, 2024). В качестве возможного решения исследователи (Zhu, 2023; Li, 2024; Chai, 2024) предложили обогащать наборы данных для инструктивной настройки задачами перевода, которые предназначены для согласования новых языковых знаний с существующими английскими семантическими представлениями. Однако, как показали Ranaldi (2023) и Husain (2024), причина проблемы согласования, вероятно, кроется в неэффективности алгоритма токенизации, которую можно решить либо путем создания нового языково-специфического словаря токенов, либо путем повторного использования английских токенов для романизированного представления языка.

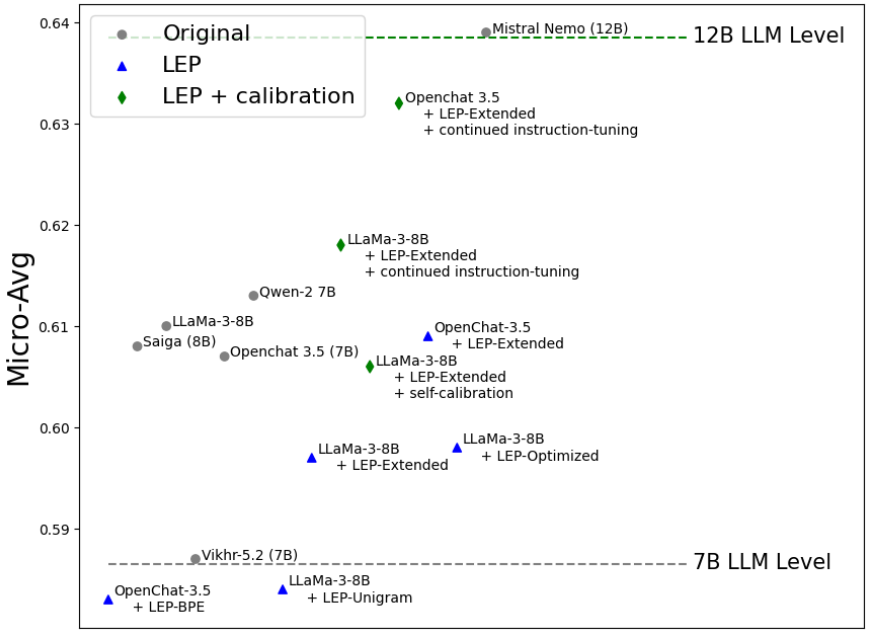
Вдохновленные работами Lakew (2018), Kuratov (2019), Rust (2021) и Yang (2022) по адаптации словаря для моделей энкодеров, Cui и др. (2023) предложили схему дообучения для полной языковой адаптации больших языковых моделей. В сочетании с инструктивной настройкой на синтезированных данных этот подход позволил создать ChineseLLaMa — первую открытую модель, достигшую уровня производительности ChatGPT на китайском при существенно улучшенной вычислительной эффективности благодаря токенизатору, адаптированному к структурам китайского языка. Этот подход был детально изучен Тихомировым (2023) в контексте адаптации LLaMa-2 (Touvron, 2023b) для русского языка, где показано, что эффективность семантического выравнивания может быть дополнительно улучшена с помощью более морфологически точного алгоритма токенизации. Более того, Nguyen (2023) продемонстрировал, что полная схема языковой адаптации больших языковых моделей превосходит современные закрытые аналоги для малоресурсных языков из-за их смещения в сторону популярных языков.

Несмотря на то что текущая итерация алгоритма языковой адаптации относительно экономична, польза от разработки языково-адаптированных LLM снижается на фоне стремительного развития технологии LLM и многоязычной специализации открытых моделей. В то же время становится обычной практикой выпуск моделей, обученных с использованием инструкций (Jiang, 2023; Dubey 2024), они работают наравне с современными закрытыми аналогами, не раскрывая данных инструктивной настройки, качество этих данных является основным фактором, определяющим итоговые возможности LLM по решению задач (Zhou 2024). Создание наборов данных такого качества требует значительных инвестиций в ручную разметку, что под силу только крупным организациям (Dubey 2024). Если высококачественный набор инструкций для конкретного языка недоступен, то преимущество модели, прошедшей полную языковую адаптацию, будет заключаться только в более высокой вычислительной эффективности, поскольку использование данных инструктивной настройки более низкого качества приведет к ухудшению производительности при решении задач.

Чтобы сократить затраты на языковую адаптацию и обеспечить прямую языковую адаптацию инструктированных LLM, мы предлагаем обновленную схему для языковой адаптации — проекцию предобученных эмбеддингов (Learned Embedding Propagation, LEP). В отличие от схемы языковой адаптации LLM (Cui, 2023), наш метод требует меньше данных и вычислительных ресурсов благодаря ограниченному влиянию предварительного обучения на параметры модели. Это компенсируется новой специализированной процедурой проецирования эмбеддингов, позволяющей пропустить этап дообучения инструкциям и вместо этого внедрить знания нового языка в любой существующий инструктированный вариант модели. Для дальнейшего содействия русской адаптации мы разработали новый облегченный бенчмарк для оценки качества LLM во время обучения — Darumeru. Мы протестировали LEP на моделях Mistral-7B и LLaMa-3-8B с четырьмя вариантами русской токенизации. Результаты оценки (Рисунок 1) демонстрируют, что, несмотря на меньшую параметризацию, наш метод языковой адаптации не только восстанавливает исходное качество инструктированной модели, но и в некоторых случаях даже превосходит его со значительным отрывом. Дополнительные экспериментальные исследования по улучшению адаптированных моделей с помощью дополнительных шагов калибровки и дообучения инструкциям также подтверждают эффективность методологии, повышая производительность моделей выше существующих аналогов.

Рисунок 1

Сравнение качества предлагаемого метода адаптации на основе бенчмарка Darumeru



МЕТОДОЛОГИЯ

Адаптация моделей к языку

В рамках продолжения предыдущих исследований по языковой адаптации больших языковых моделей (Cui, 2023; Tikhomirov, 2023) мы провели оптимизацию словаря модели для лучшего соответствия морфологии русского языка, а затем продолжили процесс предобучения на большом корпусе русскоязычных текстов различных жанров и тематической направленности.

Формально адаптация модели состоит из трех этапов:

(1.) Обучение токенизатора;
(2.) Инициализация эмбеддингов модели;
(3.) Дообучение на корпусе текстов новых эмбеддингов (как входных, так и выходных).

Обучение токенайзера

Поскольку не существует общепринятых эталонных практик для оптимизации словаря, мы рассмотрели четыре варианта обучения токенизации:

BPE — полная замена исходного токенизатора новым токенизатором, обученным на основе алгоритма BPE (Vries, 2021), который используется в большинстве современных больших языковых моделей.
Unigram — полная замена исходного токенизатора новым токенизатором, обученным на основе алгоритма Unigram (Kudo, 2018; Tikhomirov, 2023).
Extension — расширение исходного словаря BPE путем предварительного построения нового словаря BPE для русскоязычного корпуса и последующего объединения его с исходным (Cui, 2023).
Optimization — реорганизация существующего словаря BPE путем сокращения его до 50 % наиболее часто встречающихся токенов русского корпуса, с последующим расширением словаря до исходного размера за счет добавления новых токенов. Этот метод применяется только для LLM с большим английским словарем.

Инициализация эмбеддингов

Предыдущие работы по адаптации больших языковых моделей (Cui, 2023; Tikhomirov, 2023; Nguyen, 2023) показали, что простое усреднение эмбеддингов перекрывающихся сабтокенов является достаточным решением для инициализации эмбеддингов. Формально при наличии исходных векторов эмбеддингов и новых новые эмбеддинги инициализируются следующим образом:

$$v_{new}(t_i^n) = \frac{1}{K}\sum_{j=1}^{K} v_{old}(t_j^o); \quad (1)$$

$$tokenize_{old}(t_i^n) = [t_1^o, \ldots, t_K^o]. \quad (2)$$

где — исходная функция токенизации;
— токен нового словаря;
— токен исходного словаря.

В то время как существуют и более продвинутые методы инициализации, недавние исследования вариантов для языковой адаптации больших языковых моделей (Tejaswi, 2024) показали, что усреднение эмбеддингов обеспечивает наилучшее ожидаемое качество адаптации, а разрыв в про-

изводительности с методами, специально разработанными под конкретные задачи, находится в пределах стандартного отклонения. Поэтому для всех экспериментов мы используем описанную стратегию инициализации эмбеддингов путем усреднения субтокенов.

Продолженное предобучение

Основная проблема инициализации эмбеддингов заключается в том, что, несмотря на введение новых токенов, большая языковая модель сохраняет привычку использовать токены, присутствовавшие в исходной токенизации. В результате вычислительная производительность модели при генерации текста остается такой же, поскольку модель склонна использовать больше токенов на слово, чем ожидается, а также неправильно интерпретирует новые токены из-за омонимии контекста токенов.

