

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО ДИАЛОГОВОГО АССИСТЕНТА С ДИНАМИЧЕСКИМ ВЫБОРОМ НЕЙРОСЕТЕВЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ И МЕХАНИЗМОМ ИЗВЛЕЧЕНИЯ ЗНАНИЙ

Н. А. Полковникова

ФГБОУ ВО «Государственный морской университет имени адмирала Ф.Ф. Ушакова»,
Новороссийск, Российская Федерация

Статья посвящена разработке интеллектуального диалогового ассистента для поддержки пользователей при получении и уточнении расписания учебных занятий. Рассмотрена эволюция диалоговых систем — от ранних правило-ориентированных текстовых диалоговых программ, основанных на шаблонном сопоставлении входных высказываний до современных виртуальных ассистентов, реализованных на основе нейросетевых трансформерных архитектур и больших языковых моделей. Актуальность исследования обусловлена применением больших языковых моделей в сочетании с технологией извлечения контекстных знаний (Retrieval-Augmented Generation, RAG), позволяющей формировать ответы на основе извлекаемой из внешних источников информации. Разработана архитектура интеллектуального диалогового ассистента, интегрирующая механизм RAG, динамическую маршрутизацию между несколькими нейросетевыми языковыми моделями и автоматическое обновление базы знаний посредством парсинга веб-ресурсов Государственного морского университета имени адмирала Ф. Ф. Ушакова. Предложенный подход обеспечивает устойчивую обработку вариативных пользовательских запросов и генерацию ответов на основе извлеченного контекста. В отличие от существующих решений, использующих фиксированную языковую модель, предложенная архитектура реализует адаптивный выбор генеративной модели в зависимости от типа пользовательского запроса. Разработан метод доменно-ориентированного формирования контекста для обработки динамически обновляемых табличных данных расписания, включающий специализированный парсинг веб-страниц, семантическое структурирование извлекаемой информации и правила промпт-инжиниринга. Для повышения фактологической достоверности ответов предложен механизм ограничения генерации рамками извлеченных данных и терминологической фильтрации. Реализован ETL-конвейер обновления базы знаний, обеспечивающий автоматическую актуализацию информации о расписании учебных занятий. Проведено сравнительное исследование эффективности языковых моделей Qwen, GigaChat и DeepSeek при обработке пользовательских запросов в рассматриваемом предметном домене. Полученные результаты демонстрируют повышение точности ответов и подтверждают эффективность предложенной архитектуры диалогового ассистента.

Ключевые слова: диалоговый ассистент, чат-бот, виртуальный цифровой ассистент, большая языковая модель, нейросетевой трансформер, механизм извлечения знаний, извлечение контекстных знаний.

Для цитирования:

Полковникова Н. А. Разработка интеллектуального диалогового ассистента с динамическим выбором нейросетевых языковых моделей и механизмом извлечения знаний / Н. А. Полковникова // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. 2026. Т. 18. № 3. С. 538–557. EDN ZVQCVX. DOI: 10.21821/2309-5180-2026-18-3-538-557.

DEVELOPMENT OF AN INTELLIGENT CONVERSATIONAL ASSISTANT WITH DYNAMIC SELECTION OF NEURAL LANGUAGE MODELS AND A KNOWLEDGE RETRIEVAL MECHANISM

N. A. Polkovnikova

Admiral Ushakov State Maritime University, Novorossiysk,
Novorossiysk, Russian Federation

This paper presents the development of an intelligent conversational assistant designed to support users in obtaining and clarifying information about university class schedules. The evolution of dialogue systems

is considered, from early rule-based text dialogue programs based on pattern matching of input utterances to modern virtual assistants implemented using neural transformer architectures and large language models. The relevance of the study is determined by the use of large language models in combination with Retrieval-Augmented Generation (RAG), which makes it possible to generate responses based on information retrieved from external sources. An architecture of an intelligent conversational assistant has been developed that integrates a RAG mechanism, dynamic routing between several neural language models, and automatic updating of the knowledge base through parsing Admiral F. F. Ushakov State Maritime University web resources. The proposed approach ensures robust processing of variable user queries and response generation based on the retrieved context. Unlike existing solutions that rely on a fixed language model, the proposed architecture implements adaptive selection of a generative model depending on the type of user query. A method for domain-oriented context construction for processing dynamically updated tabular schedule data has been developed, including specialized web page parsing, semantic structuring of the extracted information, and prompt engineering rules. To improve the factual reliability of responses, a mechanism is proposed that constrains generation to the retrieved data and applies terminological filtering. An ETL pipeline for updating the knowledge base has been implemented, enabling automatic updating of information about university class schedules. A comparative study of the effectiveness of the Qwen, GigaChat, and DeepSeek language models in processing user queries within the considered subject domain has been conducted. The obtained results demonstrate improved response accuracy and confirm the effectiveness of the proposed conversational assistant architecture.

Keywords: conversational assistant; chatbot; virtual digital assistant; large language model; neural transformer architecture; knowledge retrieval mechanism; retrieval-augmented generation.

For citation:

Polkovnikova, Natalia A. "Development of an intelligent conversational assistant with dynamic selection of neural language models and a knowledge retrieval mechanism." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 18.3 (2026): 538–557. DOI: 10.21821/2309-5180-2026-18-3-538-557.

Введение (Introduction)

Актуальность исследования обусловлена фундаментальным сдвигом в парадигме цифровых коммуникаций: переход от жестко детерминированных алгоритмов к адаптивным диалоговым интерфейсам становится ключевым вектором развития интеллектуальных диалоговых систем. В настоящее время диалоговые системы трансформируются в полноценных интеллектуальных ассистентов, интегрируемых в критически важные бизнес-процессы. Вместе с тем задача разработки эффективных, эмпатичных и одновременно точных, контекстно-ориентированных, фактологически достоверных виртуальных цифровых ассистентов остается далекой от окончательного решения. В частности, распространенной проблемой является снижение качества обработки пользовательских обращений при отклонении формулировок от предусмотренных шаблонов и сценариев.

На современном этапе развития технологий искусственного интеллекта наблюдается противоречие между высокими когнитивными способностями больших языковых моделей (англ. Large Language Models (LLM)) и требованиями к фактической точности ответов в предметных областях с динамически обновляемыми данными [1–5]. Несмотря на значительный прогресс больших языковых моделей, их практическое применение в специализированных предметных областях ограничивается феноменом генерации недостоверной информации (англ. Hallucination), отсутствием механизмов оперативного доступа к актуальным данным и высокой зависимостью качества ответа от формулировки пользовательского запроса.

Указанные ограничения препятствуют использованию автономных LLM в задачах информационного сопровождения образовательных процессов и требуют разработки новых архитектурных подходов к обеспечению фактологической достоверности генерируемых ответов. В этом контексте особую значимость приобретает разработка диалоговых ассистентов на базе гибридной архитектуры, объединяющих генеративный потенциал нейросетевых трансформеров с технологиями семантического поиска и извлечения контекста (англ. Retrieval-Augmented Generation (RAG)).

Особую актуальность данная задача приобретает в образовательной сфере, где пользователям требуется оперативный доступ к постоянно обновляемой информации о расписании занятий, аудиториях, преподавателях и изменениях учебного процесса. Использование традиционных

чат-ботов в образовательной сфере затрудняет обработку запросов, сформулированных в произвольной форме, что снижает удобство взаимодействия пользователей с информационными системами.

Архитектуры RAG позволяют повысить актуальность и фактологическую достоверность ответов, однако эффективность таких систем в существенной степени зависит от характеристик используемой языковой модели и вычислительных ресурсов, необходимых для генерации ответа. Дополнительный интерес представляет применение механизмов динамического выбора языковых моделей, позволяющих адаптировать вычислительные ресурсы системы к сложности пользовательского запроса, обеспечивая баланс между качеством ответа, временем отклика и вычислительными затратами.

Целью работы является повышение фактологической достоверности и актуальности ответов интеллектуальных диалоговых систем посредством разработки и реализации гибридного диалогового ассистента на основе механизма извлечения контекстных знаний и динамического выбора языковых моделей, для поддержки пользователей в предметной области расписания учебных занятий.

Для достижения поставленной цели предложена архитектура интеллектуального диалогового ассистента, объединяющая механизм извлечения контекстных знаний и модуль динамического выбора языковых моделей. Эффективность предложенного решения подтверждена результатами экспериментальной оценки данных предметной области расписания учебных занятий.

Методы и материалы (Methods and Materials)

Для реализации диалогового ассистента и обоснованного выбора технических средств и подходов выполнен анализ эволюции диалоговых систем: отправной точкой развития принято считать работу А. М. Тьюринга «Computing Machinery and Intelligence» (1950 г.), в которой был предложен тест, оценивающий способность машины имитировать человеческий диалог. В 1960-е гг. появились первые прототипы текстовых собеседников ELIZA Дж. Вейценбаума (1966 г.) и PARRY К. Колби (1972 г.), основанные на правилах и шаблонах. Эти системы демонстрировали принципиальную техническую возможность имитации диалога, но оставались жестко сценарными и крайне ограниченными по лингвистическому и предметному охвату. В 1980–1990-е гг. получили распространение правило- и скриптов-ориентированные чат-боты (например, A.L.I.C.E. Р. Уоллеса, 1995 г.), использующие языки шаблонов и иерархии правил. Они легли в основу многих ранних коммерческих решений, но масштабирование таких систем требовало ручной разработки большого числа правил, а гибкость и «естественность» диалога оставались низкими. На этом этапе чат-боты рассматривались преимущественно как вспомогательные интерфейсы к простым сервисам. С начала 2000-х гг. под влиянием развития статистического обучения и обработки естественного языка диалоговые системы эволюционируют в сторону так называемых «конвейерных» (pipeline) архитектур: с модулями понимания намерений (Intent Recognition), извлечения сущностей (NER) и управления диалогом.

