УДК 004.852; 021.6
DOI: 10.26102/2310-6018/2026.53.2.008

Агентный подход к интеллектуальному поиску в библиотечных системах

И.С. Рзянкин[1]✉, Р.А. Барышев[1], А.А. Гучко[2]

[1]Сибирский федеральный университет, Красноярск, Российская Федерация
[2]Независимый исследователь, Красноярск, Российская Федерация

Резюме. В статье исследуется применение агентного подхода Retrieval-Augmented Generation (Agentic RAG) в задачах интеллектуального поиска по библиотечным фондам. Объектом исследования является архитектура Agentic RAG, объединяющая методы извлечения информации, агентное планирование и механизмы самооценки промежуточных результатов. Рассматриваемая проблема связана с ограничениями классического Retrieval-Augmented Generation при обработке сложных тематических и контекстных запросов в условиях семантически насыщенных библиотечных данных. В отличие от традиционного RAG, агентная архитектура позволяет итеративно уточнять стратегию поиска, адаптироваться к контексту запроса и пересматривать промежуточные результаты. Методология исследования основана на разработке программного прототипа Agentic RAG и его экспериментальном сравнении с классическим RAG на корпусе реальных данных университетской библиотеки, включающем библиографические метаданные, аннотации и фрагменты полных текстов. Для оценки эффективности использованы количественные метрики информационного поиска (Precision@k, Recall@k, MRR, nDCG) и экспертная оценка релевантности итоговых ответов. Результаты демонстрируют устойчивое превосходство Agentic RAG по показателям точности, полноты и качества ранжирования, особенно при обработке сложных запросов. При этом интерпретация выводов ограничена выбранным набором метрик и параметрами экспериментального корпуса. Практическая значимость заключается в возможности внедрения агентной архитектуры в библиотечно-информационные системы без радикальной перестройки инфраструктуры.

Ключевые слова: агентный поиск, Retrieval-Augmented Generation, библиотечные информационные системы, интеллектуальный поиск, семантический поиск, нейросетевые технологии, агентные архитектуры.

Для цитирования: Рзянкин И.С., Барышев Р.А., Гучко А.А. Агентный подход к интеллектуальному поиску в библиотечных системах. *Моделирование, оптимизация и информационные технологии*. 2026;14(2). URL: https://moitvivt.ru/ru/journal/article?id=2199 DOI: 10.26102/2310-6018/2026.53.2.008

Agent-based approach to intelligent search in library systems

I.S. Rzyankin[1]✉, R.A. Baryshev[1], A.A. Guchko[2]

[1]Siberian Federal University, Krasnoyarsk, the Russian Federation
[2]Independent Researcher, Krasnoyarsk, the Russian Federation

Abstract. The article explores the application of an agent-based Retrieval-Augmented Generation (Agentic RAG) approach to intelligent search tasks in library collections. The object of the study is the Agentic RAG architecture, which integrates information retrieval mechanisms with agent-based planning and self-evaluation of intermediate results. The addressed problem concerns the limitations of classical Retrieval-Augmented Generation in handling complex thematic and contextual queries within semantically rich library data environments. Unlike traditional RAG pipelines, the agent-based architecture enables iterative refinement of search strategies, adaptive decision-making, and reassessment of intermediate outcomes. The research methodology is based on the development of a

software prototype implementing Agentic RAG and its experimental comparison with a classical RAG baseline using a real university library corpus comprising bibliographic metadata, annotations, and full-text fragments. The evaluation framework includes standard information retrieval metrics (Precision@k, Recall@k, MRR, nDCG) as well as expert-based assessment of answer relevance. The results demonstrate a consistent superiority of Agentic RAG in terms of retrieval accuracy, recall, and ranking quality, particularly for complex queries. However, the interpretation of findings is constrained by the selected evaluation metrics and the characteristics of the experimental corpus. The practical significance lies in the potential integration of agent-based architectures into library information systems without requiring substantial infrastructural changes.

Keywords: agent-based search, Retrieval-Augmented Generation, library information systems, intelligent search, semantic search, neural network technologies, agent architectures.

For citation: Rzyankin I.S., Baryshev R.A., Guchko A.A. Agent-based approach to intelligent search in library systems. *Modeling, Optimization and Information Technology*. 2026;14(2). (In. Russ.) 2URL: https://moitvivt.ru/ru/journal/article?id=2199 DOI: 10.26102/2310-6018/2026.53.2.008

Введение

По мере выхода новых исследований и научных трудов, наполнение библиотечных фондов неуклонно увеличивается, появляется огромное количество новых данных, по которым ориентируются пользователи библиотек. Данную проблему активно решает цифровизация библиотек, которая сейчас идет полным ходом и по которой опубликовано немало научных трудов [1], цифровизация стала трендом современного библиотечного дела [2].

Современные библиотечные фонды – это высокоструктурированные, но семантически сложные данные, которые включают в себя множество метаданных – авторы, аннотации, ключевые слова. По мере увеличения фондов, пользователям все чаще требуется не просто искать в них материалы по названию и ключевым словам, а требуется получать тематическую подборку материалов по смыслу своего запроса, или получить взаимосвязанный тематический контент. Традиционные поисковые алгоритмы уже не способны удовлетворить такие задачи [3].

С момента начала активного использования нейронных сетей проводились активные обсуждения и попытки внедрить их использование в библиотечно-издательскую сферу, но такое внедрение было очень осторожным, так как LLM неизбежно свойственны галлюцинации, а библиотечно-издательская сфера оперирует исключительно точными данными, и любая ошибка в них, совершенная по вине LLM, может быть критичной. Но с момента появления технологии RAG (Retrieval-Augmented Generation), попытки внедрить нейронные сети в библиотеки возобновились. Дело в том, что нейросети, действующие по технологии RAG, перед генерацией своего ответа производят поиск нужной информации во внешней базе, вносят ее в свое контекстное окно, а затем, используют контекст, чтобы сгенерировать более точный и обоснованный ответ. RAG особенно хорошо подходит для работы с библиотечными данными, так как в них важно объединять обширные источники знаний и контекст запросов пользователей. Такая архитектура позволяет системе сначала искать релевантные книги, статьи или каталожные записи, а затем формировать ответ, учитывающий найденные материалы и смысл вопроса. Благодаря этому RAG способен соединять точность фактографического поиска с гибкостью генеративных моделей, предоставляя осмысленные, контекстно богатые ответы, основанные на реальных данных библиотечного фонда.

