Н. А. Шешко

АЛГОРИТМИЧЕСКИЕ МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО СБОРА И ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

В работе исследуются алгоритмические подходы к интеллектуальной подготовке данных для систем поддержки принятия решений на базе архитектуры RAG. Рассматриваются методы преодоления ограничений больших языковых моделей, таких как «галлюцинации» и отсутствие актуальных контекстных знаний. Подробно анализируются стратегии мультимодального парсинга PDF-документов, семантического и пропозиционального сегментирования текста, а также математические основы векторизации. Особое внимание уделено методам оптимизации поиска: алгоритмам HyDE, HNSW и семантическому кэшированию, обеспечивающим высокую точность и производительность интеллектуальных агентов в корпоративных средах.

Введение

Развитие систем поддержки принятия решений (СППР) прошло путь от простых аналитических инструментов, оперирующих структурированными базами данных, до интеллектуальных экосистем, способных интерпретировать сложные неструктурированные массивы информации. Традиционные СППР, опирающиеся на жесткие алгоритмы и заранее определенные правила, сталкиваются с непреодолимыми трудностями при попытке анализа текстовых отчетов, юридических документов или научных публикаций, которые составляют до 80% корпоративных данных. Появление больших языковых моделей (LLM) открыло возможность создания «когнитивного слоя» в СППР, способного рассуждать на естественном языке, однако прямая интеграция таких моделей в бизнес-процессы выявила ряд критических ограничений.

Одной из центральных проблем является феномен «галлюцинаций», когда модель генерирует фактически неверную информацию, что недопустимо в медицине, финансах или юриспруденции. Кроме того, языковые модели имеют «горизонт знаний», ограниченный датой их обучения, и не обладают доступом к закрытой проприетарной информации организации.

Решением стала архитектура генерации с расширенным поиском (Retrieval-Augmented Generation, RAG). Данный подход трансформирует роль языковой модели из универсального справочника в интеллектуальный процессор, который сначала находит релевантные факты во внешнем доверенном источнике, а затем синтезирует ответ на их основе. Эффективность такой архитектуры критически зависит от качества этапа сбора и предварительной обработки данных, который превращает сырой контент в машиночитаемый поисковый индекс [1].

I. Архитектура и уровни подготовки данных

Интеллектуальная подготовка данных в современных СППР представляет собой многослойный конвейер. Типовая архитектура включает уровень данных, уровень индексирования (векторизации), уровень поиска и уровень генерации. На уровне данных происходит сбор информации из гетерогенных источников: клинических рекомендаций, финансовых отчетов, внутренних политик и нормативных актов. Интеллектуальный сбор предполагает не только извлечение файлов, но и их очистку от шума – рекламных баннеров, колонтитулов и дублирующихся фрагментов. Важным аспектом является сохранение контекстуальной целостности при парсинге, чтобы таблицы и списки оставались связанными структурами.

Уровень индексирования преобразует очищенные данные в векторные представления (эмбеддинги). Этот процесс позволяет перевести задачу поиска из области сопоставления ключевых слов в область семантической близости. Вместо того чтобы искать точное совпадение слова «автомобиль», система находит фрагменты со словами «транспортное средство» или «машина», понимая их смысловое родство. Математические методы кодируют нюансы контекста и отношений между объектами в многомерных пространствах.

Для эффективного хранения и управления многомерными эмбеддингами в промышленных СППР применяются специализированные векторные базы данных (Vector DB), такие как Pinecone, Milvus или Weaviate. В отличие от реляционных СУБД, векторные хранилища оптимизированы для выполнения поиска по сходству (similarity search) на аппаратном уровне [2]. Использование индексации типа IVF (Inverted File Index) в сочетании с квантованием векторов позволяет не только сократить объем занимаемой оперативной памяти, но и обеспечить горизонтальное масштабирование системы при росте объема корпоративной базы знаний до терабайтных значений. Это критически важно для СППР, работающих в режиме реального времени, где задержка при извлечении контекста напрямую влияет на скорость принятия управленческих решений

II. Мультимодальный анализ и распознавание структуры

Современные корпоративные документы обладают сложной топологией, включающей многоколоночную верстку, встроенные таблицы и графические элементы, что обуславливает необходимость перехода к глубокому мультимодальному анализу. Для детекции логических блоков, таких как заголовки, параграфы и списки, применяются алгоритмы распознавания структуры (Layout Analysis) на базе моде-

лей компьютерного зрения (например, LayoutLM или YOLOv8-doc). Использование специализированных функций парсинга, таких как partition_pdf, позволяет извлекать табличные данные в форматах HTML, JSON или Markdown, полностью сохраняя иерархическую связь между ячейками и заголовками столбцов, что критически важно для корректной последующей векторизации.