Появление голосовых ассистентов (Apple Siri с 2011 г., а затем Amazon Alexa, Google Assistant) делает диалоговые интерфейсы массовыми, но они по-прежнему жестко связаны с заранее определенным набором сценариев и намерений, а расширение функционала требует значительных инженерных усилий. На рис. 1 представлена структурная схема эволюции диалоговых ассистентов (чат-ботов) за период 1950–2026 гг.: от продукционных правил до больших языковых моделей и мультимодальных агентов. На схеме выделены следующие этапы: эра правил и ранних концепций (1950–1979 гг.), эра статистических моделей и экспертных систем (1980–2009 гг.), эра глубокого обучения и нейросетевых подходов (2010–2018 гг.), а также эра больших языковых моделей (LLM) и мультимодальности (2019–2026 гг.).

Фундаментальный сдвиг в развитии технологий диалоговых ассистентов связан с внедрением глубоких нейронных сетей, в частности архитектуры Sequence-to-Sequence (seq2seq), обеспечившей возможность прямой генерации ответов в виде связных лексических последовательностей (последовательности слов). Однако первые нейросетевые диалоговые модели имели ограниченный контекст, тривиальные ответы и были пригодны преимущественно для исследовательских прототипов. В 2017 г. произошел фазовый сдвиг с появлением нейросетевой архитектуры трансформера

и парадигмы предобученных языковых моделей большого масштаба. В 2018 г. появились модели BERT, GPT-1, а затем GPT-2 и GPT-3 было продемонстрировано, что единая модель предобученная на больших объемах данных (Common Crawl, The Pile, RedPajama, FineWeb и др.) способна решать широкий спектр языковых задач без специализированного обучения для каждой задачи [6–9].

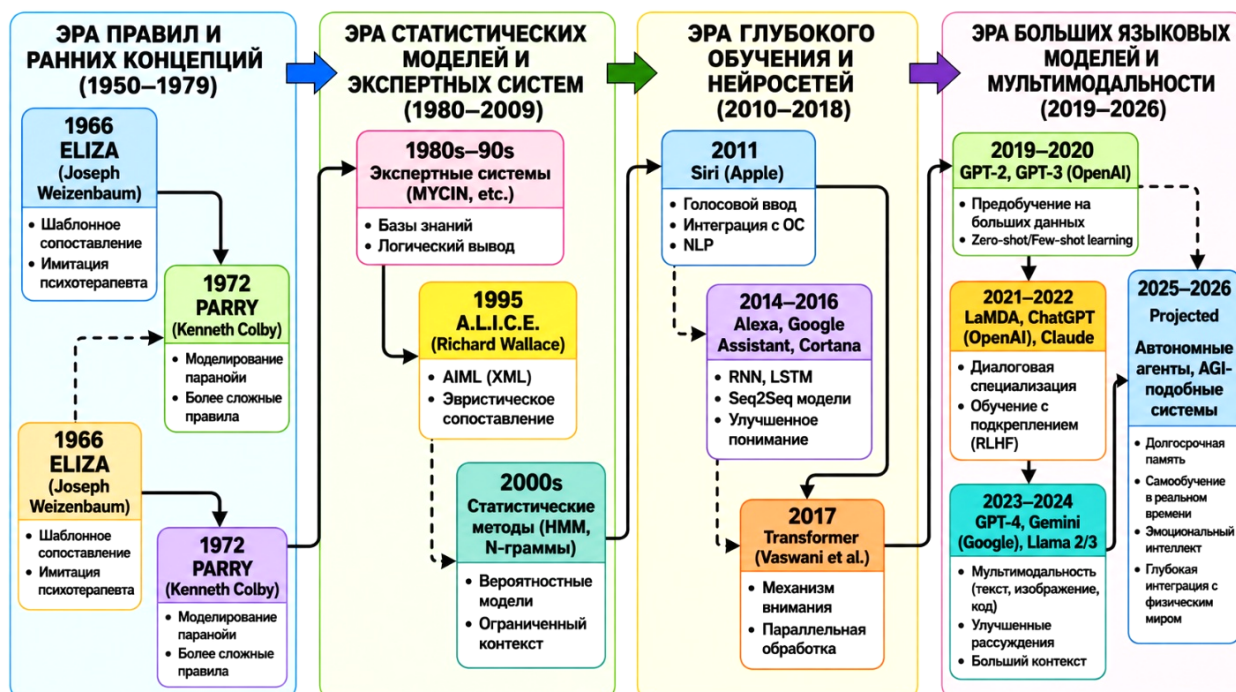


Рис. 1. Эволюция диалоговых ассистентов
Fig. 1. Evolution of Conversational Assistants

Появление в 2022 г. нейросетевой архитектуры GPT-3.5 с использованием обучения с подкреплением на основе обратной связи человека (RLHF — Reinforcement learning from human feedback) инициировало процесс массового внедрения LLM-чат-ботов: диалоговые системы превратились из вспомогательного сервиса в универсальный интерфейс доступа к знаниям и цифровым услугам.

Выход моделей нового поколения (GPT-4, Claude 3, Gemini, Llama-3 и др.) закрепил эту тенденцию, обеспечив многомодальность, поддержку длинного контекста, сложные рассуждения и адаптацию к прикладным доменам [10, 11]. В 2025–2026 гг. диалоговые ассистенты трансформируются в автономных агентов с элементами AGI (Artificial General Intelligence — общий искусственный интеллект) — ассистентов с интеллектом уровня человека.

Автономные агенты — это программные системы, способные самостоятельно ставить цели и выполнять задачи без постоянного контроля со стороны человека [12–16]. Развитие таких технологий направлено на создание AGI-подобных систем, которые стремятся к универсальности человеческого мышления и умению решать нестандартные задачи в условиях неопределенности. Ключевым компонентом их архитектуры является долгосрочная память, позволяющая сохранять опыт и контекст диалогов за пределами текущей сессии. Эффективность работы повышается за счет самообучения в реальном времени, когда система адаптируется к новым данным «на лету», не требуя полной перетренировки. Дополняет этот набор эмоциональный интеллект, обеспечивающий способность агента распознавать настроение пользователя и выстраивать эмпатичное взаимодействие.

Таким образом, за семь десятилетий развития чат-боты прошли путь от простых шаблонных программ до универсальных нейросетевых ассистентов на базе больших языковых моделей, способных вести многоходовой диалог, решать сложные когнитивные задачи и интегрироваться в критически важные информационные системы.

Большие языковые модели (LLM) — это нейросетевые модели обработки естественного языка, которые обучаются на очень больших корпусах текстов и затем способны генерировать связный текст, отвечать на вопросы, суммировать, переводить, извлекать информацию и выполнять другие языковые задачи. Обычно это крупные трансформерные модели, дополнительно дополненные под диалог и инструкции Instruction Tuning, RLHF и др.).

В основу современных LLM заложена нейросетевая архитектура трансформер (англ. Transformer). В отличие от рекуррентных нейросетевых архитектур (RNN, LSTM, GRU), трансформеры не имеют рекуррентных слоев, они заменены на механизм внимания (self-attention). Поэтому нейросетевые трансформеры не требуют строго последовательного прохождения по токенам при обучении, что обеспечивает высокую степень параллелизма вычислений. Параллельная обработка последовательностей делает возможным обучение моделей на значительно больших объемах данных и с существенно большим числом параметров при использовании современных GPU/TPU-кластеров [17–22]. Таким образом, применение нейросетевых трансформерных архитектур для больших языковых моделей позволяет получать следующие результаты:

- параллелизация обучения (существенное ускорение по сравнению с LSTM-архитектурой);
- эффективная обработка длинных контекстов;
- масштабируемость при увеличении объема обучающих данных и числа параметров модели.

В совокупности указанные свойства представляют собой ключевое архитектурное достижение, во многом определившее появление и практическую реализуемость современных больших языковых моделей. Фактически данный переход стал фундаментальным этапом развития, без которого широкое распространение LLM в их текущем виде было бы затруднено. Трансформерные языковые модели демонстрируют устойчивый рост качества по мере увеличения масштаба модели, данных и вычислений, что подтверждается эмпирическими закономерностями масштабирования. Масштабирование, в свою очередь, способствует формированию более универсальных языковых представлений и улучшению переноса на различные прикладные задачи [23–26].

В принципе, языковая модель большого масштаба может быть реализована и на альтернативных архитектурах, однако на текущем этапе именно трансформеры обеспечивают наиболее выгодный компромисс между качеством, устойчивостью обучения и вычислительной эффективностью. Кроме того, трансформеры хорошо интегрируются с методами последующей настройки под инструкции и диалоговые сценарии, что повышает их практическую ценность для построения ассистентов. По этой причине большинство современных диалоговых ассистентов, ориентированных на открытые формулировки запросов, опираются на трансформерные большие языковые модели как на базовый компонент. Таким образом, распространенность трансформеров в роли основы для больших языковых моделей объясняется не терминологическим определением, а совокупностью архитектурных, вычислительных и инженерных преимуществ, проявляющихся при обучении и применении моделей в крупном масштабе.

Большие языковые модели (LLM) радикально изменили парадигму человеко-машинного взаимодействия, сделав естественный язык основным интерфейсом доступа к цифровым сервисам, корпоративным знаниям и автоматизированным процессам. Однако в реальных производственных и институциональных сценариях использование публичных LLM показало фундаментальные ограничения: фактическую недостоверность (галлюцинации), статичность встроенных знаний и отсутствие прозрачной привязки генерируемых утверждений к авторитетным источникам. Именно эти ограничения определяют научную и практическую актуальность разработки диалоговых ассистентов на базе технологии RAG-метода, объединяющего возможности больших языковых моделей с механизмами поиска информации во внешних источниках. Технология RAG была сформулирована и популяризована в 2020 г. исследователями Facebook AI Research (компания Meta, в то время Facebook).

