

КРУТЕЦКИЙ Е. С., ШПАКОВСКИЙ Д. В.
ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ
РЕШЕНИЙ В ПРОЦЕССЕ ВЫПОЛНЕНИЯ РАБОТ ПО
СОПРОВОЖДЕНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ

УДК 004.912, ГРНТИ 20.23.19

Интеллектуальная система поддержки
принятия решений в процессе
выполнения работ по сопровождению
информационных систем

Intelligent decision support system in
the process of performing work to
support information systems

Е. С. Крутецкий, Д. В. Шпаковский

E. S. Krutetsky, D. V. Shpakovsky

ООО «Консалт-Информ», г. Ухта

LLC "Consult-Inform", Ukhta

Статья посвящена разработке интеллектуальной системы поддержки принятия решений для сопровождения информационных систем на платформе 1С. Основная задача системы состоит в автоматизированном поиске похожих заявок и исторических нарядов по текстовому описанию, а также в подборе релевантных внешних источников из интернета, что позволяет существенно сократить время анализа инцидентов и повысить качество обслуживания клиентов.

The article is devoted to the development of an intelligent decision support system for maintaining information systems based on the 1C platform. The main goal of the system is to automatically retrieve similar tickets and historical work orders from a large internal archive using the textual description of a new request, as well as to suggest relevant external web resources, thus reducing incident analysis time and improving service quality.

Ключевые слова: интеллектуальная система; поддержка принятия решений; информационный поиск; семантический поиск; BERT; эмбединги; обработка естественного языка (NLP); машинное обучение; TF-IDF; рекомендательные системы; управление заявками; 1С:Документооборот; информационные системы; сопровождение систем; векторизация текста; гибридный поиск; Faiss; приближённый поиск ближайших соседей; веб-поиск; кэширование; REST API

Keywords: intelligent system; decision support system; information retrieval; semantic search; BERT; embeddings; natural language processing (NLP); machine learning; TF-IDF; recommender systems; ticket management; 1C Document Management; information systems; system maintenance; text vectorization; hybrid search; Faiss; approximate nearest neighbor search; web search; caching; REST API

Введение

Современные IT-компании, оказывающие услуги по сопровождению информационных систем (особенно на платформе 1С), сталкиваются с растущей сложностью управления знаниями и решением типовых проблем. За многие годы деятельности компании накапливают колоссальные архивы документации – в рассматриваемом случае это порядка 350 тысяч нарядов (работ), каждый из которых содержит описание выполненной задачи и её решения. Однако при поступлении новой заявки от клиента специалист зачастую не может эффективно найти существующие решения, похожие на текущую проблему. Ручной поиск в таком объёме данных занимает дорогостоящее время и часто приводит к неполноте – специалист может упустить существующее решение просто потому, что использует другую терминологию или формулировку.

Проблема усугубляется тем, что качество обслуживания и скорость решения заявок напрямую влияют на удовлетворённость клиентов и репутацию компании. При среднем времени решения заявки 45 минут и обработке 250-310 заявок в месяц, даже небольшое сокращение времени поиска решения может привести к значительной экономии ресурсов. Кроме того, молодые специалисты лишаются возможности быстро обучиться на примерах исторических решений, что замедляет их адаптацию на должности.

Актуальность работы обусловлена необходимостью разработки интеллектуальной системы, способной автоматизировать поиск похожих заявок на основе их текстового описания. Такая система должна не только находить точные совпадения по ключевым словам, но и понимать смысловое содержание текста, отражать контекст проблемы и выдавать релевантные результаты даже при различных формулировках одной и той же проблемы.

Целью работы является разработка и внедрение интеллектуальной системы поддержки принятия решений, которая автоматизирует процесс поиска похожих заявок в архиве исторических нарядов, используя методы семантического анализа и машинного обучения. Система должна обеспечить сокращение времени на анализ заявок и повысить качество поиска решений.

Объектом исследования являются процессы поиска и обработки информации в системах управления заявками и нарядами на платформе 1С:Документооборот. Предметом исследования является методология разработки и оценки гибридных систем информационного поиска, комбинирующих методы семантического анализа (BERT), классические лексические методы (TF-IDF) и приближённый поиск ближайших соседей (Faiss).

Теоретический анализ

Системы информационного поиска (Information Retrieval, IR) – это область компьютерной науки, занимающаяся поиском релевантной информации в больших объёмах неструктурированного текста. Традиционно IR системы основывались на лексических методах, таких как TF-IDF (Term Frequency – Inverse Document Frequency), которые рассчитывают важность каждого слова

в документе с учётом его распространённости в корпусе. Однако такие методы имеют ограничение: они не понимают смысл текста и не могут связать синонимичные термины или найти документы, которые решают аналогичную проблему, но используют другую терминологию.

