

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории упругости и биомеханики

Разработка RAG-системы поддержки пользователей

цифровых ресурсов и сервисов СГУ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы

направления 09.03.03 – Прикладная информатика

механико-математического факультета

Беззубова Андрея Владимировича

Научный руководитель
доцент, к.ю.н.

Р.В. Амелин

Зав. кафедрой
зав. кафедрой, д.ф.-м.н., профессор

Л.Ю. Коссович

Саратов 2025

Введение. Современные университеты предоставляют доступ к множеству цифровых ресурсов и сервисов, необходимых для обучения, административного взаимодействия и научной деятельности. Однако расширение этой экосистемы сопровождается ростом сложности и количества обращений от студентов и сотрудников. Нехватка единого, интуитивного интерфейса, сложные формулировки в документах, разрозненные источники информации — всё это снижает эффективность взаимодействия с университетской цифровой средой и создаёт дополнительную нагрузку на службы поддержки.

Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского (СГУ) обладает развитой цифровой инфраструктурой, включающей: официальный сайт, образовательный портал "Система дистанционного обучения Ipsilon Uni", систему создания и управления курсами Moodle, электронную библиотеку и другие административные сервисы. Однако в условиях постоянного обновления документов, отсутствия единого поиска по системам и ограниченности ресурсов поддержки, возникает потребность в интеллектуальной системе, способной оперативно и точно отвечать на вопросы пользователей, опираясь на внутренние документы вуза.

С увеличением числа студентов, ростом объёма внутренней документации и переходом к смешанным форматам обучения нагрузка на административные органы университетов постоянно возрастает. В таких условиях автоматизация информационного сопровождения становится не просто инструментом повышения удобства, а необходимым элементом эффективной работы образовательной инфраструктуры. Интеллектуальные системы поддержки позволяют сократить время ответа на типовые запросы, разгрузить административный персонал и повысить удовлетворённость пользователей.

Одним из наиболее перспективных подходов для реализации такой системы является Retrieval-Augmented Generation (RAG) — архитектура, сочетающая поиск по базе знаний с генерацией ответов на естественном языке. Метод RAG позволяет формировать осмысленные ответы, подтверждённые источниками, что особенно актуально в образовательной и административной среде.

Цель бакалаврской работы — разработка интеллектуальной RAG-системы поддержки пользователей, способной отвечать на вопросы на основе

документов СГУ, и оценка её эффективности в контексте цифровой инфраструктуры вуза.

Задачи работы:

- проанализировать цифровые ресурсы СГУ и выделить типичные сценарии обращений пользователей;
- изучить существующие подходы к автоматической поддержке и выявить их ограничения;
- сформулировать требования к интеллектуальной системе поддержки;
- спроектировать архитектуру RAG-системы;
- реализовать прототип системы с использованием современных технологий обработки естественного языка;
- провести тестирование системы и оценить её точность, стабильность и практическую применимость.

Структура работы и объём работы. Работа состоит из введения, четырёх разделов, заключения, четырёх приложений и содержит 64 страницы. Список использованных источников включает 20 наименований.

Раздел 1. Анализ предметной области.

Раздел 2. Проектирование архитектуры RAG-системы

Раздел 3. Реализация RAG-системы

Раздел 4. Тестирование и оценка качества

Основное содержание работы. Работа посвящена разработке интеллектуальной RAG-системы поддержки пользователей цифровых и информационных сервисов СГУ.

Во введении бакалаврской работы описываются особенности цифровой инфраструктуры современных университетов и связанные с этим проблемы взаимодействия пользователей с цифровыми сервисами. Обоснована актуальность темы исследования, поставлена цель работы и сформулированы задачи.

В первом разделе рассматриваются особенности цифровой инфраструктуры СГУ и основные проблемы пользователей при взаимодействии с ней, обоснована актуальность применения метода Retrieval-Augmented Generation (RAG) для создания интеллектуальной системы поддержки, способной отвечать на вопросы на естественном языке на основе внутренних

документов вуза.

Цифровизация университетов сопровождается значительным ростом числа сервисов: от электронного расписания до систем подачи заявлений и документооборота. В СГУ эта цифровая инфраструктура включает официальный сайт, образовательный портал "Система дистанционного обучения Ipsilon Uni", систему создания и управления курсами Moodle, электронную библиотеку и другие административные сервисы.

Несмотря на разнообразие и доступность сервисов, пользователи сталкиваются с рядом затруднений:

- отсутствием единой точки входа к информации;
- сложностью поиска и навигации по различным платформам;
- разрозненностью и несогласованностью нормативных данных;
- отсутствием интерактивной обратной связи;
- перегрузкой сотрудников справочных служб.