Стандартные большие языковые модели, несмотря на свою мощь, имеют следующие фундаментальные ограничения:

– проблема устаревших знаний: LLM обучаются один раз на фиксированном наборе данных (они не знают о событиях, произошедших после завершения обучения);

– галлюцинации и фактические ошибки: если модель не знает точного ответа, она склонна генерировать правдоподобный, но ложный текст, опираясь на статистические вероятности, а не на факты;

– отсутствие доступа к закрытым данным: публичные LLM не имеют доступа к корпоративным базам знаний, личным документам или закрытым исследованиям организации;

– сложность и дороговизна дообучения: постоянное переобучение огромной модели для внесения новых знаний требует колоссальных вычислительных ресурсов и времени, что экономически нецелесообразно для динамично меняющихся данных.

Технология RAG изменяет парадигму генерации ответа: вместо того чтобы полагаться исключительно на свою внутреннюю (параметрическую) память, языковая модель получает доступ к внешней (непараметрической) памяти. Процесс выглядит следующим образом:

– поиск (retrieval): система принимает запрос пользователя и сначала производит поиск релевантной информации во внешней базе данных (векторной базе, корпоративном хранилище, интернете);

– обогащение (augmentation): найденная информация (контекст) добавляется к исходному запросу пользователя;

– генерация (generation): LLM получает обогащенный запрос и генерирует ответ, используя найденные факты как первоисточник.

Совокупность указанных механизмов позволяет трансформировать LLM из системы, опирающейся исключительно на знания, полученные на этапе обучения, в интеллектуального агента, способного осуществлять поиск, анализ и обобщение актуальной информации из внешних источников. Можно выделить следующие ключевые преимущества применения технологии RAG:

1. Актуальность информации: система может отвечать на вопросы, используя данные, появившиеся секунду назад (например, биржевые сводки, последние новости, новые приказы по компании), без необходимости переобучения модели.

2. Снижение уровня галлюцинаций: поскольку модель получает инструкцию формировать ответ на основе предоставленного контекста, вероятность выдумки резко снижается; ответы становятся фактологически обоснованными.

3. Прозрачность и интерпретируемость: RAG позволяет указывать источники информации. Система может не просто дать ответ, но и сослаться на конкретный документ или абзац, что критически важно для многих сфер.

4. Безопасность данных и приватность: при обучении данные остаются в весовых коэффициентах нейросети. Корпоративные данные хранятся в защищенном внешнем контуре, и модель получает к ним доступ только в момент генерации ответа в рамках конкретной сессии.

5. Экономическая эффективность: внедрение RAG значительно дешевле, чем постоянное дообучение: легче обновить документ в базе данных, чем переобучать нейросеть с миллиардами параметров. RAG позволяет использовать сравнительно компактные модели, компенсируя ограниченность их параметрических знаний доступом к внешнему хранилищу, что уменьшает вычислительные затраты и упрощает внедрение в продуктивные системы.

6. Персонализация контекста: возможность динамически подменять базы знаний для разных пользователей или задач, используя одну и ту же базовую модель (LLM) как механизм рассуждений.

Архитектура диалогового ассистента на основе большой языковой модели с механизмом извлечения контекстных знаний (RAG) строится как конвейер, в котором генерация ответа опирается не только на «память» модели, но и на внешнюю базу документов / данных, релевантных текущему запросу. На рис. 2 представлена разработанная архитектура диалогового ассистента для поддержки пользователей при получении и уточнении расписания учебных занятий на основе большой языковой модели и механизма извлечения контекстных знаний.

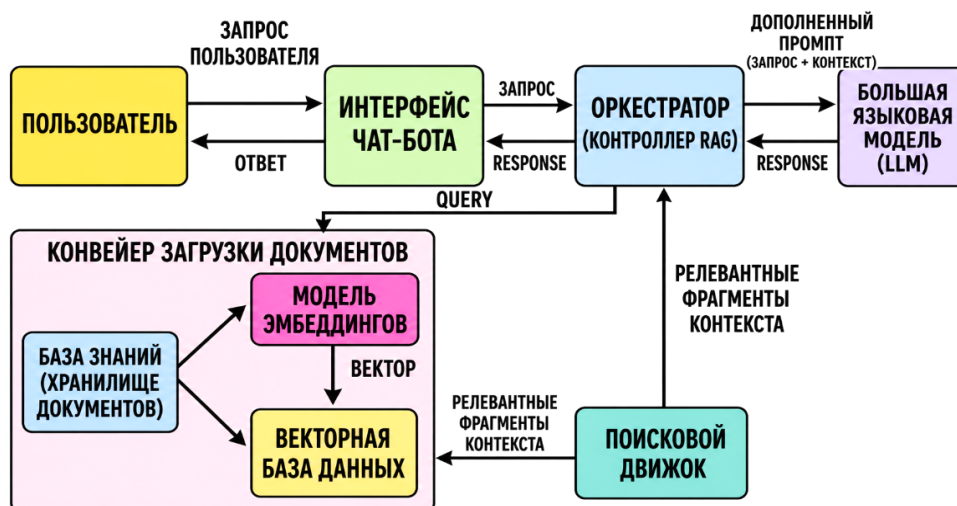


Рис. 2. Архитектура диалогового ассистента на основе большой языковой модели и механизм извлечения контекстных знаний

Fig. 2. Architecture of a Conversational Assistant Based on an LLM and Contextual Knowledge Retrieval

Пользователь отправляет текстовый запрос через клиентский интерфейс (веб / мессенджер / приложение). Оркестратор (контроллер RAG) принимает запрос и при необходимости выполняет предобработку: очистку текста, исправление опечаток или переформулирование запроса для улучшения качества поиска. Затем оркестратор RAG определяет режим обработки запроса (маршрутизацию запроса): контур RAG или поисковый движок общей базы знаний языковой модели, используя только ее параметрическую память. Далее модуль предобработки нормализует текст, определяет язык, извлекает сущности (например, группа, преподаватель, аудитория, дата), а также формирует контекст диалога, используя хранилище истории сообщений.

На этом этапе часто выполняется классификация намерения и оценка сложности запроса, чтобы выбрать стратегию обработки: прямой ответ, вызов инструментов или запуск RAG-контура. Запрос пользователя преобразуется в поисковое представление: либо строится векторное представление (модель эмбедингов), либо формируется текстовый поисковый запрос для полнотекстового поиска. Поисковый модуль обращается к базе знаний (хранилищу документов), которая создается из заранее заданных документов, расписаний, регламентов или FAQ. Индексация включает этапы ETL: извлечение данных, очистку, разбиение на фрагменты (chunking), генерацию эмбедингов, добавление метаданных (источник, дата, права доступа) и сохранение в векторное хранилище / поисковый движок.

Основные компоненты диалогового ассистента на основе большой языковой модели и механизм извлечения контекстных знаний:

1. База знаний (хранилище документов Knowledge Base): внешние документы (pdf, docx, html, текст), разбиваемые на фрагменты (chunks).
2. Эмбединги и векторное хранилище: текст преобразуется в числовые векторы и сохраняется в специальной базе для семантического поиска.
3. Поисковик («retriever»): при поступлении вопроса система производит поиск наиболее похожих по смыслу фрагментов в векторной базе. Основа технологии RAG — это процесс («Knowledge Retrieval») поиска и выборки релевантных фрагментов информации из большой базы данных в ответ на запрос пользователя.

4. Генератор (LLM): языковая модель получает дополненный промпт (вопрос пользователя + найденные фрагменты текста) и формирует точный ответ на основе предоставленных данных.

Большая языковая модель с механизмом извлечения контекстных знаний формализуется как композиция параметрической вероятностной модели (нейросетевого трансформера) и внешнего

стохастического оператора доступа к знаниям, где генерация текста условно зависит не только от входной последовательности, но и от извлеченного релевантного контекста из внешнего пространства знаний. Рассмотрим формализацию LLM через словарь токенов v и последовательность токенов:

$$X = (x_1, x_2, \dots, x_T), x_t \in V, \quad (1)$$

где x_t — t -й токен в последовательности, $t \in \{1, \dots, T\}$; t — индекс позиции токена в последовательности (текущий шаг генерации); T — общая длина последовательности (общее количество токенов в последовательности); v — словарь токенов (vocabulary).

Токен в LLM — это минимальная дискретная единица представления текста, с которой непосредственно работает модель; это не обязательно слово, а элемент формального алфавита модели, получаемый процедурой токенизации. Большая языковая модель (LLM) формально задает условное распределение:

$$P_\theta(x_1, \dots, x_T) = \prod_{t=1}^T P_\theta(x_t | x_1, \dots, x_{t-1}), \quad (2)$$

где θ — совокупность обучаемых параметров языковой модели; $P_\theta(x_t | x_1, \dots, x_{t-1})$ представляет собой условное распределение вероятностей следующего токена, формируемое глубокой нейронной сетью архитектуры трансформер и аппроксимирующее статистические закономерности естественного языка.

Каждое условное распределение реализуется через скрытое представление:

$$h_t = f_\theta(x_1, \dots, x_{t-1}), P_\theta(x_t | \cdot) = \text{Softmax}(Wh_t). \quad (3)$$

При реализации механизма извлечения контекстных знаний введем внешнее пространство знаний K ,

$$K = \{k_1, k_2, \dots, k_N\}, \quad (4)$$

где каждый элемент k_i — это структурированный или неструктурированный фрагмент информации (документ, абзац, факт).

Определим отображение кодирования знаний, которое переводит знания в векторное представление (эмбединги):

$$\varphi: K \rightarrow \mathbb{R}^d, \quad (5)$$

где φ — функция кодирования (эмбединга), отображающая элемент пространства знаний K в вектор фиксированной размерности d , сохраняющий его семантическое содержание.

Далее рассмотрим механизм извлечения контекстных знаний (Retrieval). Пусть q_t — входной запрос или текущий контекст:

$$q_t = g(h_t), \quad (6)$$

где g — проекция скрытого состояния модели в пространство запросов.

