КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2025 Т. 17 № 5 С. 871–888
DOI: 10.20537/2076-7633-2025-17-5-871-888

Ки&М

МОДЕЛИ В ФИЗИКЕ И ТЕХНОЛОГИИ

УДК: 004.94

Применение больших языковых моделей для интеллектуального поиска и извлечения информации в корпоративных информационных системах

И. В. Антонов[a], Ю. В. Бруттан[b]

Псковский государственный университет,
Россия, 180000, г. Псков, пл. Ленина, д. 2

E-mail: [a] igorant63@yandex.ru, [b] bruttan@mail.ru

Получено 04.06.2025, после доработки — 04.08.2025.
Принято к публикации 03.09.2025.

В данной статье исследуется эффективность применения технологии Retrieval-Augmented Generation (RAG) в сочетании с различными большими языковыми моделями (LLM) для поиска документов и получения информации в корпоративных информационных системах. Рассматриваются варианты использования LLM в корпоративных системах, архитектура RAG, характерные проблемы интеграции LLM в RAG-систему. Предлагается архитектура системы, включающая в себя векторный энкодер текстов и LLM. Энкодер используется для создания векторной базы данных, индексирующей библиотеку корпоративных документов. Запрос, передаваемый LLM, дополняется релевантным ему контекстом из библиотеки корпоративных документов, извлекаемым с использованием векторной базы данных и библиотеки FAISS. Большая языковая модель принимает запрос пользователя и формирует ответ на основе переданных в контексте запроса данных. Рассматриваются общая структура и алгоритм функционирования предлагаемого решения, реализующего архитектуру RAG. Обосновывается выбор LLM для исследования и проводится анализ результативности использования популярных LLM (ChatGPT, GigaChat, YandexGPT, Llama, Mistral, Qwen и др.) в качестве компонента для генерации ответов. На основе тестового набора вопросов методом экспертных оценок оцениваются точность, полнота, грамотность и лаконичность ответов, предоставляемых рассматриваемыми моделями. Анализируются характеристики отдельных моделей, полученные в результате исследования. Приводится информация о средней скорости отклика моделей. Отмечается существенное влияние объема доступной памяти графического адаптера на производительность локальных LLM. На основе интегрального показателя качества формируется общий рейтинг LLM. Полученные результаты подтверждают эффективность предложенной архитектуры RAG для поиска документов и получения информации в корпоративных информационных системах. Были определены возможные направления дальнейших исследований в этой области: дополнение контекста, передаваемого LLM, и переход к архитектуре на базе LLM-агентов. В заключении представлены рекомендации по выбору оптимальной конфигурации RAG и LLM для построения решений, обеспечивающих быстрый и точный доступ к информации в рамках корпоративных информационных систем.

Ключевые слова: искусственный интеллект, информационные системы, семантический поиск, обработка естественного языка, векторизация документов, RAG, LLM

Статья подготовлена в рамках реализации проекта «Знание в цифре» программы развития Псковского государственного университета программы стратегического и академического лидерства «Приоритет 2030» (2024 г.).

COMPUTER RESEARCH AND MODELING
2025 VOL. 17 NO. 5 P. 871–888
DOI: 10.20537/2076-7633-2025-17-5-871-888

Ки&М

MODELS IN PHYSICS AND TECHNOLOGY

UDC: 004.94

Using RAG technology and large language models to search for documents and obtain information in corporate information systems

I. V. Antonov[a], Iu. V. Bruttan[b]

Pskov State University,
2 Lenin sq., Pskov, 180000, Russia

E-mail: [a] igorant63@yandex.ru, [b] bruttan@mail.ru

Received 04.06.2025, after completion — 04.08.2025.
Accepted for publication 03.09.2025.

This paper investigates the effectiveness of Retrieval-Augmented Generation (RAG) combined with various Large Language Models (LLMs) for document retrieval and information access in corporate information systems. We survey typical use-cases of LLMs in enterprise environments, outline the RAG architecture, and discuss the major challenges that arise when integrating LLMs into a RAG pipeline. A system architecture is proposed that couples a text-vector encoder with an LLM. The encoder builds a vector database that indexes a library of corporate documents. For every user query, relevant contextual fragments are retrieved from this library via the FAISS engine and appended to the prompt given to the LLM. The LLM then generates an answer grounded in the supplied context. The overall structure and workflow of the proposed RAG solution are described in detail. To justify the choice of the generative component, we benchmark a set of widely used LLMs — ChatGPT, GigaChat, YandexGPT, Llama, Mistral, Qwen, and others — when employed as the answer-generation module. Using an expert-annotated test set of queries, we evaluate the accuracy, completeness, linguistic quality, and conciseness of the responses. Model-specific characteristics and average response latencies are analysed; the study highlights the significant influence of available GPU memory on the throughput of local LLM deployments. An overall ranking of the models is derived from an aggregated quality metric. The results confirm that the proposed RAG architecture provides efficient document retrieval and information delivery in corporate environments. Future research directions include richer context augmentation techniques and a transition toward agent-based LLM architectures. The paper concludes with practical recommendations on selecting an optimal RAG–LLM configuration to ensure fast and precise access to enterprise knowledge assets.

Keywords: artificial intelligence, information systems, semantic search, natural language processing, document vectorization, RAG, LLM

Citation: *Computer Research and Modeling*, 2025, vol. 17, no. 5, pp. 871–888 (Russian).

We acknowledge Pskov State University for financial support of the present study (2024).

1. Введение

Современные большие языковые модели (LLM) демонстрируют высокую эффективность в обработке естественного языка и генерации связного текста. Однако их способность предоставлять точную и актуальную информацию в специализированных областях, таких как корпоративные данные и научные исследования, ограничена. Одной из ключевых проблем является проблема «галлюцинаций», которая связана с генерацией ответов системами на базе LLM недостоверных или несоответствующих источникам данных.

Для повышения достоверности информации активно развивается подход Retrieval-Augmented Generation (RAG), который сочетает генеративные возможности LLM с механизмами поиска и выборки релевантных документов. Этот метод особенно перспективен для реализации интеллектуальных систем доступа к корпоративной и научной информации, так как позволяет моделям опираться на достоверные источники знаний и предоставлять более точные ответы на запросы пользователей.

Несмотря на активное внедрение технологии RAG, остается открытым вопрос, насколько различные LLM подходят для работы в этой архитектуре при поиске и обработке корпоративных и научных данных. Сравнительный анализ моделей с учетом таких параметров, как точность выборки, качество генерации, соответствие исходным документам и вычислительная эффективность, необходим для выбора оптимального решения.

Настоящее исследование направлено на оценку степени соответствия современных LLM задаче реализации функций интеллектуального ассистента в составе корпоративной информационной системы. В качестве конкретного примера реализации такого ассистента рассмотрена задача создания навигатора по ресурсам университетской библиотеки. Полученные результаты позволят определить сильные и слабые стороны различных моделей в этом контексте и сформулировать рекомендации по их применению в системах корпоративного поиска информации.

2. Использование LLM в корпоративных информационных системах

Использование больших языковых моделей [Weng, 2023] позволяет значительно повысить эффективность работы с корпоративными данными — документами, отчетами, перепиской, автоматизируя поиск информации, генерацию отчетов и анализ бизнес-процессов. На основе использования LLM реализуются быстрый поиск информации в корпоративных базах знаний, инструкциях, нормативных документах, автоматическая категоризация и аннотирование документов, распознавание и извлечение ключевых фактов из большого массива текстов.

Интеграция LLM в корпоративные системы часто выполняется на основе создания виртуальных ассистентов и чат-ботов [Овсянников, Сарычев, 2024], обеспечивающих эффективный доступ к корпоративной информации.

Универсальные системы искусственного интеллекта на базе LLM, представленные на рынке и используемые в таких задачах, обучаются на общедоступных данных из открытых источников. Это обстоятельство обусловливает существование принципиальных ограничений на использование LLM в качестве источника информации, имеющей корпоративную специфику и не предназначенной для открытого распространения.

