УДК 004.912

«МЕТОДЫ РАЗДЕЛЕНИЯ И ИНДЕКСИРОВАНИЯ ТЕКСТОВЫХ ДОКУМЕНТОВ ДЛЯ СИСТЕМ НА ОСНОВЕ RAG»

Ажигалиев Дамир Русланұлы
damir.azhigaliyev@gmail.com
Магистрант 2 курса образовательной программы «Программная инженерия»
Атырауский университет имени Х. Досмухамедова, г. Атырау, Республика Казахстан
Научный руководитель, PhD, ассоц.проф. – Шармуханбет С.Р.

Аннотация. В этой статье рассматриваются современные методы разделения (разбиения) и индексирования текстовых документов, используемые при создании систем генерации ответов с расширенным извлечением информации (RAG). Был проведен сравнительный анализ шести основных стратегий группировки: фиксированного размера, рекурсивного, скользящего окна, семантического, структурного и агентного окна. Рассматриваются методы векторной, гибридной и иерархической индексации, а также алгоритмы приближения к ближайшим соседям. Описаны практические аспекты построения конвейера обработки документов и сформулированы рекомендации по выбору оптимальной стратегии в зависимости от характеристик основного документа и целевой задачи.

Ключевые слова: поиск и расширенная генерация, RAG, фрагментация, индексация, векторный поиск, встраивание, языковые модели, обработка естественного языка.

Самый простой и распространенный подход к разделению документов на фрагменты фиксированного размера. Этот метод используется для разделения текста на фрагменты определенной длины, измеряемые символами или токенами, с возможностью определения области перекрытия между соседними фрагментами. Типичный размер фрагмента составляет от 256 до 1024 токенов, а диапазон перекрытия обычно составляет от 10 до 20 процентов длины фрагмента. Преимуществом этого метода является простота реализации и предсказуемость объема полученных фрагментов, но при этом полностью игнорируется семантическая и синтаксическая структура текста, что часто приводит к разрыву логически связанных предложений и абзацев.

Рекурсивное разделение текста на символы является улучшением предыдущего подхода, в котором разделение происходит последовательно в соответствии с иерархией разделителей: сначала двойными разделителями строк (границами абзацев), затем разделителями отдельных строк, затем предложениями и, при необходимости, словами. Алгоритм рекурсивно применяет каждый последующий уровень разделения, только если результирующий фрагмент превышает заданный максимальный размер. Это обеспечивает более разумное разделение, которое учитывает естественную структуру текста, сохраняя при этом контроль над размером фрагментов.

Разделение фрагментов на скользящие окна дополняет метод фиксированного размера, создавая значительные области перекрытия, которые достигают 50 и более процентов длины фрагмента. Основная цель этой стратегии - минимизировать потерю информации на границах между соседними фрагментами. При поиске один и тот же релевантный отрывок может быть полностью включен по крайней мере в один из перекрывающихся отрывков, что увеличивает полноту извлечения. В то же время метод приводит к значительному увеличению объема индекса за счет дублирования контента, а также может генерировать избыточные результаты поиска, которые требуют дополнительной дедупликации во время последующей обработки. На практике скользящее окно часто сочетается с другими методами: например, рекурсивное разделение фрагментов

используется для разделения первичных фрагментов, а скользящее окно используется для генерации дополнительных вариантов фрагментов, охватывающих граничные области.

Семантическое деление основано на анализе семантической связности текста и формирует границы фрагментов в точках существенных изменений текстового содержания. Технически этот метод реализуется путем последовательного вычисления вложений для каждого предложения или группы предложений и определения косинусного сходства между соседними фрагментами. Если значение сходства падает ниже установленного порогового значения, в этот момент между фрагментами образуется граница. Преимуществом метода является возможность сохранения тематической целостности каждого фрагмента, что значительно повышает релевантность результатов поиска. Недостатками являются более высокие вычислительные затраты, связанные с необходимостью создания активов на этапе подготовки для каждого предложения.

Структурное разбиение на фрагменты (разбиение на фрагменты на основе структуры документа) использует явную разметку документа - заголовки, подзаголовки, абзацы, списки и таблицы - для определения границ фрагментов. Такой подход особенно эффективен при работе с документами с четкой иерархической структурой: техническими документами, научными статьями и юридическими актами. При реализации структурного разделения фрагментов документ предварительно анализируется, чтобы выделить его структурные элементы, и каждый элемент или группа связанных элементов преобразуются в отдельный фрагмент. Метаданные контекста структуры (путь в иерархии заголовков, номер раздела) добавляются к каждому фрагменту, чтобы обогатить его информацией о его местоположении во всей структуре документа.