Механизм извлечения формализуется как стохастический оператор:

$$R(q_t, K) \rightarrow C_t = \{k_{t,1}, \dots, k_{t,m}\}, \quad (7)$$

где $C_t \subset K$ — релевантный контекст.

На уровне вероятностей полное распределение имеет вид

$$P(k|q_t) = \frac{\exp(q_t, \varphi(k))}{\sum_{k' \in K} \exp(\langle q_t, \varphi(k') \rangle)}, \quad (8)$$

где $k \in K$ — элемент внешнего хранилища знаний (документ, абзац, факт); $q_t \in \mathbb{R}^d$ — вектор запроса, полученный из состояния LLM; $\varphi(k) \in \mathbb{R}^d$ — векторное представление элемента знания; $\langle q_t, \varphi(k) \rangle$ — скалярное произведение.

Математический смысл формулы $P(kq_i)$ заключается в том, что механизм извлечения выбирает знания с вероятностью, экспоненциально возрастающей по мере семантической близости между текущим запросом модели и векторным представлением знания.

Далее рассмотрим интеграцию извлеченного контекста в LLM, кодируемого как c_i :

$$c_i = \psi(C_i), \quad (9)$$

где ψ — агрегирующая функция (конкатенация, attention, pooling).

Обновленное скрытое состояние $\tilde{h}_i$:

$$\tilde{h}_i = \mathcal{F}(h_i, c_i), \quad (10)$$

где $\mathcal{F}$ — функция слияния (например, cross-attention).

Соответственно итоговое распределение:

$$P_\theta(x_i | x_{<i}, C_i) = \text{Softmax}(W\tilde{h}_i), \quad (11)$$

где $\tilde{h}_i \in \mathbb{R}^d$ — итоговое скрытое состояние; $W \in \mathbb{R}^{v \times d}$ — матрица выходных весовых коэффициентов; $\text{Softmax}(W\tilde{h}_i)$ — математический оператор нормализации, который преобразует вектор логитов (набор необработанных числовых предсказаний), полученных линейной проекцией скрытого состояния модели, в корректное дискретное распределение вероятностей по всем токенам словаря.

В итоге полная (совокупная) вероятностная модель может быть записана как вероятность последовательности токенов $x_{1:T}$ с учетом механизма извлечения знаний:

$$P(x_{1:T}) = \prod_{t=1}^T \sum_{C_t \in K} P_\theta(x_t | x_{<t}, C_t) \cdot P(C_t | x_{<t}), \quad (12)$$

где $P(C_t | x_{<t})$ — распределение извлечения, модель является иерархической латентной моделью с внешней памятью.

Следовательно, генерация не осуществляется одномоментно, а представляется в виде разложения полной вероятности на цепочку условных вероятностей:

$$P(x_{1:T}) = P(x_1) \cdot P(x_2 | x_1) \cdot P(x_3 | x_1, x_2) \dots \quad (13)$$

Таким образом, это позволяет интерпретировать математическую модель LLM как параметрическую аппроксимацию условного распределения языка, в которой механизм извлечения знаний $R(q_i, K)$ — внешний нелокальный оператор памяти модели, а вся совокупная вероятностная модель системы — гибридная вероятностная модель с разделением: долгосрочных знаний (хранящиеся в K) и краткосрочного контекста (представленного в $\tilde{h}_i$). Тем самым модель LLM не кодирует все знания непосредственно в параметрах, а условно подключает их из внешней памяти, корректно учитывая этот процесс в полной вероятностной формализации.

Результаты (Results)

Разработка интеллектуального диалогового ассистента выполнена для поддержки пользователей при получении и уточнении расписания учебных занятий. В условиях цифровой трансформации образовательного процесса оперативный доступ к актуальной информации, в частности к расписанию занятий, становится критически важным фактором организации учебной деятельности. В связи с этим возникает острая необходимость в разработке интеллектуальных диалоговых ассистентов нового поколения, способных понимать семантический контекст запроса и проявлять устойчивость к вариативности формулировок пользователей.

Актуальность предлагаемого подхода обусловлена интеграцией больших языковых моделей с технологией извлечения контекстных знаний RAG. Отличительной особенностью предлагаемого решения является устойчивость к вариативным формулировкам запросов (в том числе при отклонении от заранее заданных шаблонов), обеспечиваемая динамическим выбором нейросетевых языковых моделей и использованием механизма извлечения знаний для формирования точных

ответов. Такое сочетание позволяет преодолеть главную проблему генеративных моделей: обеспечить формирование ответов исключительно на основе верифицированных данных из базы расписания университета. На рис. 3 представлена архитектура разработанного диалогового ассистента с технологией RAG и голосовым интерфейсом.

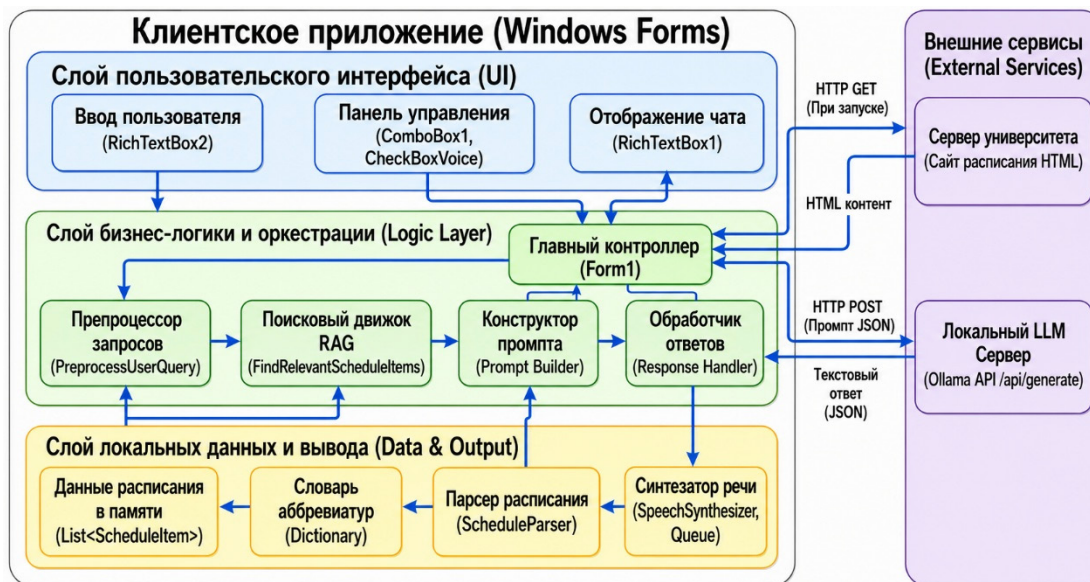


Рис. 3. Архитектура разработанного диалогового ассистента с технологией RAG и голосовым интерфейсом

Fig. 3. Architecture of the Developed Conversational Assistant with RAG Technology and a Voice Interface

Разработанный диалоговый ассистент представляет собой клиентское приложение (Windows Forms), интерфейс которого содержит следующие элементы управления: текстовое поле для отображения диалога пользователя и ассистента (RichTextBox1), текстовое поле для ввода вопросов пользователя (RichTextBox2), выпадающий список доступных языковых моделей (ComboBox1), кнопку отправки сообщений (Button1).

Класс Form1 выполняет роль оркестратора (контроллер RAG): принимает запрос пользователя и принимает решение, где находится информация: во внутренней векторной базе знаний (данные расписания в памяти), во внешних источниках (обновление данных с сайта) или в общих знаниях языковой модели, выбранной пользователем (локальный LLM-сервер). Для обновления данных о расписании на сайте университета в диалоговом ассистенте реализован парсинг. В контексте технологии RAG парсинг является фундаментом всей системы.

Парсинг веб-сайтов — это автоматизированное извлечение веб-данных, которое представляет собой алгоритмический процесс извлечения семантически значимой информации из гипертекстовых документов, размещенных в сети интернет. Данная процедура базируется на программной имитации клиентских запросов по протоколу HTTP к целевому серверу, в результате чего система получает исходный код веб-страницы, как правило, представленный на языке разметки HTML. Парсинг применяется практически везде, где необходимо автоматизировать сбор, обработку и анализ больших объемов информации. Это инструмент, который позволяет превратить хаотичные данные из сети Интернет или файлов в структурированную базу данных для принятия решений.

В архитектуре систем с технологией RAG модуль парсинга выполняет функцию фундаментального компонента ETL-конвейера, обеспечивающего первичную обработку и нормализацию гетерогенных массивов данных перед их помещением в векторную базу знаний. Данный процесс предполагает алгоритмическую трансформацию неструктурированной информации из различных форматов, таких как pdf, docx или HTML, в унифицированное текстовое представление, пригодное для последующей токенизации и генерации эмбедингов.

Ключевой задачей парсинга в контексте RAG является обеспечение семантически корректной сегментации текста, известной как чанкинг (chunking). В отличие от примитивного разбиения по количеству символов, интеллектуальные алгоритмы парсинга анализируют структуру документа, идентифицируя логические блоки, такие как заголовки, параграфы и списки. Это позволяет формировать контекстные единицы, сохраняющие смысловую целостность, что критически важно для корректной работы языковых моделей при генерации ответов.

Особое значение имеет обработка табличных данных, где парсер должен не просто извлечь текст, но и сериализовать структуру таблицы в формат, интерпретируемый нейросетью, сохраняя реляционные связи между ячейками и заголовками столбцов. Таким образом, парсинг трансформирует «сырые» данные в структурированную базу знаний, выступая необходимым промежуточным звеном между источником информации и генеративной нейросетью.