Одним из возможных путей преодоления этих ограничений является дополнительное обучение открытых моделей на собственных данных. Дополнительное обучение больших языковых моделей на корпоративных данных представляется привлекательным решением, поскольку оно позволяет адаптировать модель к специфике компании, терминологии и внутренним процессам. Однако на практике этот процесс сопровождается рядом сложностей, делающих его труднореализуемым или нецелесообразным в ряде случаев.

- *Высокие вычислительные затраты*. Дообучение LLM требует значительных вычислительных ресурсов, особенно если модель содержит миллиарды параметров. Полноценное дообучение на специализированных данных требует GPU- или TPU-кластеров, что приводит к высоким затратам на оборудование и электроэнергию. Для крупных моделей, таких как GPT-4 или Llama 70B, даже частичное дообучение может занимать недели или месяцы.

- *Проблемы конфиденциальности и безопасности*. Корпоративные данные могут содержать конфиденциальную, коммерческую и персональную информацию, что накладывает ограничения на их обработку. Передача данных в облачные сервисы для дообучения может нарушать требования законодательства, в частности Федеральный закон о персональных данных (ФЗ-152). Сам процесс дообучения может привести к утечкам информации, если модель случайно воспроизведет приватные данные в ответах.

- *Ограниченный объем корпоративных данных*. Дообучение LLM требует больших объемов качественных данных, но в реальных условиях корпоративные данные часто разрознены между разными системами (CRM, ERP, базы знаний). Внутренние документы могут быть неструктурированными, содержать ошибки, устаревшую или неполную информацию. Объем доступных текстов может быть недостаточен для значимого влияния на модель. Требуется предобработка и аннотация данных, что требует дополнительных усилий со стороны специалистов.

Альтернативным путем решения проблемы интеграции LLM и корпоративных источников данных является применение технологии Retrieval-Augmented Generation (RAG) [Федоров, Поляков, 2023].

3. Технология Retrieval-Augmented Generation (RAG)

Технология Retrieval-Augmented Generation (RAG), впервые предложенная в работе [Lewis et al., 2020], является архитектурой, предназначенной для повышения качества и достоверности ответов больших языковых моделей. Основная цель RAG — преодолеть принципиальные ограничения стандартных LLM, такие как статичность их знаний и склонность к генерации фактически недостоверной информации («галлюцинациям»), путем динамического предоставления модели релевантной информации из внешних авторитетных источников в момент генерации ответа [Cheng et al., 2025]. Это особенно актуально для корпоративных приложений, где требуется оперировать актуальными и специфическими данными.

Технология RAG основана на трех ключевых компонентах [Gao et al., 2023].

1. Модуль поиска (Retriever): извлекает релевантные документы или фрагменты текста из внешней базы знаний, индексированных данных или библиотек документов.

2. Модуль генерации (Generator): получает найденные документы в качестве дополнительного контекста и генерирует ответ, учитывая предоставленную информацию.

3. База знаний: внешний источник информации, откуда поисковый модуль извлекает релевантные данные.

Процесс работы RAG можно разделить на несколько этапов.

1. Формирование запроса: пользователь вводит текстовый запрос, который может быть автоматически уточнен или преобразован.

2. Извлечение релевантных данных: система выполняет поиск по базе данных, выбирая наиболее релевантные документы.

3. Формирование расширенного контекста: найденные документы объединяются и передаются в LLM в качестве дополнительных входных данных.

4. Генерация ответа: модель формирует осмысленный ответ, используя расширенный контент в качестве основной информации.

Основой для RAG служит корпус документов или данных (статьи, отчеты, веб-страницы, записи в базах данных и т. д.), релевантных для решаемой задачи. Корпус проходит следующие этапы обработки:

1) чанкинг (Chunking) — разделение исходных документов на более мелкие семантически связанные фрагменты текста для улучшения гранулярности поиска и соблюдения ограничений на размер контекстного окна LLM;

2) векторизацию (Embedding) — преобразование каждого текстового чанка в числовой вектор (эмбеддинг) с использованием специализированных моделей, учитывающих семантическое содержание текста;

3) индексацию (Indexing) — сохранение полученных векторов и связанных с ними метаданных в оптимизированной для поиска по сходству векторной базе данных (Vector Database), такой как FAISS, ChromaDB или Elasticsearch с поддержкой векторного поиска [Зупарова, 2024].

Модуль поиска во время обработки запроса пользователя отвечает за извлечение релевантной информации. Запрос пользователя сначала преобразуется в вектор с помощью той же embedding-модели, что использовалась для индексации базы знаний. Затем вектор запроса сравнивается с векторами чанков в векторной базе данных с использованием метрики семантической близости, чаще всего косинусного сходства (Cosine Similarity). Система отбирает Top-K (где K — настраиваемый параметр) наиболее релевантных чанков, которые будут использованы на следующем этапе.

В качестве модуля генерации в RAG используется большая языковая модель, отвечающая за синтез итогового ответа. Ключевым аспектом является то, что LLM получает на вход не исходный запрос в чистом виде, а специально сформированный расширенный промпт (augmented prompt). Этот промпт обычно включает системные инструкции, K релевантных чанков, извлеченных ретривером, и оригинальный запрос пользователя. Модель генерирует ответ, основываясь на предоставленном контексте и своих внутренних знаниях. Именно выбор и характеристики этого компонента — конкретной LLM — оказывают решающее влияние на итоговую производительность RAG-системы по таким параметрам, как скорость, точность, полнота и стиль ответа. Сравнительный анализ эффективности различных LLM в роли генератора в RAG-архитектуре и является основным предметом исследования в данной работе.

Таким образом, технология RAG представляет собой синтез механизмов информационного поиска и возможностей генеративных языковых моделей. Динамическое извлечение релевантного контекста позволяет LLM генерировать более точные, фактически обоснованные и актуальные ответы по сравнению со стандартным подходом. Однако, как будет показано далее, практическая реализация RAG требует тщательного подбора и оценки всех компонентов системы, в особенности используемой большой языковой модели.

Модуль поиска может быть реализован на основе традиционного полнотекстового поиска по ключевым словам либо на основе векторного поиска, использующего векторные представления текстов (эмбеддинги) для поиска семантически близких текстов. Второй вариант обеспечивает необходимую гибкость интерпретации запросов пользователя и является основным используемым практически способом реализации модуля поиска в RAG-системах.

В качестве модуля генерации может использоваться локальная LLM, либо облачный онлайн-сервис, обрабатывающий запросы на основе API, предназначенного для удаленного доступа к LLM.

Корпоративные базы знаний обычно представлены наборами неструктурированных или структурированных текстовых документов. При использовании RAG-технологии общепринятым подходом является сегментация документов, входящих в базу знаний, и их индексирование.

Использование Retrieval-Augmented Generation дает несколько ключевых преимуществ по сравнению с автономным использованием LLM.

- Актуальность информации: RAG позволяет работать с постоянно обновляющимися данными, что критично для научных исследований [Lewis et al., 2020].
- Снижение количества «галлюцинаций»: модель использует реальные источники, уменьшая вероятность выдачи недостоверных фактов.
- Лучшее понимание узкоспециализированных предметных областей: подключение к тематическим библиотекам повышает качество обработки специализированной информации [Бородулин, 2024].
- Компактность модели: вместо хранения всего объема знаний внутри параметров модели RAG позволяет использовать внешние хранилища, снижая требования к объему данных, необходимых для обучения.

4. Характерные проблемы при интеграции LLM в RAG-систему

Интеграция больших языковых моделей (LLM) в архитектуру Retrieval-Augmented Generation (RAG) открывает новые возможности для интеллектуального поиска и генерации текстов на основе внешних знаний [Изосимова и др., 2024]. Однако этот процесс сопровождается рядом сложностей, связанных как с особенностями работы LLM, так и с ограничениями механизмов поиска и обработки информации [Береснев, 2024].