Среди типичных обращений — вопросы вроде: «Где найти расписание?», «Как взять академический отпуск?», «Какие требования к оформлению диплома?». Большинство ответов содержатся в регламентирующих документах, но они либо труднодоступны, либо не адаптированы под пользовательский язык запросов.

Классические решения — разделы FAQ, статические инструкции и чат-боты со сценариями — оказываются недостаточно гибкими. Они не учитывают контекст запроса, не умеют работать с естественным языком и не поддерживают обращение к внутренним документам. Полнотекстовый поиск по сайту также часто неэффективен из-за формального языка документов.

Традиционные методы поддержки пользователей в вузах — такие как разделы FAQ, статические инструкции и сценарные чат-боты — имеют ряд очевидных преимуществ: простота внедрения, низкая стоимость и минимальные требования к поддержке. Однако их эффективность сильно ограничена. Статичные FAQ быстро устаревают, не учитывают изменения в регламентах и не адаптируются к естественным формулировкам запросов. Скриптовые чат-боты, основанные на ключевых словах и жёстких сценариях, плохо справляются с нетипичными вопросами и не умеют работать с внутренними документами. Полнотекстовый поиск по сайтам часто не даёт нужных ре-

зультатов из-за расхождений между языком запроса и официальными формулировками в документах.

Таким образом, существующие решения обеспечивают лишь базовый уровень поддержки и не решают проблему контекстного поиска и интерпретации запроса на естественном языке. В условиях постоянно обновляющихся информационных ресурсов и высокой нагрузки на службы поддержки университетов требуется более интеллектуальная система, способная сочетать точность поиска с гибкостью генерации ответов.

Важно подчеркнуть, что без интеллектуальной поддержки пользователи зачастую вынуждены самостоятельно разбираться в сложной структуре внутренних регламентов. Это увеличивает время поиска информации, снижает оперативность принятия решений и может приводить к ошибкам в оформлении документов.

Современные интеллектуальные подходы используют трансформерные модели (например, BERT, GPT), которые позволяют анализировать смысл запроса. Однако без доступа к базе знаний такие модели склонны к генерации неточных или вымышленных ответов.

Метод Retrieval-Augmented Generation (RAG) устраняет основные недостатки классических решений, объединяя точный поиск информации с генерацией осмысленных ответов на естественном языке. Вместо жестких сценариев или поиска по ключевым словам RAG сначала находит релевантные фрагменты документов, а затем формирует ответ, опираясь только на найденные данные. Это позволяет давать обоснованные ответы, сохранять ссылку на источник и минимизировать риск ошибок. В университетской среде, где важно точное следование регламентам и быстрый доступ к актуальной информации, архитектура RAG оказывается наиболее эффективной: она повышает качество поддержки пользователей без увеличения нагрузки на административные органы.

В вузовской среде RAG особенно актуален — он способен автоматически отвечать на вопросы, опираясь на регламенты, приказы и инструкции, что делает его ценным инструментом цифровой поддержки.

Таким образом, анализ цифровой инфраструктуры СГУ и существующих подходов к поддержке пользователей подтвердил необходимость созда-

ния интеллектуальной системы, способной работать с естественными языковыми запросами и внутренними документами университета.

Второй раздел работы посвящен проектированию архитектуры RAG-системы.

Retrieval-Augmented Generation (RAG) — это подход, сочетающий поиск по базе знаний с генерацией ответа языковой моделью. Такой метод позволяет формировать точные и проверяемые ответы на основе конкретных фрагментов документов, что особенно важно в образовательной среде.

Архитектура проектируемой системы учитывает требования университета к достоверности, адаптивности и масштабируемости. Она построена по модульному принципу, где каждый компонент выполняет строго определённую функцию. Общая схема включает:

- **интерфейс взаимодействия с пользователем (API)** — отвечает за приём запросов, кеширование, маршрутизацию и возврат результата;
- **поисковый модуль (Retriever)** — сочетает классический полнотекстовый поиск и векторный поиск. Это обеспечивает как точное, так и семантическое совпадение запросов с документами;
- **генератор ответов (LLM)** — языковая модель, принимающая на вход релевантные фрагменты и исходный вопрос. Сформулированный ответ основан только на найденной информации;
- **индексатор документов** — обрабатывает документы в различных форматах. Разбивает их на фрагменты, очищает текст, извлекает метаданные и формирует векторные представления;
- **хранилище векторов (Vector Store)** — хранит эмбединги (векторные представления данных) и обеспечивает быстрый доступ к ним при поиске;
- **фронтенд-интерфейс** — прототип, созданный для тестирования системы. Позволяет отправлять запросы, просматривать ответы и источники.

