УДК 004.89
DOI: 10.26102/2310-6018/2026.62.11.009

Применение эмбеддингов и больших языковых моделей для построения интеллектуальной системы технической поддержки

Д.Ю. Сузьмин✉, Б.В. Усманов, А.И. Егунова, С.А. Ямашкин, С.А. Федосин, А.А. Аббакумов

Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация

Резюме. Актуальность исследования обусловлена необходимостью автоматизации технической поддержки программных продуктов при ограниченности знаний универсальных больших языковых моделей и недостаточной эффективности поиска только по точному совпадению слов. Цель работы состоит в проектировании и функциональной проверке интеллектуальной системы, формирующей ответы на основе релевантных фрагментов разнородной базы знаний. Основным подходом является генерация с дополнением извлеченными данными, объединяющая обработку документации, исходного кода и решенных обращений пользователей в систему технической поддержки, построение нормализованных векторных представлений, полнотекстовый и семантический поиск, объединение результатов ранжирования и формирование ответа большой языковой моделью. Для проверки сформирован набор из 100 вопросов по пяти категориям и использована база знаний из 148 источников, разделенных на 201 фрагмент. Для вопросов с подтвержденными сведениями корректные ответы получены в 76,25 % случаев. Общая доля корректно обработанных запросов составила 67 %, а при отсутствии подтверждающих сведений система корректно отказалась от ответа в 6 из 20 случаев. Полученные результаты подтверждают работоспособность разработанной системы при наличии необходимых сведений в базе знаний и показывают необходимость совершенствования обработки запросов с отсутствующей информацией. Практическая значимость решения состоит в возможности создавать специализированных агентов технической поддержки и обновлять их предметные знания посредством переиндексации источников без дополнительного обучения языковой модели.

Ключевые слова: большая языковая модель, эмбеддинг, генерация с дополнением извлеченными данными, семантический поиск, векторная база данных, техническая поддержка, веб-приложение.

Для цитирования: Сузьмин Д.Ю., Усманов Б.В., Егунова А.И., Ямашкин С.А., Федосин С.А., Аббакумов А.А. Применение эмбеддингов и больших языковых моделей для построения интеллектуальной системы технической поддержки. *Моделирование, оптимизация и информационные технологии.* 2026;14(11). URL: https://moitvivt.ru/ru/journal/article?id=2620 DOI: 10.26102/2310-6018/2026.62.11.009

Application of embeddings and large language models for building an intelligent technical support system

D.Yu. Suzmin✉, B.V. Usmanov, A.I. Egunova, S.A. Yamashkin, S.A. Fedosin, A.A. Abbakumov

National Research Ogarev Mordovia State University, Saransk, the Russian Federation

Abstract. The relevance of this study is determined by the need to automate technical support for software products, given the limited knowledge of general-purpose large language models and the insufficient effectiveness of search methods based solely on exact word matching. The aim of this work

© Сузьмин Д.Ю., Усманов Б.В., Егунова А.И., Ямашкин С.А., Федосин С.А., Аббакумов А.А., 2026

is to design and functionally evaluate an intelligent system that generates responses based on relevant fragments retrieved from a heterogeneous knowledge base. The proposed approach is based on retrieval-augmented generation, which integrates the processing of documentation, source code, and resolved user support requests, the construction of normalized vector representations, full-text and semantic search, ranking result fusion, and response generation using a large language model. For evaluation, a dataset of 100 questions divided into five categories was created, together with a knowledge base containing 148 sources split into 201 fragments. For questions supported by relevant information, correct answers were obtained in 76.25 % of cases. The overall proportion of correctly processed queries was 67 %. When supporting information was absent, the system correctly refused to provide an answer in 6 out of 20 cases. The results confirm the effectiveness of the developed system when the required information is available in the knowledge base and indicate the need to improve the processing of queries for which relevant information is unavailable. The practical significance of the proposed solution lies in its ability to create specialized technical support agents whose domain knowledge can be updated by re-indexing information sources without additional training of the language model.

Keywords: large language model, embedding, retrieval-augmented generation, semantic search, vector database, technical support, web application.