Одной из ключевых проблем является качество выборки релевантных документов. Алгоритмы поиска могут извлекать либо недостаточно релевантные, либо избыточные данные, что снижает точность ответов модели. Кроме того, не все LLM способны эффективно учитывать предоставленный контекст, корректно интерпретировать данные и различать достоверные и ошибочные сведения.

Важным вызовом остается проблема «галлюцинаций» — генерация LLM информации, не содержащейся в исходных документах. Даже при наличии релевантных источников модель может выдавать ответы, основанные на ее внутренних вероятностных представлениях, а не на реальных данных. Это особенно критично в области научных исследований, где требуются высокая точность и соответствие первоисточникам.

Дополнительные сложности связаны с оптимизацией вычислительных ресурсов. Интеграция LLM в RAG-системы требует значительных вычислительных мощностей, особенно при обработке больших массивов данных. Выбор между локальным развертыванием моделей и использованием облачных решений также влияет на производительность, задержки в выдаче ответов и стоимость эксплуатации.

Наконец, важной проблемой является адаптация моделей под специфические технические и научные предметные области [Жигалов, Болодурина, 2024]. Большинство LLM обучены на обширных корпусах данных общего назначения и могут демонстрировать недостаточную точность при работе с узкоспециализированной терминологией. Это требует дообучения или адаптации моделей для корректной обработки научных текстов.

Успешная интеграция LLM в RAG для работы с научными материалами требует решения множества технических и методологических вопросов, включая выбор оптимальной архитектуры поиска, повышение достоверности ответов и снижение вычислительных затрат [Олейник и др., 2024]. Настоящее исследование направлено на анализ этих проблем и выявление возможных подходов для их решения.

5. Цель исследования и постановка задачи

Цель данного исследования — провести сравнительный анализ возможности использования различных больших языковых моделей (LLM) в контексте их интеграции в архитектуру Retrieval-Augmented Generation (RAG) для предоставления точной и достоверной информации на основе содержания корпоративной библиотеки документов.

Для достижения этой цели необходимо:

- оценить, насколько различные LLM эффективно обрабатывают запрашиваемую информацию при использовании RAG;
- исследовать влияние параметров моделей на качество генерации ответов;
- определить, какие LLM обеспечивают наибольшую точность и достоверность извлеченной информации, минимизируя эффект «галлюцинаций»;
- проанализировать производительность моделей в условиях реальной нагрузки, включая скорость обработки запросов и потребление вычислительных ресурсов;
- разработать рекомендации по выбору и настройке LLM для интеграции в системы интеллектуального поиска научных и технических данных.

Для решения указанных задач был сформирован список исследуемых LLM. Конкретные модели были включены в список на основании достаточно высоких позиций в рейтингах, связанных с обработкой текстов, и при условии наличия в моделях поддержки русского языка, либо в оригинальной версии, либо в результате дополнительного обучения оригинальной модели на русскоязычных текстах.

В публикуемых исследованиях характеристик LLM, как правило, используют автоматизированное тестирование с использованием стандартных датасетов и формализованных критериев оценки качества полученных результатов. Рассматриваемая задача требует оценки адекватности извлечения полезной информации из произвольного контекста. Формализованные метрики не позволяют в полной мере учесть все семантические аспекты в отношениях между запросом, контекстом и итоговым ответом. В силу изложенных обстоятельств, для оценки полученных результатов применялся метод экспертных оценок.

Оценка полученных результатов выполнялась группой из трех специалистов лаборатории LAMBDA-PSKOV Псковского государственного университета, каждый из которых имеет опыт в области обработки естественного языка и предметной области «Веб-программирование». Каждому ответу выставлялись оценки по четырем критериям (точность, полнота, лаконичность, грамотность) по шкале от 1 до 10, где 1 — полное несоответствие, а 10 — идеальное соответствие.

В процессе оценки ответа экспертами учитывался предоставленный LLM контекст запроса. Итоговая оценка по каждому критерию для каждой модели рассчитывалась как среднее арифметическое оценок всех экспертов. Хотя данный метод не исключает возможной субъективности оценок, он позволяет более гибко оценить семантическую адекватность ответов, что затруднительно при использовании формальных метрик.

Для проведения тестирования был сформирован банк вопросов, включающий в себя 40 вопросов по веб-программированию — одной из тем, представленных в использованном фонде оцифрованных текстов. Содержательная информация для ответов на часть этих вопросов отсутствует в используемом фонде документов. Данное обстоятельство позволяет определить, насколько модели способны выявлять отсутствие запрошенной у них информации в предоставленном им контексте.

6. Формирование векторной базы данных

В качестве источника документов для формирования базы знаний был использован цифровой архив выпускных квалификационных работ Псковского государственного университета. Архив включает 11 000 файлов в формате pdf, содержащих тексты выпускных работ различных направлений подготовки и учебно-методические материалы.

Для индексации этой базы документов была использована библиотека FAISS [Johnson et al., 2019], поддерживающая формирование векторных представлений текстов, сохранение их в базе данных и эффективный семантический поиск фрагментов текстов, релевантных запросам пользователей.

Для получения векторных представлений документов и запросов использовалась предобученная модель multilingual-e5-small из семейства E5, доступная на Hugging Face Hub (https://huggingface.co/intfloat/multilingual-e5-small). Данная модель содержит 6 слоев трансформера, 66M обучаемых весов и формирует 384-разрядные векторные представления текстов. Модель характеризуется как компактная и производительная, русский язык входит в число поддерживаемых языков.

В процессе индексации исходные документы были разделены на блоки (чанки). Размер блока был установлен равным 512 токенам, что обусловлено характеристиками выбранной для векторизации модели multilingual-e5-small. Максимальная длина входной последовательности, которую способна обработать данная модель, составляет 512 токенов. Размер блока был выбран аналогичным для наиболее эффективного использования возможностей векторного энкодера. Для минимизации потери контекста на границах сегментов было дополнительно установлено перекрытие соседних блоков в 100 токенов.

Была выполнена индексация использованной базы документов, в результате которой была сформирована векторная база данных, позволяющая получать релевантные запросам фрагменты документов и информацию о файлах документов, которым принадлежат эти фрагменты. Размер полученной базы данных векторов фрагментов документов составил 8,5 Гб. Скрипт генерации векторной базы данных (make_faiss.py) доступен в репозитории проекта на github.com.

7. Взаимодействие с LLM

В рамках реализации RAG-системы из построенной векторной базы данных документов извлекаются фрагменты, семантически близкие к клиентскому запросу. В данном исследовании запрашивался Топ-5 наиболее релевантных результатов. Для найденных фрагментов запрашивается их исходный текст и включается в контекст запроса, передаваемого в LLM. Промпт запроса указывает LLM использовать в процессе генерации ответа исключительно информацию из

предоставленного контекста. В результате обработки запроса LLM формирует и возвращает ответ на переданный клиентский запрос, содержащий скомпонованную на основе предложенного контекста информацию. В случае отсутствия релевантной запросу информации LLM возвращает сообщение об отсутствии необходимой информации в предоставленном контексте. Данный сценарий взаимодействия является предполагаемым и может реализовываться конкретными LLM с той или иной степенью успешности.

В качестве предварительного фильтра, отсеивающего запросы, для которых отсутствует релевантный контент, в реализованной системе используется показатель семантической близости к запросу найденных в векторной базе фрагментов документов. Если он ниже заданного порога (0,8), то запрос к LLM не формируется и пользователь системы получает уведомление об отсутствии в базе данных релевантной запросу информации.

Для исследования были отобраны 26 доступных на конец 2024 года моделей LLM, имеющих положительные отзывы и высокие рейтинги: 22 модели были загружены с портала Hugging Face, 4 модели использовались в режиме онлайн-доступа через предоставляемый API к серверам LLM в интернете.