Тем не менее, несмотря на потенциал данной технологии, в ее применении сохраняется ряд проблем. Во-первых, они сталкиваются с проблемой контекстной

интеграции – даже при корректном извлечении релевантных данных RAG часто не способен органично включить их в итоговый ответ, что приводит к фрагментированным, непоследовательным или чрезмерно общим результатам. Во-вторых, RAG слабо справляется с многошаговым рассуждением: такие системы не умеют уточнять поиск на основе промежуточных результатов и затрудняются при обработке сложных, составных запросов, требующих объединения данных из разных источников. В-третьих, существует проблема масштабируемости и задержек – при росте объемов данных операции поиска и ранжирования становятся вычислительно затратными, снижая скорость отклика и ограничивая применение в реальном времени [4].

Но несмотря на все вышеперечисленное, эти ограничения не являются тупиком развития: они послужили отправной точкой для появления Agentic RAG, который объединяет RAG с автономными агентами, обеспечивая динамическое принятие решений, итеративное уточнение запросов и адаптацию рабочих процессов в реальном времени. Эта парадигма вводит рассуждения на основе агентов, позволяя модели планировать, отражать и совершенствовать свой поисковый процесс итеративно. Такая модель не только извлекает и использует внешние данные, но и планирует собственные действия, оценивает качество промежуточных результатов и итеративно совершенствует процесс поиска и генерации. Благодаря внедрению механизмов саморефлексии и адаптивного планирования Agentic RAG обеспечивает более глубокое понимание контекста и устойчивость к ошибкам. Уже показано, что подобные подходы демонстрируют высокую эффективность в таких областях, как анализ научных коллекций, образовательные системы и интеллектуальные рекомендательные платформы [4].

По состоянию на конец 2025 года технология RAG уже применялась в библиотечных системах, тогда как Agentic RAG – только в смежных областях [3, 4].

Цель данного исследования – оценить эффективность Agentic RAG в задачах библиотечного поиска и сравнить ее с классическим RAG.

Гипотеза исследования состоит в том, что Agentic RAG обеспечивает более высокую релевантность и лучшую интерпретируемость результатов за счет итеративного и самооценочного характера своей архитектуры.

Подход Retrieval-Augmented Generation (RAG) стал одной из ключевых архитектур для объединения генеративных языковых моделей с внешними источниками знаний.

Изначально RAG был предложен как способ повысить достоверность ответов нейросетевых моделей за счет включения этапа поиска релевантных документов перед генерацией текста [5].

Такой механизм позволяет модели не только опираться на собственные параметры, но и использовать актуальные данные из внешних баз знаний, что обеспечивает более точные и обоснованные ответы [6].

За последние годы подход RAG успешно применялся в задачах интеллектуального поиска, вопросно-ответных систем, научной аналитики и образовательных сервисов [7].

Однако классические реализации RAG имеют ряд ограничений:

– во-первых, они используют фиксированные стратегии извлечения данных;

– во-вторых, генерация ответа осуществляется в один проход, без возможности корректировки на основе промежуточных результатов;

– в-третьих, отсутствует механизм самооценки качества поиска [8].

Эти ограничения послужили основой для появления новых адаптивных, агентных вариантов RAG, способных к планированию и рефлексии.

В последние годы появились исследования, посвящённые применению RAG в задачах библиотечного поиска и информационного обслуживания.

Так, в работе Designing Question-Answer Based Search System in Libraries (2024) предложена система вопросно-ответного поиска на основе RAG с использованием фреймворков LangChain и ChromaDB, что позволило улучшить доступ пользователей к метаданным библиотечных каталогов [9].

Другое исследование – Prospects of RAG for Academic Library Search and Retrieval (2024) – рассмотрело перспективы интеграции RAG в академические библиотеки, показав, что данный подход способен объединить точность фактографического поиска с гибкостью семантической генерации [3].

Тем не менее, все эти системы использовали только классический вариант RAG, без агентной логики и итеративного уточнения запросов.

Авторы отмечают трудности при обработке сложных, составных запросов, требующих объединения данных из разных источников, а также ограниченную способность моделей к адаптации под контекст пользователя.

Появление агентного подхода (Agentic RAG) стало важным этапом развития RAG-архитектур.

Согласно Singh и соавт. (2025), Agentic RAG объединяет традиционный механизм поиска с агентной архитектурой, включающей планирование, самооценку и итеративное уточнение запросов [4].

Благодаря этому модель способна не только искать и использовать внешние данные, но и самостоятельно формировать стратегию поиска, оценивать промежуточные результаты и переформулировать запросы в процессе решения задачи.

Agentic RAG уже показал эффективность в ряде направлений:

– научные коллекции – система CollEX применяет мультимодальный Agentic RAG для анализа и связывания исследовательских данных [10];

– гибридные аналитические системы – Agentic Hybrid RAG Framework объединяет RAG с графовыми структурами знаний для анализа научных цитирований [11];

– образовательные и рекомендательные платформы – агентные модели используются для адаптивного обучения и контекстной фильтрации материалов [12].

Анализ приведенных работ показывает, что архитектуры RAG доказали свою эффективность в задачах интеллектуального поиска и генерации контекстуальных ответов, включая библиотечные сценарии.

В то же время, агентные модификации RAG, обладающие механизмами планирования и самооценки, активно развиваются в научных и образовательных системах, но еще не применялись в библиотечно-информационной среде.

Таким образом, наблюдается очевидный научный зазор: несмотря на семантическое сходство между научными коллекциями и библиотечными фондами, потенциал Agentic RAG для библиотечного поиска остается не исследованным.

Настоящее исследование направлено на восполнение этого пробела и включает количественное и качественное экспериментальное сравнение Agentic RAG и классического RAG при решении задач информационного поиска в библиотечных системах на реальных данных.

Материалы и методы

Общая архитектура исследования. В рамках данного исследования было разработано программно-нейросетевое решение, основанное на архитектуре Retrieval-Augmented Generation (RAG), но при этом реализующее функциональность интеллектуального агента. Такой подход сочетает принципы классического RAG и

агентных систем искусственного интеллекта. Он является относительно новым направлением: по состоянию на октябрь 2025 года агентные реализации RAG остаются слабо исследованными и находятся на стадии активного экспериментирования [4].

Разработанная система функционирует не как статический пайплайн, а как агент, который самостоятельно принимает решения о стратегии поиска и последовательности действий. В отличие от классического RAG, где модель просто получает фрагменты текста на основе семантической близости и использует их для генерации ответа, агентный RAG анализирует содержание запроса, определяет, какой тип поиска более уместен – векторный (по эмбеддингам) или полнотекстовый (по BM25), – и формирует собственный план выполнения. Таким образом, RAG в данном решении выступает не просто как retrieval-компонент, а как часть агентной среды, где нейросеть способна к элементарному планированию и самооценке промежуточных результатов. Визуализация данного подхода представлена на Рисунке 1.

