

# Области применения больших языковых моделей для цифровых образовательных платформ

А. Г. Леонов<sup>1</sup>, Н. С. Мартынов<sup>2</sup>, К. А. Машенко<sup>3</sup>, М. С. Паремузов<sup>4</sup>,  
К. К. Пчелин<sup>5</sup>, А. В. Шляхов<sup>6</sup>

<sup>1</sup>НИЦ «Курчатовский институт» - НИИСИ, Москва, Россия, МГУ им. М. В. Ломоносова, Москва, Россия, МПГУ, Москва, Россия, Государственный университет управления, Москва, Россия, dr.l@vip.niisi.ru;

<sup>2</sup>НИЦ «Курчатовский институт» - НИИСИ, Москва, Россия, nikolai.martynov@math.msu.ru;

<sup>3</sup>НИЦ «Курчатовский институт» - НИИСИ, Москва, Россия, МГУ им. М. В. Ломоносова, Москва, Россия, kirill.mashchenko@niisi.ru;

<sup>4</sup>НИЦ «Курчатовский институт» - НИИСИ, Москва, Россия, matveyparem@gmail.com;

<sup>5</sup>НИЦ «Курчатовский институт» - НИИСИ, Москва, Россия, k.pchelin@gmail.com;

<sup>6</sup>НИЦ «Курчатовский институт» - НИИСИ, Москва, Россия, shlyakhov@vip.niisi.ru;

**Аннотация.** В работе рассматриваются возможности применения больших языковых моделей (LLM) для повышения эффективности взаимодействия студентов и преподавателей в рамках цифровых образовательных платформ, включая апробацию этих моделей в условиях реального учебного процесса с использованием ЦОП Мирера. Анализируются современные state-of-the-art решения – YandexGPT, Mistral, Qwen, LLaMA и их модификации, а также особенности их архитектуры, производительности и возможностей адаптации под образовательные задачи. Показано, что корректная настройка параметров моделей позволяет эффективно использовать их для автоматизации рутинных операций, персонализации обучения и расширения инструментов преподавателя.

**Ключевые слова:** цифровая образовательная платформа, Мирера, большие языковые модели, искусственный интеллект, нейросетевые технологии, AI-агенты, LLM, персонализация обучения, автоматизация образовательных процессов, безопасность данных

## 1. Введение

Развитие компьютерных технологий и значительное увеличение вычислительных ресурсов в последние годы вывели на новый уровень возможности применения систем искусственного интеллекта (ИИ), в частности – больших языковых моделей (LLM), которые ранее рассматривались преимущественно как теоретическая концепция. Современные нейросетевые архитектуры всё активнее интегрируются в различные сферы деятельности, трансформируя традиционные процессы и создавая предпосылки для комплексной автоматизации.

Образовательная сфера стала одним из ключевых направлений практического внедрения ИИ. Сегодня наблюдается рост числа цифровых образовательных платформ и проектов, использующих LLM для поддержки учащихся и преподавателей, оптимизация процесса обучения и реформирование распределения нагрузки педагога для создания заданий, требующих высокого уровня

экспертизы [1]. Эти модели способны адаптироваться к индивидуальным особенностям обучающихся, предоставлять персонализированные рекомендации, обеспечивать интерактивность взаимодействия, что, в свою очередь, способствует более глубокому усвоению материала [2]. В дополнение LLM можно использовать для анализа и переработки авторских образовательных материалов, что помогает в подготовке новых учебных курсов и тестовых заданий, повышающих объективность оценки уровня знаний учащихся.

Таким образом, внедрение больших языковых моделей в образовательный процесс позволит решить широкий спектр задач: от персонализации обучения и автоматизации рутинных операций до интеллектуальной поддержки педагога при разработке курсов и оценивании успеваемости [3]. Последнее способствует повышению эффективности образовательной среды, в том числе: снижению административной нагрузки на преподавателей, развитию коммуникации в рамках учебного

процесса, обеспечению доступа студентов к ИИ-инструментам, способствующим более продуктивному и комфорtnому обучению и т.д.

Одновременно с цифровой трансформацией растёт внимание и к вопросам безопасности и этических аспектов применения ИИ [4]. Современные языковые модели проектируются с учётом минимизации рисков, связанных с предвзятостью, манипуляцией или генерацией нежелательного контента, что повышает доверенность к ним и дает надежду на успешность в использовании в образовательных целях.