Взаимодействие со всеми моделями было реализовано на основе библиотеки LangChain, позволяющей организовать работу с LLM различных моделей и локальных, и размещаемых в облаке, через единый программный интерфейс.

8. Общая архитектура системы на базе RAG

Общая архитектура системы, реализованной в данном исследовании, приведена на рис. 1.

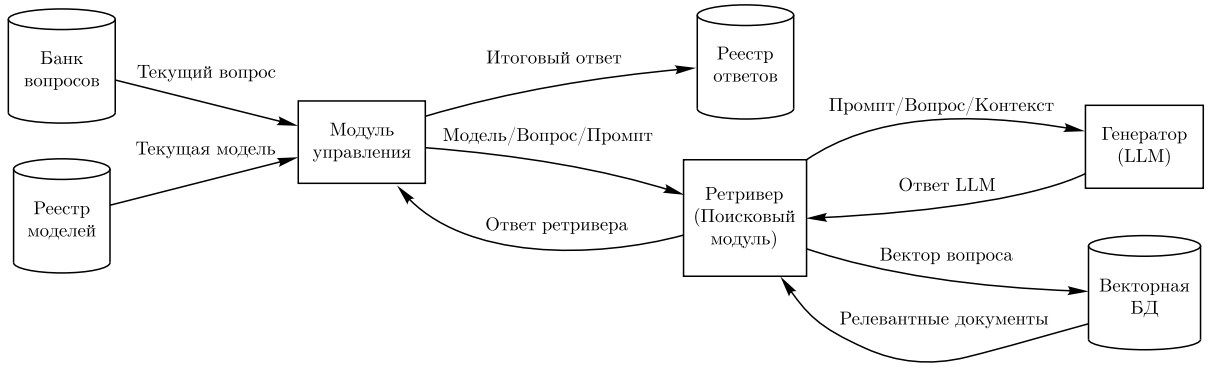

Рис. 1. Архитектура системы на базе RAG

Модуль управления системы реализует следующий сценарий.

1. Загружает и инициализирует список моделей LLM из реестра моделей.
2. Загружает список вопросов из банка вопросов.
   (a) Для каждой модели.
      i. Для каждого вопроса.
         A. Ретриверу передаются модель LLM, текущий вопрос и шаблон промпта.
         B. Ретривер формирует вектор запроса и отбирает, используя векторную базу, пять наиболее релевантных вопросу фрагментов документов.
         C. Ретривер формирует и передает LLM полный запрос, включающий в себя промпт, вопрос, и контекст запроса, сформированный из релевантных запросу фрагментов.

D. Ретривер принимает ответ LLM и передает его модулю управления вместе с данными о документах контекста.

E. Модуль управления сохраняет полученную информацию в реестре ответов.

Основным инструментом, обеспечивающим реализацию этого сценария, является класс RetrievalQA из библиотеки LangChain. Он реализует цепочку операций, включающую пункты A–D пункта 2.a.i изложенного выше сценария. При инициализации цепочки операций указывается ряд параметров, позволяющих задать модель LLM, число запрашиваемых из векторной базы документов, порог их релевантности, шаблон генерации промпта на основе запроса и другие настройки.

Исследование выполнялось на аппаратной базе, включающей компьютер на базе процессора AMD Ryzen 9 (3,8 ГГц, оперативная память 128 Гб) и видеоадаптера NVIDIA GeForce 4060 Ti, 16 Gb. На показателях времени реакции LLM в данном исследовании сказалось то обстоятельство, что полностью загружаются в память используемого видеоадаптера только модели, размер которых меньше 8 миллиардов коэффициентов. Локально загружаемые модели большего размера существенно увеличат свою производительность при увеличении размера видеопамяти, доступной в системе.

9. Полученные результаты

На начальных этапах исследования было обнаружено существенное влияние используемого промпта на качества ответов, получаемых от LLM. Несмотря на то что смысл альтернативных промптов был очень близок, характерное влияние их на качество результатов наблюдалось явно и не было однозначным. Для различных LLM лучшие результаты обеспечивали различные промпты. В итоге для каждой LLM предварительно был выбран один из двух промптов, обеспечивающий более успешную генерацию ответов.

Далее приведем примеры используемых промптов.

Промпт A.

Ответь исключительно на основе предоставленного контекста на следующий далее вопрос.

Не включай в ответ абсолютно никакой информации по данному вопросу, которой нет в предоставленном контексте. Если информация отсутствует в контексте, укажи: «*Ответ не может быть дан на основе предоставленного контекста*». *Отвечай на русском языке.*

Промпт B.

1. *Используй только предоставленный контекст для ответа на вопрос.*

2. *Не добавляй информацию, которая не содержится в предоставленном контексте.*

3. *Если информация отсутствует в контексте, укажи*: «*Ответ не может быть дан на основе предоставленного контекста*».

4. *Отвечай на русском языке.*

Тестирование с использованием приоритетного для модели промпта прошли следующие LLM.

Промпт A:

Saiga-Mistral-7B, Mistral-7B-it, Saiga-MistralNemo-12B, Gemma2-9B-it, Saiga-LLama3-8B, LLama-3.2-3B-it, Vikhr-Gemma-2B-it, Qwen2.5-7B-it, Neural-Chat-v3-3-it, Phi-3.5-mini-it, LLama-3.1-70B, Vikhr-Llama3.1-8B-it, Granite-3.0-2B, Nemotron-Mini-4B-it, T-lite-1.0-it, Gemma2-2B-it, Mistral-Nemo-it, Llama-3-8B-it, LLama-3.1-8B-it, Aya-Expanse-8B, Cotype-Nano, Kolibri-Vikhr-Mistral, RQwen-v0.1.

Промпт B:

GigaChat-Lite, YandexGPT, ChatGPT-4o-mini.

Все представленные в исследовании модели объединяет архитектура decoder-only Transformer. Индивидуальные особенности отдельных моделей связаны с технологиями, использованными при их обучении, числом слоев трансформеров, вариациями в используемых слоях нормализации и во внутренней организации слоев трансформера. Поддерживаемый размер контекста варьируется от 4k до 32k. У компактных моделей (~ 2B весов) число блоков трансформера составляет 16–22, у более крупных моделей (7B–9B) используется от 28 до 32 слоев.

Окончание it (сокращение от instruct) в названиях моделей маркирует те из них, которые проходили дополнительное обучение на датасетах «вопрос – ответ».

Доступ к четырем моделям выполнялся с использованием их API доступа через онлайн-сервисы в сети. Это модели GigaChat-Lite, YandexGPT, ChatGPT-4o-mini и LLama-3.1-70B. Остальные 22 модели были загружены с портала Hugging Face и в процессе исследования загружались локально в память видеоадаптера и компьютера.

Ответы моделей оценивались по четырем критериям на основе десятибалльной системы оценок. Для оценки ответов использовались следующие метрики:

1) точность: насколько корректны ответы;

2) полнота: насколько полно раскрывается тема;

3) лаконичность: насколько в ответе отсутствует избыточная информация;

4) грамотность: насколько полно соблюдены правила русского языка.

В таблице 1 приведены оценки моделей, сформированные на основе экспертных оценок их ответов на тестовые вопросы.

Модели в таблице 1 перечислены в порядке убывания значения метрики «Точность».

На рис. 2 представлена диаграмма распределения оценок LLM по совокупности метрик.

Для моделей был рассчитан комплексный показатель качества, в котором использовались следующие веса для отдельных показателей: «Точность» — 0,40; «Полнота» — 0,35; «Лаконичность» — 0,15; «Грамотность» — 0,10. Диаграмма ранжирования моделей на основе комплексного показателя качества представлена на рис. 3.