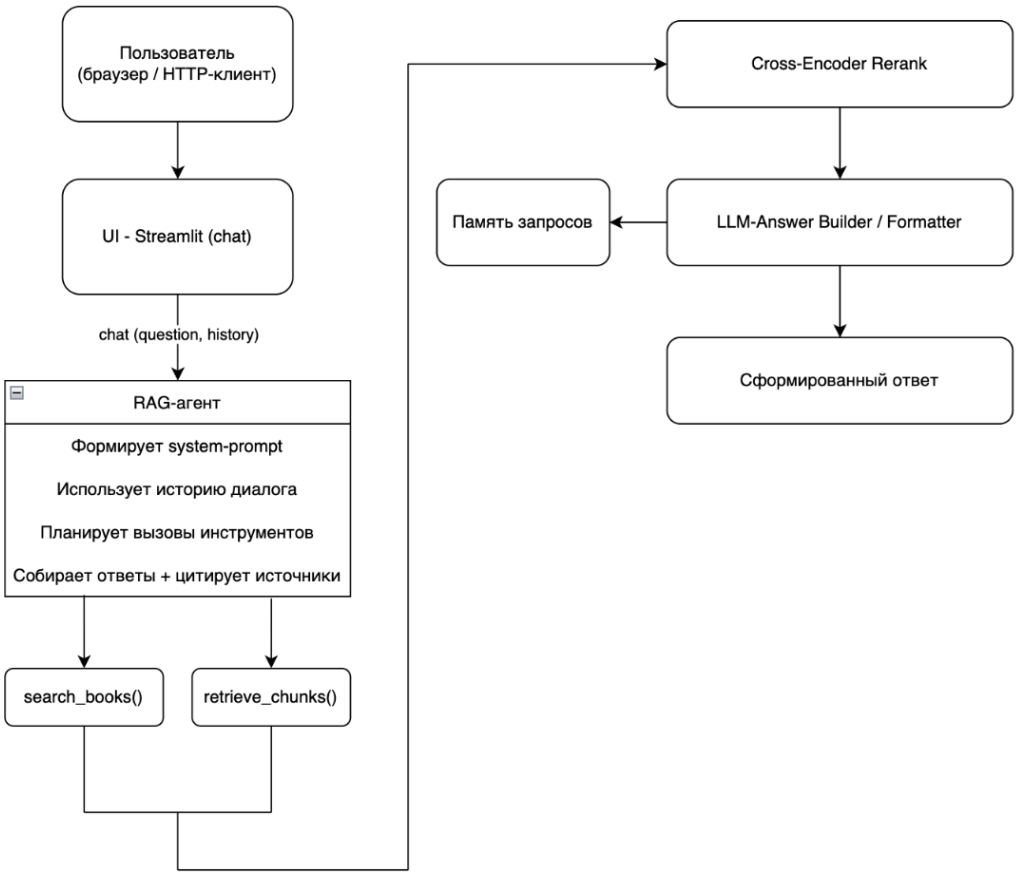

Рисунок 1 – Архитектура системы библиотечного поиска на основе агентного RAG
Figure 1 – Architecture of the library search system based on agent-based RAG

Классический RAG (Baseline Model). В качестве отправной точки для эксперимента была реализована классическая архитектура Retrieval-Augmented Generation (RAG), на основе которой построен прототип библиотечного поиска.

Данный вариант системы представляет собой неагентную последовательность модулей, где поиск и генерация ответа выполняются в один проход, без итераций и без планирования действий.

Архитектура включает два параллельных канала поиска – векторный и полнотекстовый. Первый использует векторное хранилище ChromaDB, где сохранены эмбеддинги аннотаций и фрагментов книг. Каждый документ имеет отдельный вектор, а

текст книги разбивается на чанки по ≈ 1000 символов. Второй канал построен на SQLite + BM25, обеспечивая поиск по точным совпадениям слов и устойчивость к частотным запросам.

Оба канала дополняют друг друга: векторный поиск эффективен при семантических запросах, тогда как BM25 дает преимущество при формальных или терминологических обращениях. После извлечения документов из обоих каналов результаты объединяются в общий пул, который обрабатывается компонентом CrossEncoder Rerank – он уточняет порядок релевантности фрагментов и снижает долю нерелевантных ответов. Затем модуль LLM-Answer Builder формирует итоговый ответ на естественном языке, опираясь на найденные фрагменты и добавляя ссылки на источники.

На практике классический RAG был дополнен рядом технических усовершенствований:

1. В системе применялись два эмбеддера с кросс-голосованием для повышения точности семантического поиска, а также дублирование данных в SQLite для ускорения полнотекстового доступа.
2. Логика конвейера реализована в виде пятиэтапного пайплайна (Рисунок 2):
3. Поиск ближайших соседей (k-NN) по всем текстовым чанкам.
4. Определение book_id, в рамках которого найдено наибольшее количество совпадений или суммарных очков релевантности.
5. Повторный поиск чанков, но уже только внутри определенной книги.
6. Реранжирование результатов через CrossEncoder.
7. Генерация и стриминг ответа LLM-моделью с выводом названия книги-источника и числа использованных фрагментов.

Для более наглядного представления логики извлечения и ранжирования фрагментов текста в классической архитектуре RAG был реализован многоэтапный конвейер поиска. Он объединяет результаты векторного и полнотекстового поиска, выполняет агрегацию по источникам и последующее уточнение релевантности с помощью реранжирования. Фрагмент реализации данного конвейера, используемого в качестве базовой модели (baseline) в эксперименте, представлен на Рисунке 2.

```
"""
RAG-конвейер v3

1.  k-NN по ВСЕМ текстовым чанкам (type="content_chunk").
2.  Считаем, для какого `book_id` чаще всего (или с наибольшей суммой очков)
    встречаются эти чанки → «победитель».
3.  Повторно ищем чанки, но уже **только** внутри найденной книги.
4.  Cross-encoder ранжирует выбранные чанки.
5.  GPT-4o-mini стримит ответ; на экран выводятся название книги-источника
    и число чанков, попавших в контекст.

Коллекция Chroma та же:
    ├─ {book_id}_ann        ┐ аннотация (type="annotation")
    └─ {book_id}_c{idx}     ┘ чанки (type="content_chunk")
"""
```

Рисунок 2 – Фрагмент реализации классического RAG-конвейера v3 (поиск и ранжирование чанков)
Figure 2 – Fragment of the implementation of the classic v3 RAG pipeline (search and ranking of chunks)

Этот конвейер обеспечивает корректную работу при простых и точных запросах, демонстрируя высокий уровень фактографической точности.

Однако его принципиальным ограничением остается отсутствие механизма адаптации к контексту: система не оценивает, насколько найденные фрагменты действительно отвечают информационному намерению пользователя и не умеет изменять стратегию поиска при неудачных результатах.

Эти наблюдения послужили основанием для перехода к агентной архитектуре RAG, в которой добавлены планирование, самооценка и итеративное уточнение поиска.

Данный вариант использовался как baseline-модель для последующего сравнительного анализа с Agentic RAG.