Для смягчения этой проблемы распространенной тактикой является обучение вновь инициализированных эмбеддингов на корпусах языка адаптации, используя ту же задачу предобучения, что и для большой языковой модели, а именно языковое моделирование. Для этой задачи входной текст разбивается на последовательности токенов, и модели предлагается предсказать для каждой последовательности следующий возможный токен. Оптимизация модели выполняется с использованием простой функции потерь перекрестной энтропии, поэтому для задачи предобучения можно использовать любой текстовый корпус.

Продолженное предобучение только эмбеддингов позволяет модели адаптировать их для внутренней семантики, перераспределяя существующие языковые знания среди вновь введенных токенов. Однако некоторые исследователи (Cui, 2023; Tikhomirov 2024) отмечают, что предобучения только эмбеддингов может быть недостаточно для правильного согласования модели и словаря — необходимо также обучать промежуточные слои модели. С другой стороны, увеличение числа параметров, подлежащих оптимизации, снижает стабильность процесса обучения, что, в свою очередь, существенно повышает требования к размеру данных и вычислительным ресурсам. В качестве компромиссного решения мы дополняем предобучение эмбеддингов процедурой последующего выравнивания слоев, которая повторно использует существующие дообученные версии адаптированной модели.

Проекция предобученных эмбеддингов

Вопрос эффективного переноса знаний для моделей, адаптированных к языку, ранее изучался в контексте моделей-кодировщиков. Чтобы решить проблему отсутствия набора данных для настройки задач на целевом языке, Artetxe и др. (2019) предложили простой алгоритм для переноса знаний в модели BERT:

(1.) Предварительно обучить с нуля полную языковую модель на доступных больших моноязычных текстовых корпусах (например, на английском языке), используя в качестве целевой задачи языковое моделирование (для BERT это маскированное языковое моделирование);
(2.) Создать копию предобученной модели и заменить эмбеддинги оригинала новыми эмбеддингами для целевого языка;
(3.) Продолжить обучение модифицированной модели на одноязычных корпусах целевого языка только для эмбеддингов, замораживая (не обновляя) все остальные слои. Используется та же задача языкового моделирования;
(4.) Дообучить копию на наборе данных целевой задачи, сохраняя эмбеддинги замороженными;
(5.) Заменить эмбеддинги дообученной копии эмбеддингами оригинальной модели, полученными после продолженного предобучения (continued pre-training) на корпусах целевого языка.

Основным преимуществом описанного алгоритма является то, что этап продолженного предобучения требует гораздо меньше данных, чем исходное предобучение (pre-training) с нуля, поскольку требует обучения только части параметров модели. Это снижает сложность задачи оптимизации модели и, таким образом, обеспечивает более быструю сходимость (Kaplan, 2020). Основная гипотеза заключается в том, что знания для решения задач не зависят от языка, что было подтверждено в оригинальных экспериментах (Artetxe, 2019) для задач понимания естественного языка и классификации документов. Однако авторы отметили, что тонкой настройки для целевых задач с замороженными эмбеддингами недостаточно для правильного выравнивания при замене эмбеддингов. Для максимизации эффективности обработки целевого словаря требуются дополнительные преобразования или введение специальных штрафов в процесс обучения. В качестве возможного решения проблемы выравнивания эмбеддингов Chen и др. (2023) предложили использовать специальный режим предварительного обучения с активным «забыванием» эмбеддингов, чтобы побудить языковую модель накапливать знания в промежуточных слоях. Недостатком такого подхода является то, что мы должны иметь полный контроль над исходным предварительным обучением, что невозможно для современных больших языковых моделей, полученных путем обучения на высококачественных проприетарных наборах данных с огромными вычислительными затратами.

Мы утверждаем, что выравнивание при замене эмбеддингов может быть достигнуто без специальных процедур обучения, используя траекторию обновления параметров при тонкой настройке. Ilharco и др. (2023) показали, что траектория тонкой настройки может быть аппроксимирована линейными преобразованиями параметров базовой модели, которые могут быть получены из декомпозиции параметров дообученных вариантов моделей. Следовательно, посредством определения соответствующих линейных

преобразований параметров эмбеддингов мы можем аппроксимировать результаты полной схемы языковой адаптации без использования набора данных для обучения с инструкциями.

Формально обозначим входные и выходные эмбеддинги большой языковой модели (LLM) как *I, O*, а W — как псевдолинейное приближение композиции промежуточных слоев LLM:

$$LLM_{base} = I_{base} W_{base} O_{base} \tag{3}$$

Пусть D, U — это линейные преобразования эмбеддингов, которые выравнивают исходные эмбеддинги с настроенными слоями:

$$LLM_{base \to inst} = I_{inst} W_{inst} O_{inst} = \\ = I_{base} D_{inst} W_{inst} U_{inst} O_{base} \tag{4}$$

Поскольку наша стратегия инициализации эмбеддингов целевого языка усредняет эмбеддинги перекрывающихся токенов в и , мы можем формализовать процесс инициализации с помощью операции преобразования словаря :

$$LLM_{base \to ru} = T_{ru} I_{base} W_{base} O_{base} T_{ru}^T = I_{ru} W_{base} O_{ru} \tag{5}$$

Следуя описанной выше логике, тонкая настройка базовой модели, адаптированной к языку, может быть представлена следующим образом:

$$LLM_{ru \to inst} = T_{ru} I_{base} D_{inst}^{ru} W_{ru \to inst} U_{inst}^{ru} O_{base} T_{ru}^T \tag{6}$$

Предполагая, что оптимальное , мы приходим к окончательному уравнению для отображения дообученных эмбеддингов $I_{ru/cpt}, O_{ru/cpt}$:

$$LLM_{ru/cpt \to inst} = I_{ru/cpt} D_{inst}^{ru} W_{inst} U_{inst}^{ru} O_{ru/cpt} \tag{7}$$

Оставшиеся переменные $D_{inst}^{ru}, U_{inst}^{ru}$ определяются предположениями о свойствах выравнивания эмбеддингов. В наших экспериментах мы рассматриваем три варианта:

1. Прямая замена эмбеддингов.
2. Коррекция перекрывающихся токенов.
3. Проекция преобразования словаря.

Прямая замена эмбеддингов

Учитывая, что большинство современных больших языковых моделей обучаются на многоязычных наборах данных, можно ожидать, что их внутренние представления адаптированы для независимой от языка обработки текста. Аналогично оригинальным работам по переносу знаний на основе эмбеддингов для моделей-кодировщиков, мы предполагаем, что слой эмбеддингов содержит только концептуальную информацию, то есть мы предполагаем, что , где $D_{inst}^{ru} = U_{inst}^{ru} = E$ where E — единичная матрица.

Коррекция перекрывающихся токенов

Поскольку рассматриваемые большие языковые модели изначально были разработаны для многоязычной генерации текста, они имеют базовый набор наиболее распространенных токенов для популярных языков, включая русский. Идея заключается в том, чтобы найти объединение исходного словаря и адаптированного к языку словаря и использовать это подмножество для сокращения до общих компонентов инициализации эмбеддингов , где . Это позволяет аппроксимировать проекции эмбеддингов как $D_{inst}^{ru} \approx D_{inst}$ and $U_{inst}^{ru} \approx U_{inst}$ и :

$$D_{inst}^{ru} = I_{base/com}^{-1} I_{inst/com} \tag{9}$$

$$I_{X/com} = [I_X^{idx(t)}]_{t \in C} \tag{10}$$

$$O_{X/com} = [O_X^{idx(t)}]_{t \in C} \tag{11}$$

где — функция, которая возвращает индекс токена *t* в матрице эмбеддингов. Следует отметить, что матрицы , вероятно, не будут обратимыми, и поэтому их обращение должно быть аппроксимировано с помощью метода наименьших квадратов.

Проекция преобразования словаря

Поскольку преобразование инициализации эмбеддингов является универсальным как для базовых, так и для тонко настроенных моделей, мы можем вывести альтернативное уравнение для получения адаптированной на язык инструктивной LLM:

$$LLM_{inst \to ru} = T_{ru} I_{inst} W_{inst} O_{inst} T_{ru}^T \tag{12}$$

Предполагая, что оба варианта инструктивно настроенных адаптаций эквиваленты $LLM_{ru \to inst} = LLM_{inst \to ru}$, мы получаем следующие формулы выравнивания эмбеддингов:

$$D_{inst}^{ru} = (T_{ru} I_{base})^{-1} T_{ru} I_{inst} \tag{13}$$

$$U_{inst}^{ru} = O_{inst} T_{ru}^T (O_{base} T_{ru}^T)^{-1} \tag{14}$$

Аналогично предыдущему методу выравнивания, вычисление матриц преобразования включает метод наименьших квадратов для нахождения псевдообратных матриц. Это основная причина, по которой преобразование словаря не должно быть изолировано. Предварительные эксперименты показали, что такое упрощение увеличивает погрешность преобразований выравнивания, что снижает качество процедуры распространения эмбеддингов.