Диаграмма распределения среднего времени отклика различных моделей на запрос представлена на рис. 4. Вдоль горизонтальной оси в диаграмме указано время отклика в секундах. По скорости реакции ожидаемо лидируют мощные модели, работающие в режиме онлайн-сервисов. Большие задержки отклика ряда моделей связаны с ограниченным размером памяти использованного в исследовании графического адаптера. Если использовать в аналогичном решении видеоадаптеры с размером памяти от 32–48 гигабайт, то модели с числом весов от 8 миллиардов и более покажут существенно большую производительность.

Материалы и результаты данного исследования размещены на портале Github и доступны по ссылке https://github.com/igorant63/RAG_LLM. 26 файлов формата JSON содержат ответы каждой LLM на отправленные запросы, информацию о времени обработки каждого запроса и экспертные оценки каждого ответа. В файле files.txt находится список исследуемых LLM, файл questions.txt содержит список отправляемых вопросов. Файл llm_queries.log содержит вопросы, объединенные с контекстом, включаемым в RAG-запрос. Скрипт report3.py формирует усредненные оценки для каждой LLM и сохраняет их в отчет report3.txt. Скрипт make_faiss.py использовался для создания векторной базы данных на основе использованного датасета документов в формате PDF. Скрипт review.py позволяет открыть JSON-файл модели и просмотреть ее ответы на вопросы и оценки.

Таблица 1. Итоговые оценки ответов моделей

№ п/п	Модель	Точность	Полнота	Лаконичность	Грамотность
1	ChatGPT-4o-mini	**9,25**	**9,39**	9,81	9,89
2	Gemma2-9B-it	**9,25**	8,72	**9,89**	9,67
3	LLama-3.1-70B	9,11	8,75	9,75	9,94
4	GigaChat-Lite	8,89	8,89	9,17	**9,97**
5	Phi-3.5-mini-it	8,83	8,86	8,86	9,28
6	Gemma2-2B-it	8,64	8,25	9,83	9,36
7	YandexGPT	8,61	8,92	9,03	9,81
8	Kolibri-Vikhr-Mistral	8,61	8,31	9,56	9,69
9	Neural-Chat-v3-3-it	8,56	8,5	8,94	9,83
10	Mistral-Nemo-it	8,39	8,39	9,47	9,78
11	Mistral-7B-Instruct	8,36	8,25	9,22	9,58
12	Aya-Expanse-8B	8,22	8,47	8,28	9,44
13	Nemotron-Mini-4B-it	8,19	8,08	8,75	9,58
14	T-lite-1.0-it	7,81	8,47	7,47	8,81
15	Saiga-MistralNemo-12B	7,72	8,28	8,22	9,72
16	RQwen-v0.1	7,69	8,33	5,39	8,39
17	Granite-3.0-2B	7,69	7,89	7,86	8,75
18	Qwen2.5-7B-it	7,53	8,17	7,33	8,53
19	Saiga-Mistral-7B	7,5	7,75	8,64	8,92
20	Vikhr-Llama3.1-8B-it	7,08	8,03	5,39	7,78
21	Cotype-Nano	7,03	7,75	6,92	8,75
22	Llama-3-8B-it	6,64	7,42	7,31	5,81
23	LLama-3.2-3B-it	6,64	7,31	6,69	6,47
24	Saiga-LLama3-8B	6,39	7,11	5,28	7,03
25	LLama-3.1-8B-it	5,67	6,42	4,31	5,86
26	Vikhr-Gemma-2B-it	4,67	4,56	9,03	7,72

10. Обсуждение

Выполненное исследование продемонстрировало возможность успешной реализации корпоративной системы поиска информации и документов на основе технологий LLM и RAG в рамках рассмотренной выше архитектуры системы. Лучшие результаты сочетали высокую точность и грамотность ответов. Исследование подтвердило способность LLM успешно различать собственную информацию и информацию, получаемую ими из контекста, обнаруживать в предоставленном контексте запрашиваемую информацию и предоставлять ее в упорядоченном виде в ответе на запрос.

Следует отметить, несмотря на то что для тестирования использовались вопросы из области веб-программирования, основным объектом экспертной оценки являлись не внутренние знания моделей в этой сфере, а их способность работать с предоставленным контекстом. Ключевые оцениваемые параметры — точность и полнота — отражали, насколько эффективно модель может извлекать и синтезировать информацию, релевантную запросу, строго придерживаясь данных из предложенных ей фрагментов документов. Эта способность является универсальной и не зависит от конкретной предметной области. Умение корректно обрабатывать и обобщать информацию из ограниченного набора источников — ключевая задача для RAG-систем в любой корпоративной среде, будь то работа с технической документацией, юридическими договорами или финансовыми отчетами. Таким образом, полученные результаты и рейтинг моделей следует интерпретировать как оценку их общей эффективности в решении типовых задач извлечения знаний, что имеет значение для широкого круга технических и деловых текстов.