For citation: Suzmin D.Yu., Usmanov B.V., Egunova A.I., Yamashkin S.A., Fedosin S.A., Abbakumov A.A. Application of embeddings and large language models for building an intelligent technical support system. *Modeling, Optimization and Information Technology.* 2026;14(11). (In Russ.). URL: https://moitvivt.ru/ru/journal/article?id=2620 DOI: 10.26102/2310-6018/2026.62.11.009

Введение

Развитие программных продуктов сопровождается ростом объема документации, исходного кода и истории обращений пользователей. Эти данные могут использоваться при решении повторяющихся проблем, однако поиск ответа требует анализа нескольких разнородных источников. Традиционный поиск ориентирован преимущественно на совпадение слов и не всегда позволяет находить материалы, близкие запросу по смыслу. Большие языковые модели способны формировать связные ответы на естественном языке, но их знания ограничены данными предварительного обучения и могут не содержать актуальной информации о конкретном программном продукте. Кроме того, при отсутствии необходимых сведений модель может генерировать правдоподобные, но неподтвержденные утверждения [1].

Одним из способов решения этой проблемы является генерация с дополнением извлеченными данными. В таких системах пользовательский запрос используется для поиска релевантных фрагментов во внешней базе знаний, которые затем передаются большой языковой модели в качестве контекста [2, 3]. Для семантического поиска запросы и документы представляются в виде эмбеддингов, близость которых отражает смысловое сходство текстов и позволяет находить релевантные материалы без точного совпадения формулировок [1, 2].

Подобные системы применяются для интеллектуального поиска в корпоративных информационных системах и анализа специализированных текстов [1, 4]. Качество извлечения контекста зависит от метода поиска и структуры базы знаний [5]. При построении системы технической поддержки необходимо также обеспечить совместную работу с документацией, исходным кодом и историей решенных обращений, исключить смешивание данных разных программных продуктов, поддерживать обновление эмбеддингов при изменении источников и ограничивать объем контекста, передаваемого языковой модели.

Таким образом, проблема исследования состоит в необходимости обеспечить поиск сведений в документации, исходном коде и решенных обращениях пользователей

для формирования ответов технической поддержки без дополнительного обучения большой языковой модели.

Цель работы – проектирование и реализация интеллектуальной системы технической поддержки, использующей эмбеддинги для семантического поиска и большую языковую модель для формирования ответов на обращения пользователей.

Для достижения поставленной цели необходимо решить следующие задачи:

1) определить состав и способ организации базы знаний, объединяющей документацию, исходный код и обезличенные решенные обращения;

2) разработать архитектуру системы с изоляцией знаний отдельных программных продуктов;

3) разработать гибридный поиск и ранжирование фрагментов с последующей передачей релевантного контекста большой языковой модели;

4) провести функциональную проверку разработанной системы и оценить корректность ответов, релевантность найденных источников информации, обработку запросов с отсутствующими сведениями.

Практическим результатом работы является веб-приложение, позволяющее создавать специализированных агентов поддержки для отдельных программных продуктов и обновлять их знания посредством переиндексации источников.

Материалы и методы

Архитектура системы. Разработанная система технической поддержки является частью веб-приложения с сервисной архитектурой. Пользователь взаимодействует с ней через веб-интерфейс, а программный шлюз проверяет запрос и направляет его в сервис технической поддержки. Сервис управляет агентами, источниками знаний, диалогами, поиском информации и формированием ответов. Для каждого программного продукта создается отдельный агент с собственной базой знаний. При поиске идентификатор агента используется как обязательное условие, поэтому сведения одного продукта не могут попасть в ответ по другому продукту.

Сведения об агентах, источниках, фрагментах и диалогах хранятся в PostgreSQL. Расширение pgvector применяется для хранения векторных представлений и поиска по косинусному расстоянию. Архив исходного кода размещается в закрытом объектном хранилище MinIO. Подготовка базы знаний выполняется фоновым обработчиком, а построение эмбеддингов и генерация ответа осуществляются двумя отдельными внешними моделями. Архитектура системы представлена на Рисунке 1.

