Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 002.304

СРАВНИТЕЛЬНЫЙ И КРИТИЧЕСКИЙ АНАЛИЗ АРХИТЕКТУР RETRIEVAL-AUGMENTED GENERATION: МЕТОДОЛОГИЧЕСКИЕ ОСНОВАНИЯ И ПРАКТИЧЕСКИЕ АСПЕКТЫ ПРИМЕНЕНИЯ GRAPHRAG, KAG, CAG, HYDE, FID И ATLAS

Валадов Антон Сергеевич,
студент, магистрант
2 курс, факультет "Отдел магистратуры", кафедра «Институт сквозных технологий»
Донской государственный технический университет
Россия, г. Ростов-на-Дону
coolmrant@yandex.com

Павловский Денис Валерьевич,
студент, магистрант
2 курс, факультет "Отдел магистратуры", кафедра «Институт сквозных технологий»
Донской государственный технический университет
Россия, г. Ростов-на-Дону
PavlovskiyD.V@yandex.ru

Аннотация

Retrieval-Augmented Generation (RAG) рассматривается как практический способ повысить обоснованность ответов больших языковых моделей за счёт обращения к внешним источникам знаний. В работе выполнен сравнительный и критический анализ архитектур GraphRAG, Knowledge-Augmented Generation (KAG), Cache-Augmented Generation (CAG), Hypothetical Document Embeddings (HyDE), Fusion-in-Decoder (FiD) и Atlas. Сопоставление проведено по ряду прикладных критериев: точность ответов, устойчивость к галлюцинациям, задержка и вычислительные затраты, оперативность обновления источников и сложность интеграции. Показано, что универсально лучшего решения не существует: FiD и Atlas обеспечивают высокое качество на сложных запросах ценой ресурсоёмкости и инженерной сложности; GraphRAG эффективен в доменах с выраженными связями между сущностями; HyDE повышает полноту извлечения при отсутствии данных для обучения ретривера; CAG снижает латентность при условии ограниченного и относительно статичного корпуса; KAG минимизирует ошибки за счёт логического вывода, но имеет более узкую применимость. Сформулированы практические рекомендации по выбору варианта RAG с учётом ограничений инфраструктуры и динамики базы знаний, включая сценарии использования в русскоязычных правовых и медицинских системах.

Ключевые слова: Retrieval-Augmented Generation; большие языковые модели; семантический поиск; граф знаний; логический вывод; кэширование контекста; Fusion-in-Decoder.

COMPARATIVE AND CRITICAL ANALYSIS OF RETRIEVAL-AUGMENTED GENERATION ARCHITECTURES: METHODOLOGICAL FOUNDATIONS AND PRACTICAL ASPECTS OF GRAPHRAG, KAG, CAG, HYDE, FID, AND ATLAS APPLICATIONS

Valadov Anton Sergeevich,
Student, Master's degree candidate
2nd year, Faculty of Master's Studies, Department of the Institute of End-to-End Technologies
Don State Technical University
Rostov-on-Don, Russia

Pavlovskij Denis Valerevich,
Student, Master's degree candidate
2nd year, Faculty of Master's Studies, Department of the Institute of End-to-End Technologies
Don State Technical University
Rostov-on-Don, Russia

ABSTRACT

Retrieval-Augmented Generation (RAG) is widely used to improve the factual grounding of large language models by retrieving relevant evidence from external knowledge sources. This paper provides a comparative and critical review of GraphRAG, Knowledge-Augmented Generation (KAG), Cache-Augmented Generation (CAG), Hypothetical Document Embeddings (HyDE), Fusion-in-Decoder (FiD), and Atlas. The approaches are evaluated using practice-oriented criteria: answer accuracy, hallucination robustness, latency and computational cost, knowledge freshness, integration complexity, and suitability for Russian-language data. The analysis shows that no single architecture dominates across all dimensions: FiD and Atlas typically deliver strong performance on complex queries at the expense of higher compute and engineering overhead; GraphRAG is advantageous when relational structure between entities is essential; HyDE improves retrieval coverage in low-supervision settings; CAG reduces latency for bounded and relatively static corpora; KAG relies on structured reasoning and can minimize hallucinations but is less universal. Based on these findings, we outline selection guidelines for RAG variants under different operational constraints, including Russian legal and medical document retrieval scenarios.