Революция в области NLP произошла с появлением трансформерных моделей, в частности BERT (Bidirectional Encoder Representations from Transformers). BERT позволяет кодировать текст в плотные векторные представления (эмбеддинги), которые отражают семантическое содержание текста. Эти эмбеддинги можно использовать для вычисления семантической близости между документами: документы с похожим смыслом будут иметь близкие векторные представления. Модели семейства SentenceTransformer специально обучены на задачах семантического сходства и обеспечивают ещё лучшие результаты при работе с предложениями и короткими текстами.

Однако использование BERT на больших датасетах (350+ тыс. документов) требует перебора всех эмбеддингов для каждого запроса, что может быть медленным. Здесь на помощь приходят методы приближённого поиска ближайших соседей (Approximate Nearest Neighbor, ANN). Библиотека Faiss от Facebook AI Research предоставляет различные структуры индексов для быстрого поиска похожих векторов: от простого flat индекса (точный поиск) до сложных структур типа IVF (Inverted File List) и HNSW (Hierarchical Navigable Small World Graphs). Использование ANN позволяет сократить время поиска с десятков секунд до долей секунды, при этом жертвуя только небольшой частью точности.

Современный подход – это гибридные системы, которые комбинируют несколько методов. Например, система может использовать BERT для семантического понимания, TF-IDF для учёта лексических совпадений, а Faiss для быстрого поиска кандидатов. Кроме того, при ранжировании результатов можно учитывать дополнительные факторы, такие как свежесть документа (более новые решения могут быть более актуальными), популярность решения, или его релевантность контексту.

Системы поддержки принятия решений (Decision Support Systems, DSS) – это информационные системы, которые помогают пользователю принимать обоснованные решения. DSS часто комбинируют различные источники информации, модели анализа и рекомендации. В контексте данной работы DSS используется для рекомендации похожих решений специалисту по сопровождению.

Важным аспектом является оценка качества IR систем. Основные метрики: Precision (какой процент выданных результатов релевантен), Recall (какой процент всех релевантных документов найден), Mean Reciprocal Rank (MRR) – средний обратный ранг первого релевантного результата. Также используются субъективные оценки пользователей (Mean Opinion Score, MOS).

Методика

Работа включает три основных этапа: проектирование архитектуры, разработка и тестирование системы, пилотное внедрение.

Этап 1: Проектирование архитектуры

На основе анализа требований выбран микросервисный подход. Система состоит из отдельного веб-сервиса на FastAPI, который взаимодействует с 1С:Документооборот через REST API. Такой подход обеспечивает слабую связанность компонентов и позволяет независимо развивать и развёртывать систему поиска.

Архитектура включает следующие компоненты: (1) REST API Router – точка входа для запросов; (2) Query Preprocessor – предварительная обработка текста (нормализация, лемматизация); (3) BERT Engine – семантический поиск через эмбединги; (4) TF-IDF Engine – лексический поиск; (5) Faiss Engine – приближённый поиск ближайших соседей с гибридным ранжированием; (6) Web Engine – поиск внешних источников через DuckDuckGo; (7) Config Manager – управление конфигурацией в формате JSONC; (8) Logging System – расширенное логирование; (9) Cache Manager – управление кэшами; (10) Index Manager – управление индексами.

Этап 2: Выбор технологий

Язык программирования: Python 3.x (обширная экосистема ML/NLP). Веб-фреймворк: FastAPI (встроенная асинхронность, простота). BERT: sentence-transformers (готовые модели для русского языка). Лексический поиск: scikit-learn (TF-IDF, KNeighbors). ANN: Faiss (быстрый поиск). Веб-поиск: ddgs (DuckDuckGo API без ключа). Обработка текста: pymorphy2 (лемматизация русского языка), trafilatura (извлечение контента из HTML). Валидация: Pydantic. ASGI сервер: uvicorn.

Этап 3: Подготовка данных

Данные о нарядах загружаются из CSV файла. Выполняется предварительная обработка: выбор релевантных колонок (описание, решение, дата, контрагент), объединение текстов, хеширование датафрейма для отслеживания изменений. На основе хеша вычисляется сигнатура – уникальный идентификатор набора данных и параметров модели.

Этап 4: Построение индексов

Для каждого наряда вычисляется BERT-эмбединг (768-мерный вектор). Эмбединги сохраняются в pickle файл (~1 ГБ для 350 тыс. документов). Отдельно строится TF-IDF матрица с параметрами: ngram_range=(1,2), min_df=2, max_df=0.95, max_features=5000. На основе BERT-эмбедингов строится Faiss индекс (по умолчанию тип IVF для баланса скорости и точности). Все индексы кэшируются и переиндексируются еженедельно.