Реализация парсинга для RAG-ассистента необходима, прежде всего, для построения надежного ETL-конвейера (Extract, Transform, Load). Расписание занятий — это динамически изменяемые данные, и статичная база знаний здесь неприменима. Парсинг обеспечивает регулярный сбор обновлений с веб-ресурса, гарантируя, что векторный индекс содержит актуальную информацию, а модель не генерирует ответы на основе устаревших данных. С технической точки зрения парсинг критичен для нормализации данных перед их отправкой в LLM. Веб-страницы структурируют информацию через HTML-теги, которые ориентированы на визуальное отображение в браузере, а не на машинное понимание. Если передать в LLM веб-страницу в HTML-формате, модель может утратить связь между заголовками колонок и содержимым ячеек.

Парсер выполняет сериализацию этой структуры: он преобразует вложенные таблицы в линейные, структурированные записи, где четко прописаны связи «время – предмет – аудитория». Кроме того, парсинг решает проблему зашумленности контекста. Веб-страница содержит множество служебных элементов: скрипты, стили, навигацию. Передача служебных элементов в контекстное окно модели не только тратит лимиты токенов, но и снижает качество поиска, так как релевантный сигнал теряется на фоне технического кода. Парсер извлекает исключительно полезную нагрузку, делая работу ассистента более точной и экономичной.

На рис. 4 представлена реализация логики программного класса ScheduleParser для парсинга расписания с сайта ФГБОУ ВО «ГМУ имени адмирала Ф. Ф. Ушакова» (далее — ГМУ имени адмирала Ф. Ф. Ушакова). Разработанный класс ScheduleParser предназначен для извлечения и обработки данных о расписании с веб-страницы, используя HTML-парсинг и обработку данных для дальнейшего использования в диалоговом ассистенте.

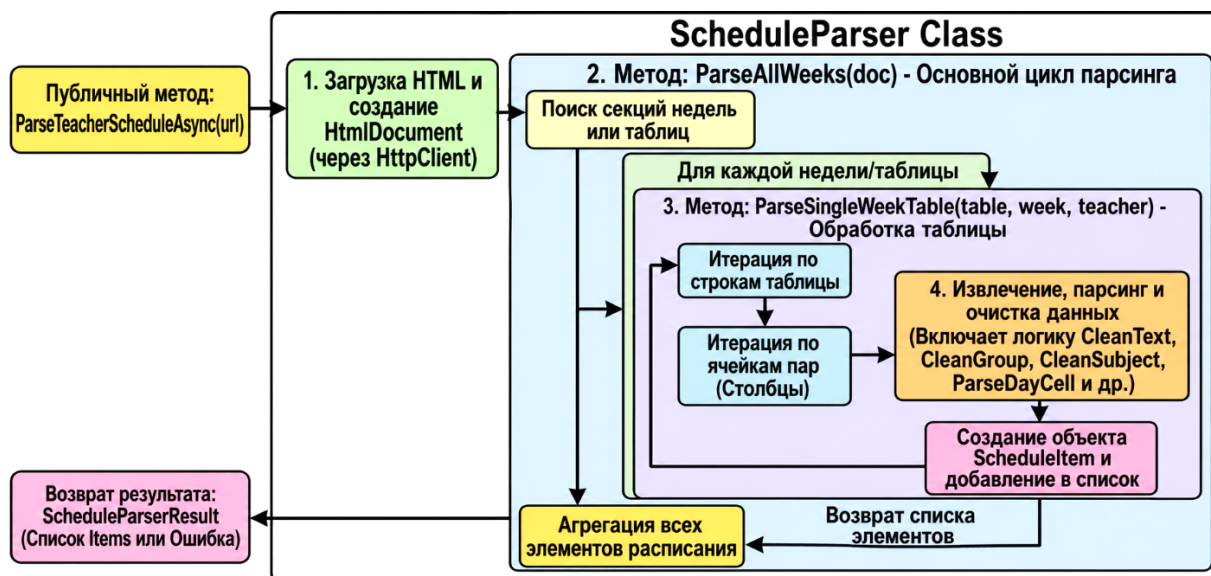


Рис. 4. Реализация логики программного класса ScheduleParser для парсинга расписания

Fig. 4. Implementation of the ScheduleParser Class Logic for Schedule Parsing

В конструкторе класса `SheduleParser` создается объект `HttpClient`, который используется для загрузки веб-страниц. Настроены заголовки HTTP-запросов, чтобы они выглядели как запросы от браузера (для обхода блокировок). Настроены кодировки для работы с текстом на русском языке в кодировке Windows-1251.

Расписание представлено на сайте в виде таблиц: для каждой недели семестра отдельная таблица с расписанием. Поэтому в классе `SheduleParser` реализована обработка таблиц расписания методом `ParseSingleWeekTable`, который извлекает следующие данные из таблицы для одной недели:

- строки таблицы, представляющие дни недели;
- время занятий извлекается из второй строки таблицы;
- каждая ячейка в таблице представляет занятие для определенной пары, из которой извлекается информация о группе, предмете и аудитории.

Методом `ParseSingleWeekTable` также извлекаются день недели и дата, которые переводят знания в векторное представление (эмбединги), обрабатывая ячейки с помощью регулярных выражений. Время пары автоматически добавляется из заранее определенного списка.

Рассмотрим реализованный фрагмент программного кода асинхронной функции `CallOllamaWithContextAsync` — «интеллектуальный центр» разработанного диалогового ассистента. Данная функция отвечает за подготовку текста (промпта), который отправляется нейросети для генерации ответа:

```
private async Task<string> CallOllamaWithContextAsync(string modelName,
string userPrompt)
{
    var contextBuilder = new StringBuilder();
    contextBuilder.AppendLine("Ты – помощник по расписанию Полковниковой Н.А. из ГМУ им. адмирала Ф.Ф. Ушакова. Отвечай на русском, будь вежлив и структурируй информацию.");
    // --- ДОБАВЛЕНА СТРОГАЯ ИНСТРУКЦИЯ ПО ТЕРМИНОЛОГИИ ---
    contextBuilder.AppendLine("СТРОГОЕ ПРАВИЛО: никогда не употребляй слова 'урок', 'уроки' или 'уроков'. В университете проводятся 'пары' и 'занятия'. Всегда используй слова 'пара', 'пары', 'занятие' или 'занятия'.");
    contextBuilder.AppendLine($"Справочник: {string.Join(", ", _abbreviations.Select(kv => $"{kv.Key} = {kv.Value}"))}");
    List<ScheduleItem> relevantItems = FindRelevantScheduleItems(userPrompt);
    if (relevantItems.Count > 0)
    {
        contextBuilder.AppendLine("Основывай ответ исключительно на данных ниже. Не придумывай информацию.");
        contextBuilder.AppendLine("\n--- ДАННЫЕ О РАСПИСАНИИ ---");
        foreach (var item in relevantItems.Take(15)) { contextBuilder.AppendLine($"• {item.DayName}, {item.DateText}, {item.TimeRange} ({item.WeekPeriod}): '{item.SubjectName}', гр. '{item.GroupNumber}', ауд. '{item.RoomNumber}'."); }
        contextBuilder.AppendLine("--- КОНЕЦ ДАННЫХ ---\n");
        contextBuilder.AppendLine("Ответь на вопрос пользователя, основываясь на этих данных.");
    }
    else { contextBuilder.AppendLine("В расписании нет информации. Ответь на вопрос, используя общие знания, но помни, что ты ассистент по расписанию."); }

    string finalPrompt = contextBuilder.ToString() + "\nВопрос: " + userPrompt;
    return await CallOllamaAsync(modelName, finalPrompt);
}
```

В строке (`contextBuilder.AppendLine(...)`) задается настройка роли и правил для нейросети ее назначение и правила поведения: «нейросеть является помощником по расписанию Н. А. Полковниковой, которая работает в “ГМУ имени адмирала Ф. Ф. Ушакова”». Далее задается поведение нейросети: «отвечай на русском, вежливо, структурируй». Затем указывается строгий запрет: запрещается использовать школьную лексику («урок») и заставлять использовать университетскую («пара», «занятие»). Это гарантирует, что диалоговый ассистент будет звучать профессионально.

Далее производится подключение «словаря»: строка «Справочник: {string.Join ...}»: берет заданный в базе знаний список аббревиатур (СППР, БД, СИИ и т. д.) и добавляет их расшифровки прямо в инструкцию. Теперь, если пользователь спросит про «СППР», нейросеть будет знать, что это дисциплина «Системы поддержки принятия решений». Затем методом `FindRelevantScheduleItems` (`userPrompt`) оркестратор RAG осуществляет поиск информации. Это поисковик по базе знаний расписания: он ищет занятия, которые соответствуют вопросу пользователя (по дате, предмету и т. д.).

Далее осуществляется внедрение данных в промпт: если данные найдены (строка `relevantItems.Count > 0`), то нейросети задается команда: «основывай ответ исключительно на данных, указанных ниже. Не придумывай информацию». Затем перечисляются найденные пары (день, время, аудитория). Это превращает нейросеть в точного информатора. Если же данные не найдены, то разрешаем нейросети использовать общие знания (например, чтобы просто поддержать разговор), но напоминаем ей о ее роли ассистента («В расписании нет информации. Ответь на вопрос, используя общие знания, но помни, что ты ассистент по расписанию»). Чтобы уменьшить риск галлюцинаций, в промпте задается требование отвечать только на основании предоставленного контекста и явно сообщать о недостатке данных.

В конце функции `CallOllamaWithContextAsync` производится соединение всей инструкции с конкретным вопросом пользователя (`string finalPrompt = ...`) и отправка в Ollama (`await CallOllamaAsync`). Таким образом, реализованный метод обрабатывает общий вопрос пользователя (например, «когда начинаются занятия по дисциплине СППР?», «когда начинаются занятия по Информатике?») и превращает его в подробное техническое задание для нейросети, снабженное правилами, терминами и конкретными данными из расписания для того, чтобы получить точный ответ (рис. 5).