Таким образом, интеграция больших языковых моделей в цифровое образование не только отражает ключевые тенденции технологического развития, но и открывает новые возможности для трансформации традиционных подходов к обучению в сторону большей гибкости, адаптивности и эффективности. Исследование и совершенствование этих технологий становится приоритетной задачей как для академической науки, так и для прикладных разработок.

Помимо аналитического рассмотрения, модели были апробированы в условиях реального учебного процесса с использованием ЦОП Мирера, что позволило сопоставить теоретические выводы с практическими результатами.

## 2. Обзор моделей и их архитектур

### 2.1 Критерии выбора модели для образовательной платформы

Выбор подходящей большой языковой модели (LLM) для внедрения в образовательную платформу является сложной многокритериальной задачей, требующей одновременного учёта большого набора факторов:

- производительности,
- требований к вычислительным ресурсам,
- качества генерации текстового контента,
- адаптируемости к различным предметным областям,
- аспектов безопасности,
- совокупной стоимости владения.

От выбора модели существенно зависят эффективность применения искусственного интеллекта в учебных процессах и устойчивость его работы в реальном образовательном процессе.

Одним из первичных факторов является вычислительная сложность модели, проявляющаяся в числе параметров, размере контекстного окна и требованиях к аппаратным

ресурсам. Многие образовательные организации не располагают серверами с графическими ускорителями, позволяющими запускать крупные модели. В таких условиях целесообразно выбирать либо облегчённые модели (7-8В параметров) в квантованных версиях [5] (с пониженной численной точностью хранения параметров), либо решения с внешним доступом через API, которые полностью нивелируют ограничения по собственным вычислительным ресурсам.

Локальное развёртывание обеспечивает значительно больший контроль над моделью, её интерфейсом и данными, что особенно важно для соблюдения требований законодательства о персональных данных пользователей. Вместе с тем такой подход потребует закупки (и/или) аренды вычислительных ресурсов необходимой производительности и наличие высоких технических компетенций специалистов поддержки для корректной настройки и надежной эксплуатации. Облачное (API) развёртывание снимает проблему собственной инфраструктуры, однако сопряжено с рисками передачи данных третьим сторонам и зависимостью системы от внешнего провайдера (стабильность подключения, тарифы, политика использования). Поэтому в учреждениях в качестве практического компромисса зачастую выступают гибридные решения, в которых чувствительные операции выполняются локально, а ресурсоёмкие – в облаке.

Архитектурные особенности моделей и состав обучающих наборов данных, на которых они обучались, в конечном итоге определяют пригодность для выполнения конкретных задач. Так, ориентированные на программный код модификации LLM (например, семейства Qwen и Mistral) демонстрируют более высокое качество в курсах по программированию и инженерных дисциплинах, где требуется генерация и верификация кода. В то же время с задачей проверки текстов на русском языке эффективнее справляются модели, обученные на широких русскоязычных корпусах данных.

Образовательные проекты существенно варьируются по требованиям к формам и методике учебного взаимодействия: одни курсы для заданий используют краткие регламентированные ответы, другие, напротив, базируются на аналитических рассуждениях. Поэтому важным критерием для выбора архитектуры становится способность модели быстро подстраиваться под разные сценарии использования (системные инструкции, параметры инференса, шаблоны подсказок) и наличие гибких интерфейсов системной интеграции.

Критерии выбора модели для образовательной платформы можно условно разделить на технические (аппаратные требования, масштабируемость, архитектура) и прикладные (соответствие предметной области, безопасность, стоимость, гибкость внедрения).

В совокупности рассмотренные факторы формируют методологическую основу для дальнейшего анализа конкретных решений. Оптимальная модель должна учитывать баланс между вычислительной сложностью и доступностью ресурсов, обеспечивать адаптируемость к специфике предметной области, гарантировать защиту данных и оставаться экономически оправданной для использования. При этом разные архитектуры LLM реализуют эти критерии по-разному: от ориентации на максимальную производительность крупных моделей до акцента на компактность и энергоэффективность, от специализации на генерации кода до усиленных возможностей для сложных рассуждений и текстового анализа.

Для обеспечения формирования полноценного представления о современных возможностях и ограничениях LLM в образовательной среде целесообразным является рассмотрение конкретных архитектур, применяемых в современной практике.