Agentic RAG (Proposed Approach). В предложенном решении агент RAG выполняет серию шагов, направленных на автономное управление процессом поиска и генерации ответов.

При поступлении запроса агент формирует system prompt, в который включаются указания о необходимых действиях и ссылки на доступные инструменты. Эти инструменты представлены как функции: search_books() – обращение к векторной базе ChromaDB, и retrieve_chunks() – полнотекстовый поиск через SQLite и BM25. Агент использует историю диалога для сохранения контекста и уточнения намерений пользователя.

Алгоритм работы включает несколько этапов. Сначала агент строит план действий – определяет, к каким источникам обратиться и в каком порядке. Затем он выполняет поиск, анализирует полученные результаты и оценивает их релевантность. После этого формируется новый план, и при необходимости поиск повторяется с уточненным запросом.

Векторное хранилище ChromaDB содержит эмбеддинги аннотаций и фрагментов текста книг. Аннотации индексируются как отдельные объекты с собственными векторами, а полные тексты разбиваются на фрагменты (чанки) размером около 1000 символов, каждый из которых также представлен отдельным эмбеддингом. Это позволяет агенту комбинировать результаты по смыслу и по контексту, а также находить связи между аннотацией и соответствующим текстом книги.

Параллельно используется модуль полнотекстового поиска на основе BM25 и SQLite, который обеспечивает прямое соответствие слов из пользовательского запроса текстам библиографических записей. Оба канала поиска объединяются в результирующем пуле, после чего применяется CrossEncoder Rerank для уточнения ранжирования документов. Итоговый ответ формируется модулем LLM-Answer Builder, который комбинирует результаты и формирует связный текст с указанием источников. Для анализа стоит отметить, что в обоих вариантах использовался один и тот же CrossEncoder для справедливого сравнения.

Таким образом, реализованный Agentic RAG сочетает возможности классического извлечения по эмбеддингам с гибкостью агентного планирования и возможностью итеративного уточнения запроса. Это обеспечивает более точную релевантность и устойчивость к ошибкам по сравнению с традиционным RAG.

В отличие от классического конвейера, реализующего фиксированную последовательность операций поиска и генерации ответа, предложенная в работе архитектура Agentic RAG функционирует как автономный интеллектуальный агент. Такая система не ограничивается однократным извлечением релевантных фрагментов, а включает этапы планирования, оценки промежуточных результатов и итеративного уточнения стратегии поиска. Обобщенная схема авторской агентной архитектуры Retrieval-Augmented Generation, использованной в эксперименте, представлена на Рисунке 3.

```
Query → Planner → Retriever → Generator → Evaluator → (Loop) → Final Answer
```

Рисунок 3 – Архитектура авторского агентного варианта RAG (Agentic RAG)
Figure 3 – Architecture of the author's agent-based version of RAG (Agentic RAG)

Подробное описание данных, метрик и экспериментальной среды приведено в следующих разделах.

Следует отметить, что эффективность Agentic RAG зависит от нескольких технологических факторов, которые ограничивают его применимость. Во-первых, агентная стратегия чувствительна к глубине итераций: увеличение числа шагов приводит к росту времени отклика и накоплению погрешности при оценке промежуточных результатов. Во-вторых, работа агента предполагает высокую детерминированность структурированного вывода модели; отклонения от ожидаемого формата или вариативность генерации могут нарушать корректность принятия решений. В-третьих, гибридная схема извлечения требует стабильности эмбеддингов и точной работы BM25, что делает систему чувствительной к качеству данных и единообразию токенизации. Эти особенности не влияют на валидность эксперимента, но определяют технологические пределы масштабирования Agentic RAG в продуктивных системах.

В рамках эксперимента была проанализирована работа Agentic RAG на типичном пользовательском запросе «что можно почитать по физике». На первом шаге агент инициировал гибридный поиск (BM25 + векторная модель), получив подборку учебных и научных изданий по базовым и специальным разделам физики. Далее был выполнен SQL-поиск по полю title, позволивший дополнительно выявить тематически подходящие источники, включая материалы по гидрофизике и молекулярной физике. На основании совокупности результатов агент сформировал итоговый список рекомендаций, включающий учебные пособия, практикумы и научные сборники, доступные в электронной библиотеке СФУ.

Дополнительно была проанализирована работа Agentic RAG на более контекстном и слабо формализованном запросе: «психология для широкого круга читателей, популярные и интересные книги». В отличие от тематического запроса по дисциплине, данный запрос содержит указание на целевую аудиторию и характер изданий, что требует от системы интерпретации пользовательского намерения. На первом этапе агент инициировал гибридный поиск (BM25 + векторная модель) по ключевым словам «психология», «широкий круг читателей», «популярные книги». В результате была получена подборка изданий, среди которых, в частности, выявлено учебное пособие «Социальная психология личности, общения и группы» (2024), в аннотации которого прямо указано, что издание предназначено для широкого круга читателей, интересующихся вопросами взаимодействия с людьми. Далее агент сформировал итоговый список рекомендаций, дополнив его изданиями по психологии труда и юридической психологии, доступными в электронной библиотеке СФУ.

Характерно, что агент не ограничился формальным совпадением по слову «психология», а учитывал контекст аннотаций и целевое назначение изданий. Это демонстрирует способность Agentic RAG интерпретировать семантические признаки аудитории и адаптировать стратегию отбора источников в соответствии с предполагаемым информационным намерением пользователя. В классическом RAG подобный запрос чаще приводил к доминированию формальных учебных изданий без учета целевой аудитории.

Для оценки работы классического и агентного вариантов RAG была сформирована тестовая коллекция библиотечных данных, представляющая собой подмножество электронного фонда университетской библиотеки.

Коллекция была отобрана специально для моделирования типичных поисковых задач, с которыми сталкиваются пользователи научных библиотек: тематический поиск, уточнение контекста, извлечение определений, цитат и сравнение понятий в смежных дисциплинах.

В основу набора данных легли 3248 наименования из университетского каталога, охватывающих гуманитарные, технические и естественно-научные дисциплины.

Для каждой записи в корпус были включены три уровня данных:

1. Библиографические метаданные – автор, название, ключевые слова, год издания и УДК-код.

2. Аннотация – краткое описание содержания книги или статьи (средний объем ~900 символов).

3. Фрагменты полного текста – выдержки из электронных версий изданий, доступных в репозитории в формате PDF и DOCX.

Таким образом, корпус объединяет как структурированные поля (метаданные), так и неструктурированный текст, что позволяет тестировать оба подхода RAG – векторный и полнотекстовый.

Для повышения чистоты эксперимента из коллекции были исключены тексты, содержащие менее 200 слов или дублирующие аннотации.