Для обработки сканированных копий документов обязательным этапом конвейера остается оптическое распознавание символов (OCR) с применением инструментов Tesseract, EasyOCR или Amazon Textract [3]. Однако современный мультимодальный подход не ограничивается извлечением текста: графики, диаграммы и фотографии подвергаются семантическому описанию (captioning) с помощью моделей типа BLIP-2 или GPT-4o. Сгенерированные текстовые метаданные индексируются вместе с основным контентом, обеспечивая возможность поиска документов по визуальному содержанию иллюстраций.

Дополнительную точность при анализе неструктурированных данных обеспечивают алгоритмы типа Table Transformer, которые преобразуют визуальную сетку таблицы в графовое представление. Это позволяет формировать целостную «карту документа», где каждый текстовый фрагмент жестко привязан к своей иерархической позиции в структуре файла. Такой подход предотвращает семантические разрывы на этапе сегментации, когда данные из разных разделов или колонок могут быть ошибочно объединены в один поисковый чанк, что в конечном итоге повышает достоверность ответов интеллектуальных агентов в СППР.

III. Алгоритмические основы очистки и векторизации

Технологический стек подготовки данных в рамках RAG-архитектуры (рис. 1) требует внедрения алгоритмических фильтров на этапе, предшествующем индексации. Ключевой задачей здесь становится интеллектуальная очистка контента от избыточности и шума.

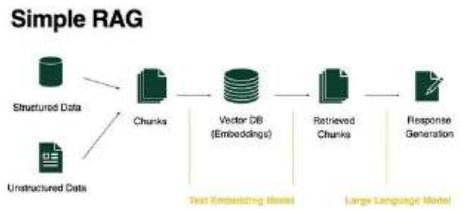

Рис. 1 – Схема конвейера обработки данных и генерации ответа (RAG)

Для борьбы с избыточностью наиболее эффективным является использование алгоритма MinHash в сочетании с мерой сходства Жаккара. Процесс реализации данного метода состоит из следующих этапов: создание признакового вектора для каждого документа (через n-граммы), применение MinHash-модели для хеширования векторов, выполнение поиска сходства на основе полученных хешей и последующая фильтрация документов, чей коэффициент сходства превышает заданный порог (например, 0.95).

Формально модель эмбеддингов можно представить как функцию g, преобразующую текст T в вектор V:

$$V = g(T) \tag{1}$$

Когда пользователь вводит запрос Q, система вычисляет вектор запроса eq и сравнивает его с векторами фрагментов знаний e_{c_j}. Основным инструментом является косинусное сходство, измеряющее угол между векторами. Формула косинусного сходства между запросом q и фрагментом c_j:

$$s(q, c_j) = \frac{e_q \times e_{c_j}}{\|e_q\|\|e_{c_j}\|} = \frac{\sum_{i=1}^{d} e_{q,i} e_{i,c_j}}{\sqrt{\sum_{i=1}^{d} e_{q,i}^2}\sqrt{\sum_{i=1}^{d} e_{c,j}^2}} \tag{2}$$

Для обеспечения работы СППР в реальном времени при объемах данных в миллионы документов прямое вычисление косинусного сходства становится вычислительно затратным. Решением является использование алгоритмов приблизительного поиска ближайших соседей (ANN), таких как HNSW (Hierarchical Navigable Small World) [4]. Это позволяет сократить время поиска с линейного до логарифмического, обеспечивая мгновенный доступ к релевантному контексту без значительной потери точности семантического сопоставления.

IV. Методы сегментации текста (Чанкинг)

Эффективность ретривера зависит от того, как текст был разделен на фрагменты (чанки). Слишком крупные фрагменты содержат избыточный шум, а слишком мелкие теряют контекст.