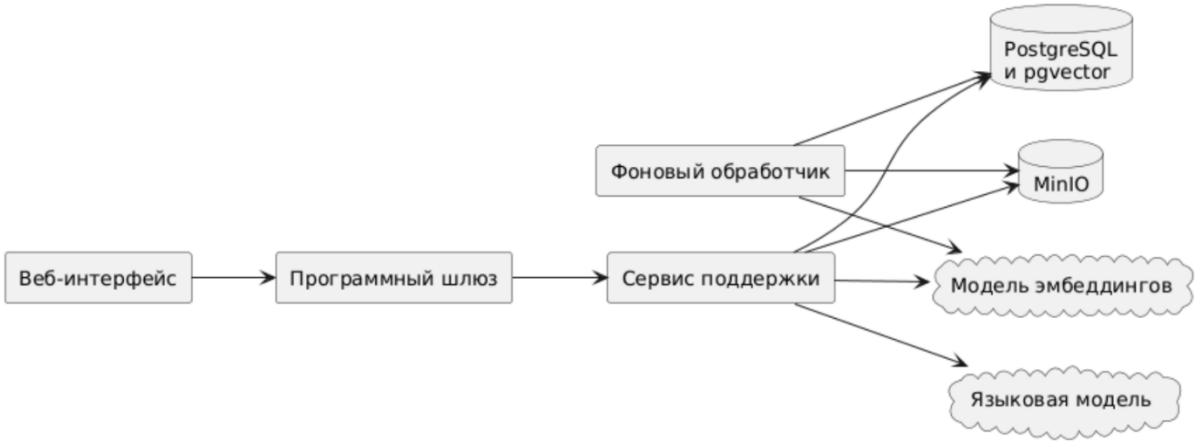


Рисунок 1 – Архитектура интеллектуальной системы технической поддержки
Figure 1 – Architecture of the intelligent technical support system

Подготовка базы знаний и построение эмбеддингов. База знаний формируется из статей в формате Markdown, исходного кода и базы уже решенных обращений пользователей. Для каждого источника вычисляется контрольная сумма SHA-256. Если содержимое или модель эмбеддингов изменяются, источник индексируется повторно. Вместе с фрагментом сохраняются его тип, заголовок, порядковый номер и идентификатор модели. Для исходного кода дополнительно указываются путь к файлу и номера строк.

Статьи разделяются на фрагменты длиной до 1800 символов с перекрытием 200 символов. Файлы исходного кода длиной до 160 строк индексируются целиком, а более крупные файлы делятся на блоки до 120 строк с перекрытием 20 строк. Одно решенное обращение сохраняется как единая смысловая единица, чтобы не разделять описание проблемы и итоговое решение.

Построение эмбеддинга определяется как отображение текста в векторное пространство размерности 384:

$$E: T \rightarrow \mathbb{R}^{384}, \tag{1}$$

где T – текст запроса или фрагмента, $E(T)$ – его векторное представление.

Для построения эмбеддингов используется внешняя модель qwen3-embedding-4b, подключенная через OpenAI-совместимый программный интерфейс [6]. Полученные векторы сокращаются до 384 измерений путем сохранения первых 384 координат, после чего нормализуются по евклидовой норме:

$$\hat{v} = \frac{v}{\|v\|_2}, \tag{2}$$

где $v \in \mathbb{R}^{384}$ – сокращенный вектор признаков, $\hat{v} \in \mathbb{R}^{384}$ – его нормализованное представление, $\|v\|_2$ – евклидова норма вектора v.

Нормализованные представления сохраняются в поле vector(384). Близость запроса q и фрагмента d определяется косинусной мерой [7]:

$$sim(q, d) = \frac{E(q) \cdot E(d)}{\|E(q)\|_2 \cdot \|E(d)\|_2}, \tag{3}$$

где q – запрос пользователя, d – фрагмент базы знаний, $E(q)$ и $E(d)$ – их векторные представления, $sim(q,d)$ – значение их косинусной близости.

Для ускорения поиска используется индекс HNSW [8]. В дальнейшее ранжирование включаются результаты с косинусной близостью не менее 0,55.

Извлечение контекста и формирование ответа. Поиск сочетает полнотекстовый и векторный методы. Полнотекстовый поиск обнаруживает прямые совпадения терминов, а векторный поиск находит смысловое соответствие между запросом и фрагментами базы знаний.

Ранжированные списки объединяются методом Reciprocal Rank Fusion [9]:

$$RRF(d) = \sum_{r \in R} \frac{1}{k + rank_r(d)}, \tag{4}$$

где R – множество списков результатов, $rank_r(d)$ – позиция фрагмента d в списке r, $k = 60$ – константа сглаживания. В итоговый контекст включаются до трех фрагментов документации и до двух решенных обращений. Для исходного кода выполняется отдельный поиск, возвращающий до четырех фрагментов. Объем текста одного результата поиска ограничивается 6000 символами.