Подготовка и очистка данных. На этапе предобработки каждый документ прошел серию процедур:

– удаление HTML-разметки, символов форматирования и гиперссылок;

– нормализация регистра и унификация пунктуации;

– лемматизация русского текста с помощью библиотеки pymorphy2;

– токенизация и разбиение на логические фрагменты (чанки) размером ≈ 1 000 символов с перекрытием 100 символов для сохранения контекста между блоками;

– удаление повторяющихся чанков и контроль средней длины предложений.

Для аннотаций и метаданных чанки не создавались – каждая запись обрабатывалась как единый текстовый блок, что позволило сохранить цельность смысловой структуры краткого описания книги.

После очистки корпус составил:

– около 5,2 млн символов текста,

– 24 180 чанков,

– средний объем текста книги – ~4 100 символов.

Общий объем данных после индексации – 203 МБ.

Индексирование и хранилища. Для проведения экспериментов использовались два типа хранилищ, соответствующих двум стратегиям поиска:

1. Векторное хранилище (ChromaDB).

– Используемая модель эмбеддингов: sergeyzh/rubert-mini-frida.

– Индексировались аннотации и текстовые чанки отдельно.

– Для каждой книги сохранялись связи book_id ↔ chunk_id, что позволяло агрегировать фрагменты в контексте одного источника.

– Векторный поиск осуществлялся по косинусному сходству с выборкой top_k = 10.

2. Полнотекстовое хранилище (SQLite + BM25).

– Использовался встроенный механизм FTS5.

– В индекс включались поля title, keywords, annotation, text_chunk.

– Для каждого запроса рассчитывался рейтинг релевантности BM25 с порогом отсечения 0,3.

– Результаты сохранялись в промежуточную таблицу и далее направлялись в модуль реранжирования.

Такое раздельное хранение позволило агенту комбинировать оба источника в зависимости от контекста запроса – обращаться к ChromaDB для семантических запросов и к BM25 при поиске формальных терминов или цитат.

Обогащение и формирование обучающих пар. Для проверки качества поиска часть корпуса (около 120 записей) была вручную размечена экспертами-библиографами.

Каждому документу сопоставлялись релевантные запросы (в среднем 3-4 на запись) и оценка по шкале релевантности от 0 до 3:

– 0 – нерелевантный ответ;
– 1 – частично релевантный;
– 2 – релевантный по теме, но не по контексту;
– 3 – полностью релевантный.

Эта разметка использовалась не для обучения, а для объективной оценки точности retrieval-компоненты обеих систем.

Таким образом, часть корпуса стала своеобразным «тестовым стендом», по которому измерялись метрики Precision@k и Recall@k.

Для наглядности структура одной библиографической записи приведена в Таблице 1.

Таблица 1 – Структура типовой библиографической записи
Table 1 – The structure of a standard bibliographic record

Поле	Пример содержимого
book_id	345684
title	*Технические средства автоматизации и управления*
authors	В. Н. Жуков, А. А. Климов
keywords	автоматизация, геодезия, координаты
annotation	Рассматриваются основы автоматизации процессов измерения и систем координат в инженерных приложениях.
text_chunks	24 чанка, средний объем – 980 символов

Разделение на векторное и полнотекстовое хранилище позволило напрямую сравнивать стратегии поиска – статическую (в классическом RAG) и динамическую (в Agentic RAG), что стало ключевым элементом последующих экспериментов.

Модели и конфигурация поиска. Для обеспечения воспроизводимости результатов эксперимента конфигурация retrieval-компонент и моделей генерации была жестко зафиксирована. В качестве эмбеддинговой модели использовалась русскоязычная модель *sergeyzh/rubert-mini-frida*, обеспечивающая оптимальное соотношение скорости и точности при работе с научными и библиотечными текстами. Векторный поиск сочетался с полнотекстовым поиском BM25 в гибридной схеме при весах 0,6 (векторный канал) и 0,4 (лексический канал).

Параметры извлечения задавались симметрично для обоих вариантов RAG:

– количество возвращаемых кандидатов: $k = 5$ (стандартный режим) и $k = 15$ (расширенный режим),

– для полнотекстового поиска BM25: $k = 5 / 10$ соответственно,

– для SQL-доступа через FTS5: limit = 5 / 15.

Для генерации ответов и выполнения агентных рассуждений применялась локально запущенная языковая модель *Qwen 3* (*30B*) через провайдер llama.cpp. Взаимодействие с моделью осуществлялось локально через API-совместимый интерфейс по адресу http://localhost:8020/v1, что позволило полностью контролировать параметры генерации и исключить внешние сетевые задержки. Все параметры модели и пороговые значения поиска оставались неизменными на протяжении всех экспериментальных серий.

Для обеспечения детерминированного поведения агентной архитектуры в эксперименте использовался режим структурированного вывода (Structure Output) в JSON-формате. Все промежуточные решения агента – выбор инструмента, оценка релевантности, планирование дальнейших шагов – генерировались в заранее заданных структурах, что исключало неоднозначность при интерпретации ответов модели и обеспечивало воспроизводимость результатов. Форматы структур были фиксированы для всех запусков эксперимента.

Метрики оценки. Для оценки качества извлечения и релевантности ответов использовались четыре метрики: Precision@k, Recall@k, MRR и Expert Relevance Score (ERS).

Следует отметить, что шкала 0–3 использовалась исключительно для разметки релевантности документов на этапе оценки retrieval-компоненты, тогда как итоговый показатель Expert Relevance Score (ERS) рассчитывался по шкале 1–5 и применялся для оценки качества сформированного ответа.

Оценка проводилась на 100 запросах, одинаковых для классического и агентного RAG. Используемый набор поисковых запросов был сформирован на основе реальных информационных потребностей пользователей университетской библиотеки. Запросы составлялись совместно с профильными специалистами и включали три типа формулировок: тематические (40 %), фактографические (35 %) и контекстные, требующие уточнения исходного вопроса (25 %). Такая структура выборки отражает типовой набор обращений в научных библиотеках и обеспечивает репрезентативность оценки.

Количественные метрики. 1. Precision@k – доля релевантных документов среди топ-k результатов:

$$Precision@k = \frac{|\text{релевантные документы среди топ}-k|}{k}. \quad (1)$$

Использовались значения k = 3, 5, 10.

2. Recall@k – полнота поиска:

$$\text{Recall@k} = \frac{|\text{релевантные документы среди топ}-\text{k}|}{|\text{все релевантные документы}|}. \quad (2)$$

3. MRR (Mean Reciprocal Rank) – среднее обратное значение позиции первого релевантного результата:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}. \quad (3)$$

4. Expert Relevance Score (ERS) – экспертная оценка итогового ответа по шкале 1–5. Два библиотечных специалиста оценивали ответы независимо.

5. Normalized Discounted Cumulative Gain (nDCG@k) – оценивает убывание ценности документов по мере упорядочивания результата. Использовались значения k = 5.