Ключевыми принципами архитектуры стали:

- **модульность** — независимость компонентов облегчает отладку, обновление и масштабирование;
- **прозрачность** — каждый ответ сопровождается указанием источников;

- **гибкость** — возможность переиндексации документов без остановки сервиса;

- **безопасность и мониторинг** — базовые механизмы логирования, контроля ошибок и ограничения нагрузки.

Процесс обработки запроса в системе построен по следующей схеме: пользовательский запрос передаётся через API в модуль поиска, который извлекает релевантные фрагменты из базы знаний. Далее отобранные документы передаются генератору ответов, который формирует итоговый текст с ссылкой на источник. Сформированный ответ возвращается пользователю через API-интерфейс. Такая цепочка обеспечивает модульность и позволяет легко масштабировать каждый компонент системы независимо.

При проектировании особое внимание уделялось поддержке горизонтального масштабирования. Архитектура позволяет увеличивать производительность системы за счёт добавления дополнительных узлов поиска или генерации без необходимости полной переработки существующих компонентов.

Важно отметить, что использование модульного подхода облегчает также процессы отладки и обновления системы. При необходимости можно вносить изменения в отдельные компоненты (например, заменять поисковый модуль на более эффективный) без риска нарушить работу всей системы, что особенно ценно в условиях ограниченных ресурсов университетских ИТ-служб.

При проектировании RAG-системы особое внимание было уделено выбору технологий, обеспечивающих поддержку русского языка, открытость, стабильность и лёгкость интеграции. В качестве основной языковой модели выбрана Vikhr-YandexGPT-5-Lite-8B-it, благодаря её способности работать с русскоязычными запросами и встроенной поддержке двухэтапной фильтрации документов. Для построения векторного поиска использована модель multilingual-e5-large, хорошо справляющаяся с сопоставлением смыслов в русскоязычных текстах. В качестве хранилища эмбедингов выбрана ChromaDB за счёт удобной работы с метаданными и возможности динамического обновления индекса без остановки сервиса. Связующим элементом системы стал фреймворк LangChain, позволивший объединить поиск, генерацию ответов и работу с базой знаний в единую структуру. Бэкенд построен на FastAPI, что

обеспечило высокую скорость работы и лёгкость масштабирования.

Также была учтена безопасность работы системы. Все процессы обработки данных происходят локально, без обращения к внешним API, что исключает риски утечки информации и обеспечивает соответствие требованиям внутренней ИТ-безопасности университета.

В результате проектирования была сформирована архитектура RAG-системы, обеспечивающая модульность, прозрачность работы, масштабируемость и удобную интеграцию в существующую цифровую среду университета.

В третьем разделе описана разработка RAG-системы с использованием современных open-source инструментов и фреймворков.

Реализация интеллектуальной RAG-системы поддержки пользователей проводилась с использованием современных библиотек и фреймворков: `transformers`, `langchain`, `ChromaDB`, `FastAPI`, `Streamlit`. Все компоненты системы объединены в REST-сервис с модульной структурой.

Документы, служащие источником данных, поступают в формате PDF, DOCX и TXT. В процессе индексации из них извлекается текст, проводится его очистка и разбиение на логические фрагменты. Для каждого фрагмента вычисляется эмбединг на основе модели `multilingual-e5-large`, после чего данные сохраняются в векторное хранилище `ChromaDB`.

Особое внимание было уделено обработке текстов с сохранением структуры документов. Для файлов PDF использовалась библиотека `PyMuPDF` с постраничной разбивкой текста, что позволило учитывать расположение информации. Для файлов DOCX применялась библиотека `python-docx` с сохранением заголовков и форматирования, что облегчает дальнейший поиск и формирование ответов. Предобработка данных включала удаление лишних пробелов, исправление разрывов строк и нормализацию кодировок.

Дополнительно при разработке была предусмотрена возможность расширения системы за счёт подключения новых источников данных без необходимости полной переиндексации существующей базы знаний. Это позволяет гибко адаптировать систему под изменяющиеся потребности университета.

Поисковый модуль реализует гибридный подход, при котором особое внимание уделялось балансировке между скоростью поиска и точностью

результатов. Векторный поиск использовался для обработки семантически сложных запросов, тогда как полнотекстовый BM25-поиск позволял точно находить документы при совпадении формулировок. Комбинирование этих методов обеспечивало высокую релевантность при различных типах пользовательских вопросов.

Генерация ответов осуществляется языковой моделью `Vikhr-YandexGPT-5-Lite-8B-it`, которой на вход передаются пользовательский запрос и отобранные фрагменты. Специально подобранные промпты позволяют формировать краткие, обоснованные ответы с ссылкой на источник. Если система не находит релевантной информации, она сообщает об этом пользователю без дальнейшей генерации недостоверных данных.