## 2.2 YandexGPT-5-Lite-8B-Instruct

YandexGPT-5-Lite [6] содержит около 8 млрд параметров и представляет собой плотную (Dense) инструкционную модель с контекстным окном до 32 тыс. токенов, что позволяет обрабатывать крупные образовательные материалы в рамках одной сессии. Используется эффективный токенизатор, оптимизированный под особенности русского языка и учитывающий его морфологические и синтаксические особенности, что обеспечивает корректное разбиение текста и высокое качество генерации.

Ключевое преимущество модели – её глубокая адаптация к русскоязычной среде и возможность локального запуска на видеокартах с 12-24 GB VRAM благодаря квантизации в форматах GGUF/GPTQ. Такая оптимизация снижает аппаратные требования и уменьшает задержки отклика. Отсутствие механизма Mixture of Experts (MoE) [7] упрощает инфраструктурную интеграцию и повышает стабильность работы.

Сочетание высокой поддержки русского языка и длинного контекста делает модель подходящей для создания образовательных агентов, персонализации ответов, автоматизированного формирования учебных материалов и предоставления детализированной

обратной связи. Локальное развёртывание обеспечивает контроль над безопасностью и приватностью данных, что критически важно для образовательных платформ.

Таким образом, YandexGPT-5-Lite предоставляет сбалансированное решение для образовательных проектов, где требуются качественная обработка русского языка, работа с большими контекстами и гибкость локального развёртывания. Благодаря оптимизациям и настройкам модель отлично подходит для задач интерактивного обучения и поддержки преподавателей в автоматизации рутинных операций.

## 2.3 Mistral-Small-3.2-24B-Instruct

Mistral-Small [8] оснащена плотной архитектурой, содержащей 24 млрд параметров, а также имеет расширенное контекстное окно до 128 тыс. токенов. Модель поддерживает мультимодальные входные данные и функцию вызова внешних процедур (function calling), что расширяет сферу применения за пределы простой текстовой генерации. Токенизация основана на Byte Pair Encoding (BPE) и ориентирована на сценарии с поддержкой нескольких языков, что обеспечивает высокую степень универсальности.

Хотя MoE не используется, успешно были реализованы усовершенствованные методы инструкционной донастройки, повышающие точность следования командам и уменьшающие повторяемость ответов. Оптимизация квантизации (Q4\_K\_M) позволяет запускать модель на GPU с 24 GB VRAM.

Большой контекст и мультимодальность способствуют созданию многофункциональных образовательных агентов, способных анализировать тексты и изображения и обеспечивать высокий уровень интерактивности. Устойчивость модели к генерации недопустимого и нежелательного контента повышает надёжность её применения в образовательной среде.

Mistral-Small-3.2-24B является высокопроизводительной универсальной моделью для реализации продвинутых образовательных решений с мультизадачностью, особенно в сложных AI-системах, требующих работы с разнородными данными и длинным контекстом.

## 2.4 Qwen2.5-Coder-7B/14B

Семейство больших языковых моделей с плотной архитектурой Qwen2.5-Coder [9] (7B и 14B параметров) ориентировано на задачи генерации и анализа программного кода, а также связанного с ним естественного языка. Архитектура трансформера с инструкционным

обучением обеспечивает точное следование запросам пользователя, а разнообразие размеров модели позволяет подобрать оптимальный вариант в зависимости от вычислительных ресурсов и требований задач. Таким образом достигается необходимый для поставленной задачи баланс между производительностью и качеством.

В модели используется продвинутый специализированный токенизатор для корректной обработки кода и технических текстов, учитывающий точный синтаксис языков программирования и комментариев. Поддержка квантизации до 4 бит обеспечивает возможность эффективного запуска модели на видеокартах с 12-24 GB VRAM без существенного ухудшения качества.

Отличительной чертой модели Qwen2.5-Coder является её специализация на языках программирования. Она способна генерировать код «с нуля», выполнять его рефакторинг, проводить верификацию написанных программ, анализировать ошибки, разъяснять сложные конструкции и вести диалог с пользователем, оставаясь в технической предметной области. В сочетании с инструкционным обучением эти функции обеспечивают её высокую применимость в образовательных проектах в области ИТ и инженерии, предоставляя возможности для незамедлительной помощи студентам в решении программных заданий, а также помочь преподавателю при подготовке материалов и проверке домашних работ.

Qwen2.5-Coder-7B/14B – это оптимальное решение для интеграции в цифровые образовательные платформы, где требуется качественная поддержка курсов программирования и автоматизация учебных процессов, связанных с программным кодом. Высокая точность и глубина обработки при разумных требованиях к вычислительным ресурсам делают эти модели привлекательным выбором для образовательных учреждений, стремящихся к внедрению ИИ для повышения качества обучения.