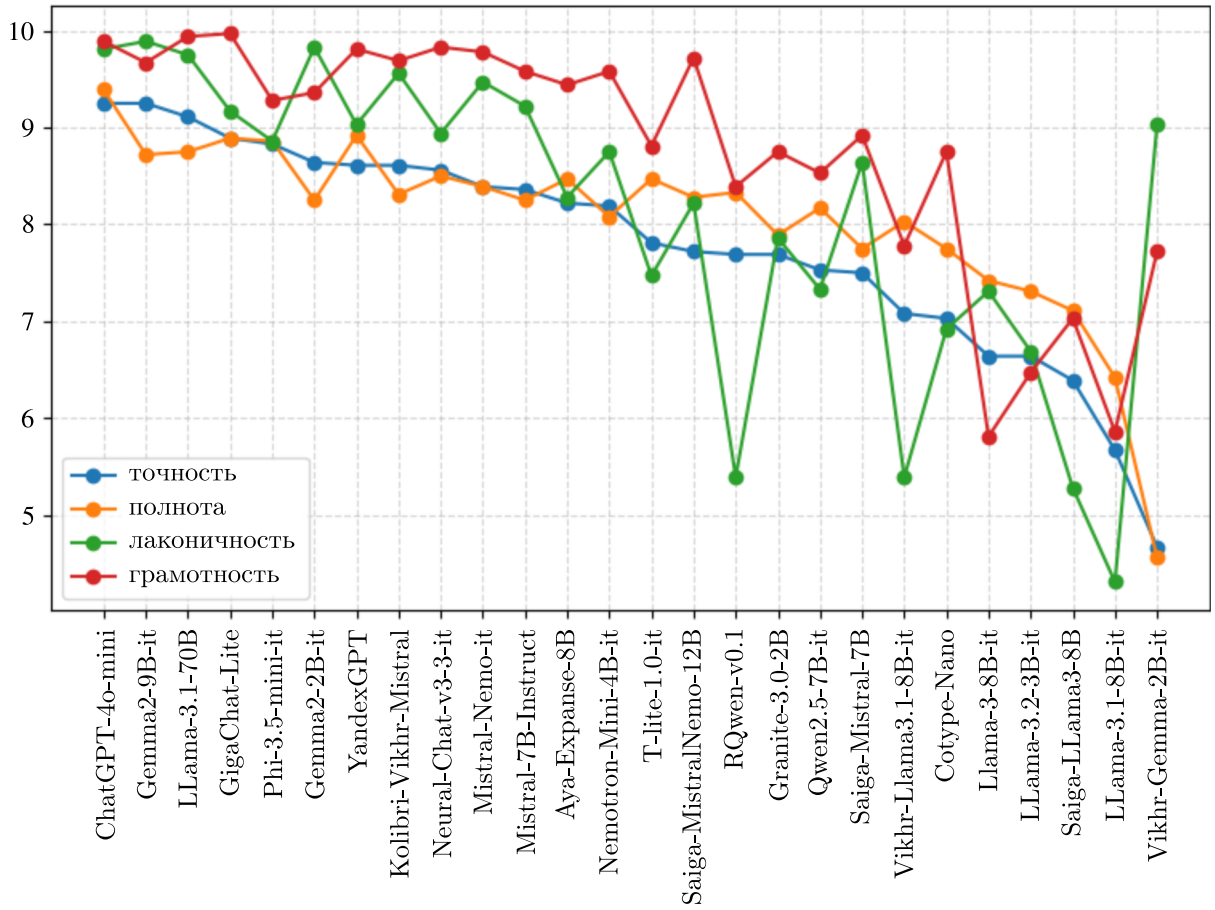

Рис. 2. Итоговые оценки моделей по совокупности метрик

Рассматриваемые LLM продемонстрировали существенный разброс качества ответов и времени отклика. Ожидаемо, что в полученном рейтинге лидируют мощные системы, доступные для использования через онлайн-сервисы. Так, ChatGPT-4o-mini показывает отличные результаты по всем качественным метрикам (точность: 9,25; грамотность: 9,89; полнота: 9,39; лаконичность: 9,81), LLama3.1-70B и GigaChat-Lite также демонстрируют очень высокие оценки ($> 9{,}9$) по грамотности и хорошие показатели по точности и полноте. YandexGPT выделяется высокой полнотой (8,92) и грамотностью (9,81) при высокой скорости отклика.

В то же время использование в рабочих процессах онлайн-сервисов может быть приемлемым не для всех организаций, прежде всего из соображений закрытости внутренней информации. Кроме того, доступ к онлайн-ресурсам может быть нестабильным и сталкиваться с административными ограничениями. Поэтому во многих случаях актуальным вариантом построения систем такого рода является использование локальной LLM для обработки запросов. Ряд локальных моделей, несмотря на существенно меньшее количество обучаемых весов по сравнению с онлайн-системами, также продемонстрировали хорошие результаты. Прежде всего, обращает на себя внимание модель Gemma2-9B-it, которая заняла второе место в общем рейтинге, имея равный с ChatGPT показатель точности и незначительно уступая ему по остальным показателям. Относительно большое среднее время отклика этой модели, превышающее в данном исследовании 20 секунд, может значительно сократиться при использовании видеоадаптеров с объемом памяти, превышающим 16 Гб. При наличии памяти GPU в пределах 16 Гб разумным решением может быть использование локальной модели Phi-3.5-mini-it, которая заняла 6-е место в общем рейтинге. Ее среднее время отклика составляло 11 секунд. В случае ограниченных ресурсов

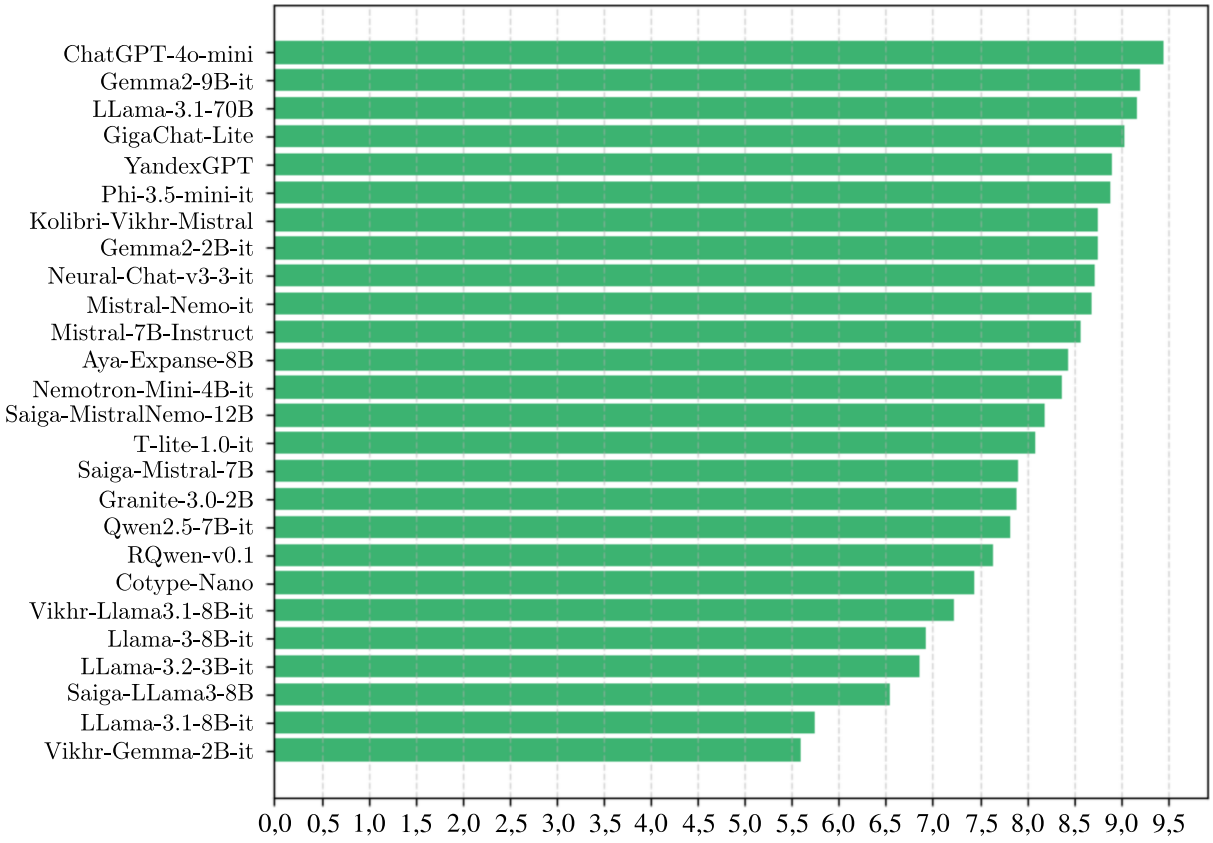

Рис. 3. Ранжирование LLM по интегральному показателю качества

памяти видеоадаптера (6 Гб) пригодная к практическому использованию система может быть построена на основе локальной модели Gemma2-2B-it, которая заняла 8-е место в общем рейтинге со средним временем отклика 3,2 с.

Важно отметить, что оценка времени отклика локальных моделей напрямую зависит от используемой аппаратной платформы, а именно видеоадаптера с 16 Гб видеопамяти. Такая конфигурация не всегда репрезентативна для промышленного внедрения. Типичный корпоративный заказчик может использовать как более мощные серверные решения (например, с GPU объемом памяти 32–48 Гб и выше), так и менее производительные стандартные рабочие места. В первом случае время отклика более крупных моделей (например, Gemma2-9B-it) значительно сократится (ориентировочно в 2–3 раза), что сделает их предпочтительным выбором для локального развертывания. Во втором случае, при ограниченных ресурсах, именно компактные и быстрые модели, такие как Gemma2-2B-it, становятся единственным практически применимым вариантом. Соответственно, выбор оптимальной локальной LLM должен производиться не только на основе показателей качества, но и с учетом имеющихся у заказчика вычислительных мощностей.

Для российских пользователей важным практическим аспектом при внедрении систем на базе LLM является уровень поддержки ими русского языка. Несмотря на то что не для всех из рассмотренных моделей официально заявлена поддержка русского языка, в целом исследуемые модели продемонстрировали хороший уровень грамотности. При этом у некоторых моделей наблюдались эпизодические артефакты в форме спонтанного перехода в тексте ответа на английский или китайский язык.