При поступлении вопроса сервис передает большой языковой модели qwen3.6-35b-a3b инструкцию агента, историю диалога и доступные поисковые операции. Модель запрашивает сведения из документации, обращений или исходного кода, после чего

найденные фрагменты возвращаются ей вместе с метаданными. Системная инструкция требует основывать утверждения о продукте только на найденных материалах. При отсутствии подтверждающих сведений система должна сообщить о недостатке информации или предложить обращение к специалисту. Последовательность обработки показана на Рисунке 2.

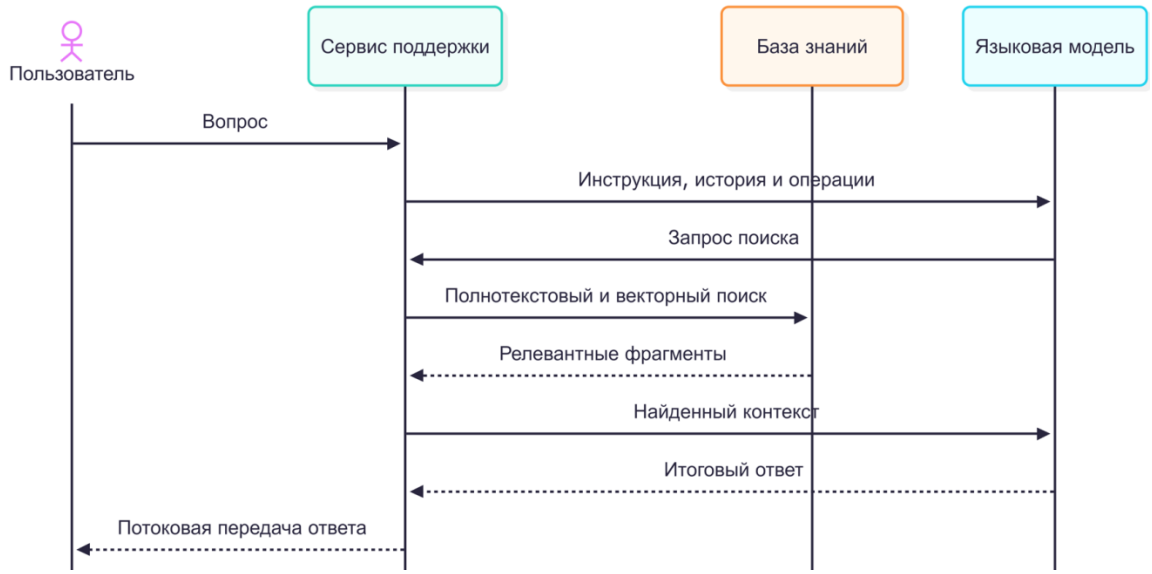


Рисунок 2 – Последовательность обработки обращения пользователя
Figure 2 – Sequence of processing a user support request

Методика функциональной проверки. Для функциональной проверки формируется 20 вопросов по пяти категориям: установка, настройка, типовые ошибки, исходный код и отсутствие информации. На каждую категорию приходится двадцать вопросов. Для вопросов первых четырех категорий заранее фиксируются ожидаемый источник и обязательные факты ответа. Для пятой категории предварительно подтверждается отсутствие ответа в базе знаний.

Каждый вопрос отправляется в отдельном диалоге при неизменной конфигурации системы. Для него сохраняются найденные источники, итоговый ответ и время обработки. Источник считается релевантным, если он содержит заранее определенные сведения. Ответ считается корректным, если передает обязательные факты, не противоречит источнику и не содержит неподтвержденных утверждений. При отсутствии информации корректным результатом считается явный отказ от фактического ответа [10].

Доля запросов с найденным релевантным источником вычисляется только для вопросов, имеющих ответ в базе знаний:

$$P_{source} = \frac{N_{relevant}}{N_{answerable}} \cdot 100\ \%, \quad (5)$$

где $N_{relevant}$ – число запросов с найденным релевантным источником, $N_{answerable}$ – число вопросов, имеющих подтвержденный ответ в базе знаний.