Keywords: Retrieval-Augmented Generation; large language models; semantic retrieval; knowledge graphs; logical reasoning; context caching; Fusion-in-Decoder; Russian-language embeddings.

Введение

Архитектура Retrieval-Augmented Generation (RAG) была изначально предложена для повышения фактичности и обоснованности ответов больших языковых моделей за счёт интеграции механизмов извлечения релевантных фрагментов из внешних источников знаний. Такая интеграция уменьшает зависимость от параметрической памяти модели и позволяет использовать актуальную информацию на момент инференса. RAG применяется во множестве доменов, включая здравоохранение, где он демонстрирует повышение

точности и согласованности ответов за счёт условной генерации, основанной на достоверных источниках и объективных данных [7].

У базового RAG есть инженерные и качественные ограничения: поиск добавляет задержку и усложняет систему; итоговый ответ зависит от качества извлечения (нерелевантные или неполные документы приводят к ошибкам, качество векторного представления напрямую влияет на эффективность ретривера. Эти проблемы особенно заметны в доменах, где требуется проверяемость и высокая точность (право, медицина).

Цель работы — сравнить и критически оценить архитектуры GraphRAG, KAG, CAG, HyDE, FiD и Atlas. Сопоставление проводится по прикладным критериям: точность, устойчивость к галлюцинациям, ресурсоёмкость/задержка, оперативность обновления источников и сложность внедрения. дополнительно рассматриваются кейсы использования в юридических и медицинских системах.

Критерии и методика сравнительного анализа

Для сопоставления подходов GraphRAG, KAG, CAG, HyDE, FiD и Atlas используются следующие прикладные критерии: (1) точность и полнота ответов на фактологические и многосвязные запросы; (2) устойчивость к галлюцинациям (насколько метод «заземляет» ответ на извлечённые знания); (3) ресурсоёмкость и производительность (дополнительная задержка на retrieval, требования к памяти и вычислениям, рост стоимости при увеличении числа документов); (4) оперативность обновления источников (чувствительность к обновлению базы/индекса, риск устаревания контекста); (5) сложность интеграции и поддержки (число компонентов, инфраструктурные требования, необходимость дообучения и/или экспертной разметки);

Методика. Оценивание выполняется в качественной форме по шкале “+++ / ++ / + / ± / –”, где более высокий знак соответствует лучшему проявлению свойства, а “–” означает слабую выраженность или отсутствие преимущества по критерию. Итоговые оценки сведены в сравнительную таблицу; они опираются на результаты, описанные в источниках и отчётах, а также на анализ практических сценариев внедрения.

Правило интерпретации результатов. При сравнении фиксируется не «абсолютно лучший» метод, а компромисс между качеством ответа, стоимостью вычислений и сложностью внедрения.

Обзор и критический разбор архитектур RAG

Классический RAG

Retrieval-Augmented Generation (RAG) — это архитектура, в которой генерация текста языковой моделью усиливается за счёт извлечения релевантных фрагментов из внешней базы знаний в момент инференса. Такой подход позволяет модели обращаться к актуальным и глубоко релевантным данным вне параметрической памяти, что повышает фактичность и обоснованность ответов по сравнению с моделями, опирающимися только на предобученные параметры. Помимо увеличения точности, RAG снижает риск генерации неверных утверждений (hallucinations) за счёт использования проверяемых внешних источников [6].

Общая схема представлена на рисунке 1.

Рисунок 1 – индексация и ответ на запрос RAG

Сильная сторона — минимальная инженерная сложность и возможность обновлять знания обновлением корпуса и индекса без переобучения модели. Основная слабость — зависимость от качества retrieval: нерелевантные/неполные фрагменты напрямую ухудшают ответ, а увеличение top-k повышает стоимость и латентность из-за роста контекста.

GraphRAG

Graph Retrieval-Augmented Generation (GRAG) расширяет классическую архитектуру Retrieval-Augmented Generation за счёт использования графовых структур для организации процесса извлечения знаний. В GRAG документы и сущности представляются в виде графа, что позволяет выполнять многошаговый retrieval с учётом связей между найденными фрагментами и их контекстом. Такой подход обеспечивает более связный и полный контекст для генерации ответа при обработке сложных запросов, требующих композиционного рассуждения и объединения информации из нескольких источников [2].