Разделение агентства на фрагменты - это относительно новый подход, который напрямую использует языковую модель для определения оптимальных границ фрагмента. Алгоритм последовательно обрабатывает предложения исходного текста и предлагает языковой модели решить, продолжит ли следующее предложение текущую семантическую единицу или начнет новую тему. Этот метод демонстрирует высокое качество разбиения, поскольку языковая модель может учитывать глубокие семантические взаимосвязи, но его практическое применение ограничено значительными вычислительными затратами и временем обработки, что делает его непригодным для систем с большими объемами документов.

Методы разделения текстового документа на фрагменты. Простейшим и наиболее распространённым подходом к разделению документов является чанкинг по фиксированному размеру (Fixed-Size Chunking). При данном методе текст делится на фрагменты заданной длины, измеряемой в символах или токенах, с возможностью задания области перекрытия (overlap) между соседними фрагментами. Типичный размер фрагмента варьируется от 256 до 1024 токенов, а область перекрытия обычно составляет от 10 до 20 процентов длины фрагмента. Достоинством данного метода является простота реализации и предсказуемость объёма получаемых фрагментов, однако он полностью игнорирует семантическую и синтаксическую структуру текста, что нередко приводит к разрыву логически связанных предложений и абзацев.

Рекурсивный чанкинг (Recursive Character Text Splitting) представляет собой усовершенствование предыдущего подхода, при котором разделение выполняется последовательно по иерархии разделителей: сначала по двойным переносам строк (границы абзацев), затем по одинарным переносам, далее по предложениям и, при необходимости, по словам. Алгоритм рекурсивно применяет каждый следующий уровень разделителя только в том случае, если полученный фрагмент превышает заданный максимальный размер. Благодаря этому достигается более осмысленное разбиение, учитывающее естественную структуру текста, при сохранении контроля над размером фрагментов.

Чанкинг на основе скользящего окна (Sliding Window Chunking) дополняет метод фиксированного размера введением значительных областей перекрытия, достигающих 50 и более процентов длины фрагмента. Главной целью данной стратегии является

минимизация потери информации на границах между соседними фрагментами. При поиске один и тот же релевантный пассаж может оказаться полностью включённым хотя бы в один из перекрывающихся фрагментов, что повышает полноту извлечения. Вместе с тем метод приводит к существенному увеличению объёма индекса вследствие дублирования контента, а также может порождать избыточные результаты поиска, требующие дополнительной дедупликации на этапе постобработки. На практике скользящее окно часто комбинируется с другими методами: например, рекурсивный чанкинг применяется для первичного разбиения, а скользящее окно — для генерации дополнительных вариантов фрагментов, покрывающих граничные области.

Семантический чанкинг (Semantic Chunking) опирается на анализ смысловой связности текста и формирует границы фрагментов в точках существенного изменения тематики. Технически данный метод реализуется путём последовательного вычисления эмбеддингов для каждого предложения или группы предложений и определения косинусного сходства между соседними фрагментами. Если значение сходства опускается ниже установленного порога, в данной точке проводится граница между фрагментами. Достоинством метода является способность сохранять тематическую целостность каждого фрагмента, что значительно повышает релевантность результатов поиска. К недостаткам относятся более высокие вычислительные затраты, связанные с необходимостью генерации эмбеддингов для каждого предложения на этапе предобработки.

Структурный чанкинг (Document-Structure-Based Chunking) использует явную разметку документа — заголовки, подзаголовки, параграфы, списки и таблицы — для определения границ фрагментов. Данный подход особенно эффективен при работе с документами, обладающими чёткой иерархической структурой: техническая документация, научные статьи, юридические акты. Реализация структурного чанкинга предполагает предварительный синтаксический разбор документа с выделением его структурных элементов, после чего каждый элемент или группа связанных элементов преобразуется в отдельный фрагмент. Метаданные структурного контекста (путь в иерархии заголовков, номер раздела) присоединяются к каждому фрагменту, обогащая его информацией о месте в общей структуре документа.