Бенчмарк Darumeru

Существующие бенчмарки оценки качества LLM для русского языка (Fenogenova, 2024) не раскрывают ответы на тестовых данных для локальной оценки. С одной стороны, такая инициатива разумна на фоне растущей тенденции обучения на тестовых данных, что делает результаты ранжирования LLM бессмысленными. С другой стороны, скрытые ответы на тестовых данных означают, что оценка требует наличия онлайн-подключения к системе бенчмарка, что препятствует оценке в офлайн-вычислительных средах, тем самым откладывая оценку до конца сессии обучения. Более того, отсутствие доступа к тестовым ответам делает невозможным классификацию типов ошибок предсказания, вследствие чего ограничивается анализ качества после обучения.

Для решения этой проблемы мы разработали новую систему бенчмарков, которая фокусируется на быстрой и информативной оценке качества генерации текста LLM. Этот бенчмарк состоит из комбинаций открытых частей наборов данных из MERA (Fenogenova, 2024), mmlu_ru / mmlu_en, RuCoLA (Mikhailov, 2022), а также новых наборов данных для оценки генерации текста — всего 17 наборов данных. Более подробное описание каждого набора данных приведено в следующих разделах.

Фреймворк

Разработанный фреймворк использует формат сообщений для обеспечения совместимости как с предварительно обученными, так и с инструктивными LLM. Это означает, что все данные задач для моделей преобразуются в последовательность пар «роль пользователя» — «содержание сообщения», из которых формируется итоговый запрос. Система поддерживает задачи, требующие оценки вероятности следующего токена, генерации или logsoftmax для всей сгенерированной последовательности. Оценка может проводиться непосредственно в обычной среде обучения моделей transformers или через специализированные фреймворки для использования моделей vLLM.

DaruMERA и DaruMMLU

Мы составили DaruMERA из следующих наборов данных MERA: MultiQ, PARus, RCB, RWSD, USE, ruOpenBookQA, ruWorldTree. Для лучшей оценки понимания языка мы также добавили валидационную выборку набора данных RuCoLA.

Для части DaruMMLU мы выделили ruMMLU (MERA) и дополнили его наборами данных MMLU из репозитория NLP-Core-Team[1].

В оригинальные наборы данных внесены следующие изменения:

(1.) Версия MultiQ была дополнена эталонными ответами. Существующие варианты ответов не соответствовали форме вопросов, так как они были извлечены из текста без надлежащей предобработки. Для исправления этой ситуации пары «вопрос — эталонный ответ» были переданы модели LLaMa-3-70B-Instruct для перефразирования ответа с учетом формулировки вопросов.
(2.) Версия ruMMLU отличается от аналогичной в репозитории NLP-Core-Team тем, что имеет примеры few-shot, общие для всех запросов, независимо от домена, а также использует не один фиксированный шаблон, а несколько вариантов в качестве инструкций.
(3.) При расчете PARus для каждого примера был сгенерирован такой же пример, но с другим порядком вариантов, и успешным считался только случай, когда модель предсказывала правильный вариант как для прямого, так и для обратного порядка.

Для измерения качества моделей на наборах данных PARus, RWSD, MMLU мы использовали метрику точности (accuracy). Для RCB, ruOpenBookQA и ruWorldTree мы усреднили точность и F1-macro. Для RuCoLa мы использовали среднее значение точности и коэффициента корреляции Мэтьюса (MCC). Для MultiQ мы использовали среднее значение метрик F1 и точного совпадения (exact match). Для USE использовалась нормализованная общая оценка.

DaruSum

Большинство задач оценки направлены на измерение способностей модели к пониманию текста и глобальных контекстуальных знаний, которые необходимы для правильной обработки запросов. Однако для генерации текста модель также должна уметь фильтровать входной текст для выделения содержимого, релевантного запросу, чтобы гарантировать получение пользователем желаемого ответа независимо от формата или размера ввода. Суммаризация текста является идеальной задачей оценки для такого случая, поскольку она требует как фильтрации входного содержимого, так и составления ответа из значимых фрагментов.

Существуют два подхода к суммаризации: экстрактивный и абстрактивный. Экстрактивная суммаризация — это по сути задача ранжирования на основе важности предложений, где аннотация формируется путем выбора топ-k ранжированных предложений. В свою очередь, абстрактивная суммаризация представляет собой задачу генерации текста, где ранжирование значимости интегрировано в процесс выборки токенов, поскольку модель сама направляет себя к наиболее краткой аннотации. Несмотря на то что

[1] https://github.com/NLP-Core-Team/mmlu_ru

абстрактивный подход считается более предпочтительным, автоматически отличить ошибки, связанные с субоптимальной фильтрацией содержимого, от ошибок генерации текста трудно. В то же время ограничение процесса генерации текста входными фрагментами, такими как предложения, по сути сводит задачу к экстрактивной суммаризации. Таким образом, для оценки точности фильтрации содержимого и качества генерации текста достаточно оценить абстрактивную суммаризацию в свободных и ограниченных условиях генерации.

Для набора данных по суммаризации мы выбрали Gazeta (Gusev, 2020), который зарекомендовал себя как стандарт для оценки автоматической суммаризации на русском языке. Для повышения точности процедуры оценки мы разработали протокол фильтрации примеров, чтобы все содержимое эталонной аннотации можно было вывести из входного документа. Поскольку LLaMa-3-70B показала высокую согласованность с человеческой оценкой, мы использовали ее в качестве оценщика корректности примеров и поручили ей найти все цитаты, подтверждающие предложения в аннотациях. Мы отфильтровали все примеры, в которых более 20 % предложений аннотации не имели подтверждения, и сопоставили найденные цитаты с предложениями документа, тем самым создав точные экстрактивные метки. Чтобы адаптировать задачу для настройки с несколькими примерами при ограниченном размере контекстного окна, мы сжали документы, удалив абзацы, не имеющие экстрактивных меток аннотации. Учитывая вариативность длины генерации текста LLM (Dubois, 2024), в качестве метрики для абстрактивных и экстрактивных аннотаций мы выбрали среднее значение полноты ROUGE-1 и ROUGE-2 и R-точности соответственно.

DaruCopy

При замене словаря LLM важно, чтобы модель научилась полностью использовать новые токены. Эмбеддинги входных токенов отвечают за передачу смысла текста, что можно оценить с помощью задач понимания естественного языка, таких как MMLU. Напротив, эмбеддинги выходных токенов используются для поиска ближайшего семантического значения к текущему состоянию нейронной сети, оно зависит от контекстной истории. Как следствие, в творческих задачах это состояние нестабильно, и LLM чаще генерирует более редкие токены. В то же время в задачах, где от LLM требуется повторное использование входного контекста, ожидается, что состояние сети попадет в семантические кластеры токенов, присутствующие во входной последовательности. Следуя этой логике, предложив LLM создать копию входного текста, мы можем оценить эффективность генерации токенов.

Мы использовали статьи «Википедии» разных жанров для формирования наборов данных задачи копирования на английском и русском языках. Они включают два варианта задачи копирования: по предложениям и абзацам. Первый вариант оценивает соответствие LLM алгоритму токенизации путем расчета отношения длины исходного текста к сгенерированному тексту в токенах. В варианте по абзацам мы оцениваем общую стабильность генерации текста, измеряя процент генераций, в которых отношение токенов наибольшей общей подпоследовательности (lcs) ко всем токенам абзаца превышает 99 % (1 % оставлен для ошибок пробелов). Отклонение от 99 % при высоких показателях копирования предложений указывает на то, что модель имеет тенденцию путать токены, что приводит к генерации недостоверного контекста в творческих задачах —это основная проблема надежности для практических приложений.

Параметры оценки на бенчмарке

При оценке качества на бенчмарке мы использовали следующие гиперпараметры: размер батча 8, длина последовательности 4096 токенов, пять примеров в few-shot-сценарии для оценки качества базовых моделей и zero-shot для оценки качества инструктивных моделей.

Постановка экспериментов

Мы произвели экспериментальную оценку разработанной методологии на двух моделях: Mistral-7B-v0.1 (Jiang, 2023) и LLaMa-3-8B (Dubey, 2024).

Продолженнное предобучение

Обучающий набор данных для токенизации и продолжения предварительного обучения включает документы из следующих областей: русскоязычная «Википедия», англоязычная «Википедия», «Хабрахабр», «Пикабу», художественная литература, новости, учебная литература.