6. Итоговая оценка – среднее арифметическое. Итоговая оценка рассчитывалась как среднее арифметическое значений используемых метрик.

Для обеспечения единообразия и воспроизводимости экспертной оценки качества ответов была сформирована шкала интерпретации баллов, используемая при расчёте показателя Expert Relevance Score (ERS). Описание уровней релевантности и соответствующих им значений приведено в Таблице 2.

Таблица 2 – Оценочная шкала экспертной релевантности итоговых ответов
Table 2 – Evaluation scale of expert relevance of final responses

Балл	Интерпретация
1	нерелевантный ответ
2	частично релевантный
3	релевантен по теме
4	релевантен по контексту
5	полностью релевантен

Условия измерений:

- количество запросов – 100;
- типы запросов – фактографические, тематические, контекстные;
- модели и параметры поиска идентичны в обоих вариантах;
- для каждого запроса вычислялись средние значения всех метрик.

Назначение и интерпретация метрик. Для систематизации используемых показателей качества и уточнения их функционального назначения в рамках экспериментального сравнения была составлена сводная таблица метрик оценки. Назначение каждой метрики и ее роль в анализе эффективности решений представлены в Таблице 3.

Таблица 3 – Назначение используемых метрик оценки
Table 3 – Purpose of the evaluation metrics

Метрика	Цель
Precision@k	оценка точности отбора
Recall@k	оценка полноты поиска
MRR	позиция первого релевантного документа
ERS	экспертная оценка смысловой релевантности

Набор метрик позволил провести прямое сравнение двух систем по точности поиска, полноте и качеству текстовых ответов.

Результаты представлены в разделе «Результаты».

Экспериментальная среда. Все вычислительные эксперименты проводились на выделенном сервере под управлением операционной системы Windows 10 Pro, оснащенном высокопроизводительной многоядерной архитектурой и дискретными графическими ускорителями. Конфигурация аппаратной платформы включала следующие компоненты:

Процессор:

2 × Intel Xeon Gold 5220R (24 физических ядра и 48 потоков каждый, тактовая частота 2.20 ГГц). Совокупно сервер предоставляет 48 физических ядер и 96 потоков,

что обеспечивает высокую степень параллелизма при обработке поисковых запросов и выполнении задач индексации.

Оперативная память:

256 ГБ DDR4. Такой объём RAM позволил хранить векторные индексы, промежуточные результаты поиска и кеши без использования подкачки, что исключило влияние IO-задержек на результаты экспериментов.

Графические ускорители:

Сервер оснащен двумя профессиональными видеокартами NVIDIA RTX A5000 (24 ГБ GDDR6 каждая). Параметры GPU по данным nvidia-smi:

– Драйвер: 561.17.

– Версия CUDA: 12.6.

– GPU0: 10 602 MiB задействовано.

– GPU1: 9 652 MiB задействовано.

Графические ускорители использовались для ускорения работы CrossEncoder-реренкеров и локального запуска LLM-модулей при тестировании Agentic RAG.

Дисковая подсистема:

Сервер имеет два NVMe-накопителя:

– SSD NVMe 1.92 ТБ.

– SSD NVMe 1.00 ТБ.

Высокая скорость чтения и записи обеспечила отсутствие дисковых задержек при индексации корпуса из 24 180 текстовых чанков.

Программное обеспечение:

– Python 3.10.

– ChromaDB 0.5+.

– SQLite FTS5.

– PyTorch с поддержкой CUDA 12.6.

– LM Studio (локальный запуск LLM).

– NVIDIA CUDA Toolkit 12.x.

На данном сервере были развернуты обе исследуемые модели – классический RAG и предложенный Agentic RAG. Все эксперименты выполнялись в идентичной среде с фиксированными версиями библиотек и детерминированными параметрами поиска (top-k, длина контекста, набор метрик). Это позволяет исключить влияние аппаратных различий на сравнение результатов.

Результаты

В этом разделе представлены результаты сопоставительного экспериментального исследования традиционной системы RAG и разработанной системы Agentic RAG для задач библиографического поиска. Исследование проводилось на корпусе из 3248 документов (24 180 текстовых чанков) и 100 специализированных запросов, сформулированных экспертами. Для оценки качества использованы стандартные метрики для RAG-систем, включая Precision@k, Recall@k, Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (nDCG) и экспертную оценку релевантности (ERS).

По результатам проведенного экспериментального сравнения были рассчитаны значения всех выбранных метрик для классического RAG и предложенного агентного варианта. Сводные результаты количественной и экспертной оценки эффективности обеих систем, полученные при идентичных условиях эксперимента, представлены в Таблице 4.

Таблица 4 – Результаты сравнительного анализа классического RAG и Agentic RAG по основным метрикам
Table 4 – Comparative results of classical RAG and Agentic RAG based on core evaluation metrics

Метрика	**Классический RAG**	**Agentic RAG**
Precision@5	0,46	0,61
Recall@5	0,49	0,68
MRR	0,39	0,54
nDCG@5	0,52	0,70
ERS	2,10	2,60

Из таблицы видно, что Agentic RAG существенно превосходит классическую систему по всем показателям. Прецизионность при k = 5 увеличилась с 0,46 до 0,61, а полнота – с 0,49 до 0,68. Ранговые показатели также значительно улучшились: MRR вырос с 0,39 до 0,54, а nDCG@5 – с 0,52 до 0,70. Экспертная оценка релевантности (ERS, шкала 1–5) поднялась с 2,1 до 2,6, что подтверждает более качественное ранжирование.

Обсуждение

Повышение качества связано с особенностями предложенной архитектуры: агентная модель выполняет итеративный поиск, планирует стратегию и проводит самооценку результатов, что позволяет избегать лишнего шума и улучшать покрытие запроса. Однако за увеличение точности приходится платить задержкой: среднее время ответа Agentic RAG составило 1200 мс, в то время как классическая система отвечала в среднем за 400 мс. Такое увеличение обусловлено необходимостью выполнять несколько итераций поиска и переоценки контекста.

Качественный анализ показал, что Agentic RAG особенно полезен для сложных тематических запросов, требующих последовательного уточнения контекста. Например, при поиске литературы о цифровой консервации архивных коллекций классический RAG извлекал поверхностные упоминания оцифровки, тогда как Agentic RAG после нескольких итераций находил статьи с детальными методологиями и примерами внедрения. В некоторых случаях агент возвращал избыточный контекст, но это можно минимизировать ограничением глубины поиска.

В части генерации ответов оценка показала, что Agentic RAG обеспечивает более высокую фактическую точность и полноту благодаря лучшему подбору источников. Это согласуется с данными из смежных областей, где агентные модели достигают высоких показателей (например, агентный CyberRAG для классификации кибератак получил точность 94,92 % и высокие оценки объяснений по BERTScore и экспертным рецензиям [13]. Таким образом, использование агентного подхода оправдано для информационных систем, ориентированных на научно-библиографический поиск.