Агентный чанкинг (Agentic Chunking) представляет относительно новый подход, при котором для определения оптимальных границ фрагментов используется непосредственно языковая модель. Алгоритм последовательно обрабатывает предложения исходного текста, предлагая языковой модели принять решение о том, продолжает ли очередное предложение текущую смысловую единицу или начинает новую тему. Данный метод демонстрирует высокое качество разбиения, поскольку языковая модель способна учитывать глубинные семантические связи, однако его практическое применение ограничивается значительными вычислительными затратами и временем обработки, что делает его малопригодным для систем с большими объёмами документов.

Методы индексации сегментов. По завершении этапа декомпозиции полученные фрагменты индексируются, что позволяет организовать данные в структуры, обеспечивающие быстрый и точный поиск. Векторное индексирование является доминирующим подходом в современных системах RAG. Каждый фрагмент текста преобразуется в числовой вектор фиксированной размерности с использованием модели встраивания, и все полученные векторы сохраняются в специальном векторном хранилище. Поиск релевантных фрагментов выполняется путем вычисления расстояния или сходства между вектором запроса и векторами памяти.

Среди алгоритмов аппроксимации (ANN) наиболее часто использовались несколько подходов. Алгоритм иерархического навигируемого малого мира (HNSW) создает многоуровневую диаграмму, в которой верхние уровни содержат разреженные соединения для быстрой глобальной навигации, а нижние уровни содержат плотные соединения для точного локального поиска. Этот алгоритм обеспечивает субмиллисекундное время отклика при обработке коллекций из миллионов документов и реализован в большинстве

современных векторных баз данных, таких как FAISS, Milvus, Weaviate и Qdrant. Алгоритм обратного индексирования файлов (IVF) использует алгоритм кластеризации для предварительного разделения векторного пространства на кластеры и доступа только к ближайшим кластерам при поиске, что значительно сокращает объем поиска. Количественное определение продукта (PQ) позволяет сжимать векторы, разбивая их на подсекторы и кодируя каждый с ограниченным набором центроидов, что снижает требования к оперативной памяти с небольшой потерей точности.

Гибридное индексирование сочетает в себе векторный поиск с классическими методами поиска информации, в частности полнотекстовой индексацией, основанной на алгоритме BM25. Эта комбинация позволяет учитывать как семантическую близость (с использованием векторных представлений), так и точное лексическое соответствие ключевых терминов (с использованием BM25). На практике результаты обоих методов оцениваются и объединяются с помощью алгоритма взаимного ранжирования (RRF) или модели обучения с перекрестным кодированием и повторным ранжированием. Гибридный подход демонстрирует неизменно более высокую полноту и точность по сравнению с любым методом, что подтверждается многочисленными экспериментальными исследованиями.

Иерархическое индексирование создает многоуровневую структуру индексов, в которой краткие описания документов или большие разделы индексируются на верхнем уровне, а подробные разделы - на нижнем. При обработке запроса поиск сначала выполняется с использованием аннотаций для идентификации соответствующих документов или разделов, а затем он детализируется на уровне определенных фрагментов. Такой подход значительно улучшает качество поиска коллекций с тематически разными документами, поскольку позволяет заблаговременно отсеять нерелевантные области содержимого.

Оптимизация параметров сегментации и индексации. Оптимизация разделения и индексации - это многоступенчатая задача, решение которой зависит от характеристик корпуса целевого документа и требований конкретного сценария применения. Одним из наиболее важных параметров является размер фрагмента. Экспериментальные данные показывают, что оптимальный размер фрагмента для вопросов с точным ответом находится в диапазоне от 256 до 512 токенов, в то время как размеры фрагмента от 512 до 1024 токенов более эффективны для задач обобщения и анализа. Степень совпадения между соседними фрагментами также играет важную роль: при отсутствии совпадения существует риск потери контекста на границах фрагмента, а при чрезмерном совпадении объем индекса увеличивается, а затраты на хранение и обработку увеличиваются.

Обогащение метаданными - эффективный способ повысить точность поиска. К каждому фрагменту можно добавить атрибуты, характеризующие источник документа, дату создания, автора, раздел, к которому относится фрагмент, а также ключевые термины, извлеченные при автоматическом выборе именованных объектов. При обработке запроса метаданные используются для предварительной фильтрации, ограничения области поиска и повышения скорости и релевантности результатов. Кроме того, включение контекстного поиска — когда к каждому фрагменту добавляется краткое описание его положения в общем контексте документа — значительно улучшает качество извлечения, что было продемонстрировано в ряде недавних исследований.