Документы были дедуплицированы с использованием алгоритма Locality Sensitive Hashing Minhash. Мы удалили метаданные, ссылки, разделы комментариев и плохо отформатированные документы для улучшения распределения токенов и уменьшения количества грамматически неправильных примеров. Чтобы уменьшить семантический шум, мы ограничили словарь кириллицей и латиницей, а также удалили нестандартные символы, такие как эмодзи или логограммы (например, китайские иероглифы), используя нормализацию UTF-8.

Для обучения тексты отбирались с повышенными весами для «Википедии», учебной и научной литературы. Кроме того, для подачи текстов в языковую модель мы добились того, чтобы каждый пример начинался либо с нового документа, либо с нового абзаца.

Параметры токенизации. Мы обучили токенизаторы BPE и Unigram для двух вариантов: 32 000 и 128 000 токенов для Mistral-7B и LLaMa-3-8B соответственно. Для расширения токенизации мы увеличили исходные токеназаторы до 55 328 и 174 816 токенов соответственно, используя при этом

новые дообученные BPE-словари на русском языке. Поскольку токенайзер LLaMa-3-8B уже достаточно объемный, мы решили сделать Optimized-версию токенайзера. Для этого исходный токенайзер был сначала сокращен до 64000 токенов, а затем расширен за счет добавления новых 64 000 токенов с учетом возможных пересечений, что в итоге дало 114 504 токена.

Гиперпараметры. Во время продолженного предобучения мы использовали следующие гиперпараметры: общий размер батча — 256; длина последовательности — 1024; «затухание» весов (weight decay) — 0,1; планировщик — косинусный; количество шагов разогрева (warmup steps) — 100; количество эпох — 1.

Мы протестировали четыре различных скорости обучения: 2e-5, 5e-5, 1e-4, 2e-4 для каждой модели и варианта токенизации на 20 % от всего набора данных. На основе результатов тестирования мы выбрали скорость обучения 1e-4 для всех моделей Mistral-7B, и скорость обучения 2e-4 для моделей LLaMa-3-8B. Важно отметить, что эффективность адаптации модели показала значительную зависимость от скорости обучения, особенно для моделей на базе LLaMa-3-8B.

Исследование влияния: самокалибровка

В случаях полной замены словаря, когда модель учится перестраивать все новые эмбеддинги практически с нуля, процесс проецирования может иметь более низкую эффективность, поскольку разница между инструктивными и адаптированными к языку эмбеддингами может быть значительной. Логичным решением является синтез данных с применением оригинальной инструктивной LLM, а затем использование их для калибровки адаптированной версии. Для генерации примеров мы использовали промпты из набора данных инструкций Saiga и применили «жадное» декодирование, чтобы получить наиболее вероятный ответ с точки зрения инструктивной LLM. Затем мы попросили LLaMa-3-70B оценить качество синтезированных пар с точки зрения грамматики и релевантности по 5-балльной шкале. Все примеры, получившие оценку ниже 4, были отброшены, в результате чего осталось 13 531 пример для калибровки.

Поскольку примеры калибровки являются родными для внутренних семантических представлений LLM, существует риск, что вместо выравнивания модель может вернуться к исходной токенизации во время генерации, отдавая предпочтение меньшим, но более знакомым фрагментам токенизации. Чтобы предотвратить такой сценарий, мы воспользовались тем фактом, что все современные LLM предварительно обучены на статьях «Википедии» таким образом, что их представления эмбеддингов согласованы с концепциями «Википедии». При обращении к дообученной модели с задачей повторить статью из «Википедии» токен за токеном, мы заставляем модель вспомнить свою предварительно обученную память и таким образом распространить сигналы активации, соответствующие концепциям в статье, на эмбеддинги оптимальных токенов новой токенизации. Следуя этой логике, мы дополнили набор данных для калибровки 10 000 примерами задач копирования статей, полученных из части «Википедии», которая не пересекается с нашими наборами данных для предварительного обучения или тестирования.

Мы обнаружили, что следующие настройки LoRA-дообучения являются оптимальными для процедуры калибровки: ранг — 8; альфа — 1; скорость обучения — 2.5e-5; «затухание» весов — 0.1; целевые модули LoRa — первый и последний слои трансформера; модули LoRa для сохранения — lm_head, embed_tokens; максимальная длина последовательности — 8096; общий размер батча — 64; количество эпох — 1.

Исследование влияния: дополнительное дообучение инструкциям

В дополнение к экспериментам по самокалибровке мы решили проверить, как дополнительное обучение с использованием инструкций из высококачественного русскоязычного набора данных повлияет на конечную производительность. Для этого эксперимента мы выбрали набор данных Saiga, который считается лучшим открытым вариантом для русского языка. Мы также исследовали влияние добавления небольшого количества (2000) специальных инструкций в набор данных, целью которых является копирование большого текста из «Википедии».

Для дообучения моделей мы использовали адаптеры LoRA с рекомендованными Saiga настройками гиперпараметров: ранг — 32; альфа — 16; скорость обучения — 5e-5; «затухание» весов — 0,05; целевые модули LoRa — attention, mlp; модули LoRa для сохранения — lm_head; максимальная длина последовательности M 4096; общий размер батча — 128; количество эпох — 1.

РЕЗУЛЬТАТЫ

Оценка качества открытых LLM на Darumeru

Для установления базового уровня мы провели тестирование популярных инструктивных LLM (Таблица 1): Openchat 3.5, LLaMa-3 (instruct) (Dubey, 2024), Saiga (Gusev, 2023), Vikhr (Nikolich, 2024), Qwen-2, Mistral Nemo (Jiang, 2023). Как и ожидалось, самая большая модель, Mistral Nemo, показала наилучшую производительность. Однако модель Qwen-2 7B превосходит Mistral Nemo в выполнении задач MMLU, при этом отставая в тестах на устойчивость генерации текста DaruSum и DaruCopy. Vikhr-5.2 аналогично имеет такую же оценку по DaruMERA, как и Mistral Nemo. Учитывая законы

масштабирования LLM (Kaplan, 2020) и разрыв в производительности с современными LLM с менее чем 10 миллиардами параметров, эти наблюдения позволяют предположить, что некоторые части наборов данных MMLU и MERA попали в обучающие данные Vikhr-5.2 и Qwen-2 7B.

Адаптация словаря и продолженное предобучение

После получения первоначальных результатов тестирования мы сосредоточились на русской адаптации базовых моделей Mistral-7B и LLaMa-3-8B. Для оценки результатов языковой адаптации мы использовали обучение в контексте с несколькими примерами (few-shot), поскольку модели не привыкли напрямую интерпретировать инструкции.

Рисунок 2 показывает динамику оценки Darumeru на протяжении процесса продолженного предобучения. В случае Mistral-7B методы замены словаря, такие как BPE и Unigram, почти исчерпывают обучающие примеры, сходясь к оптимуму на последних 10 тысячах шагов обучения. В отличие от этого, LLaMa-3-8B более устойчива к методам адаптации словаря, поскольку все они имеют тенденцию сходиться в середине сессии обучения на 20–30 тысячах шагов. Поскольку полный размер набора данных составляет 96 ГБ, мы можем сделать вывод, что 40 ГБ текстов — это минимум, необходимый для хорошей производительности русскоязычных адаптированных эмбеддингов.

В Таблице 2 мы приводим подробные результаты лучших контрольных точек (чекпойнтов). Как и ожидалось, методы расширения словаря, такие как Extended и Optimized, имеют наименьшую сложность оптимизации, поскольку они показывают наивысшие показатели языковой адаптации. Для Mistral-7B все языковые адаптации значительно превосходят оригинальную базовую модель, однако разница между их эффективностью токенизации (символов на токен) и средней производительностью задач может считаться незначительной. Для LLaMa-3-8B только варианты Extended смогли достичь исходных показателей качества LLM, в основном отставая по задачам DaruMMLU. Наиболее эффективные с точки зрения токенизации варианты, BPE и Unigram, значительно отстают, проигрывая по DaruMERA и DaruSum. Мы предполагаем, что замена словаря в случае BPE и Unigram оказывает значительное влияние на понимание языка, и в случае их продолженного предобучения для правильного семантического выравнивания будет недостаточно только эмбеддингов, но потребуются дополнительные процедуры настройки.

Проекция предобученных эмбеддингов

Результаты отработки полной схемы проекции предобученных эмбеддингов (Learned Embedding Propagation, LEP) представлены в Таблице 3. Для каждого варианта построения адаптированного словаря (BPE, Unigram, Extended и Optimized) мы тестируем три метода: прямую замену эмбеддингов (Swap), коррекцию перекрывающихся токенов (Overlap) и проекцию преобразования словаря (Conversion). В качестве адаптированной модели эмбеддингов мы использовали лучшие чекпойнты продолженного предобучения (Таблица 2).