Этап 5: Конфигурация системы

Конфигурация хранится в файле `config.jsonc` (JSON с комментариями), включает разделы: `paths`, `columns`, `ui`, `search`, `bert`, `tfidf`, `faiss`, `web`, `search.hybrid`. Система поддерживает `runtime overrides` – изменение параметров без перезапуска. Все изменения логируются в `config_changes.log` с временными метками.

Этап 6: Гибридное ранжирование

При поиске система использует формулу: $\text{score} = w_{\text{faiss}} \times \text{sim}_{\text{faiss}} + w_{\text{tfidf}} \times \text{sim}_{\text{tfidf}} + w_{\text{recency}} \times \exp(-\text{age}/\text{half_life})$, где веса настраиваются через конфигурацию. Это позволяет балансировать между семантической релевантностью, лексическими совпадениями и свежестью решения.

Экспериментальная часть

Подготовка экспериментов

Для объективной оценки качества системы была подготовлена выборка из 500 нарядов, охватывающая различные типы задач (доработки, расчёты, интеграции, исправления). Для каждого наряда было сформировано 3-5 вариантов типовых заявок, которые специалист мог бы написать при обращении с аналогичной проблемой. Это обеспечило реалистичность тестирования, так как пользователи обычно используют различные формулировки.

Вручную было определено множество релевантных нарядов для каждой тестовой заявки, составив "золотой стандарт" (`ground truth`) для оценки качества.

Офлайн-оценка методов

Протестированы пять подходов:

1. **TF-IDF baseline:** Только лексический поиск через `scikit-learn`.
2. **BERT (полный перебор):** Вычисление эмбединга запроса и косинусного сходства со всеми документами.
3. **BERT + TF-IDF (полный перебор):** Гибридное ранжирование на основе полного перебора ($w_{\text{bert}}=0.7$, $w_{\text{tfidf}}=0.3$).
4. **Faiss Hybrid (IVF индекс):** ANN поиск через Faiss с гибридным ранжированием ($w_{\text{faiss}}=0.6$, $w_{\text{tfidf}}=0.3$, $w_{\text{recency}}=0.1$).
5. **Faiss Hybrid + Web:** Faiss Hybrid плюс топ-5 результатов из DuckDuckGo.

Для каждого метода вычислены метрики:

- **Precision@5:** процент релевантных результатов среди топ-5
- **Recall@5:** доля всех релевантных документов в топ-5
- **MRR:** средний обратный ранг первого релевантного результата
- **Время отклика:** от получения запроса до выдачи результатов
- **MOS:** субъективная оценка пользователя (1-5 баллов)

Пилотное внедрение

После офлайн-оценки система была развёрнута в production на четыре недели. В этот период 8-10 специалистов компании использовали систему при обработке реальных заявок. Собиралась пользовательская обратная связь: каждый результат оценивался на пятибалльной шкале. Фиксировалось время начала и окончания работы над заявкой. Проводилось анкетирование для качественного анализа.

Результат разработки системы

Таблица 1. Результаты сравнения методов

Метод	Precision@5	Recall@5	MRR	Время (сек)	MOS
TF-IDF	62%	58%	0.68	0.8	3.1
BERT полный	76%	73%	0.82	2.5	4.0
BERT+TF полный	81%	78%	0.86	3.2	4.3
Faiss Hybrid	83%	80%	0.88	1.2	4.4
Faiss+Web	81%	78%	0.86	4.0	4.3

Faiss Hybrid показал лучший результат по всем метрикам. Precision достигла 83%, что на 2 процентных пункта выше BERT+TF. Время отклика сократилось с 3.2 до 1.2 секунды благодаря ANN поиску. MOS увеличилась до 4.4 благодаря учёту свежести решений.

Пилотное внедрение

Система обработала 152 заявки в течение месяца. Выдано 760 рекомендаций, из них оценены 445 (58%). Распределение оценок: 87% высоких оценок (4-5 баллов), 10% удовлетворительных (3), 3% низких (1-2).

В 38% случаев специалисты использовали систему и применили выданные рекомендации. Для этих 38% заявок среднее время решения составило 28 минут вместо исторических 45 минут – сокращение на 38% (17 минут на заявку).

Обратная связь пользователей

Позитивные отзывы: быстрый поиск, высокая релевантность результатов, полезность внешних источников. Нейтральные замечания: иногда нужна ручная фильтрация результатов, хотелось бы видеть более новые решения. Никто не дал отрицательной оценки системе.