Исследование выявило неоднозначные результаты дополнительного обучения официальных моделей с целью улучшения поддержки ими русского языка. Три модели в данном обзоре

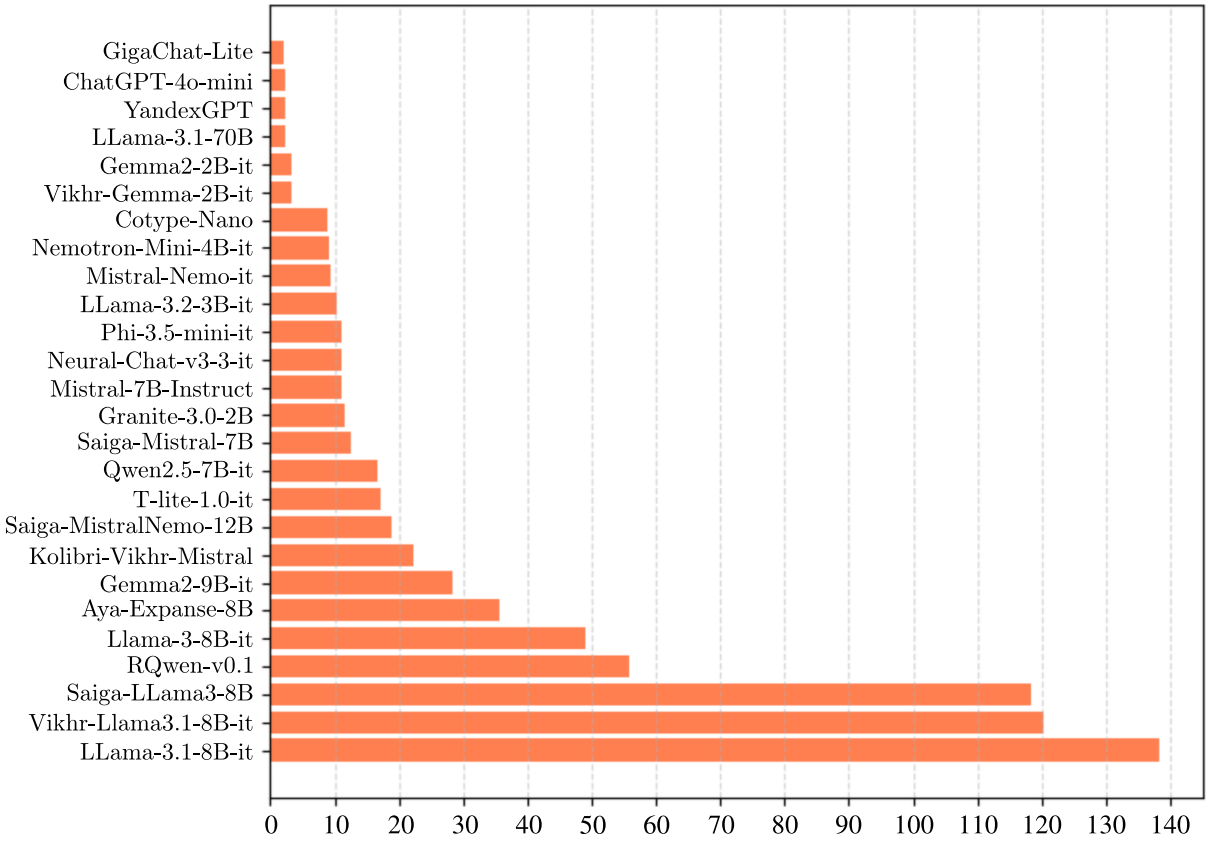

Рис. 4. Среднее время отклика моделей

представляют проект «Вихрь» [Vikhr models], в рамках которого выполнялось дополнительное обучение (fine-tuning) официальных моделей Mistral, Gemma и Llama на русскоязычных текстах для адаптации этих моделей к русскому языку. Модель Kolibri-Vikhr-Mistral несколько улучшила все показатели исходной модели Mistral-7B, но при этом увеличилось время отклика модели. Модель Vikhr-Llama3.1-8B-it аналогично несколько улучшила показатели своего прототипа Llama-3.1-8B, но модели Llama-3.1 в целом продемонстрировали невысокое качество ответов. Однако модель Vikhr-Gemma-2B-it в рассматриваемой задаче существенно уступила по всем показателям своему прототипу Gemma-2B-it. Другое семейство адаптированных к русскому языку моделей содержит в своем названии префикс Saiga [Gusev]. В рассматриваемой задаче модель Saiga-Mistral-7B по всем показателям уступила своему прототипу Mistral-7B, а Saiga-Llama3-8B уступила модели Llama3-8B.

Рассмотренное в данной работе решение может быть взято за основу при построении информационной системы, обеспечивающей поиск документов и получение информации из корпоративных информационных ресурсов. В то же время применяемая в рассмотренной выше системе фрагментация исходных документов на небольшие фрагменты, используемые в качестве контекста запросов к LLM, создает некоторые ограничения в отношении способности таких систем предоставлять связную и целостную информацию по объемным и сложным вопросам. То есть такая система обеспечит быстрый и эффективный доступ к информации о конкретных фактах и содержащих их документах. Однако связное изложение некоторой объемной концепции или теории останется для такой системы проблематичным. При необходимости преодолеть это ограничение перспективными могут оказаться подходы, реализующие дополнение фрагментов, найденных в векторной базе, их собственным контекстом в документе. Таким образом, для формирования более содержательного ответа может предоставляться расширенный контекст найден-

ного документа, если поддерживаемый LLM размер входного контекста позволяет реализовать данный подход. Такое расширение контекста повышает вероятность более содержательного ответа на исходный запрос.

Наиболее перспективным и универсальным путем преодоления ограничений стандартной RAG-архитектуры является использование агентной парадигмы в данной сфере. В рамках архитектуры на базе агентов LLM может выполнять ряд последовательных шагов, содержащих формирование дополнительных запросов к индексу документов и включение результатов этих запросов в контекст. Такой подход усложнит архитектуру решения и потребует дополнительной работы по созданию сценариев для агентов и тестированию достигнутых результатов. Но RAG-система такого типа потенциально может обеспечить дополнительный уровень полноты и качества предоставляемой информации.

11. Заключение

Современные большие языковые модели открыли новые возможности для построения корпоративных информационных систем семантического поиска и извлечения знаний. Интеграция LLM с технологией RAG превращает традиционный поиск документов в интеллектуальный сервис, способный:

- понимать естественный язык пользователя и вопросы, сформулированные в произвольной форме;
- оперативно извлекать релевантные фрагменты из корпоративных хранилищ информации;
- формировать грамотный и связный ответ на поставленный вопрос, даже если нужная информация распределена между множеством документов.

Важным аспектом построения RAG-системы является рациональный выбор используемой LLM. Не все модели, имеющие хорошие показатели в публикуемых тестах, в равной мере подходят для успешного построения RAG-системы. Для задач, допускающих использование онлайн-сервисов, такие модели, как GigaChat и YandexGPT, являются наиболее перспективными кандидатами. Они обеспечивают быстрые ответы и имеют высокие показатели качества. При построении систем с использованием локальных LLM прежде всего заслуживают внимание модели Gemma2-9B-it и Phi-3.5-mini-it, которые показывают хорошую точность и полноту ответов. Высокая грамотность ответов наблюдается у большинства моделей, что говорит о зрелости современных LLM в плане генерации синтаксически и орфографически корректного текста. Однако точность и полнота остаются ключевыми дифференцирующими факторами.

Список литературы (References)

Береснев А. Д. Применение открытых LLM с RAG-архитектурой для организации технической поддержки ИТ-сервисов // Альманах научных работ молодых ученых Университета ИТМО: Материалы Пятьдесят третьей (LIII) научной и учебно-методической конференции, Санкт-Петербург, 29 января – 2 февраля 2024 года. — СПб.: Национальный исследовательский университет ИТМО, 2024. — С. 44–48.