Взаимодействие между компонентами организовано через FastAPI. Реализованы эндпоинты для приёма запросов, получения ответа, запуска переиндексации и мониторинга состояния системы. Для тестирования разработан простой веб-интерфейс на Streamlit, обеспечивающий наглядную демонстрацию работы.

Для повышения эффективности генерации ответов и предотвращения перегрузки языковой модели был реализован механизм ограничения объема текста, передаваемого на вход генератору. Из релевантных фрагментов формируется ограниченная по размеру выборка, максимально соответствующая запросу пользователя. Это позволяет сохранять качество генерации и стабильное время отклика.

Для оптимизации работы системы и снижения времени отклика был реализован механизм кэширования результатов запросов. Это позволяет значительно ускорить выдачу ответов на повторные или схожие запросы, а также снижает нагрузку на генеративную модель и векторную базу данных.

Отдельное внимание при разработке уделялось обеспечению отказоустойчивости сервиса. Были реализованы механизмы обработки ошибок, перезапуска процессов при сбоях и журналирования основных событий, что повышает надёжность работы системы в условиях реальной эксплуатации.

Таким образом, был создан полнофункциональный прототип системы, готовый к внедрению в реальных условиях и дальнейшему расширению. Он

демонстрирует практическую применимость архитектуры RAG для задач информационной поддержки в вузовской среде.

Четвертый раздел посвящен описанию проверки работоспособности разработанного продукта на основе реальных пользовательских запросов .

Для оценки работоспособности и эффективности RAG-системы было проведено комплексное тестирование, включающее проверку качества ответов, времени отклика и устойчивости при многократных запросах. Использовались реальные сценарии, характерные для пользователей СГУ: поиск информации о порядке оформления документов, регламентных сроках, действиях в нестандартных ситуациях.

Проверка проводилась с использованием выборки из 20 пользовательских запросов. Оценка релевантности показала, что 95% ответов точно соответствовали содержанию университетских документов. Полнота охвата составила 80%. Выявленные ограничения касаются главным образом недостатка данных в предоставленных системе документах.

Среднее время генерации ответа составило от 10,3 секунды. При наличии закешированных результатов оно снижалось до 0,05 секунд. Работа системы оставалась стабильной при последовательной подаче запросов, утечек памяти и критических сбоев не зафиксировано.

В целом, результаты тестирования подтверждают, что система успешно решает поставленные задачи — она понимает естественный язык запросов, извлекает релевантную информацию, формирует обоснованные ответы и корректно сообщает об отсутствии данных. Эти характеристики делают её пригодной для использования в учебной и административной среде вуза.

Заключение. Бакалаврская работа была посвящена разработке интеллектуальной RAG-системы поддержки пользователей цифровых ресурсов и сервисов СГУ. Проведённое исследование подтвердило актуальность задачи автоматизации информационного сопровождения в вузовской среде, где пользователи часто сталкиваются с труднодоступной или неструктурированной информацией.

В ходе работы был реализован прототип системы, способный воспринимать естественно-языковые запросы, находить релевантные фрагменты в университетских документах и формировать обоснованные ответы с указа-

нием источника. Система построена на современной архитектуре Retrieval-Augmented Generation и использует гибридный подход к поиску, совмещая точность и гибкость.

Результаты тестирования показали высокую точность, стабильную работу и достаточную скорость отклика для практического применения. Разработка опирается на открытые технологии и может быть интегрирована в существующую цифровую инфраструктуру университета.

Практическая значимость работы заключается в возможности снизить нагрузку на административные органы, ускорить доступ к регламентированной информации и повысить удовлетворённость пользователей. Предложенное решение может быть масштабировано и адаптировано под нужды любых организаций с развитой внутренней документацией.

Таким образом, созданный прототип RAG-системы уже на текущем этапе представляет собой полноценное решение для автоматизации поддержки пользователей цифровой инфраструктуры университета. Он может стать основой для дальнейших исследований в области применения технологий обработки естественного языка в образовательных организациях.

В перспективе возможно расширение функциональности системы за счёт добавления поддержки диалоговых сценариев, что позволит пользователям уточнять вопросы в процессе общения. Также представляют интерес развитие мультязычной поддержки, интеграция с внутренними сервисами вуза для автоматического получения актуальных данных и улучшение обработки сложных документов с вложенной структурой. Эти направления делают систему ещё более полезной для студентов и сотрудников университета, обеспечивая быстрый и интуитивный доступ к необходимой информации.