Для Mistral-7B и OpenChat 3.5 результаты проекции эмбеддингов имеют большую вариативность в зависимости от выбранного алгоритма токенизации для русского словаря. В случае BPE, который является тем же алгоритмом, что используется в оригинале, обученные эмбеддинги для нового словаря имеют наивысшее соответствие с инструктивным аналогом при прямой замене эмбеддингов. В случае более морфологически корректной токенизации для русского языка, Unigram, проекция с перекрытием показывает наивысшую среднюю производительность по набору задач. Однако, если посмотреть на групповые оценки, становится

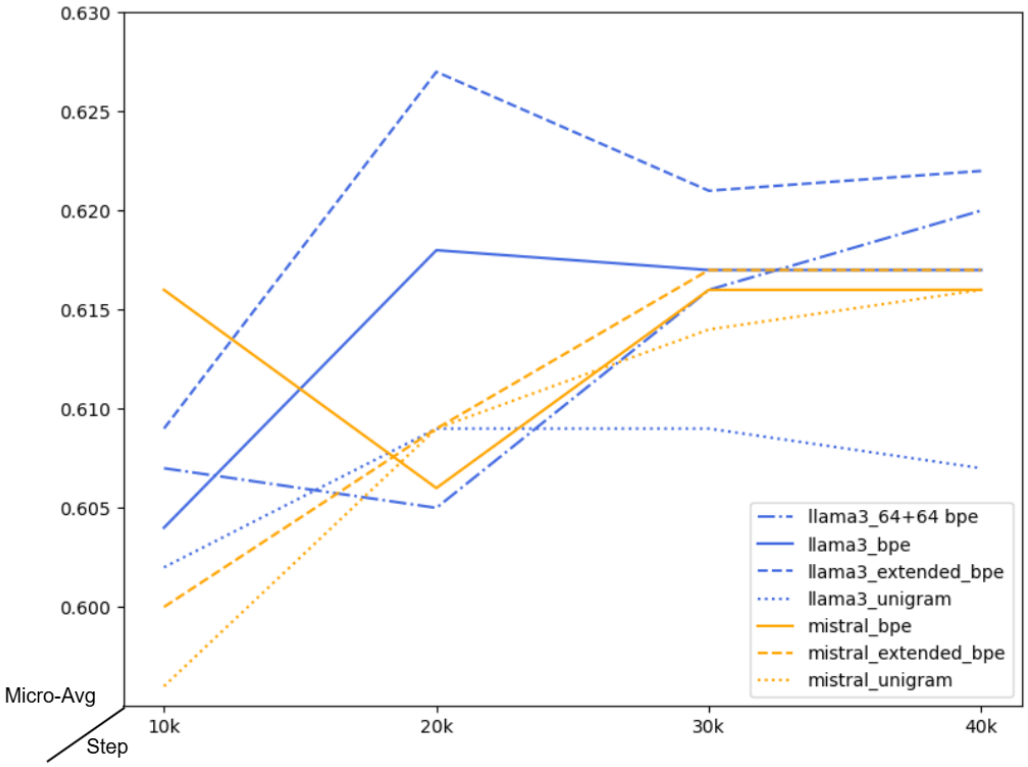
Таблица 1

Результаты zero-shot-оценки популярных инструктивных открытых моделей на наборе данных Darumeru

Модель	Micro-Avg	DaruMMLU	DaruMERA	DaruSum	DaruCopy (EN)	DaruCopy (RU)
Openchat 3.5 (Mistral-7B)	0,607	0,543	0,526	0,322	0,999	0,917
LLaMa-3-8B (Instruct)	**0,610**	0,571	0,510	0,322	**1,000**	0,972
Saiga (LLaMa-3-8B)	0,608	0,574	0,514	0,320	0,995	0,939
Vikhr-5.2 (Mistral-7B)	0,587	0,494	**0,573**	0,308	0,959	0,693
Qwen-2 7B	0,613	**0,624**	0,548	0,300	0,938	0,842
Mistral Nemo (12B)	**0,639**	0,592	**0,576**	0,320	0,998	0,924
Ours						
Openchat 3.5 + LEP-Extended + calibration (best)	0,632	0,541	0,563	0,321	**1,000**	**0,989**
LLaMa-3-8B (Instruct) + LEP-Extended + calibration (best)	0,618	0,565	0,521	**0,339**	**1,000**	0,984

Рисунок 2

Микроусреднение метрик качества на бенчмарке на протяжении обучения



очевидно, что проекция преобразования является лучшим вариантом, так как она лидирует в решении всех задач, кроме DaruCopy (Ru), где все варианты конвертации Unigram испытывают проблемы. Традиционное расширение словаря также склоняется к проекции преобразования и демонстрирует лучшую общую производительность среди вариантов, превосходящую оригинальный OpenChat 3.5.

Для LLaMa-3-8B проекция эмбеддингов работает более определенно. Для всех вариантов токенизации проекция преобразованием дает наилучшие результаты, однако, в отличие от Mistral-7B LEP, ни один из вариантов проекции эмбеддингов не достигает качества оригинального LLaMa-3-8B (instruct). Значительное снижение производительности наблюдается во всех группах задач, причем DaruCopy страдает больше всего. Более того, несмотря на то, что BPE является оригинальным алгоритмом токенизации, русский словарь, построенный на BPE, имеет наименьшую совместимость эмбеддингов с инструктивной версией, показывая наибольший разрыв в оценках. Несмотря на меньший размер словаря, оптимизированный вариант сохраняет тот же уровень качества, что и расширенный, и при сравнении проекций преобразованием первый лучше справляется с задачами DaruSum и DaruCopy.

Из наблюдений следует несколько выводов. Во-первых, алгоритм проекции преобразованием, вероятно, является наиболее эффективным решением для большинства сценариев проекции эмбеддингов, в некоторых случаях даже достаточным для восстановления производительности оригинальной инструктивной модели. Во-вторых, несмотря на то, что словарь токенизации Unigram можно считать морфологически правильным для русского языка, он уступает вариантам Extended и Optimization, поскольку требует полной замены словаря, что, учитывая нестабильную производительность BPE, приводит к наибольшему несоответствию между эмбеддингами и семантическим представлением внутренних слоев. Токены, удаленные в оптимизированном варианте, кажутся несущественными для способности решения русскоязычных задач, поскольку этот вариант превосходит токенизацию Extended, которая полностью сохраняет оригинальный словарь. Разрыв в производительности в LEP LLaMa-3-8B (instruct), вероятно, является следствием проприетарного набора данных для инструктивной настройки, который был достаточно большим для согласования семантики эмбеддингов с задачами следования инструкциям (Dubey, 2024). Другая гипотеза заключается в том, что оригинальная LLM прошла процедуру выравнивания с учетом человеческих предпочтений, направленную на блокировку генерации вредоносных ответов за счет необходимых ограничений рассуждений. В результате модель имеет тенденцию блокировать потенциально вредоносную семантику, исходящую из входных эмбеддингов, что, в свою очередь, подавляет способности понимания текста.

Таблица 2

Оценка качества на Darumeru во few-shot-постановке

Модель	Vocab	Symbols per token	Micro-Avg	DaruMMLU	DaruMERA	DaruSum	DaruCopy (EN)	DaruCopy (RU)
Mistral-7B	**original**	2,44	0,604	**0,545**	0,504	0,307	**1,000**	**1,000**
	BPE	**3,76**	0,616	0,528	0,537	**0,316**	0,995	0,984
	Unigram	**3,78**	0,614	0,516	**0,544**	0,311	0,995	0,960
	Extended	**3,77**	**0,617**	**0,538**	0,532	0,314	**1,000**	**0,995**
LLaMa-3-8B	**original**	2,89	**0,629**	**0,582**	0,547	0,326	0,980	0,982
	BPE	**4,40**	0,618	**0,561**	0,532	0,321	1,000	0,963
	Unigram	**4,35**	0,609	0,560	0,517	0,316	1,000	0,951
	Extended	3,78	**0,627**	0,560	0,550	0,325	0,980	0,983
	Optimized	3,40	**0,620**	0,552	0,536	0,323	0,981	0,989

Таблица 3

Результаты на бенчмарке Darumeru в zero-shot-постановке для схемы проекции предобученных эмбеддингов