## 2.5 Llama 3.1 8B Instruct

Llama 3.1 8B [10] имеет плотную архитектуру, 8 млрд параметров и динамическое контекстное окно до 32 тыс. токенов. Токенизация основана на алгоритме SentencePiece, что обеспечивает эффективность как на английском, так и на других языках [11]. Модель поддерживает квантизацию (Q4/Q5/Q6) для оптимизированной работы на GPU с 12–24 GB VRAM.

Сильной стороной модели является развитая экосистема интерфейсов и

инструментов (llama.cpp, Ollama, vLLM), упрощающих масштабирование и кастомизацию. Инструкционная версия улучшает следование заданным запросам и снижает вероятность ошибок в сложных рассуждениях.

Модель является универсальной, поэтому она применима для широкого круга образовательных агентов: от помощи в написании сочинений и анализе текстов до генерации вопросов и интерактивного взаимодействия. Активно поддерживаемое сообщество и широкие возможности дообучения расширяют потенциал модели.

Llama 3.1 8B Instruct – мощное универсальное решение с отличной поддержкой сообщества и инструментов. Оно отлично подходит для образовательных платформ с международным уклоном, где важна гибкость, масштабируемость и широкая применимость.

## 2.6 SeedCoder

Seed-Coder [12] – семейство LLM (~8B параметров), разработанное с акцентом на открытость, прозрачность и высокую производительность при работе с анализом и генерацией программного кода. Доступны варианты базовая (base), инструкционная (instruct) и модель для многозадачного логического рассуждения (reasoning), оптимизированные для различных сценариев использования.

Плотная архитектура трансформера обучена на крупном корпусе кода из GitHub и других источников по «модельно-центричному» подходу, когда сами LLM участвуют в отборе, оценке и фильтрации обучающих данных. Это снижает человеческое вмешательство, минимизирует предвзятость и повышает качество корпуса. Контекст модели поддерживает работу с длинными последовательностями (реализуются варианты с удлинённым контекстом), а токенизатор адаптирован для точной обработки кода и связанного с ним естественного языка, что улучшает генерацию и понимание сложных синтаксических конструкций.

Отличительной чертой Seed-Coder является использование продвинутых методов дообучения и оптимизации, таких как supervised fine-tuning и direct preference optimization (DPO) [13] для instruct версии, а также reinforcement learning с Long-Chain-of-Thought (LongCoT) [14] для модели reasoning. Эти методы позволяют значительно повысить качество следования инструкциям, улучшить способность к многозадачным логическим рассуждениям и решению сложных кодовых задач, что особенно важно для образовательных сценариев,

тестирования и отладки программ.

Модель эффективна для создания интерактивных AI-агентов и ассистентов в программировании: генерация и исправление кода, поиск ошибок, объяснение алгоритмов. Поддержка квантизации позволяет запуск модели на 12-24 GB VRAM без значимой потери качества.

Seed-Coder является современным и масштабируемым решением для образовательных платформ с акцентом на программирование и инженерные задачи. Благодаря инновационному подходу к подготовке данных и продвинутым механизмам обучения, модель обеспечивает высокую производительность и качество генерации, открытость разработки, что повышает привлекательность её интеграции в системы AI-помощников, автоматизированных проверок заданий и интерактивного обучения программированию.

### 3. Особенности моделей в задачах образования

#### 3.1. Образовательные AI-агенты

Современные образовательные агенты на основе LLM часто используют модульную архитектуру, включающую интеграцию с внешними источниками данных и специализированными инструментами (tools). Одним из эффективных подходов является Retrieval-Augmented Generation (RAG) – схема, при которой агент, получив запрос, сначала извлекает релевантную информацию из базы знаний или контент-хранилища, а затем формирует ответ с учётом этих данных [15, 16].

Подобная архитектура позволяет:

- работать с актуальными учебными материалами без необходимости повторного масштабного обучения модели;
- адаптировать систему под специфику конкретного образовательного курса;
- обеспечивать более высокую точность и релевантность ответов;
- реализовывать сложные сценарии, требующие анализа внешних структурированных или неструктурных данных.

В таких конфигурациях LLM выступает как центральный управляемый компонент, координирующий взаимодействие между инструментами (поиск, генерация тестов, анализ кода) и базами знаний.