Общая доля корректно обработанных запросов определяется с учетом правильных отказов:

$$P_{correct} = \frac{(N_{correct} + N_{refusal})}{N_{all}} \cdot 100\ \%, \quad (6)$$

где $N_{correct}$ – число корректных содержательных ответов, $N_{refusal}$ – число корректных отказов при отсутствии информации, N_{all} – общее число проверочных вопросов.

Результаты группируются по категориям и представляются в сводной таблице. В статье подробно рассматриваются один корректный содержательный ответ и один корректный отказ. Полный набор вопросов и ответов сохраняется отдельно как материал проверки.

Результаты

Функциональная проверка проведена на базе знаний программного продукта, для которого была настроена разработанная система технической поддержки. На момент проверки база содержала 148 источников, разделенных на 201 текстовый фрагмент. В ее состав входили 4 статьи, 25 файлов исходного кода и 119 закрытых обращений пользователей.

Проверочный набор включал 100 вопросов, поровну распределенных между пятью категориями: установка, настройка, типовые проблемы, исходный код и отсутствующие сведения. Каждый вопрос отправлялся в отдельном диалоге. Часть вопросов представляла собой уточненные или перефразированные варианты запросов к одному и тому же факту. Одновременно обрабатывалось не более четырех вопросов.

Для 80 вопросов, ответы на которые присутствовали в базе знаний, заранее были определены ключевые факты. Остальные 20 вопросов относились к возможностям, сведения о которых в базе знаний не были подтверждены. В таких случаях корректным считался ответ о невозможности подтвердить информацию без последующего утверждения об отсутствии самой функции. Ответ считался некорректным, если наряду с ожидаемым фактом содержал сведения, противоречащие базе знаний, или неподтвержденные указания.

Таблица 1 – Результаты функциональной проверки по категориям
Table 1 – Functional test results by category

Категория вопросов	Число вопросов	Корректные ответы	Доля корректных ответов, %
Установка продукта	20	15	75
Настройка продукта	20	16	80
Типовые проблемы с продуктом	20	14	70
По исходному коду продукта	20	16	80
Отсутствующие сведения	20	6	30
Всего	*100*	*67*	*67*

Среди 80 вопросов с подтвержденными сведениями корректно обработан 61 запрос, что составляет 76,25 %. Хотя бы одна цитата присутствовала в ответах на 67 из 80 таких вопросов, или в 83,75 % случаев. Наличие цитаты фиксировалось отдельно и само по себе не являлось основанием для признания ответа корректным.

В категориях настройки и исходного кода корректно обработано по 16 из 20 вопросов. Для вопросов об установке получено 15 корректных ответов, для типовых проблем – 14. Ошибки включали искажение названий параметров и функций, выбор другой функции вместо указанной в источнике, а также формирование инструкций, отсутствовавших в базе знаний. В нескольких случаях ответ содержал подходящую цитату, но неверно интерпретировал приведенный в ней фрагмент. При отсутствии цитат система в отдельных ответах предлагала неподтвержденные параметры конфигурации, каталоги и сторонние компоненты.

При проверке поведения в условиях отсутствия подтверждающих сведений корректный отказ получен в 6 из 20 случаев, или в 30 %. В остальных ответах система либо преобразовывала отсутствие информации в категоричное утверждение об отсутствии функции, либо добавляла неподтвержденный способ ее настройки. В одном случае были сформированы несуществующие параметры интеграции со сторонним сервисом.

Обсуждение

Для запросов с подтвержденными сведениями доля корректных ответов составила 76,25 %. Наиболее высокие показатели получены для настройки и исходного кода – по 80 %, наиболее низкий – для типовых проблем, 70 %. Зависимость качества генерации от релевантности извлеченного контекста рассматривается в методиках оценки систем генерации с дополнением извлеченными данными [10]. Совместное применение семантического и полнотекстового поиска позволяло обрабатывать как смысловые формулировки, так и точные названия файлов, функций и сообщений об ошибках. Полученные показатели характеризуют работу системы в целом и не позволяют отдельно определить вклад каждого способа поиска, поскольку их раздельная проверка не проводилась.