Vocab	LEP method	Micro-Avg	DaruMMLU	DaruMERA	DaruSum	DaruCopy (En)	DaruCopy (Ru)
			OpenChat-3.5				
BPE	Swap	**0,587**	**0,528**	**0,526**	0,277	0,988	**0,829**
	Overlap	0,584	0,525	0,523	0,281	0,986	0,818
	Conversion	0,583	0,526	0,524	**0,284**	**0,993**	0,791
Unigram	Swap	0,556	**0,517**	0,517	0,282	0,985	0,614
	Overlap	**0,572**	0,514	0,534	0,297	0,981	**0,68**
	Conversion	0,565	**0,515**	**0,519**	**0,301**	**0,999**	0,651
Extended	Swap	**0,608**	0,535	0,540	0,298	0,999	**0,907**
	Overlap	0,607	0,535	0,539	**0,307**	0,999	0,898
	Conversion	**0,609**	**0,535**	**0,541**	**0,306**	0,999	**0,909**
			LLaMa-3-8B (instruct)				
BPE	Swap	0,565	0,544	0,486	0,317	0,999	0,729
	Overlap	0,569	0,546	0,489	0,314	0,999	0,753
	Conversion	**0,570**	0,546	0,490	0,318	0,999	0,754
Unigram	Swap	0,582	0,545	0,488	0,313	0,999	0,865
	Overlap	0,580	0,545	0,482	0,314	0,999	0,876
	Conversion	**0,584**	**0,545**	**0,488**	**0,315**	0,994	0,889
Extended	Swap	0,592	0,557	0,498	0,319	0,969	0,921
	Overlap	**0,597**	0,556	0,504	**0,321**	0,964	**0,936**
	Conversion	**0,597**	0,556	0,501	0,318	**0,994**	0,921
Optimized	Swap	0,594	0,554	0,499	**0,327**	0,970	0,928
	Overlap	0,586	0,553	0,495	0,323	0,925	0,925
	Conversion	**0,598**	**0,555**	**0,500**	**0,324**	**0,995**	**0,928**

Исследование влияния: самокалибровка

В экспериментах по самокалибровке мы сосредоточились на сокращении разрыва в производительности лучших LEP LLaMa-3-8B instruct моделей (см. Таблицу 4, самокалибровка). Как и ожидалось, производительность в решении задач DaruCopy значительно улучшилась, практически достигнув уровня идеальной надежности. DaruSum также показал улучшения, поскольку повышенные возможности цитирования способствуют созданию более точных и кратких резюме. Однако в других задачах значительного прогресса не наблюдалось, а в случае с наименее удачной адаптацией словаря, Unigram, результаты в бенчмарках значительно ухудшились.

Мы предполагаем, что данные для самокалибровки способствуют формированию узконаправленного стиля рассуждений, так как обучение на наиболее вероятных ответах смещает модель в сторону использования более общего словаря, который встречался в обучающих данных с наибольшей частотой. В результате понимание редких специализированных концепций, представленных в наборах данных MMLU и MERA, может ухудшаться из-за повышенной склонности модели к использованию более распространенного языка. Эту проблему можно смягчить с помощью более сложных процедур выбора примеров, таких как поиск с разбиением на множества (beam search) или генерация нескольких кандидатов с последующей постгенерационной сортировкой. Эти методы особенно эффективны при использовании более мощных, современных LLM, таких как GPT-4 или LLaMa-3-405B.

Исследование влияния: дополнительное дообучение инструкциям

Наши эксперименты по калибровке методом продолженного инструкционного дообучения, представленные в Таблице 4, показали, что дополнительно дообученные LEP-адаптированные модели достигают, а в некоторых случаях даже превосходят оригинальные модели. Добавление 2 000 инструкций для копирования длинных текстов в обучающий инструкционный набор данных положительно влияет практически во всех случаях.

Примеры

Мы также исследовали, как изменяются ответы моделей в зависимости от этапа: оригинальная модель, LEP, LEP + калибровка (Рисунок 3). На данном примере видно, что оригинальная модель восприняла вопрос совершенно неправильно. Модель LEP уже отвечает более корректно, но не учитывает, что это фразеологизм. Калиброванная модель дает наиболее правильный ответ среди всех трех версий, обращая внимание на истинное значение фразы.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Оценка качества на новом бенчмарке для русского языка

Результаты, представленные в Таблице 1, демонстрируют, что дообучение передовых открытых LLM на русскоязычных наборах инструктивных данных зачастую приводит к снижению производительности в понимании языка. Это явление первоначально было замечено в бенчмарках Ru-Arena-General[2] и Chatbot Arena[3], однако из-за их открытого формата вопросов было сложно отделить ошибки генерации от некорректных пользовательских запросов.

Бенчмарки с закрытыми вопросами, такие как MERA (Fenogenova, 2024), который использовался в качестве основы для Darumeru, не позволяют надежно выявлять ухудшение языковой обработки из-за возможности так называемого взлома бенчмарка. Под этим подразумевается дообучение модели на решениях бенчмарка или схожих данных, что рассматривается как форма «читерства» в контексте оценки LLM.

Как правило, разработчики LLM не прибегают к подобным практикам и, напротив, прилагают дополнительные усилия, чтобы исключить любые возможные данные бенчмарков из общего обучающего набора модели. Однако выявление утечек данных, связанных с бенчмарками, является трудоемкой задачей, поскольку требует проверки обучающих данных не только на точные совпадения, но и на возможные перефразирования, включая переводы примеров на другие языки.

Наш бенчмарк Darumeru снимает ограничения формата закрытых вопросов с помощью новых введенных задач для обобщения текста (DaruSum) и диагностики токенизации (DaruCopy). DaruSum требует двух ключевых элементов решения задач: правильного анализа текста и хороших навыков написания текста. Любое снижение производительности в этом подмножестве бенчмарка указывает на проблемы с пониманием текста или генерацией текста. DaruCopy различает эти два аспекта, но оценивает только последний, упрощая задачу до явной трансляции исходного контекста без какого-либо анализа или перефразирования. Следовательно, более низкие оценки DaruCopy указывают на конфликт в логике LLM, поскольку модель не следует простейшей инструкции задачи копирования текста. Эти два подмножества бенчмарка Darumeru показывают, что LLaMa-3-8B является более надежным выбором для задач обработки русского языка, чем Saiga или Vikhr-5.2, несмотря на их специализацию на русском языке, что контрастирует с результатами бенчмарка MERA (Fenogenova, 2024). В то время как результаты MERA для Saiga находятся в пре-

[2] https://huggingface.co/spaces/Vikhrmodels/arenahardlb

[3] https://lmarena.ai/

Таблица 4

Результаты оценки качества на бенчмарке Darumeru для оценки влияния схем калибровки после LEP

Модель	Fine-tuning data	Micro-Avg	DaruMMLU	DaruMERA	DaruSum	DaruCopy (EN)	DaruCopy (RU)
			OpenChat 3.5				
Original model	-	0,607	**0,543**	0,526	0,322	0,999	0,917
	saiga d7	0,611	0,540	**0,528**	**0,325**	0,999	0,945
	+copy task	**0,615**	0,541	0,524	0,324	**1,000**	**0,995**
Unigram	-	0,565	0,515	0,519	0,301	0,999	0,651
	saiga d7	0,599	**0,532**	0,556	0,316	0,999	0,754
	+copy task	**0,630**	0,530	**0,559**	**0,321**	**1,000**	**0,999**
Extended	-	0,609	0,535	0,541	0,306	0,999	0,909
	saiga d7	0,616	**0,543**	**0,566**	0,319	0,999	0,845
	+copy task	**0,632**	0,541	0,563	**0,321**	**1,000**	**0,989**
			LLaMa-3-8B instruct				
Original model	-	0,610	0,571	0,510	0,322	1,000	0,972
	saiga d7	0,615	**0,576**	0,512	0,329	1,000	0,983
	+copy task	**0,616**	0,575	**0,513**	**0,332**	**1,000**	**0,995**
Extended	-	0,597	0,556	0,501	0,318	0,994	0,921
	self-calibration	0,606	0,552	0,512	0,321	1,000	0,958
	saiga d7	0,614	**0,568**	0,519	0,338	0,995	0,961
	+copy task	**0,618**	0,565	**0,521**	**0,339**	**1,000**	**0,984**
Optimized	-	0,598	0,555	0,500	0,324	0,995	0,928
	self-calibration	0,601	0,550	0,501	0,325	1,000	0,95
	saiga d7	0,611	**0,555**	0,515	0,336	**1,000**	0,971
	+copy task	**0,617**	**0,555**	**0,522**	**0,339**	**1,000**	**0,989**

делах стандартного отклонения, результаты Vikhr-5.2 явно указывают на случай взлома бенчмарка.