#### 3.2. Пример реализации

В рамках цифровой образовательной платформы Мирера разработан набор специализированных сервисов,

иллюстрирующих практическое применение LLM в образовательных задачах. Эти функции демонстрируют, как комбинация языковых моделей, интеграции с базами данных и генеративных алгоритмов позволяет автоматизировать широкий спектр учебных процессов.

Примеры образовательных инструментов:

- Цифровой помощник студента для задач программирования. Модель получает доступ к данным о текущем решении (код, состояние, сообщения об ошибках, текст задания и эталонного решения) и формирует краткую обучающую рекомендацию, указывая, на какие аспекты решения следует обратить внимание для исправления ошибок;

- Помощник преподавателя. На основе тех же данных более крупная модель формирует углублённый анализ затруднений студента и рекомендации преподавателю по выстраиванию дальнейших шагов обучения;

- Генерация вариантов ответов для тестов [17, 18]. По сформулированному заданию и правильному ответу агент создаёт несколько содержательных неверных вариантов, применимых в качестве отвлекающих ответов в тестовых заданиях;

- Автоматическая генерация тестов по материалам контекста. После анализа изученного материала агент генерирует задания, соответствующие тематике пройденного курса;

- Генерация задач по программированию. По аналогии с тестами, но с формированием полного условия задачи для практического программирования;

- Построение эталонного решения. На основе текстового описания задачи модель разрабатывает оптимальный алгоритм, впоследствии проверяемый посредством тестовых прогонов (валидаторов);

- Генерация тестовых данных. Имея эталонное решение и текст задачи, агент создаёт набор входных данных для проверки альтернативных решений;

- Создание аналогичных тестовых работ. По исходному набору тестовых заданий формируются вариации с тем же количеством заданий и темами, но с изменённым содержанием;

- Декомпозиция задач. Сложная задача разбивается на последовательность более простых подзадач, которые помогают поэтапно освоить необходимые навыки перед переходом к основной.

- Расширение формулировок задач. На базе краткого описания агент генерирует полное, структурированное и методически корректное условие задачи.

Таким образом, сочетание генеративных алгоритмов с доступом к актуальным учебным данным может существенно повысить эффективность обучения студентов и работы преподавателей.

### 3.3 Выбор модели для сервисов с ИИ

В процессе разработки интеллектуальных агентов для цифровой образовательной платформы Mirera возникла задача подбора оптимальных языковых моделей для различных типов заданий. Проведённые сравнительные испытания показали, что YandexGPT демонстрирует наилучшее понимание русского языка и высокую точность при создании и проверке текстовых заданий, однако его эффективность при решении задач, связанных с программированием, оказалась ограниченной. Для кодо-ориентированных сценариев предпочтение было отдано модели Mistral 3.2 Small в квантизированной конфигурации, что позволило разместить две модели на одном графическом ускорителе с объёмом памяти 32 ГБ и достигнуть высокой скорости инференса без существенной потери качества. Модель Qwen показала достойные результаты, однако продемонстрировала меньшую точность выполнения инструкций по сравнению с Mistral. Модель семейства LLaMA по суммарным показателям уступила всем вышеупомянутым решениям и поэтому не была выбрана для внедрения.

С учётом полученных результатов была принята гибридная схема распределения ролей между моделями: YandexGPT задействуется в задачах генерации формулировок на русском языке, нормализации и рецензирования ответов студентов; Mistral 3.2 Small обслуживает контур программирования (постановка и разбор задач,

объяснение ошибок, рефакторинг, генерация тестов и подсказок). Для сохранения целостности пользовательского опыта введён маршрутизатор запросов и унифицированные промпты-шаблоны с явной спецификацией роли и формата ответа, что позволило избавиться от непредсказуемого поведения моделей и стабилизировать ответы.

## 4. Заключение

В ходе проведённого исследования были проанализированы критерии выбора и особенности архитектур современных больших языковых моделей, применимых в образовательных цифровых платформах. Показано, что компетентный подбор модели с учётом аппаратных ограничений, предметной специализации и требований к безопасности позволяет существенно повысить эффективность взаимодействия между студентами и преподавателями.

Рассмотренные архитектуры YandexGPT, Mistral, Qwen и LLaMA продемонстрировали различный потенциал в зависимости от задач и условий внедрения. Установлено, что сочетание локальных и облачных решений при оптимальной квантизации обеспечивает баланс между производительностью и приватностью данных. Результаты работы могут служить практическими рекомендациями для образовательных организаций, планирующих интеграцию LLM в учебный процесс с учётом доступных ресурсов и целей обучения.