Общая схема представлена на рисунке 2.

Рисунок 2 – индексация и ответ на запрос GraphRAG

Преимущество — более устойчивое покрытие контекста в «связанном» домене (персоны–организации–события), а также возможность целенаправленного поиска по графовым окрестностям. Критический риск — высокая цена подготовки и сопровождения: нужны извлечение сущностей, корректное связывание (entity linking) и регулярное обновление графа; ошибки на этих этапах приводят к систематическим провалам retrieval. Метод оправдан, когда домен структурирован и связи важнее «похожести текста».

KAG (Knowledge-Augmented Generation)

KAG делает акцент на формализованные знания и вывод: ответ строится не только по извлечённым фрагментам, но и через правила/онтологии/структуры, которые ограничивают пространство допустимых утверждений. Это снижает вероятность «догадки»

модели и повышает проверяемость в задачах, где важна логическая корректность. Общая схема представлена на рисунке 3.

Подготовка знаний (offline)

Главный недостаток — узкая универсальность и трудоёмкая подготовка: нужно формализовать знания, поддерживать согласованность схемы и правил, обеспечивать качество фактологической базы. KAG целесообразен там, где знания можно поддерживать как систему (регламенты, классификаторы, нормы) и где критична объяснимость.

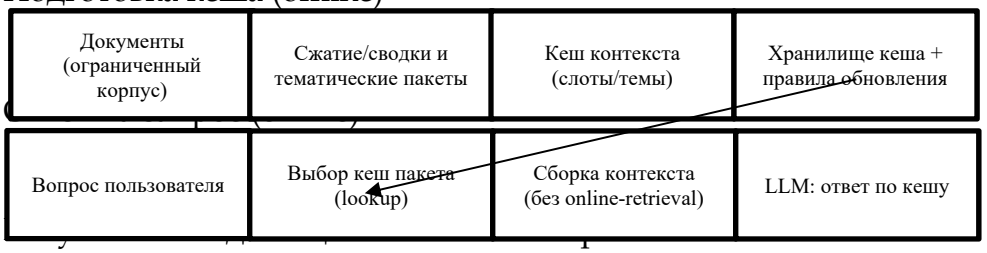
CAG (Cache-Augmented Generation)

CAG (Cache-Augmented Generation) представляет собой альтернативный подход к Retrieval-Augmented Generation, при котором релевантный корпус знаний предварительно загружается в расширенный контекст языковой модели, а состояние key-value кеша вычисляется до этапа инференса. Отказ от online-retrieval позволяет снизить задержку генерации и упростить архитектуру системы за счёт исключения компонента поиска во время ответа на запрос [1].

Подход наиболее эффективен в сценариях с ограниченным и управляемым корпусом знаний, однако чувствителен к обновлению данных и требует пересборки кеша при изменении источников [1].

Общая схема представлена на рисунке 4.

Подготовка кеша (offline)



Плюс — низкая латентность и предсказуемая стоимость ответа, что важно для сервисов с жёсткими SLA или ограниченным бюджетом на retrieval. Минус — актуальность: при частых изменениях корпуса кэш быстро устаревает и требует пересборки; кроме того, метод хуже покрывает «длинный хвост» редких вопросов, если они не предусмотрены структурой кэша. CAG логичен для ограниченных и относительно статичных баз знаний.

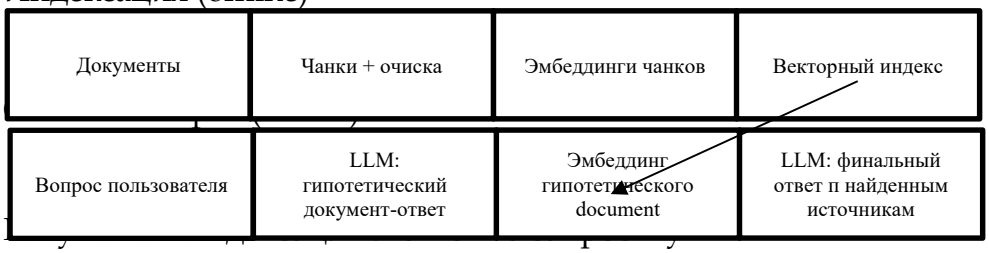
HyDE (Hypothetical Document Embeddings)

Hypothetical Document Embeddings (HyDE) представляет собой приём улучшения dense retrieval, при котором перед этапом поиска языковая модель генерирует гипотетический текст, семантически близкий предполагаемому ответу на запрос. Эмбеддинг этого текста используется в качестве поискового запроса, что позволяет снизить влияние лексического расхождения между формулировкой вопроса и документами корпуса.