Для наглядного сопоставления эффективности классического и агентного вариантов Retrieval-Augmented Generation по основным показателям качества была выполнена графическая визуализация результатов эксперимента. Сравнение значений метрик точности, полноты, ранжирования и экспертной релевантности представлено на Рисунке 4.

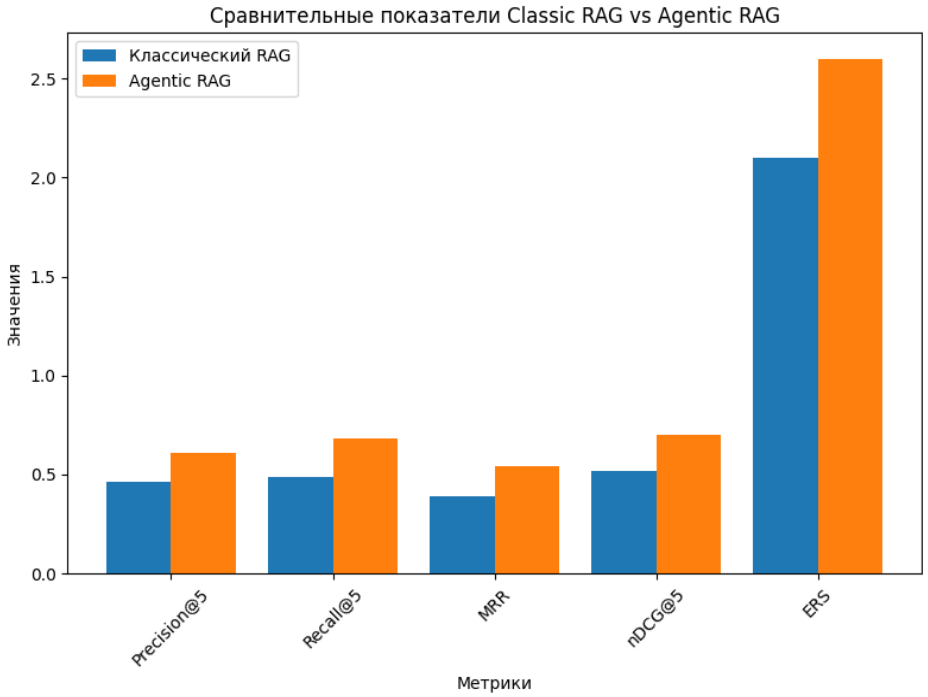

Рисунок 4 – Сравнение основных метрик извлечения для классического и авторского Agentic RAG

Figure 4 – Comparison of the main extraction metrics for classic and author's Agentic RAG

Комплексное сравнение показало, что Agentic RAG существенно улучшает эффективность и качество поиска в библиотечной среде по сравнению с классическим RAG. Увеличение задержки ответа компенсируется ростом точности, полноты и качества ранжирования. Поэтому агентный подход можно рекомендовать для внедрения в библиографические системы, ориентированные на высокую точность и глубину поиска.

Несмотря на полученные положительные результаты, проведенное исследование имеет ряд ограничений, которые необходимо учитывать при интерпретации выводов и планировании дальнейших работ. Во-первых, качество поиска в разработанных системах существенно зависит от характеристик используемых моделей представления текста – как эмбеддингов, так и ранжирующих моделей. В эксперименте применялись конкретные версии трансформерных моделей, доступные на момент исследования, что неизбежно ограничивает обобщаемость результатов. Возможные отличия в архитектуре, размере или настройках альтернативных моделей могут привести как к улучшению, так и к ухудшению показателей. Кроме того, размер доступного контекста и особенности токенизации могут влиять на полноту извлечения и способность системы корректно учитывать длинные или структурно сложные документы.

Ограничения накладываются и используемым корпусом данных. Эксперименты проводились на коллекции объемом 3248 документов, формирующей репрезентативную, но все-таки ограниченную выборку в рамках крупной университетской библиотеки. Несмотря на то, что документы были тщательно очищены и сегментированы, корпус не охватывает весь спектр дисциплин, типов публикаций и форматов источников, характерных для современных библиотечных фондов. Некоторые научные области могут быть представлены с различной плотностью, что способно влиять на результаты извлечения и ранжирования. Экспертная разметка релевантности, хоть и выполнялась специалистами, остается ограниченной по объему и неизбежно включает элементы субъективности.

Отдельного рассмотрения заслуживают ограничения применяемых метрик. Используемые в исследовании показатели – Precision@k, Recall@k, MRR, nDCG и экспертная оценка релевантности – эффективно описывают поведение систем с точки зрения извлечения, но не охватывают всех аспектов качества итогового ответа. В частности, метрики не фиксируют нюансы логической структуры ответа, устойчивость модели к неоднозначным запросам и способность правильно обрабатывать комплексные междисциплинарные вопросы. Кроме того, экспертная оценка ERS, несмотря на стандартизированные критерии, опирается на субъективное восприятие релевантности и может отличаться при расширении экспертной группы.

Ограничения касаются и временных характеристик систем. Агентная архитектура обеспечивает повышение качества извлечения, но требует большего числа итераций поиска, планирования и самооценки, что приводит к увеличению задержки ответа. Зафиксированное в эксперименте различие (в среднем 1200 мс у Agentic RAG против 400 мс у классического RAG) имеет значение для систем с высокими требованиями по времени отклика. При этом исследование не включало нагрузочного тестирования, поэтому влияние конкурентных запросов, параллельной индексации и долгосрочного роста фонда на стабильность задержек и производительность остается неизученным.

Наконец, экспериментальные сценарии охватывали преимущественно запросы, сформулированные экспертами и характерные для научно-образовательной среды. Хотя такой набор обеспечивает высокую содержательную ценность оценки, он не учитывает разнообразие реальных пользовательских запросов, включая неточные формулировки, смешанные информационные потребности, повествовательные запросы или задания, предполагающие глубокий контекст и последовательное уточнение цели. Кроме того, пользовательская оценка удобства взаимодействия с системой не проводилась – исследование сосредоточено на измеримых параметрах качества извлечения.

Перечисленные ограничения не умаляют значимости полученных результатов, но подчеркивают необходимость дальнейшего расширения экспериментальной базы, включения дополнительных метрик и сценариев, а также проверки масштабируемости и устойчивости Agentic RAG в условиях реальных библиотечных систем.

Заключение

В работе был рассмотрен практический подход к поиску по библиотечным коллекциям, основанный на агентной модификации RAG. Основное внимание уделялось тому, как такая архитектура ведет себя на реальных русскоязычных данных и насколько она способна компенсировать слабые места классического извлечения. На примере корпуса университетской библиотеки удалось показать, что использование двух различных механизмов поиска вместе со структурированным выводом и встроенной самооценкой дает системе больше свободы в выборе стратегии и позволяет ей лучше ориентироваться в неоднородных запросах. В таком виде архитектура, по сути, представляет собой новый вариант решения привычной задачи и ранее в библиотечном контексте не изучалась.