Beresnev A. D. Primenenie otkrytykh LLM c RAG-arkhitekturoi dlya organizatsii tekhnicheskoi podderzhki IT-servisov [The use of open LLM with RAG architecture for the organization of technical support for IT services] // Al'manakh nauchnykh rabot molodykh uchenykh Universiteta ITMO: Materialy Pyat'desyat tret'ei (LIII) nauchnoi i uchebno-metodicheskoi konferentsii, Sankt-Peterburg, 29 yanvarya – 2 fevralya 2024 goda. — Sankt-Peterburg: Natsional'nyi issledovatel'skii universitet ITMO, 2024. — P. 44–48 (in Russian).

Бородулин И. В. Увеличение точности больших языковых моделей с помощью расширенной поисковой генерации // Вестник науки. — 2024. — Т. 3, № 3 (72). — С. 400–405.
Borodulin I. V. Uvelichenie tochnosti bol'shikh yazykovykh modelei s pomoshch'yu rasshirennoi poiskovoi generatsii [Increasing the accuracy of large language models with advanced search generation] // Vestnik nauki. — 2024. — Vol. 3, No. 3 (72). — P. 400–405 (in Russian).

Жигалов А. Ю., Болодурина И. П. Использование RAG-метода для анализа образовательных документов // Цифровые технологии в образовании, науке, обществе: Материалы XVIII Всероссийской научно-практической конференции, Петрозаводск, 3–5 декабря 2024 года. — Петрозаводск: Петрозаводский государственный университет, 2024. — С. 54–56.
Zhigalov A. Yu., Bolodurina I. P. Ispol'zovanie RAG-metoda dlya analiza obrazovatel'nykh dokumentov [Using the RAG method to analyze educational documents] // Tsifrovye tekhnologii v obrazovanii, nauke, obshchestve: Materialy XVIII Vserossiiskoi nauchno-prakticheskoi konferentsii, Petrozavodsk, 3–5 dekabrya 2024 goda. — Petrozavodsk: Petrozavodskii gosudarstvennyi universitet, 2024. — P. 54–56 (in Russian).

Зупарова В. В. Применение гибридного подхода на основе RAG и LLM для повышения точности ответов в интеллектуальных системах поддержки // Синтез науки и образования как инструмент решения глобальных проблем современности: Сборник статей по итогам Международной научно-практической конференции, Омск, 15 августа 2024 года. — Стерлитамак: ООО «Агентство международных исследований», 2024. — С. 79–81.
Zuparova V. V. Primenenie gibridnogo podkhoda na osnove RAG i LLM dlya povysheniya tochnosti otvetov v intellektual'nykh sistemakh podderzhki [Applying a hybrid approach based on RAG and LLM to improve response accuracy in intelligent support systems] // Sintez nauki i obrazovaniya kak instrument resheniya global'nykh problem sovremennosti: Sbornik statei po itogam Mezhdunarodnoi nauchno-prakticheskoi konferentsii, Omsk, 15 avgusta 2024 goda. — Sterlitamak: OOO "Agentstvo mezhdunarodnykh issledovanii", 2024. — P. 79–81 (in Russian).

Изосимова К. С., Соловьева А. Ю., Попов В. В., Чернышева Т. Ю. Использование поисковой дополненной генерации в рекомендательных книжных системах // Информационные технологии в образовании. — 2024. — № 7. — С. 120–125.
Izosimova K. S., Solov'eva A. Yu., Popov V. V., Chernysheva T. Yu. Ispol'zovanie poiskovoi dopolnennoi generatsii v rekomendatel'nykh knizhnykh sistemakh [Using augmented search generation in recommendation book systems] // Informatsionnye tekhnologii v obrazovanii. — 2024. — No. 7. — P. 120–125 (in Russian).

Овсянников И. В., Сарычев С. П. Применение локальных языковых моделей для разработки чатбота технической поддержки пользователей по работе с реестрами счетов МИС ЕГИСЗ НСО // Политранспортные системы: Материалы XIII Всероссийской научно-технической конференции с международным участием, Новосибирск, 24–25 октября 2024 года. — Новосибирск: Сибирский государственный университет путей сообщения, 2024. — С. 480–484.
Ovsyannikov I. V., Sarychev S. P. Primenenie lokal'nykh yazykovykh modelei dlya razrabotki chat-bota tekhnicheskoi podderzhki pol'zovatelei po rabote s reestrami schetov MIS EGISZ NSO [Application of local language models for the development of a chatbot for technical support of users working with registers of accounts of the The Unified State information system in the field of healthcare of the Novosibirsk region] // Politransportnye sistemy: Materialy XIII Vserossiiskoi nauchno-tekhnicheskoi konferentsii s mezhdunarodnym uchastiem, Novosibirsk, 24–25 oktyabrya 2024 goda. — Novosibirsk: Sibirskii gosudarstvennyi universitet putei soobshcheniya, 2024. — P. 480–484 (in Russian).

Олейник А. Г., Федоров А. М., Датьев И. О., Зуенко А. А., Шестаков А. В., Вишняков И. Г. Использование RAG-технологии для создания интеллектуальной информационной системы поддержки исследовательского поиска // Труды Кольского научного центра РАН. Сер. Технические науки. — 2024. — Т. 15, № 3. — С. 5–26.
Oleynik A. G., Datyev I. O., Zuenko A. A., Fedorov A. M., Shestakov A. V., Vishnyakov I. G. Ispol'zovanie RAG-tekhnologii dlya sozdaniya intellektual'noi informatsionnoi sistemy podderzhki issledovatel'skogo poiska [Using RAG technology to create an intelligent information system to support research search] // Trudy Kol'skogo nauchnogo tsentra RAN. Ser. Tekhnicheskie nauki. — 2024. — Vol. 15, No. 3. — P. 5–26 (in Russian).

Федоров В. О., Поляков Р. А. Большие языковые модели с поисковой расширенной генерацией: обзор и перспективы // Оригинальные исследования. — 2023. — Т. 13, № 12. — С. 43–47.
Fedorov V. O., Polyakov R. A. Bol'shie yazykovye modeli s poiskovoi rasshirennoi generatsiei: obzor i perspektivy [Large language models with advanced search generation: overview and prospects] // Original'nye issledovaniya. — 2023. — Vol. 13, No. 12. — P. 43–47 (in Russian).

Cheng M., Luo Y., Ouyang J., Liu Q., Liu H., Li L., Yu S., Zhang B., Cao J., Ma J., Wang D., Chen E. A survey on knowledge-oriented retrieval-augmented generation // arXiv preprint. — 2025. — arXiv:2503.10677

Gao Y., Xiong Y., Gao X., Jia K., Pan J., Bi Y., Dai Y., Sun J., Guo Q., Wang M., Wang H. Retrieval-augmented generation for large language models: a survey // arXiv preprint. — 2023. — arXiv:2312.10997

Gusev I. Russian fine-tunes of different base LLMs. — [Electronic resource]. — https://huggingface.co/collections/IlyaGusev/saiga-6505d4ccc3d1e53166b636cd (accessed: 01.06.2025).

Hugging Face Hub. — [Electronic resource]. — https://huggingface.co/ (accessed: 01.06.2025).

Hugging Face: The AI community building the future. — [Electronic resource]. — https://huggingface.co/ (accessed: 01.06.2025).

Johnson J., Douze M., Jégou H. Faiss: a library for efficient similarity search and clustering of dense vectors // Advances in Neural Information Processing Systems 32 (NeurIPS 2019). — 2019. — Article 9873.

LangChain Team. LangChain. Building applications with LLMs through composability. — [Electronic resource]. — https://github.com/langchain-ai/langchain (accessed: 01.06.2025).

Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., Riedel S., Kiela D. Retrieval-augmented generation for knowledge-intensive NLP tasks // Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20). — Red Hook, NY: Curran Associates Inc., 2020. — Article 793. — P. 9459–9474.

Vikhr models. Community. — [Electronic resource]. — https://huggingface.co/Vikhrmodels (accessed: 01.06.2025).

Weng L. Large language models: a survey // arXiv preprint. — 2023. — arXiv:2303.18223