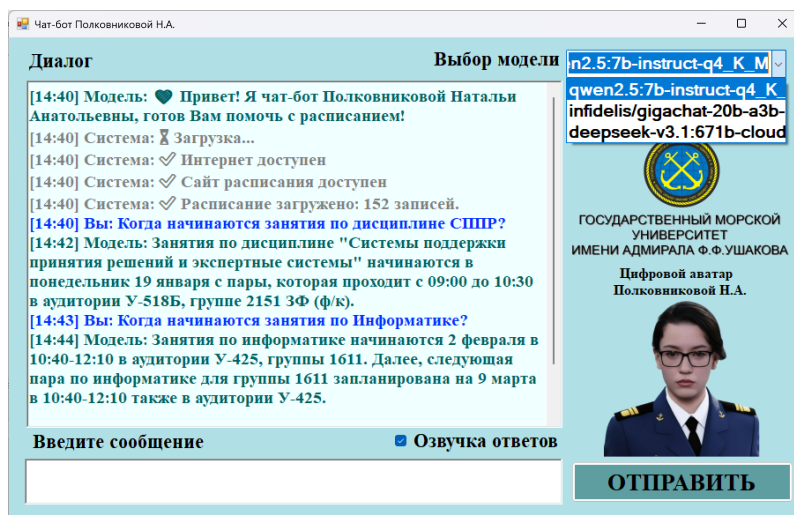


Рис. 5. Результат обработки вопросов пользователя по расписанию занятий

Fig. 5. Results of Processing User Queries Related to Class Schedules

Представленные результаты работы диалогового ассистента выполнены на примере расписания учебных занятий автора статьи доцента Н. А. Полковниковой. При запуске диалогового ассистента пользователю предлагаются на выбор три языковые модели: `qwen2.5:7b-instruct-q4_K_M`, `gigaChat-20B-A3B-instruct-v1.5` и `deepseek-v3.1:671b-cloud`. В таблице выполнено сравнение параметров данных языковых моделей.

Таблица 1

Сравнение языковых моделей

Table 1

Comparison of language models

Параметр	Языковая модель		
	Qwen2.5: 7B-Instruct-Q4_K_M	GigaChat-20B-A3B-Instruct-v1.5	DeepSeek-V3.1:671B-Cloud
Разработчик	Alibaba Group (Китай)	Sber AI (Россия)	DeepSeek (Китай)
Архитектура / тип	Декодер-трансформер, открытая модель (7 млрд параметров)	Собственная архитектура, проприетарная/закрытая модель (около 20 млрд активируемых параметров)	MoE (Mixture of Experts), условно открытая (доступ через API, 671 млрд параметров, 37 B активных)
Квантование	Да (4-битное, метод GGUF Q4_K_M)	Нет (оригинальная точность)	Нет (работает в облаке)
Контекстное окно	128 тыс. токенов	32 тыс. токенов	128 тыс. токенов
Основной язык	Многоязычная, сильна в английском и китайском	Сильно оптимизирована для русского языка	Многоязычная, сильна в английском и китайском
Доступность	Полностью открытая, можно скачать и запускать локально	Закрытый API, доступ по запросу (в основном для РФ)	Бесплатный облачный API (с лимитами), веб-интерфейс, мобильное приложение
Лицензия	Apache 2.0	Проприетарная (условия использования Sber)	Условия использования DeepSeek (для API)
Особенности и сильные стороны	1. Эффективность и скорость на локальном железе (благодаря квантованию). 2. Хорошее качество для своего размера. 3. Полная конфиденциальность (локальный запуск)	1. Лучшее качество и понимание контекста на русском среди представленных. 2. Знание российской специфики и реалий. 3. Интеграция с сервисами Сбера (поиск и т. д.)	1. Наивысшее качество и «интеллект» в сравнении (огромный размер). 2. Поддержка загрузки файлов (изображения, PDF, Word и др.). 3. Поиск в интернете (по запросу). 4. Очень большой контекст
Основные сценарии использования	Локальные приложения, приватные развертывания, эксперименты на потребительском GPU/CPU, быстрое прототипирование	Продукты и сервисы для русскоязычной аудитории, требующие глубокого понимания нюансов русского языка и культуры	Сложные аналитические и творческие задачи, работа с большими документами, программирование, исследовательские цели (где нужна максимальная мощность)
Требования для запуска модели	Низкие: работает на среднем GPU или даже CPU	Зависит от тарифов API Sber. Запуск оригинальной модели локально требует значительных ресурсов (~40 Гб + VRAM)	Бесплатный API с лимитами. Требуется подключения к Интернет. Локально не запускается

Для локального использования с хорошим балансом качества/скорости: эффективна автономная квантованная модель Qwen2.5:7B-Instruct. Для задач на русском языке, где критически важны точность, идиомы и культурный контекст: лучший выбор — GigaChat. Для максимально качественного решения сложных задач любого типа, работы с файлами и большими текстами: эффективна

модель DeepSeek-V3.1. Это самая мощная модель в списке с широкими возможностями, доступная бесплатно через API. Модель Qwen2.5 продемонстрировала высокую эффективность при локальном запуске, в то время как GigaChat-20B оказалась наилучшей для решения задач на русском языке, благодаря глубокому пониманию российской специфики и культурных особенностей. Модель DeepSeek-V3.1 с более высоким количеством параметров показала исключительные результаты в решении сложных аналитических задач и работе с большими объемами данных.

В разработанном диалоговом ассистенте реализована поддержка специальных команд пользователя для быстрого доступа к расписанию. Например, на рис. 6 представлен результат обработки команды пользователя: «дата 12 марта» — после полученной команды диалоговый ассистент выводит все занятия из расписания за указанную дату.

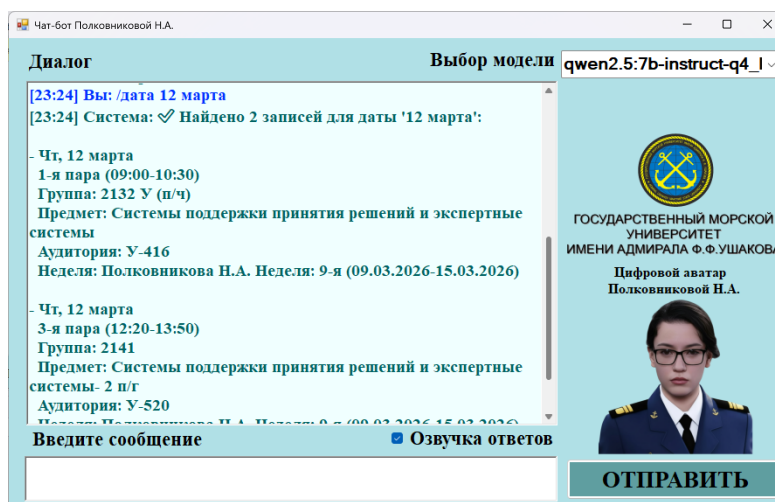


Рис. 6. Результат обработки команды пользователя
Fig. 6. Result of Processing a User Command

Если пользователь задает общий вопрос, который отсутствует в явном виде в расписании, то диалоговый ассистент использует общие знания, заложенные в выбранную языковую модель, но при этом помнит о роли ассистента и пытается связать ответ с дисциплинами в расписании. На рис. 7 представлен результат, когда пользователь задает общий вопрос: «что такое большие данные?»: сначала диалоговый ассистент связывает ответ с дисциплинами в расписании, а затем дает ответ по данному термину, используя общие знания.

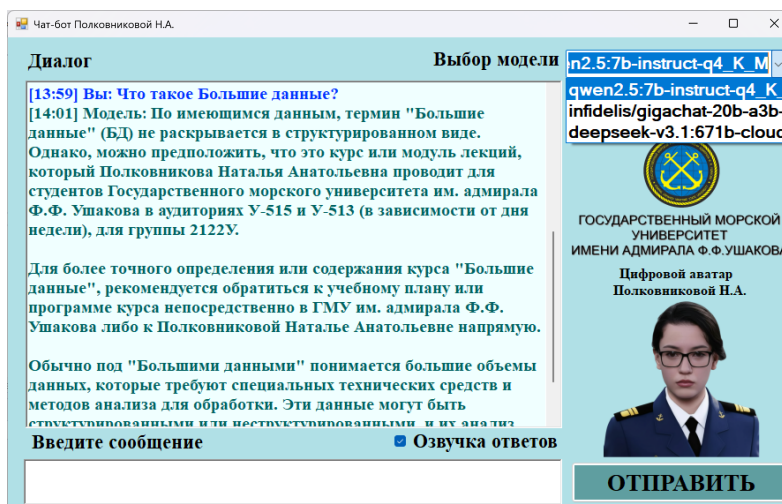


Рис. 7. Результат обработки общих вопросов пользователя
Fig. 7. Result of Processing General User Queries

В разработанном диалоговом ассистенте реализована функция озвучки ответов с помощью голосовых движков. Использование голосовых движков переводит систему из разряда классических чат-ботов в класс полноценных виртуальных цифровых ассистентов, приближая коммуникацию к формату естественного человеческого диалога. Данная функциональность особенно востребована в сценариях, когда визуальное восприятие информации с экрана затруднено или необходимо оперативное получение данных о расписании в фоновом режиме.

Заключение (Conclusion)

В работе рассмотрены этапы эволюции диалоговых систем — от ранних правил-ориентированных текстовых диалоговых программ, основанных на шаблонном сопоставлении входных высказываний, до современных виртуальных ассистентов, реализованных на основе нейросетевых трансформерных архитектур и больших языковых моделей. Описан процесс разработки диалогового ассистента с использованием нейросетевой архитектуры трансформера и технологии RAG, обеспечивающей доступ к актуальным данным через внешние базы знаний.

В рамках проведенного исследования разработан интеллектуальный диалоговый ассистент, способный эффективно поддерживать диалог и запросы пользователей при получении и уточнении расписания учебных занятий. Применение технологии извлечения контекстных знаний (RAG) в сочетании с динамическим выбором нейросетевых языковых моделей позволило значительно повысить точность ответов и устойчивость системы к вариативным формулировкам запросов. Это позволяет решить важную проблему генеративных моделей, обеспечивая ответы исключительно на основе верифицированных данных, что особенно важно в контексте работы с актуальной информацией, такой как расписание учебных занятий.