Выбор модели встраивания оказывает решающее влияние на качество векторного индекса. Современные модели, такие как BGE, E5, GTE и семейство инструментов встраивания текста OpenAI, значительно различаются по размеру генерируемых векторов, скорости работы и качеству семантического представления для разных языков и тематических областей. Для специализированных зданий (медицинская, юридическая, техническая документация) рекомендуется использовать расширенные модели встраивания, адаптированные к терминологии и стилю целевой области. Многоязычная обработка документов также является важным фактором: многоязычные модели

встраивания обеспечивают единое векторное пространство для текстов на разных языках, но их качество обычно уступает одноязычным моделям для каждого конкретного языка.

Практические аспекты внедрения конвейера обработки документов. Построение эффективного конвейера документов для системы RAG включает в себя несколько последовательных этапов, каждый из которых требует тщательного выбора инструментов и параметров. При извлечении текста из документов различных форматов (PDF, DOCX, HTML, Markdown) используются специализированные библиотеки и парсеры. Документы PDF особенно сложны, поскольку текстовый слой может отсутствовать, быть поврежден или содержать сложный макет таблицы. Эти документы обрабатываются с помощью инструментов оптического распознавания символов (OCR), а также моделей распознавания макетов, которые могут различать блоки текста, заголовки, таблицы и рисунки.

После извлечения и очистки текста выполняется этап нормализации, который удаляет повторяющиеся пробелы и служебные символы, объединяет кодировки, обрабатывает переносы и восстанавливает структуру абзаца. Качество нормализации напрямую влияет на результаты последующего разделения фрагментов и индексации, поскольку артефакты форматирования могут помешать правильному определению границ предложений и абзацев. Для многоязычных корпусов дополнительно определяется язык каждого фрагмента, что позволяет использовать модели токенизации и встраивания для конкретного языка.

Современные RAG-фреймворки, такие как LangChain, LlamaIndex и Haystack, предоставляют готовые компоненты для реализации описанных этапов обработки. Эти структуры абстрагируют детали взаимодействия с векторными репозиториями, моделями встраивания и генерирующими моделями, позволяя разработчику сосредоточиться на выборе стратегии разделения блоков и настройке параметров поиска. Это означает, что необходимо провести серию экспериментов с различными комбинациями блочных методов, моделей встраивания и алгоритмов индексирования, поскольку использование универсальных фреймворков не требует точной настройки параметров для конкретного массива документов.

Проблемы и перспективы развития.Несмотря на значительный прогресс в области группировки и индексации, ряд проблем остается нерешенным и продолжает стимулировать исследовательскую деятельность. Обработка мультимодальных документов, содержащих таблицы, графики, изображения и формулы, а также текст, требует разработки специальных стратегий группировки, которые позволяют правильно извлекать и сохранять информацию из любого представления. Существующие подходы обычно предполагают раздельную обработку текстового и табличного содержимого, что приводит к потере перекрестных ссылок между различными элементами документа.

Масштабирование RAG-систем до корпусов с миллионами документов создает серьезные технические проблемы как с точки зрения объема оперативной памяти, необходимого для хранения векторного индекса, так и с точки зрения времени отклика на поисковые запросы. Решение этой проблемы включает в себя использование комбинации методов квантования, дискового пространства для индексов и распределенных архитектур. Обеспечение релевантности индекса в динамически обновляемом корпусе документов также является нетривиальной задачей: постепенное обновление индекса при добавлении, изменении или удалении документов должно происходить без полной переиндексации и без ущерба для качества поиска.

Отдельной проблемой является формирование адекватной системы обеспечения качества для группировки и индексации. Часто используемые поисковые индикаторы - точность, отзыв и среднее взаимное ранжирование (MRR) — оценивают только результат поиска, но не выделяют вклад каждого компонента конвейера. Индикатор Recall@k измеряет долю релевантных фрагментов, включенных в первые результаты поиска k, и является наиболее актуальным для оценки стратегий группировки, поскольку неправильное

разделение приводит к распределению релевантной информации по нескольким фрагментам, и ни один из них не получает адекватной оценки. Разработка специализированных критериев, учитывающих различные типы документов и языки, остается актуальной исследовательской задачей.