Хотя бы одна цитата присутствовала в 83,75 % ответов на вопросы с подтвержденными сведениями, но корректными признаны 76,25 % ответов. В отдельных случаях система находила подходящий источник, но искажала название функции или параметра либо добавляла отсутствующую в нем инструкцию. Следовательно, наличие источника в контексте не гарантирует корректность сформированного ответа. Возможность генерации неподтвержденных сведений отмечена и при исследовании интеллектуального поиска в корпоративных информационных системах [1].

При отсутствии необходимых сведений корректный отказ сформирован в 30 % случаев. В остальных ответах отсутствие информации преобразовывалось в утверждение об отсутствии функции или дополнялось неподтвержденным способом настройки. Поэтому при оценке таких ответов необходимо учитывать не только сообщение о недостатке данных, но и последующие утверждения модели.

Ошибки могли быть связаны как с извлечением контекста, так и со способностью модели qwen3.6-35b-a3b следовать системной инструкции.

Общая доля корректно обработанных запросов составила 67 %, однако это значение зависит от состава выборки, включавшей 20 вопросов с отсутствующими сведениями. Проверка охватывала один программный продукт и одно состояние базы знаний, а часть запросов представляла разные формулировки одних фактов.

Заключение

В результате работы определена структура базы знаний и разработана архитектура интеллектуальной системы технической поддержки с изоляцией знаний отдельных программных продуктов, объединяющая обработку документации, исходного кода и обезличенных решенных обращений пользователей. Реализованы фрагментация и переиндексация источников, построение эмбеддингов, гибридный поиск и формирование ответа большой языковой моделью на основе найденного контекста.

Функциональная проверка на 100 вопросах показала, что при наличии подтвержденных сведений корректные ответы получены в 76,25 % случаев. Общая доля корректно обработанных запросов составила 67 %, а корректный отказ при отсутствии необходимых сведений сформирован в 30 % случаев. Полученные результаты подтверждают работоспособность разработанной системы и возможность ее применения

для создания специализированных агентов технической поддержки. Дальнейшее развитие системы связано с совершенствованием обработки запросов с отсутствующими сведениями и снижением числа неподтвержденных утверждений в формируемых ответах.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Антонов И.В., Бруттан Ю.В. Применение больших языковых моделей для интеллектуального поиска и извлечения информации в корпоративных информационных системах. *Компьютерные исследования и моделирование.* 2025;17(5):871–888. https://doi.org/10.20537/2076-7633-2025-17-5-871-888
   Antonov I.V., Bruttan I.V. Using RAG technology and large language models to search for documents and obtain information in corporate information systems. *Computer Research and Modeling.* 2025;17(5):871–888. (In Russ.). https://doi.org/10.20537/2076-7633-2025-17-5-871-888
2. Каширин И.Ю. Формирование LLM-ориентированных ресурсов знаний на основе генерации дополненной информации. *Вестник РГРТУ.* 2025;(94):85–97. https://doi.org/10.21667/1995-4565-2025-94-85-97
   Kashirin I.Yu. Formation of LLM-oriented knowledge resources based on generation of augmented information. *Vestnik of RSREU.* 2025;(94):85–97. (In Russ.). https://doi.org/10.21667/1995-4565-2025-94-85-97
3. Lewis P., Perez E., Piktus A., et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 06–12 December 2020, Virtual Event.* 2020. URL: https://arxiv.org/abs/2005.11401
4. Каширина И.Л., Осипов И.Р., Яковлев В.А. Разработка и оценка RAG-системы для анализа семантических связей. *Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии.* 2025;(2):114–126. https://doi.org/10.17308/sait/1995-5499/2025/2/114-126
   Kashirina I.L., Osipov I.R., Yakovlev V.A. Development and evaluation of a RAG system for analysis of semantic relations of specialised texts. *Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies.* 2025;(2):114–126. (In Russ.). https://doi.org/10.17308/sait/1995-5499/2025/2/114-126
5. Мельников А.В., Николаев И.Е., Русанов М.А., Аббазов В.Р. Сравнительный анализ методов RAG для построения русскоязычных интеллектуальных сервисов. *Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника.* 2025;25(2):5–18. https://doi.org/10.14529/ctcr250201
   Melnikov A.V., Nikolaev I.E., Rusanov M.A., Abbazov V.R. Comparative analysis of RAG methods for building Russian-speaking intelligent services. *Bulletin of the South Ural State University. Series: Computer Technologies, Automatic Control, Radioelectronics.* 2025;25(2):5–18. (In Russ.). https://doi.org/10.14529/ctcr250201
6. Zhang Y., Li M., Long D., et al. *Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models.* arXiv. URL: https://arxiv.org/abs/2506.05176 [Accessed 11th August 2026].
7. Reimers N., Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), 03–07 November 2019, Hong Kong, China.* ACL; 2019. P. 3980–3990. https://doi.org/10.18653/v1/D19-1410