Такой подход повышает полноту извлечения в условиях отсутствия обучающих данных для ретривера и при работе с редкими или абстрактными запросами. Вместе с тем

HyDE чувствителен к качеству сгенерированного гипотетического текста: семантическое смещение на этапе генерации может приводить к систематическому извлечению нерелевантных документов, что требует дополнительного ранжирования или фильтрации результатов [4].

Общая схема представлена на рисунке 5.



Практический эффект — рост полноты извлечения при разрыве лексики вопроса и корпуса и при отсутствии данных для обучения ретривера. Риск HyDE — перенос ошибок генерации на retrieval: если гипотеза смещает смысл, то извлекаются нерелевантные источники, и ошибка закрепляется. Поэтому метод требует контроля промпта и фильтрации/переранжирования результатов.

FiD (Fusion-in-Decoder)

В классическом RAG найденные фрагменты обычно объединяются в один общий контекст и целиком подаются в LLM. Это ухудшает работу на запросах, где ответ требует агрегации нескольких независимых источников: в длинном контексте часть фактов теряется, а модель нередко выбирает один «самый похожий» фрагмент и игнорирует остальные.

Fusion-in-Decoder (FiD) представляет собой архитектуру Retrieval-Augmented Generation, в которой извлечённые документы кодируются независимо, а декодер агрегирует полученные представления для формирования итогового ответа. Такой подход повышает устойчивость генерации на сложных вопросах, однако сопровождается значительным ростом вычислительных затрат на этапе инференса.

Для снижения этой нагрузки в современных работах предлагаются методы оптимизации инференса FiD. В частности, используется механизм dynamic token pruning, при котором на этапе генерации отбрасываются токены с низким вкладом в формирование ответа. Это позволяет существенно сократить время инференса при работе с длинными контекстами и многоабзацными ответами при минимальном снижении качества по сравнению с базовой архитектурой FiD [3]. Общая схема представлена на рисунке 6.

Преимущество — устойчивость на сложных вопросах, где нужно «сшить» ответ из нескольких документов, и меньшая зависимость от одного идеального фрагмента. Недостаток — высокая ресурсоёмкость: стоимость растёт с числом документов, увеличиваются требования к памяти и времени инференса. FiD оправдан, когда качество важнее задержки и требуется надёжная агрегация источников.

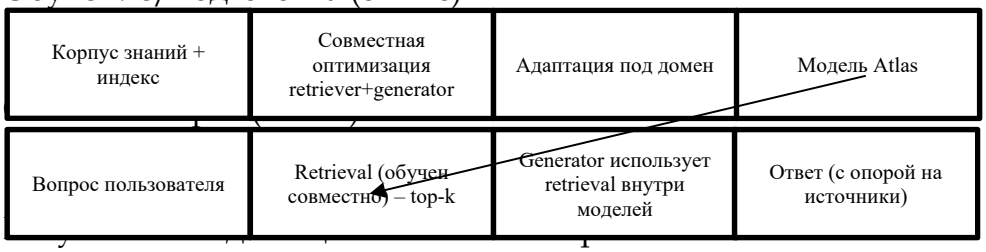
Atlas

Atlas представляет собой retrieval-augmented языковую модель, в которой компоненты извлечения и генерации обучаются и используются совместно в рамках единой архитектуры. В отличие от классического RAG, где retriever выступает как внешний и фиксированный модуль, Atlas оптимизирует процесс извлечения так, чтобы выбранные документы непосредственно способствовали повышению качества генерации ответа.