В диалоговом ассистенте реализована технология реализован RAG через механизм парсинга веб-страниц для обновления базы данных расписания с веб-ресурсов университета. Реализация парсинга веб-страниц стала важным компонентом системы, обеспечив регулярное обновление данных из внешних источников и поддержку актуальности расписания, что позволило избежать проблем с устаревшими данными и улучшить качество работы ассистента в реальных условиях. Рассмотрена реализация логики программного класса SheduleParser для парсинга расписания с сайта ФГБОУ ВО «ГМУ имени адмирала Ф.Ф. Ушакова». Разработанный класс SheduleParser предназначен для извлечения и обработки данных о расписании с веб-страницы, используя HTML-парсинг и обработку данных для дальнейшего использования в диалоговом ассистенте. Представленные результаты работы диалогового ассистента выполнены на примере расписания учебных занятий автора статьи доцента Н. А. Полковниковой.

Рассмотрен реализованный фрагмент программного кода асинхронной функции CallOllamaWithContextAsync – «интеллектуальный центр» разработанного диалогового ассистента. Реализованная функция отвечает за подготовку текста (промпта), который отправляется нейросети для генерации ответа. В функции определена настройка роли и правил для нейросети, а также назначение и правила поведения. Рассмотрены возможности динамического выбора языковой модели, а также функциональные особенности и преимущества каждой модели для различных сценариев применения. Особое внимание в работе было уделено выбору и интеграции языковых моделей, каждая из которых показала свои сильные стороны в зависимости от требований к задаче. Модель Qwen2.5:7B продемонстрировала высокую эффективность при локальном запуске, в то время как GigaChat-20B оказалась наилучшей для решения задач на русском языке благодаря глубокому пониманию российской специфики и культурных особенностей. Модель DeepSeek-V3.1, с более высоким количеством параметров, показала исключительные результаты в решении сложных аналитических задач и работе с большими объемами данных.

В результате проведенного исследования получены следующие научные результаты:

1. Предложена архитектура интеллектуального диалогового ассистента, основанная на интеграции технологии извлечения контекстных знаний (RAG) с механизмом динамического выбора нейросетевой языковой модели, что позволяет адаптивно выбирать генератор ответа в зависимости

от типа запроса и условий обработки. В известных решениях, как правило, используется фиксированная языковая модель, тогда как предложенная архитектура вводит уровень маршрутизации между несколькими LLM.

2. Выполнена формализация диалогового ассистента как гибридной вероятностной модели с внешней памятью, в которой процесс генерации зависит не только от параметров языковой модели, но и от стохастического оператора извлечения знаний из внешнего пространства. В работе получено математическое описание механизма интеграции извлеченного контекста в скрытое состояние LLM.

3. Разработан метод доменно-ориентированного формирования контекста для задач работы с динамически обновляемыми табличными данными (расписание учебных занятий), включающий специализированный парсинг веб-страниц, семантически корректный чанкинг и строгие правила промптинга, обеспечивающие формирование ответов исключительно на основе верифицированных данных.

4. Разработан метод обеспечения фактологической достоверности генеративных моделей в узком предметном домене через:

- строгую терминологическую фильтрацию на этапе промпт-инжиниринга (запрет школьной лексики «урок» в пользу университетской «пара»);
- интеграцию справочника аббревиатур непосредственно в системный промпт;
- механизм явного ограничения генерации рамками извлеченного контекста с детекцией отсутствия релевантных данных.

Эмпирически подтверждена корректность ответов диалогового ассистента при наличии данных в источнике.

5. Реализован полный ETL-конвейер RAG с парсингом веб-ресурсов университета как неотъемлемый компонент архитектуры в виде разработанного класса ScheduleParser. Это решает проблему устаревания знаний в образовательных доменах, где расписание обновляется еженедельно, — проблема, не решенная в классических статических реализациях RAG. Проведено сравнительное исследование эффективности трех LLM (Qwen, GigaChat, DeepSeek) в узком предметном домене «расписание вуза» с выявлением следующих закономерностей:

- GigaChat-20B демонстрирует преимущество в понимании русскоязычных запросов с культурно-специфической лексикой;
- квантованная модель Qwen2.5:7B обеспечивает приемлемое качество при локальном запуске;
- DeepSeek-V3.1 эффективен для сложных многоходовых запросов, но избыточен для простых фактологических вопросов.

Разработанный диалоговый ассистент продемонстрировал свою способность эффективно взаимодействовать с пользователями, обеспечивая точные и своевременные ответы на запросы, связанные с расписанием учебных занятий. Внедрение механизма извлечения контекстных знаний и интеграция различных языковых моделей открывает возможности для расширения функционала системы, включая поддержку более сложных сценариев взаимодействия и использование в других сферах. Полученные результаты подтверждают, что сочетание современных нейросетевых подходов с механизмами извлечения знаний является эффективным и перспективным решением для создания интеллектуальных диалоговых систем, способных к гибкому и точному ответу на запросы пользователей в условиях динамично меняющихся данных.

СПИСОК ЛИТЕРАТУРЫ

1. Attention is All you Need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin // Advances in Neural Information Processing Systems – 30 Curran Associates, Inc., 2017.
2. Large Language Models: A Survey [Электронный ресурс] / Minaee Sh., Mikolov T., Nikzad N. [и др.]. 2025. — Режим доступа: <https://arxiv.org/abs/2402.06196> (дата обращения: 08.01.2026).

3. A Survey of Large Language Models [Электронный ресурс] / Zhao W. X., Zhou K., Li J. [и др.]. 2026. — Режим доступа: <https://arxiv.org/abs/2303.18223> (дата обращения: 08.01.2026).
4. Challenges and Applications of Large Language Models [Электронный ресурс] / Kaddour J., Harris J., Mozes M. et al. 2023. — Режим доступа: <https://arxiv.org/abs/2307.10169> (дата обращения: 08.01.2026).
5. A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness / F. Wang, Z. Zhang, X. Zhang, Z. Wu, TH. Mo, Q. Lu, W. Wang, R. Li, J. Xu, X. Tang, Q. He, Y. Ma, M. Huang, S. Wang // ACM Trans. Intell. Syst. Technol. — 2025. — Vol. 16. — Is. 6. DOI: 10.1145/3768165.
6. History, development, and principles of large language models: an introductory survey / Z. Wang, Z. Chu, T. V. Doan, S. Ni, M. Yang, W. Zhang // AI and Ethics. — 2025. — Vol. 5. — Is. 3. Pp. 1955–1971. DOI: 10.1007/s43681-024-00583-7.
7. Datasets for large language models: a comprehensive survey / Y. Liu, J. Cao, C. Liu, K. Ding, L. Jin // Artificial Intelligence Review. — 2025. — Vol. 58. — Is. 12. Pp. 403. DOI: 10.1007/s10462-025-11403-7.
8. Efficient multimodal large language models: a survey / Y. Jin, J. Li, T. Gu, Y. Liu, B. Zhao, J. Lai, Z. Gan, Y. Wang, C. Wang, X. Tan, L. Ma // Visual Intelligence. 2025. Vol. 3. Is. 1. Pp. 27. DOI: 10.1007/s44267-025-00099-6.
9. Retrieval-Augmented Generation for Large Language Models: A Survey [Электронный ресурс] / Y. Gao, Y. Xiong, X. Gao [et al.]. — 2024. — Режим доступа: <https://arxiv.org/abs/2312.10997> (дата обращения: 08.01.2026).
10. PaperQA: Retrieval-Augmented Generative Agent for Scientific Research [Электронный ресурс] / J. Lála, O. O'Donoghue, A. Shtedritski [et al.]. — 2023. — Режим доступа: <https://arxiv.org/abs/2312.07559> (дата обращения: 08.01.2026).
11. A Comprehensive Overview of Large Language Models / H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian // ACM Trans. Intell. Syst. Technol. — 2025. — Vol. — 16. — Is. 5. DOI: 10.1145/3744746.
12. Антонов И. В. Применение больших языковых моделей для интеллектуального поиска и извлечения информации в корпоративных информационных системах / И. В. Антонов, Ю. В. Бруттан // Компьютерные исследования и моделирование. — 2025. — Т. 17. — № 5. — С. 871–888. EDN INCXRM. DOI: 10.20537/2076-7633-2025-17-5-871-888.
13. Мансур А. М. Разработка чат-бота для классификации и анализа текстов на естественном языке с использованием локальных больших языковых моделей / А. М. Мансур, Ж. Х. Мохаммад, Ю. А. Кравченко // Известия ЮФУ. Технические науки. — 2025. — № 3(245). — С. 159–171. EDN TWDTOO. DOI: 10.18522/2311-3103-2025-3-159-171.
14. Геворгян А. А. Улучшение взаимодействия с абитуриентами через чат-боты на основе ии / А. А. Геворгян, В. А. Кудинов // Auditorium. — 2024. — № 2(42). — С. 13–19. EDN MQEPFM.
15. Родионов А. В. Выбор инструментов построения чат-бота приемной комиссии учебного заведения / А. В. Родионов, Д. А. Корж // System Analysis and Mathematical Modeling. — 2024. — Т. 6. — № 3. — С. 367–376. EDN IXAVJL. DOI: 10.17150/2713-1734.2024.6(3).
16. Савкина А. В. Разработка клиентской части системы для реализации работы чат-бота / А. В. Савкина, Е. С. Матвеев // Тенденции развития науки и образования. — 2024. — № 115-15. — С. 107–112. EDN YOTZQA. DOI: 10.18411/trnio-11-2024-710.
17. Вертоградов В. В. Сравнительный анализ конкуренции генеративных чат ботов в России и мире / В. В. Вертоградов, С. В. Щелокова // Проблемы прогнозирования. — 2025. — № 5(212). — С. 73–84. EDN JWREBM. DOI: 10.47711/0868-6351-212-73-84.
18. Вольников М. С. Обзор существующих решений чат-ботов, использующих искусственные нейронные сети: их возможности, преимущества и недостатки / М. С. Вольников, Е. А. Данилов // Современные информационные технологии. — 2024. — № 39(39). — С. 9–15. EDN QZVHRL.
19. Епрынцева Н. А. Разработка чат-бота в социальной сети «ВКонтакте» для образовательной организации / Н. А. Епрынцева // Инженерный вестник Дона. — 2023. — № 9(105). — С. 656–668. EDN GSGGXV.
20. Алексеенко В. А. Разработка чат-бота для образовательных задач / В. А. Алексеенко, И. А. Татин // Информационные технологии и математическое моделирование в управлении сложными системами. — 2025. — № 2(26). — С. 8–15. EDN OWVACB.
21. Подходы к нейро-символической интеграции: большие языковые модели и базы знаний / В. А. Головкин, В. В. Голенков, М. В. Ковалев, А. А. Крошенко, Д. В. Шункевич, В. П. Ивашенко // Открытые семантические технологии проектирования интеллектуальных систем. — 2025. — № 9. — С. 87–96. EDN DSLKGD.
22. Шмат А. В. Применение больших языковых моделей и технологии Retrieval-Augmented Generation для корпоративных ассистентов / А. В. Шмат // Известия Тульского государственного университета. Технические науки. — 2024. — № 10. — С. 720–726. EDN GSNKPM. DOI: 10.24412/2071-6168-2024-10-720-721.