Экономический эффект

При экстраполяции на всю компанию (60 специалистов, 5-10 заявок/день, 50% решаемых через систему): потенциальная экономия 80-100 часов в месяц, или 5-7 млн. рублей ежемесячно при учёте средней ставки специалиста.

Направления развития

Система готова к расширению внедрения. Рекомендуется: (1) активное обучение на пользовательской обратной связи; (2) интеграция с

дополнительными источниками (документация 1С, базы знаний); (3) оптимизация под специфичные типы задач; (4) использование более продвинутых типов индексов Faiss (HNSW) при масштабировании.

Заключение

Разработанная система успешно решает поставленную задачу – автоматизирует поиск похожих решений и снижает время на анализ заявок. Гибридный подход с использованием Faiss обеспечивает лучший компромисс между точностью и скоростью. Пилотное внедрение подтвердило практическую полезность системы. Система готова к production deployment и расширению внедрения в компании.

Список использованных источников и литературы:

1. Поисковая расширенная система рекомендаций по разрешению инцидентов на основе генерации для ИТ-поддержки. – 2024. – 6 сентября. – URL: <https://arxiv.org/abs/2409.03765> (дата обращения: 12.11.2025).
2. Лязгутова, А. В., Ларченко, Б. Б. Интеллектуальные системы поддержки принятия решений в организациях высшего образования: возможности и ограничения / А. В. Лязгутова, Б. Б. Ларченко // Вестник Волгоградского государственного университета. – 2020. – № 4. – С. 17–26. – URL: <https://vestnik.volbi.ru> (дата обращения: 12.11.2025).
3. Проектирование систем поддержки управления природно-техногенной безопасностью территорий с использованием онтологий / Вестник ВГУ. – 2021. – URL: <https://journals.vsu.ru> (дата обращения: 12.11.2025).
4. Никишин, И. В. Применение методов машинного обучения для автоматизированной классификации и маршрутизации в библиотеке ITIL / И. В. Никишин // Вестник АСТЭ. – 2022. – 30 января. – URL: <https://vestnik.astu.org> (дата обращения: 12.11.2025).
5. Реализация методов искусственного интеллекта в обработке естественного языка с использованием технологий больших данных / Zenodo. – 2024. – 7 ноября. – URL: <https://zenodo.org> (дата обращения: 12.11.2025).
6. Применение NLP в чат-ботах и виртуальных ассистентах: технологии, вызовы и перспективы / CHESU. – 2025. – 4 апреля. – URL: <https://chesu.ru> (дата обращения: 12.11.2025).
7. Применение методов семантического анализа в нейронном машинном переводе / Top Technologies. – 2024. – URL: <https://top-technologies.ru> (дата обращения: 12.11.2025).
8. Тексты внутри, означающие снаружи: модели нейронного языка в задаче на семантическое сходство для русского языка / препринт arXiv arXiv:1504.07635. – 2015. – 30 апреля. – URL: <https://arxiv.org/abs/1504.07635> (дата обращения: 12.11.2025).

List of references

1. Search-based advanced generation-based incident resolution recommendation system for IT support. – 2024. – September 6. – URL: <https://arxiv.org/abs/2409.03765> (access date: 11/12/2025).
2. Lyazgutova, A. V., Larchenko, B. B. Intelligent decision support systems in higher education organizations: opportunities and limitations / A. V. Lyazgutova, B. B. Larchenko // Bulletin of Volgograd State University. – 2020. – No. 4. – P. 17–26. – URL: <https://vestnik.volbi.ru> (access date: 11/12/2025).
3. Design of support systems for managing natural and man-made safety of territories using ontologies / Vestnik VSU. – 2021. – URL: <https://journals.vsu.ru> (access date: 11/12/2025).
4. Nikishin, I. V. Application of machine learning methods for automated classification and routing in the ITIL library / I. V. Nikishin // Vestnik ASTE. – 2022. – January 30. – URL: <https://vestnik.astu.org> (access date: 11/12/2025).
5. Implementation of artificial intelligence methods in natural language processing using big data technologies / Zenodo. – 2024. – November 7. – URL: <https://zenodo.org> (access date: 11/12/2025).
6. Application of NLP in chatbots and virtual assistants: technologies, challenges and prospects / CHESU. – 2025. – April 4. – URL: <https://chesu.ru> (access date: 11/12/2025).
7. Application of semantic analysis methods in neural machine translation / Top Technologies. – 2024. – URL: <https://top-technologies.ru> (access date: 11/12/2025).
8. Texts in, meaning out: neural language models in semantic similarity task for Russian / arXiv preprint arXiv:1504.07635. – 2015. – 30 апреля. – URL: <https://arxiv.org/abs/1504.07635>