Алгоритм RecursiveCharacterTextSplitter использует иерархический список разделителей. Алгоритм сначала пытается разбить текст на параграфы, затем на предложения и слова. Это гарантирует, что семантически связанные части текста останутся вместе. Обязательным параметром является chunk_overlap (наложение фрагментов 10-20%), что предотвращает разрыв важных утверждений на стыке чанков.

Более интеллектуальный подход – SemanticChunker – анализирует смысл текста через эмбеддинги предложений: текст разбивается на предложения, вычисляется косинусное расстояние между эмбеддингами соседних предложений, граница фрагмента проводится там, где семантическая близость падает ниже динамического порога (например, 80-й процентиль).

Пропозициональное сегментирование – это передовой метод, где вместо исходных предложений система использует LLM («пропозиционайзер») для извлечения атомарных фактов – пропозиций. Каждая пропозиция является самостоятельным утверждением, сохраняющим смысл без контекста. Векторизация таких атомарных единиц радикально повышает

точность поиска, так как вектор запроса будет максимально близок к вектору конкретного факта.

V. Безопасность и конфиденциальность

Особое внимание при интеллектуальном сборе данных для корпоративных систем поддержки принятия решений уделяется вопросам обеспечения информационной безопасности и строгого соблюдения конфиденциальности. Внедрение RAG-архитектуры в бизнес-процессы неизбежно сопряжено с рисками утечки персональных данных (PII) или чувствительной коммерческой информации при передаче контекстных фрагментов во внешние облачные языковые модели. Для минимизации данных угроз этап предварительной обработки данных дополняется модулем автоматической анонимизации, который функционирует непосредственно перед процессом векторизации.

Технологической основой данного модуля выступают алгоритмы распознавания именованных сущностей (NER), позволяющие идентифицировать в неструктурированном тексте имена, адреса, финансовые идентификаторы и заменять их на обезличенные токены-заполнители. Такой подход позволяет сохранить семантическую структуру предложения, необходимую для качественной работы языковой модели, исключая при этом передачу реальных данных за периметр организации. Кроме того, на уровне векторных баз данных внедряются механизмы разграничения доступа на основе метаданных эмбеддингов. Это гарантирует, что при выполнении поиска по сходству система извлекает только те фрагменты знаний, к которым у конкретного пользователя есть соответствующие права доступа в рамках корпоративного комплаенса. Таким образом, предварительная обработка трансформируется из чисто технического этапа в стратегический инструмент обеспечения киберустойчивости, позволяя достичь баланса между глубиной интеллектуального анализа и требованиями стандартов защиты информации.

VI. Оптимизация поиска и качества генерации

Базовый RAG часто сталкивается с проблемами при обработке сложных вопросов. Для СППР применяются продвинутые методы, оптимизирующие этап подготовки контекста.

Эффективность интеллектуального сбора данных дополняется методами предварительной трансформации запросов (Query Transformation). Зачастую исходный вопрос пользователя в СППР является неполным или содержит неоднозначные термины, что затрудняет качественный поиск. Алгоритм HyDE (Hypothetical Document Embeddings) позволяет генерировать гипотетический ответ на вопрос, который затем используется как поисковый вектор. Поскольку вектор гипотетического ответа семантически ближе к реальным фрагментам знаний в базе данных, чем вектор исходного краткого вопроса, точность извлечения релевантных чанков возрастает на 15–20%. Дополнительно применяется метод Multi-Query, при котором исходный запрос перефразируется моделью в несколько вариаций, что позволяет охватить различные семантические пласты базы знаний и собрать более полный контекст для принятия решения [5].

Переранжирование (Reranking). Векторный поиск быстро выделяет топ-100 кандидатов, после чего используется Cross-Encoder модель для детального сопоставления вопроса с каждым из них. Это позволяет учитывать не только семантическую близость, но и структурные особенности запроса, что критически важно для юридических и технических СППР, где замена одного термина радикально меняет смысл рекомендации.

Гибридный поиск. Объединение векторного поиска (семантика) с классическим поиском, по ключевым словам, (BM25). Это позволяет системе учитывать как общий смысл, так и точные совпадения редких терминов или артикулов.