Работа выполнена в рамках темы государственного задания НИЦ «Курчатовский институт» - НИИСИ по теме № FNEF-2024-0001 (1023032100070-3-1.2.1).

# Applications of large language models for digital educational platforms

A. G. Leonov, N. S. Martynov, K. A. Mashchenko, M. S. Paremuзов,  
K. K. Pchelin, A. V. Shlyakhov

**Abstract.** This paper explores the potential of large language models (LLMs) to enhance interactions between students and educators within digital educational platforms. It analyzes modern state-of-the-art solutions – such as YandexGPT, Mistral, Qwen, LLaMA, and their variants – along with their architectural features, performance, and adaptability for educational tasks. The study demonstrates that proper model parameter tuning enables their effective use in automating routine tasks, personalizing learning, and expanding instructors' toolsets.

**Keywords:** digital educational platform, Mirera, large language models, artificial intelligence, neural network technologies, AI agents, LLM, personalized learning, automation of educational processes, data security

## Литература

1. Chu, Z., Wang, S., Xie, J., Zhu, T., Yan, Y., Ye, J., Zhong, A., Hu, X., Liang, J., Yu, P.S. and Wen, Q., 2025. Llm agents for education: Advances and applications. arXiv preprint arXiv:2503.11733.
2. Silva, P. and Costa, E., 2025. Assessing large language models for automated feedback generation in learning programming problem solving. arXiv preprint arXiv:2503.14630.
3. Yousef, M., Mohamed, K., Medhat, W. et al. BeGrading: large language models for enhanced feedback in programming education. *Neural Comput & Applic* 37, 1027–1040 (2025). <https://doi.org/10.1007/s00521-024-10449-y>.
4. Chen, K., Zhou, X., Lin, Y., Feng, S., Shen, L. and Wu, P., 2025. A Survey on Privacy Risks and Protection in Large Language Models. arXiv preprint arXiv:2505.01976.
5. Xunyu Zhu, Jian Li, Yong Liu, Can Ma, Weiping Wang; A Survey on Model Compression for Large Language Models. *Transactions of the Association for Computational Linguistics* 2024; 12 1556–1577. doi: [https://doi.org/10.1162/tacl\\_a\\_00704](https://doi.org/10.1162/tacl_a_00704).
6. LLM YandexGPT-5: [сайт]. – URL <https://huggingface.co/yandex/YandexGPT-5-Lite-8B-pretrain> (дата обращения 11.08.25)
7. Lo, K.M., Huang, Z., Qiu, Z., Wang, Z. and Fu, J., 2024. A closer look into mixture-of-experts in large language models. arXiv preprint arXiv:2406.18219.
8. LLM Mistral: [сайт]. – URL <https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506> (дата обращения 11.08.25)
9. LLM Qwen2.5-Coder: [сайт]. – URL <https://huggingface.co/Qwen/Qwen2.5-Coder-7B-Instruct> (дата обращения 11.08.25)
10. LLM Llama-3.1: [сайт]. – URL <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct> (дата обращения 11.08.25)
11. Kudo, T. & Richardson, J. (2018), SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing., in Eduardo Blanco & Wei Lu, ed., 'EMNLP (Demonstration)', Association for Computational Linguistics, , pp. 66-71.
12. LLM Seed-Coder: [сайт]. – URL <https://huggingface.co/ByteDance-Seed/Seed-Coder-8B-Instruct> (дата обращения 11.08.25)
13. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S. and Finn, C., 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36, pp.53728-53741.
14. Chen, Q., Qin, L., Liu, J., Peng, D., Guan, J., Wang, P., Hu, M., Zhou, Y., Gao, T. and Che, W., 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. arXiv preprint arXiv:2503.09567.
15. Thüs D, Malone S and Brünken R (2024) Exploring generative AI in higher education: a RAG system to enhance student engagement with scientific literature. *Front. Psychol.* 15:1474892. doi: 10.3389/fpsyg.2024.1474892
16. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.S. and Li, Q., 2024, August. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 6491-6501).
17. Byun, G. and Choi, J.D., 2025. D-GEN: Automatic Distractor Generation and Evaluation for Reliable Assessment of Generative Model. arXiv preprint arXiv:2504.13439.
18. Bitew, S.K., Deleu, J., Develder, C. and Demeester, T., 2023, September. Distractor generation for multiple-choice questions with predictive prompting and large language models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 48-63). Cham: Springer Nature Switzerland.