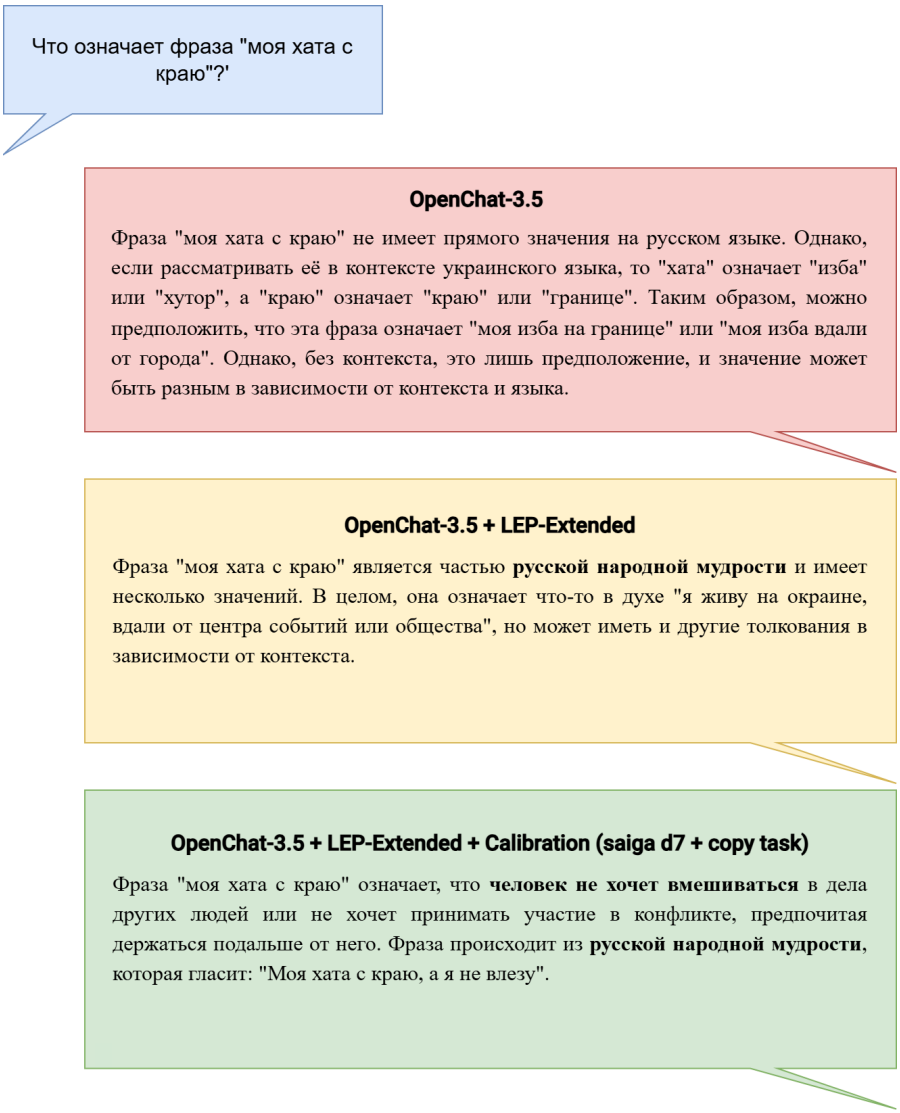
Стратегия адаптации к языку

В ходе разработки нашей схемы адаптации LLM к русскому языку мы сделали несколько проектных решений, которые были исследованы в предыдущих работах. Прежде всего, мы предположили, что знание токенизации и способность использовать новые токены хранятся во входных эмбеддингах и слоях lm_head (выходные эмбеддинги) модели. Несколько работ (Cui, 2023; Tikhomirov, 2023; Nikolich 2024; Nguyen, 2024) продемонстрировали, что языковая адаптация только этого подмножества слоев недостаточна для правильного понимания языка, и, таким образом, последующая инструктивная настройка таких моделей приводит к неоптимальным результатам. В то же время было показано (Tikhomirov, 2024), что нет существенной разницы между языковой адаптацией всех слоев и двухэтапным подходом, когда процесс обучения эмбеддингов дополняется последующим обучением других слоев. Результаты, представленные в Таблице 2, подтверждают это утверждение, поскольку первый этап двухэтапного подхода оказывается достаточно эффективным для существенного улучшения понимания русского языка моделью Mistral-7B. Однако, оценки LLaMa-3-8B после адаптации предполагают, что необходимость обучения внутренних слоев диктуется исходным уровнем владения русским языком языковой модели, который эффективно фиксируются подмножеством DaruMERA нашего бенчмарка. Результаты процедуры проекции предобученных эмбеддингов (Learned Embedding Propagation) (Таблица 3) также отражают это наблюдение, поскольку Mistral-7B показала наивысшую эффективность передачи языковых знаний.

В наших экспериментах по калибровке мы исследовали вопрос о том, может ли несоответствие слоев быть смягчено инструктивным дообучением. Инструктивное дообучение (инструктивная настройка, instruction fine-tuning) на целевом языке часто улучшает использование токенов и повышает понимание языка (Gusev, 2023; Wei, 2023; Nikolich, 2024). Аналогичная тенденция проиллюстрирована в Таблице 4. Обучая оригинальные неадаптированные инструктивные версии LLM на наборе данных Saiga (Gusev,

Рисунок 3

Пример генерации с моделью OpenChat-3.5 и ее адаптированными вариантами



2023), мы улучшили способности решения русскоязычных задач, что повысило оценки в бенчмарках. Применяя ту же процедуру к нашим моделям LEP (saiga d7), мы сохраняем положительный эффект и даже получаем результаты, превышающие результаты исходных моделей. Недостатком инструктивного дообучения на русскоязычных инструктивных наборах данных является то, что мы неизбежно нарушаем исходные знания инструктивной версии (Tejaswi, 2024). Мы попытались решить эту проблему, обучаясь на ответах, сгенерированных исходной LLM (самокалибровка), а не используя оригинальные ответы из набора данных Saiga. Однако для LLaMa-3-8B instruct мы не увидели заметного улучшения в каких-либо возможностях моделей, кроме повышения качества использования токенизации (DaruCopy). Этот результат, вероятно, связан с недостатком качества генерации наших синтезированных примеров самокалибровки, которые во время нашей ручной проверки показали значительно более простую логику русского языка и ограниченный словарный запас. Учитывая, что Saiga является ярким примером синтеза ответов из GPT-4 (Taori, 2024), мы предполагаем, что, используя более продвинутые методы выборки и лучшие протоколы оценки качества примеров, мы можем собрать референсный набор данных с аналогичными характеристиками без использования других наборов данных или моделей третьих сторон.

Ограничения ИССЛЕДОВАНИЯ

Несмотря на широкую применимость нашего метода, данное исследование имеет несколько ограничений. Во-первых, метод требует, чтобы были доступны не только инструктивные версии LLM, но и их базовые версии, что не всегда возможно. Во-вторых, в случае языков, использующих иероглифы, инициализация после замены токенизатора может быть довольно слабой из-за отсутствия общих токенов, и неизвестно, насколько адаптация эмбеддингов может помочь в этом. Еще один важный момент заключается в том, что процедура передачи знаний была ориенти-

рована на сохранение исходных знаний целевой модели, поэтому возможный объем новых знаний может быть недостаточным. Однако, поскольку методология эффективно адаптирует модель к языку, всегда возможно провести дополнительный этап продолженнного предобучения для приобретения новых знаний.

ЗАКЛЮЧЕНИЕ

В этой статье мы предложили метод проекции предобученных эмбеддингов (Learned Embedding Propagation, LEP) — улучшенный подход к языковой адаптации больших языковых моделей (LLM), который оказывает минимальное негативное влияние на исходные знания языковых моделей и при этом позволяет передавать знания языковой адаптации непосредственно любой инструктивной версии, включая обученные на проприетарных данных. Сосредоточившись на экономической эффективности нашего метода, мы разработали три специальных подхода для проецирования эмбеддингов: прямую замену эмбеддингов, коррекцию перекрывающихся токенов и проекцию преобразования словаря. Для упрощения процесса разработки адаптации мы представили Darumeru — бенчмарк для оценки качества моделей во время их обучения, который фокусируется на надежности генерации текста. Анализируя производительность популярных инструктивных LLM и четырех вариантов адаптации словаря, мы разработали рецепт для наиболее экономически эффективной процедуры. Используя этот рецепт и предложенные методы LEP, мы создали адаптированные для целевого языка варианты современных инструктивных языковых моделей с числом параметров менее девяти миллиардов — Openchat-3.5 и LLaMa-3-8B (Instruct). Результаты оценки показали, что варианты LEP с проекцией преобразования словаря воспроизводят уровни производительности оригинальной инструктивной версии, а в случае с OpenChat-3.5 даже превосходят их, сохраняя при этом все преимущества улучшенной вычислительной эффективности. Чтобы устранить оставшиеся пробелы в решении задач, мы провели эксперименты с методами самокалибровки и дополнительного дообучения по инструкциям, которые обеспечили дальнейшее улучшение понимания языка и позволили достичь новых высот в результатах на предложенном бенчмарке. Полученные результаты открывают новые перспективы для языковой адаптации LLM, обеспечивая экономически эффективное использование любых инструктивных моделей независимо от открытости их инструктивных данных со всеми достоинствами оригинальной версии.

Все наши модели, бенчмарк и фреймворк имеют открытый исходный код и доступны под оригинальными лицензиями моделей.