8. Malkov Yu.A., Yashunin D.A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2020;42(4):824–836. https://doi.org/10.1109/TPAMI.2018.2889473
9. Cormack G.V., Clarke Ch.L.A., Buettcher S. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In: *SIGIR '09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 19–23 July 2009, Boston, MA, USA.* New York: ACM; 2009. P. 758–759. https://doi.org/10.1145/1571941.1572114
10. Es Sh., James J., Espinosa Anke L., Schockaert S. RAGAs: Automated evaluation of retrieval augmented generation. In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, 17–22 March 2024, St. Julians, Malta.* ACL; 2024. P. 150–158. https://doi.org/10.18653/v1/2024.eacl-demo.16

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Сузьмин Дмитрий Юрьевич, лаборант кафедры автоматизированных систем обработки информации и управления института электроники и светотехники, Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация.
e-mail: diman.suzmin@yandex.ru

Dmitry Yu. Suzmin, Laboratory Assistant at the Department of Automated Information Processing and Control Systems, Institute of Electronics and Lighting Engineering, National Research Ogarev Mordovia State University, Saransk, the Russian Federation.

Усманов Булат Витальевич, лаборант кафедры автоматизированных систем обработки информации и управления института электроники и светотехники, Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация.
e-mail: bulat.2003.05@mail.ru

Bulat V. Usmanov, Laboratory Assistant at the Department of Automated Information Processing and Control Systems, Institute of Electronics and Lighting Engineering, National Research Ogarev Mordovia State University, Saransk, the Russian Federation.

Егунова Алла Ивановна, кандидат исторических наук, доцент, доцент кафедры автоматизированных систем обработки информации и управления института электроники и светотехники, Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация.
e-mail: vystasalla@gmail.com

Alla I. Egunova, Candidate of Historical Sciences, Docent, Associate Professor at the Department of Automated Information Processing and Control Systems, Institute of Electronics and Lighting Engineering, National Research Ogarev Mordovia State University, Saransk, the Russian Federation.

Ямашкин Станислав Анатольевич, кандидат технических наук, доцент, доцент кафедры автоматизированных систем обработки информации и управления института электроники и светотехники, Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация.

Stanislav A. Yamashkin, Candidate of Engineering Sciences, Docent, Associate Professor at the Department of Automated Information Processing and Control Systems, Institute of Electronics and Lighting Engineering, National Research Ogarev Mordovia State University, Saransk, the Russian Federation.

e-mail: yamashkinsa@mail.ru
ORCID: 0000-0002-7574-0981

Федосин Сергей Алексеевич, кандидат технических наук, профессор, профессор кафедры автоматизированных систем обработки информации и управления института электроники и светотехники, Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация.
e-mail: fedosinsa@mrsu.ru
ORCID: 0000-0003-3481-4101

Sergey A. Fedosin, Candidate of Engineering Sciences, Professor, Professor at the Department of Automated Information Processing and Control Systems, Institute of Electronics and Lighting Engineering, National Research Ogarev Mordovia State University, Saransk, the Russian Federation.

Аббакумов Андрей Александрович, кандидат технических наук, доцент, заведующий кафедрой автоматизированных систем обработки информации и управления института электроники и светотехники, Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, Саранск, Российская Федерация.
e-mail: abbakumov_aa@mail.ru

Andrey A. Abbakumov, Candidate of Engineering Sciences, Docent, Head of the Department of Automated Information Processing and Control Systems, Institute of Electronics and Lighting Engineering, National Research Ogarev Mordovia State University, Saransk, the Russian Federation.

Статья поступила в редакцию 01.09.2026; одобрена после рецензирования 30.10.2026; принята к публикации 12.11.2026.

The article was submitted 01.09.2026; approved after reviewing 30.10.2026; accepted for publication 12.11.2026.