УДК 004.89, 004.421

ПРИМЕНЕНИЕ RAG-СИСТЕМ ДЛЯ ОБРАБОТКИ БОЛЬШИХ КОРПУСОВ ДОКУМЕНТОВ В ОБРАЗОВАТЕЛЬНОЙ СРЕДЕ

Павлюченко К.А., Потоцкий Д.С.

Институт информационных технологий БГУИР, г. Минск
k.pavliuchenko@bsuir.by

Рассматриваются методы оптимизации систем извлечения дополненной генерации (RAG) при работе с большими объёмами учебных и нормативных документов. Основное внимание уделено двум этапам конвейера, определяющим качество и скорость виртуального ассистента: стратегиям разбиения документов на фрагменты и алгоритмам быстрого поиска ближайших векторов. Проанализированы метрики близости и алгоритмы приближённого поиска ближайших соседей. Приведены практические рекомендации для построения образовательных ассистентов на больших базах знаний.

Ключевые слова: RAG; векторный поиск; косинусное сходство; продуктовое квантование.

Образовательная деятельность всегда стремится использовать достижения науки, прогрессивные технологии и средства для повышения качества учебного процесса. На современном этапе всё чаще применяются интеллектуальные сервисы, в том числе виртуальные помощники, способные мгновенно отвечать на вопросы по большим массивам учебных и нормативных материалов – стандартам, методическим указаниям, конспектам и курсам. По мере роста объёма баз знаний на первый план выходят вопросы их организации и скорости поиска.

Передать в контекст языковой модели весь корпус документов невозможно: его объём многократно превышает размер контекстного окна. Технология RAG решает эту проблему, разделяя обработку запроса на два этапа: быстрое извлечение небольшого числа релевантных фрагментов из заранее проиндексированной базы и последующую генерацию ответа только на их основе [1]. Качество и скорость ассистента при этом определяются преимущественно этапом извлечения, а не самой моделью. Цель работы – систематизировать подходы к оптимизации двух ключевых компонентов этого этапа: разбиения документов на фрагменты и быстрого поиска ближайших векторов.

Разбиение документа на фрагменты (чанкинг) решающим образом влияет на полноту извлечения. Слишком крупные фрагменты размывают векторное представление и наполняют контекст модели лишним текстом. Слишком мелкие теряют смысловой контекст и увеличивают размер индекса. Основные стратегии: разбиение фиксированного размера с перекрытием соседних фрагментов на 10–20 %; рекурсивное разбиение по иерархии разделителей (абзацы, предложения, слова); структурное разбиение по границам разделов и пунктов документа; семантическое разбиение в точках смены темы; иерархическое разбиение «от малого к большому» когда для поиска индексируются мелкие фрагменты, а в контекст подставляется их крупный «родительский» фрагмент. Каждому фрагменту целесообразно присваивать метаданные (документ, раздел, страница) для точных ссылок на источник и предварительной фильтрации (таблица 1).

Таблица 1 – Стратегии разбиения документов на фрагменты

Стратегия	Характеристика
Фиксированный размер	Просто и быстро, но разрывает смысловые единицы; смягчается перекрытием
Рекурсивное	Сохраняет абзацы и предложения; разумная стратегия по умолчанию
Структурное	Целостность пунктов и точные ссылки; требует структуры документа
Семантическое	Однородные по смыслу фрагменты; высокая стоимость индексации
«От малого к большому»	Точность поиска по мелким единицам и полнота контекста крупных

Релевантность фрагмента определяется геометрической близостью его вектора к вектору запроса. Косинусное сходство (формула 1) измеряет угол между векторами и не зависит от их длины, учитывая лишь направление, которое и кодирует смысл текста – это фактический стандарт для текстовых эмбеддингов.

$$cos(a,b) = (a \cdot b) / (\|a\| \cdot \|b\|) \qquad (1)$$

Скалярное произведение $(a \cdot b)$ дешевле в вычислении, а евклидово расстояние $\|a - b\|$

измеряет геометрическую удалённость. При нормировке векторов к единичной длине все три метрики дают одинаковый порядок ранжирования. Поэтому на практике эмбеддинги нормируют однократно при индексации и далее используют скалярное произведение как наиболее быструю операцию без потери качества ранжирования.

Простейший способ поиска – вычислить близость запроса ко всем векторам базы. Он даёт стопроцентную полноту, но имеет линейную сложность порядка $N \cdot d$ (N – число фрагментов, d – размерность) и неприемлем для баз в миллионы фрагментов. Решением служит приближённый поиск ближайших соседей (Approximate Nearest Neighbor), жертвующий малой долей полноты ради ускорения на порядки.

Графовый поиск HNSW строит многослойный граф близости и спускается к запросу жадным обходом, обеспечивая почти логарифмическую сложность при высокой полноте; настраивается параметрами M и efSearch, но требует много памяти [2]. Поиск по инвертированному файлу (IVF) разбивает пространство на кластеры методом k-средних и просматривает лишь ближайшие из них.

Продуктовое квантование (PQ) сжимает векторы в короткие коды сокращая память в десятки раз, и в связке с инвертированным файлом (IVF-PQ) позволяет индексировать миллиарды векторов [3, 4]. Для повышения точности применяют двухэтапное извлечение с переранжированием кандидатов, а для запросов с точными терминами – гибридный поиск,объединяющий векторный и полнотекстовый (таблица 2).

Таблица 2 – Методы поиска ближайших векторов

Метод	Сложность поиска	Полнота	Масштаб базы
Полный перебор	Линейная	100 %	До тысяч фрагментов
HNSW	Почти логарифмическая	Высокая	Миллионы (в памяти)
IVF	Сублинейная	Настраиваемая	Десятки миллионов
IVF-PQ	Сублинейная	Умеренная	Миллиарды векторов

Для построения образовательного ассистента над большой базой знаний целесообразно применять структурное и иерархическое разбиение с перекрытием 10–20 % и метаданными. Необходимо нормировать эмбеддинги и использовать скалярное произведение, выбирать индекс по масштабу – точный перебор до тысяч фрагментов и регулировать соотношение «скорость–полнота» параметрами efSearch и nprobe. Для точных терминов добавлять гибридный поиск, а для повышения точности – переранжирование.

Таким образом, качество и скорость RAG-ассистента определяются организацией извлечения. Стратегия разбиения влияет на полноту, нормированные метрики близости взаимозаменяемы, а приближённый поиск обеспечивает работу с базами до миллиардов фрагментов. Предложенные решения имеют практическое значение для методического обеспечения учебных дисциплин в непрерывном профессиональном образовании: RAG-ассистент, построенный над выверенной базой учебно-методических и нормативных материалов позволяет оперативно сопровождать обучающихся и преподавателей, поддерживать актуальность учебного контента и снижать нагрузку при его сопровождении на разных ступенях обучения.

Литература

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. Malkov, Yu. A. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs / Yu. A. Malkov, D. A. Yashunin // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2020. — Vol. 42, № 4. — P. 824–836.

3. Jégou, H. Product Quantization for Nearest Neighbor Search / H. Jégou, M. Douze, C. Schmid // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2011. — Vol. 33, № 1. — P. 117–128.

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