БЛАГОДАРНОСТИ

Работа Михаила Тихомирова была поддержана некоммерческим фондом развития науки и образования «Интеллект». Работа Даниила Чернышева была поддержана некоммерческим фондом развития науки и образования «Интеллект». Работа выполнялась с использованием суперкомпьютера «МГУ-270» МГУ имени М.В. Ломоносова.

ВКЛАД АВТОРОВ

Михаил Тихомиров: концептуализация; сбор данных; методология; администрирование проекта; программное обеспечение; написание – первоначальный вариант.

даниил Чернышев: концептуализация; управление данными; формальный анализ; методология; валидация; визуализация; написание – первоначальный вариант; рецензирование и редактирование.

ЛИТЕРАТУРА

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., & Hashimoto, T. B. (2023). *Stanford alpaca: An instruction-following llama* model. https://github.com/tatsu-lab/stanford_alpaca

Li, H., Koto, F., Wu, M., Aji, A. F., & Baldwin, T. (2023). Bactrian-x: Multilingual replicable instruction-following models with low-rank adaptation. arXiv preprint arXiv:2305.15011. https://doi.org/10.48550/arXiv.2305.15011

Wei, X., Wei, H., Lin, H., Li, T., Zhang, P., Ren, X., Li, M., Wan, Y., Cao, Z., Xie, B., Hu, T., Li, S., Hui, B., Yu, B., Liu, D., Yang, B., & Xie, J. (2023). Polylm: An open source polyglot large language model. arXiv:2307.06018. https://doi.org/10.48550/arXiv.2307.06018

Gusev, I. (2023). *rulm: A toolkit for training neural language models*. https://github.com/IlyaGusev/rulm.

Kuulmets, H. A., Purason, T., Luhtaru, A., & Fishel, M. (2024, June). Teaching Llama a new language through cross-lingual knowledge transfer. In *Findings of the Association for Computational Linguistics: NAACL 2024* (pp. 3309-3325). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.210

Zhu, W., Lv, Y., Dong, Q., Yuan, F., Xu, J., Huang, S., Kong, L., & Li, L. (2023). *Extrapolating large language models to non-english by aligning languages.* arXiv:2308.04948. https://doi.org/10.48550/arXiv.2308.04948

Ranaldi, L., Pucci, G., & Freitas, A. (2023). *Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations*. arXiv:2308.14186. https://doi.org/10.48550/arXiv.2308.14186

Li, C., Wang, S., Zhang, J., & Zong, C. (2024, June). Improving in-context learning of multilingual generative language models with cross-lingual alignment. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (vol. 1: Long Papers, pp. 8051-8069). Association for Computational Linguistics. 10.18653/v1/2024.naacl-long.445

Chai, L., Yang, J., Sun, T., Guo, H., Liu, J., Wang, B., Liang, X., Bai, J., Li, T., Peng, Q., & Li, Z. (2024). *xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning*. arXiv preprint arXiv:2401.07037. https://doi.org/10.48550/arXiv.2401.07037

Husain, J. A., Dabre, R., Kumar, A., Puduppully, R., & Kunchukuttan, A. (2024). *RomanSetu: Efficiently unlocking multilingual capabilities of Large Language Models models via Romanization.* arXiv:2401.14280. https://doi.org/10.48550/arXiv.2401.14280

Lakew, S. M., Erofeeva, A., Negri, M., Federico, M., & Turchi, M. (2018). Transfer learning in multilingual neural machine translation with dynamic vocabulary. *Proceedings of the 15th International Conference on Spoken Language Translation* (pp. 54-61). International Conference on Spoken Language Translation. https://doi.org/10.48550/arXiv.1811.01137

Kuratov, Y., & Arkhipov, M. (2019). Adaptation of deep bidirectional multilingual transformers for Russian language. *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii* (pp. 333-339). Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. https://doi.org/10.48550/arXiv.1905.07213

Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., & Gurevych, I. (2021, August). How good is your tokenizer? On the monolingual performance of multilingual language models. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (vol. 1: Long Papers, pp. 3118-3135). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.243

Yang, Z., Xu, Z., Cui, Y., Wang, B., Lin, M., Wu, D., & Chen, Z. (2022, October). CINO: A Chinese minority pre-trained Language Model. In *Proceedings of the 29th International Conference on Computational Linguistics* (pp. 3937-3949). International Committee on Computational Linguistics. https://doi.org/10.48550/arXiv.2202.13558

Vries, W., & Nissim, M. (2021, August). As good as new. How to successfully recycle English GPT-2 to make models for other languages. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 836-846). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.74

Tikhomirov, M., & Chernyshev, D. (2023). Impact of tokenization on LLaMa Russian adaptation. *2023 Ivannikov Ispras Open Conference* (pp. 163-168). IEEE. http://dx.doi.org/10.1109/ISPRAS60948.2023.10508177

Tikhomirov, M., & Chernyshev, D. (2024). Improving Large Language Model Russian adaptation with preliminary vocabulary optimization. *Lobachevskii Journal of Mathematics, 45*, 3211-3219. 10.1134/S1995080224604120.

Cui, Y., Yang, Z., & Yao, X. (2023). *Efficient and effective text encoding for Chinese llama and alpaca.* arXiv:2304.08177. https://doi.org/10.48550/arXiv.2304.08177

Nguyen, X. P., Zhang, W., Li, X., Aljunied, M., Tan, Q., Cheng, L., Chen, G., Deng, Y., Yang, S., Liu, C., Zhang, H., & Bing, L. (2023). *SeaLLMs-Large Language Models for Southeast Asia*. arXiv:2312.00738. https://doi.org/10.48550/arXiv.2312.00738

Nikolich, A., Korolev, K., & Shelmanov, A. (2024). *Vikhr*: *The family of open-source instruction-tuned Large Language Models for Russian.* arXiv preprint arXiv:2405.13929. https://doi.org/10.48550/arXiv.2405.13929

Artetxe, M., Ruder, S., & Yogatama, D. (2020). On the cross-lingual transferability of monolingual representations. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics* (pp. 4623–4637). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.421

Chen, Y., Marchisio, K., Raileanu, R., Adelani, D., Saito Stenetorp, P. L. E., Riedel, S., & Artetxe, M. (2023). Improving language plasticity via pretraining with active forgetting. *Advances in Neural Information Processing Systems*, 36, 31543-31557. https://doi.org/10.48550/arXiv.2307.01163

Tejaswi, A., Gupta, N., & Choi, E. (2024). *Exploring design choices for building language-specific LLMs*. arXiv preprint arXiv:2406.14670. https://doi.org/10.48550/arXiv.2406.14670

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., & Levy, O. (2024). Lima: Less is more for alignment. *Advances in Neural Information Processing Systems, 36*. https://doi.org/10.48550/arXiv.2305.11206

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., ... & Ganapathy, R. (2024). *The Llama 3 Herd of Models*. arXiv:2407.21783. https://doi.org/10.48550/arXiv.2407.21783

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling laws for neural language models*. arXiv:2001.08361. https://doi.org/10.48550/arXiv.2001.08361

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., & Farhadi, A. Editing models with task arithmetic. *The Eleventh International Conference on Learning Representations*. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2212.04089

Gusev, I. (2020). Dataset for automatic summarization of Russian news. In *Artificial Intelligence and Natural Language: 9th Conference* (Proceedings 9, pp. 122-134). Springer International Publishing. https://doi.org/10.1007/978-3-030-59082-6_9

Dubois, Y., Galambosi, B., Liang, P., & Hashimoto, T. B. (2024). *Length-controlled alpacaeval: A simple way to debias automatic evaluators*. arXiv:2404.04475. https://doi.org/10.48550/arXiv.2404.04475

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A,, Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems, 35*, 27730-27744. https://doi.org/10.48550/arXiv.2203.02155

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., Roziere, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023a). *Llama: Open and efficient foundation language models*. arXiv:2302.13971. https://doi.org/10.48550/arXiv.2302.13971

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., … & Scialom, T. (2023b). *Llama 2: Open foundation and fine-tuned chat models*. arXiv:2307.09288. https://doi.org/10.48550/arXiv.2307.09288

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. D. L., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023). *Mistral 7B*. arXiv:2310.06825. https://doi.org/10.48550/arXiv.2310.06825

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., … & McGrew, B. (2023). *Gpt-4 technical report*. arXiv:2303.08774. https://doi.org/10.48550/arXiv.2303.08774

Fenogenova A. et al. (2024). *Mera: A comprehensive LLM evaluation in Russian*. arXiv:2401.04531. https://doi.org/10.48550/arXiv.2401.04531

Mikhailov, V., Shamardina, T., Ryabinin, M., Pestova, A., Smurov, I., & Artemova, E. (2022). RuCoLA: Russian corpus of linguistic acceptability. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 5207-5227). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.348