Полученные результаты подтверждают, что поставленная цель была выполнена. Agentic RAG уверенно обходит классический вариант по основным метрикам (Precision@5, Recall@5, nDCG@5, ERS), и преимущество особенно заметно там, где запрос сформулирован неидеально или требует дополнительного уточнения. Итеративная структура и возможность пересматривать промежуточные шаги играют здесь ключевую роль и позволяют системе исправлять собственные неточности.

Практическая значимость работы заключается в ее прямой применимости. Та архитектура, которая описана и протестирована, может быть внедрена в работу

университетских библиотек без существенной перестройки инфраструктуры. Повышение точности и полноты выдачи делает поиск понятнее и полезнее для пользователей, одновременно снижая нагрузку на сотрудников, которые отвечают за справочные запросы. По сути, речь идет о технологии, которую можно интегрировать в существующие сервисы и ожидать от нее ощутимого эффекта.

Вместе с тем остаются направления, которые требуют дальнейшей проработки. Перспективным выглядит включение полнотекстовых материалов, а также использование более современных моделей эмбеддингов. Возможно, дополнительные преимущества даст интеграция элементов графовых структур или специализированных инструментов тематической навигации. Агентный подход, конечно, не решает все проблемы поиска, но уже показывает, что может быть полезной основой для развития интеллектуальных библиотечных инструментов. В условиях роста объемов информации такие системы выглядят естественным следующим шагом, который позволяет улучшить качество поиска без увеличения нагрузки на персонал.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Каптерев А.И., Тикунова И.П. Отражение библиотечной проблематики в региональных стратегиях цифровой трансформации субъектов РФ. *Научные и технические библиотеки*. 2025;(3):161–180. https://doi.org/10.33186/1027-3689-2025-3-161-180
   Kapterev A.I., Tikunova I.P. The library agenda in the regional strategies of digital transformation of the RF entities. *Scientific and Technical Libraries*. 2025;(3):161–180. (In Russ.). https://doi.org/10.33186/1027-3689-2025-3-161-180
2. Тикунова И.П. Цифровизация как тренд библиотечного развития. *Труды ГПНТБ СО РАН*. 2021;(3):31–37. https://doi.org/10.20913/2618-7575-2021-3-31-37
   Tikunova I.P. Digitalization as a trend in library development. *Proceedings of SPSTL SB RAS*. 2021;(3):31–37. (In Russ.). https://doi.org/10.20913/2618-7575-2021-3-31-37
3. Bevara R.V.K., Lund B.D., Mannuru N.R., et al. Prospects of Retrieval Augmented Generation (RAG) for Academic Library Search and Retrieval. *Information Technology and Libraries*. 2025;44(2). https://doi.org/10.5860/ital.v44i2.17361
4. Singh A., Ehtesham A., Kumar S., Khoei T.T. Agentic retrieval-augmented generation: A survey on agentic RAG. arXiv. URL: https://doi.org/10.48550/arXiv.2501.09136 [Accessed 16th November 2025].
5. Lewis P., Perez E., Piktus A., et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv. URL: https://doi.org/10.48550/arXiv.2005.11401 [Accessed 16th November 2025].
6. Gao Y., Xiong Y., Gao X., et al. Retrieval-augmented generation for large language models: A survey. arXiv. URL: https://doi.org/10.48550/arXiv.2312.10997 [Accessed 26th November 2025].
7. Aytar A.Y., Kaya K., Kilic K. A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science. arXiv. URL: https://arxiv.org/html/2412.15404v1 [Accessed 26th November 2025].
8. Barnett S., Kurniawan S., Thudumu S., Brannelly Z., Abdelrazek M. Seven Failure Points When Engineering a Retrieval Augmented Generation System. In: *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN '24), 14–15 April 2024, Lisbon, Portugal.* New York: Association for Computing Machinery; 2024. P. 194–199. https://doi.org/10.1145/3644815.3644945
9. Mazumder J., Mukhopadhyay P. Designing Question-Answer Based Search System in Libraries: Application of Open Source Retrieval Augmented Generation (RAG) Pipeline.

Journal of Information and Knowledge. 2024;61(5):255–260. https://doi.org/10.17821/srels/2024/v61i5/171583

10. Schneider F., Ahmadi N.B., Ahmadi N.B., et al. CollEX – A Multimodal Agentic RAG System Enabling Interactive Exploration of Scientific Collections. In: *Proceedings of the 1st Workshop on Multimodal Augmented Generation via Multimodal Retrieval (MAGMaR 2025), 01 August 2025, Vienna, Austria*. Association for Computational Linguistics; 2025. P. 18–39. https://doi.org/10.18653/v1/2025.magmar-1.2
11. Nagori A., Casonatto R.A., Gautam A., Cheruvu A.M.S., Kamaleswaran R. Open-Source Agentic Hybrid RAG Framework for Scientific Literature Review. arXiv. URL: https://arxiv.org/abs/2508.05660 [Accessed 29th November 2025].
12. Chu Zh., Wang Sh., Xie J., et al. LLM agents for education: Advances and applications. arXiv. URL: https://arxiv.org/abs/2503.11733 [Accessed 30th November 2025].
13. Maheshwari H., Tenneti S., Nakkiran A. CiteFix: Enhancing RAG Accuracy Through Post-Processing Citation Correction. arXiv. URL: https://doi.org/10.48550/arXiv.2504.15629 [Accessed 21st November 2025].

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Рзянкин Илья Сергеевич, ведущий инженер-программист Научной библиотеки Сибирского федерального университета, Красноярск, Российская Федерация.
e-mail: i-rzyankin@yandex.ru

Ilya S. Rzyankin, Lead Software Engineer, Scientific Library of Siberian Federal University, Krasnoyarsk, the Russian Federation.

Барышев Руслан Александрович, кандидат философских наук, доцент Сибирского федерального университета, Красноярск, Российская Федерация.
e-mail: r_baryshev@bk.ru
ORCID: 0000-0002-4383-2830

Ruslan A. Baryshev, Candidate of Philosophical Sciences, Associate Professor, Siberian Federal University, Krasnoyarsk, the Russian Federation.

Гучко Алексей Андреевич, независимый исследователь, Красноярск, Российская Федерация.
e-mail: against61@gmail.com

Aleksey A. Guchko, Independent Researcher, Krasnoyarsk, the Russian Federation.

Статья поступила в редакцию 28.01.2026; одобрена после рецензирования 12.02.2026; принята к публикации 18.02.2026.

The article was submitted 28.01.2026; approved after reviewing 12.02.2026; accepted for publication 18.02.2026.