Для оптимизации эксплуатационных расходов и снижения времени отклика в высоконагруженных СППР внедряется уровень семантического кэширования запросов (Semantic Caching). В отличие от традиционного кэширования по ключу, семантический кэш сохраняет пары «запрос-ответ» и использует векторное сходство для поиска аналогичных вопросов в прошлом. Если новый запрос пользователя семантически близок к уже обработанному (например, с порогом косинусного сходства > 0.98), система мгновенно возвращает готовый ответ, минуя этапы поиска в базе знаний и генерации через LLM. Это позволяет сократить вычислительную нагрузку на инфраструктуру на 30–50% при сохранении высокой точности консультирования в рамках повторяющихся бизнес-сценариев

GraphRAG. Интеграция графов знаний. Система моделирует сущности и отношения, что помогает отвечать на вопросы, требующие синтеза информации из разных частей документов (например, связи между дочерними компаниями и транзакциями в финансовой СППР). Использование алгоритмов обнаружения сообществ (например, алгоритма Лейдена) в GraphRAG позволяет системе синтезировать ответы на глобальные вопросы по всему корпусу документов, выявляя скрытые закономерности, недоступные классическому векторному поиску.

Критическим этапом настройки конвейера в СППР является внедрение метрик автоматизированной оценки качества генерации. Использование фреймворков типа Ragas позволяет проверять систему по трем ключевым показателям: верность (соответствие ответа источнику), релевантность ответа запросу и релевантность извлеченного контекста. Такой подход позволяет математически обоснованно выбирать оптимальные параметры сегментирования (chunk size) и пороги семантической близости, минимизируя риск галлюцинаций в финальных рекомендациях.

VII. Заключение

Исследование алгоритмических методов подготовки данных подтверждает, что эффективность

современных систем поддержки принятия решений на базе RAG определяется не столько мощностью используемой языковой модели, сколько качеством формирования контекстного пространства. Переход от простых линейных конвейеров к многослойным архитектурам обработки – от мультимодального парсинга и семантического чанкинга до построения графов знаний – позволяет минимизировать риск «галлюцинаций» и обеспечить достоверность ответов в критически важных бизнес-сценариях.

Математическая точность векторных представлений в сочетании с механизмами семантического кэширования и гибридного поиска трансформирует разрозненные массивы корпоративной документации в динамическую базу знаний. Внедрение метрик автоматизированной оценки (верность, релевантность) переводит процесс настройки системы из области эмпирического подбора параметров в плоскость доказательного проектирования.

Перспективным вектором развития СППР является переход к агентским RAG-архитектурам (Agentic RAG). В этой парадигме интеллектуальный агент перестает быть пассивным потребителем контекста, обретая инструменты для автономного планирования стратегии поиска, самокоррекции и верификации фактов. Интеграция таких решений позволит бизнесу не просто извлекать информацию, а получать обоснованные рекомендации в режиме реального времени, обеспечивая необходимый баланс между глубиной интеллектуального анализа и требованиями информационной безопасности.

Список литературы

1. Lewis P. Retrieval-augmented generation for knowledge-intensive NLP tasks/ P. Lewis [et al.] // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 9459-9474.
2. Johnson J. Billion-scale similarity search with GPUs/ J. Johnson, M. Douze, H. Jégou // IEEE Transactions on Big Data. – 2019. – Vol. 7, № 3. – P. 535-547.
3. Binmakhashen G. M. Document layout analysis: a comprehensive survey/ G. M. Binmakhashen, S. A. Mahmoud // ACM Computing Surveys (CSUR). – 2019. – Vol. 52, № 6. – P. 1–36.
4. Malkov Y. A. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs/ Y. A. Malkov, D. A. Yashunin // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2018. – Vol. 42, № 4. – P. 824-836.
5. Precise zero-shot dense retrieval without relevance labels/ L. Gao [et al.] // Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. – 2023. – P. 1762–1777.

Шешко Никита Александрович, студент 4 курса факультета информационной безопасности Белорусского государственного университета информатики и радиоэлектроники, sheshko.nikita2005@gmail.com

Научный руководитель: Купрейчик Александра Сергеевна, ассистент кафедры инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники, a.kuprejchik@bsuir.by