Совместная оптимизация retriever и генератора позволяет более эффективно подбирать подтверждающий контекст в few-shot сценариях и при обработке сложных запросов, требующих точного соответствия между вопросом и источниками знаний. Вместе с тем такая интеграция увеличивает вычислительную сложность обучения и эксплуатации модели, а обновление корпуса знаний требует согласованного пересчёта индекса и контроля деградации качества по сравнению с более простыми архитектурами RAG [5].

Общая схема представлена на рисунке 7.

Обучение/подготовка (offline)



В практическом смысле Atlas отличается тем, что связка «retriever + generator» может обучаться/адаптироваться совместно, то есть retrieval оптимизируется под качество финального ответа. Это обычно повышает устойчивость на сложных запросах и снижает зависимость от ручного тюнинга (например, выбора k и правил ранжирования), но резко повышает требования к внедрению: нужен корпус, инфраструктура обучения или тонкой настройки, контроль обновления индекса и мониторинг деградации при смене домена. Поэтому Atlas оправдан в случаях, когда система строится под конкретный класс задач и есть возможность инвестировать в обучение и сопровождение; для «быстрого внедрения» чаще выбирают классический RAG.

Сравнительные результаты и обсуждение

Таблица 1 представляет качественное сопоставление подходов GraphRAG, KAG, CAG, HyDE, FiD и Atlas по прикладным критериям: (1) точность/полнота, (2) устойчивость к галлюцинациям, (3) ресурсоёмкость/производительность, (4) оперативность обновления источников, (5) сложность интеграции и поддержки. Оценки приведены по шкале “+++ / ++ / + / ± / –”, где большее значение соответствует лучшему результату по критерию; для критериев (3) и (5) “лучше” означает меньшие затраты и более простую эксплуатацию.

Критерий	RAG	GraphRAG	KAG	CAG	HyDE	FiD	Atlas
1) Точность и полнота	+	++	++	+	++	+++	+++
2) Устойчивость к галлюцинациям	±	++	+++	+	±	++	++
3) Ресурсоёмкость	+	±	±	+++	+	–	±

Критерий	RAG	GraphRAG	KAG	CAG	HyDE	FiD	Atlas
и производительность (меньше затрат = лучше)							
4) оперативность обновления источников	++	++	±	–	++	+ +	++
5) Сложность интеграции и поддержки (проще = лучше)	++ +	±	–	++	+	–	–

Таблица 1 - качественное сопоставление подходов GraphRAG, KAG, CAG, HyDE, FiD и Atlas

Результаты сравнения

Точность и полнота: По качеству ответа явными лидерами являются FiD и Atlas (оба на уровне +++), поскольку именно эти подходы лучше других извлекают пользу из нескольких источников: FiD конструктивно рассчитан на агрегацию доказательств из top-k документов без “смешивания в кашу”, а Atlas выигрывает за счёт более согласованной связки поиска и генерации, когда retrieval подбирается так, чтобы реально поддерживать итоговый ответ. Уровень ++ у GraphRAG, KAG и HyDE отражает то, что они способны существенно улучшать качество, но их эффект более условен: GraphRAG прибавляет там, где важны связи между сущностями, KAG — там, где знание можно формализовать и использовать вывод, HyDE — когда стандартный retrieval плохо “цепляет” нужные формулировки. Классический RAG и CAG остаются на уровне +: первый ограничен качеством ретривера и потерями на длинном контексте, второй — тем, что отвечает по заранее подготовленному контексту и плохо покрывает сложные “составные” запросы.

Устойчивость к галлюцинациям: Наиболее сильный результат демонстрирует KAG (+++), поскольку он ограничивает генерацию рамками формализованных знаний и вывода, снижая вероятность “догадки” без опоры на факты. GraphRAG, FiD и Atlas (++) уменьшают риск за счёт более надёжного grounding: GraphRAG — через использование графовых связей и структурированных фактов, FiD и Atlas — через лучшее подтверждение ответа несколькими релевантными источниками. Однако в этих схемах сохраняется зависимость от качества извлечения: при неполных или нерелевантных материалах ошибка может попасть в ответ. Классический RAG и HyDE (±) более уязвимы: RAG — из-за промахов retrieval, HyDE — дополнительно из-за риска неверной “гипотезы”, которая уводит поиск и подсовывает неправильные источники. CAG (+) выглядит стабильнее лишь при условии, что кэш действительно содержит нужные факты; при пробелах или устаревании кэша модель склонна компенсировать недостающее генерацией.