23. Герасименко Е. М. Алгоритм поиска и приобретения знаний на основе технологий обработки и анализа текстов на естественном языке / Е. М. Герасименко, Ю. А. Кравченко, Д. А. Шаненко // Известия ЮФУ. Технические науки. — 2024. — № 5(241). — С. 88–102. EDN GMUMBY. DOI: 10.18522/2311-3103-2024-5-88-102.
24. Понимаш З. А. Метод и алгоритм извлечения признаков из цифровых сигналов на базе нейросетей трансформер / З. А. Понимаш, М. В. Потанин // Известия ЮФУ. Технические науки. — 2024. — № 6(242). — С. 52–64. EDN RRXXMK. DOI: 10.18522/2311-3103-2024-6-52-64.
25. Интеллектуальные системы управления и университетские комплексы транспортной отрасли: тренды и перспективы / О. В. Андреева, Е. В. Хекерт, М. Л. Сомко, А. И. Епихин // Мир транспорта. — 2024. — Т. 22. — № 1(110). — С. 94–99. EDN HCAFNK. DOI: 10.30932/1992-3252-2024-22-1-12.
26. Епихин А. И. Методы машинного обучения в мониторинге рабочего поведения морского двухтактного дизельного двигателя / А. И. Епихин, С. И. Кондратьев, Е. В. Хекерт // Транспорт Российской Федерации. Журнал о науке, практике, экономике. — 2024. — № 3(112). — С. 50–53. EDN BIAZCJ.

REFERENCES

1. Vaswani, A., I. Polosukhin et al. "Attention is All you Need." *Advances in Neural Information Processing Systems* – 30. Curran Associates, Inc., 2017.
2. Minaee, Shervin, et al. "Large Language Models: A Survey." arXiv, 2025. Web. 08 Jan. 2026 . <<https://arxiv.org/abs/2402.06196>>.
3. Zhao, Wayne Xin, et al. "A Survey of Large Language Models." arXiv, 2026. Web. 08 Jan. 2026 <<https://arxiv.org/abs/2303.18223>>.
4. Kaddour, Jean, et al. "Challenges and Applications of Large Language Models." arXiv, 2023. Web. 08 Jan. 2026 .
5. Wang, F., S. Wang et al. "A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness." *ACM Trans. Intell. Syst. Technol.* 16.6 (2025). DOI: 10.1145/3768165.
6. Wang, Z., W. Zhang et al. "History, development, and principles of large language models: an introductory survey." *AI and Ethics* 5.3 (2025): 1955–1971. DOI: 10.1007/s43681-024-00583-7.
7. Liu, Y., L. Jin et al. "Datasets for large language models: a comprehensive survey." *Artificial Intelligence Review* 58.12 (2025): 403. DOI: 10.1007/s10462-025-11403-7.
8. Jin, Y., L. Ma et al. "Efficient multimodal large language models: a survey." *Visual Intelligence* 3.1 (2025): 27. DOI: 10.1007/s44267-025-00099-6.
9. Gao, Yunfan, et al. "Retrieval-Augmented Generation for Large Language Models: A Survey." arXiv, 2024. Web. 08 Jan. 2026 <<https://arxiv.org/abs/2312.10997>>.
10. Lála, Jakub, et al. "PaperQA: Retrieval-Augmented Generative Agent for Scientific Research." arXiv, 2023. Web. 08 Jan. 2026 <<https://arxiv.org/abs/2312.07559>>.
11. Naveed, H., A. Mian et al. "A Comprehensive Overview of Large Language Models." *ACM Trans. Intell. Syst. Technol.* 16.5 (2025). DOI: 10.1145/3744746.
12. Antonov, I. V. and Yu. V. Bruttan. "Using rag technology and large language models to search for documents and obtain information in corporate information systems." *Computer Research and Modeling* 17.5 (2025): 871–888. DOI: 10.20537/2076-7633-2025-17-5-871-888.
13. Mansur, A. M., Zh. Kh. Mokhammad and Yu. A. Kravchenko. "Development of a chatbot for classification and analysis of natural language texts using local large language models." *Izvestiya Sfedu. Engineering Sciences* 3(245) (2025): 159–171. DOI: 10.18522/2311-3103-2025-3-159-171.
14. Gevorgyan, A. A. and V. A. Kudinov. "Improving interaction with applicants through ai-based chatbots." *Auditorium* 2(42) (2024): 13–19.
15. Rodionov, A. V. and D. A. Korzh. "Tools for building a chatbot for the admissions commission of an university." *System Analysis and Mathematical Modeling* 6.3 (2024): 367–376. DOI: 10.17150/2713-1734.2024.6(3).
16. Savkina, A. V. and E. S. Matveev. "Razrabotka klientskoy chasti sistemy dlya realizatsii raboty chatbota." *Tendentsii razvitiya nauki i obrazovaniya* 115-15 (2024): 107–112. DOI: 10.18411/trnio-11-2024-710.
17. Vertogradov, V. V. and S. V. Shchelokova. "Comparative Competitive Analysis of Generative AI Chatbots in Russia and Worldwide." *Studies on Russian Economic Development* 36.5 (2025): 643–652. DOI: 10.1134/S1075700725700376.
18. Vol'nikov, M. S. and E. A. Danilov. "Overview of existing chatbot solutions using artificial neural networks: their capabilities, advantages and disadvantages." *Sovremennye informatsionnye tekhnologii* 39(39) (2024): 9–15.

19. Epryntseva, N. A. "The development of a chatbot in the social network "vkontakte" for an educational organization." *Engineering Journal of Don* 9(105) (2023): 656–668.
20. Alekseenko, V. A. and I. A. Tatin. "Development of a chatbot for educational purposes." *Information Technology and Mathematical Modeling in The Management of Complex Systems* 2(26) (2025): 8–15.
21. Golovko, V., V. Golenkov and M. Kovalev et al. "Approaches of Neuro-Symbolic Integration: Large Language Models and Knowledge Bases." *Information Technology and Mathematical Modeling in The Management of Complex Systems* 9 (2025): 87–96.
22. Shmat, A. V. "Application of large language models and retrieval-augmented generation technology for enterprise assistants." *Izvestiya Tula State University* 10 (2024): 720–726. DOI: 10.24412/2071-6168-2024-10-720-721.
23. Gerasimenko, E. M., Yu. A. Kravchenko and D. A. Shanenko. "Algorithm for searching and acquisition of knowledge based on technologies for processing and analyzing texts in natural language." *Izvestiya Sfedu. Engineering Sciences* 5(241) (2024): 88–102. DOI: 10.18522/2311-3103-2024-5-88-102.
24. Ponimash, Z. A. and M. V. Potanin. "Method and algorithm for extracting features from digital signals based on neural networks transformer." *Izvestiya Sfedu. Engineering Sciences* 6(242) (2024): 52–64. DOI: 10.18522/2311-3103-2024-6-52-64.
25. Andreeva, O. V., E. V. Khekert, M. L. Somko and A. I. Epikhin. "Intelligent management systems and universities in the transport industry: trends and prospects." *World of Transport and Transportation* 22.1(110) (2024): 94–99. DOI: 10.30932/1992-3252-2024-22-1-12.
26. Epikhin, A. I., S. I. Kondrat'ev and E. V. Khekert. "Machine learning methods in monitoring the operational behavior of a marine two-stroke diesel engine." *Transport of The Russian Federation* 3(112) (2024): 50–53.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Полковникова Наталья Анатольевна —
кандидат технических наук, доцент
ФГБОУ ВО «Государственный морской
университет имени адмирала Ф. Ф. Ушакова»
353918, Российская Федерация,
Краснодарский край,
г. Новороссийск, пр. Ленина, 93
e-mail: natalia-polkovnikova@mail.ru

INFORMATION ABOUT THE AUTHORS

Polkovnikova, Natalia A. —
PhD in Technical Sciences, Associate Professor
Federal State-Funded Educational Institution
of Higher Education "Admiral F. F. Ushakov
State Maritime University"
93, Lenin's avenue, Novorossiysk, Krasnodar Krai,
353918, Russian Federation
e-mail: natalia-polkovnikova@mail.ru

Статья поступила в редакцию: 03 февраля 2026 г.
Received: Feb. 3, 2026.