Еще одна многообещающая область - интеграция методов блокирования и индексирования в диаграммы знаний. При таком подходе именованные объекты и отношения между ними извлекаются из фрагментов текста, которые затем упорядочиваются в структурированную диаграмму. При обработке запроса поиск выполняется одновременно с векторным индексом и графом знаний, что позволяет находить информацию, связанную с объектом запроса, не только семантически, но и с помощью явных фактических связей. Эта архитектура, называемая GraphRAG, демонстрирует особенно высокую эффективность при ответе на вопросы, требующие обобщения информации из нескольких документов или многоуровневой аргументации.

Перспективной разработкой является адаптивная сегментация, при которой для каждого документа или даже раздела документа автоматически выбираются параметры разбиения на основе его структурных и семантических характеристик. Исследования в области разделения обученных фрагментов направлены на построение моделей, которые могут предсказать оптимальные границы фрагментов на основе выборки с отмеченными отрывками. Разработка методов графовой индексации, в которых фрагменты связаны не только пространственной близостью векторов, но и явными семантическими отношениями (причинно-следственная, общая, частная, временная последовательность), обещает качественно новый уровень точности извлечения информации.

Заключение. Анализ методов фрагментации и индексации текстовых документов для систем на основе RAG позволяет сформулировать следующие основные выводы. Выбор стратегии разделения должен определяться типом обрабатываемых документов и целевой задачей. Для однородных текстов без четкой структуры допустимо рекурсивное разбиение на фрагменты, в то время как для структурированных документов предпочтителен структурный или семантический подход. Гибридное индексирование, сочетающее векторный поиск с лексическими методами, дает наиболее стабильные результаты в большинстве практических сценариев.

Обогащение фрагментов метаданными и контекстной информацией, а также использование многоуровневых стратегий индексирования значительно повышают качество результатов поиска. Дальнейшее развитие этой области включает разработку адаптивных методов группировки, интеграцию мультимодальной обработки документов и разработку масштабируемых архитектур, способных эффективно обслуживать здания промышленного масштаба.

Практическая значимость исследования заключается в систематизации существующих подходов и формулировании рекомендаций по выбору оптимальной стратегии документооборота в зависимости от особенностей целевого корпуса и требований прикладной задачи. Предлагаемые сравнительные характеристики методов группировки могут служить отправной точкой для специалистов, занимающихся проектированием и внедрением систем RAG в образовательных, корпоративных и исследовательских приложениях. Представляется целесообразным продолжить экспериментальные исследования с целью объединения различных методов группировки в единый конвейер обработки и разработки инструментов автоматического выбора параметров, адаптированных к специфике казахстанского информационного пространства.

Список использованной литературы:

1. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems (NeurIPS). — 2020. — Vol. 33. — P. 9459–9474.

2. Gao Y., Xiong Y., Velingker A. et al. Retrieval-Augmented Generation for Large Language Models: A Survey // arXiv preprint arXiv:2312.10997. — 2023. — 43 p.

3. Malkov Y.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. — Vol. 42, No. 4. — P. 824–836.

4. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond // Foundations and Trends in Information Retrieval. — 2009. — Vol. 3, No. 4. — P. 333–389.

5. Johnson J., Douze M., Jégou H. Billion-Scale Similarity Search with GPUs // IEEE Transactions on Big Data. — 2021. — Vol. 7, No. 3. — P. 535–547.

6. Karpukhin V., Oguz B., Min S. et al. Dense Passage Retrieval for Open-Domain Question Answering // Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). — 2020. — P. 6769–6781.

7. Izacard G., Grave E. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering // Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL). — 2021. — P. 874–880.

8. Borgeaud S., Mensch A., Hoffmann J. et al. Improving Language Models by Retrieving from Trillions of Tokens // Proceedings of the International Conference on Machine Learning (ICML). — 2022. — P. 2206–2240.

9. Wang L., Yang N., Huang X. et al. Text Embeddings by Weakly-Supervised Contrastive Pre-training // arXiv preprint arXiv:2212.03533. — 2022. — 15 p.

10. Langchain Documentation. Text Splitters // LangChain Docs. — 2024. — URL: https://docs.langchain.com/docs/modules/data_connection/document_transformers/ (дата обращения: 15.03.2026).

11. Anthropic. Contextual Retrieval // Anthropic Research Blog. — 2024. — URL: https://www.anthropic.com/news/contextual-retrieval (дата обращения: 20.03.2026).

12. Камаль Р., Ли Д. Оценка стратегий чанкинга для RAG-систем // Вестник компьютерных и информационных технологий. — 2025. — № 3. — С. 45–58.