Ресурсоёмкость и производительность (меньше затрат = лучше): Безусловный победитель — CAG (+++), поскольку он минимизирует онлайн-поиск и отвечает по заранее подготовленному контексту, обеспечивая низкую задержку и предсказуемую стоимость. Классический RAG и HyDE находятся на уровне +: RAG требует выполнения retrieval при каждом запросе, а HyDE добавляет к этому шаг генерации гипотетического текста, что

увеличивает накладные расходы. GraphRAG, KAG и Atlas (±) проигрывают по производительности из-за усложнения контура: графовые операции и сопровождение графа (GraphRAG), вывод и поддержка структурированных знаний (KAG), более тяжёлая интеграция retrieval с генерацией и инфраструктурные требования (Atlas). FiD имеет наихудшую оценку (–), поскольку вычислительная стоимость растёт почти линейно с числом документов top-k, которые система вынуждена обрабатывать раздельно.

Оперативность обновления источников: Наиболее практичны RAG, GraphRAG, HyDE, FiD и Atlas (все ++): они опираются на внешние источники, и при корректно организованном обновлении корпуса и индекса достаточно быстро начинают учитывать новые данные без переобучения генератора. KAG (±) обновляется медленнее по причине природы формализованных знаний: правила, онтологии и согласованность структуры требуют более осторожного сопровождения и часто частично ручных процедур. CAG (–) закономерно хуже всех, поскольку кэш фиксирует состояние знаний на момент подготовки и без пересборки продолжит "видеть" старую версию даже при обновлённых документах.

Сложность интеграции и поддержки (проще = лучше): Самый простой и устойчивый в эксплуатации вариант — классический RAG (+++): компоненты типовые, инженерные риски понятны, поддержка сводится к обновлению индекса и контролю retrieval. CAG (++) относительно прост в онлайн-части, но требует дисциплины пересборки кэша, иначе система быстро деградирует по актуальности. HyDE (+) добавляет умеренную сложность за счёт шага генерации гипотезы и необходимости контролировать "увод" retrieval, но остаётся сравнительно компактным по инфраструктуре. GraphRAG (±) усложняется из-за отдельного графового контура (извлечение сущностей, связывание, обновление графа). Наиболее трудоёмкими остаются KAG, FiD и Atlas (–): KAG требует формализации знаний и механизма вывода, FiD — существенных вычислительных ресурсов и оптимизации инференса, Atlas — сложного контура обучения/адаптации и эксплуатации, где retrieval тесно связан с генерацией.

Заключение

В ходе работы рассмотрены и сопоставлены ключевые варианты Retrieval-Augmented Generation, различающиеся тем, как именно они организуют извлечение и использование внешних знаний в процессе генерации ответа. Наиболее "сильными" по суммарному эффекту для сложных задач являются архитектуры, которые либо лучше интегрируют несколько источников, либо глубже связывают поиск с генерацией.

На верхнем уровне по совокупности преимуществ находятся FiD и Atlas. FiD наиболее последовательно решает типичную проблему RAG — необходимость собирать ответ из нескольких независимых фрагментов: разделение обработки источников и последующее объединение при генерации повышает устойчивость на многодокументных вопросах, где простая "склейка" контекста часто теряет факты. Atlas концептуально близок по цели, но реализует её через более тесную связку поиска и генерации: retrieval встраивается в модельный контур и может быть согласован с генератором, что потенциально даёт стабильное качество при корректной подготовке корпуса и эксплуатации. При этом оба подхода предъявляют повышенные требования к инфраструктуре: FiD — прежде всего по вычислительным ресурсам, Atlas — по сложности внедрения и сопровождения.

Следующий уровень составляют методы, усиливающие RAG за счёт структуры знаний или улучшения retrieval без тяжёлой перестройки модели: GraphRAG, KAG и HyDE. GraphRAG наиболее оправдан в доменах, где критичны связи между сущностями и требуется не просто "похожий текст", а контекст, собранный по отношениями (например, "кто связан с чем и почему"). KAG выделяется как наиболее строгий подход по контролю

фактов: он полезен там, где знания можно формализовать и поддерживать в виде правил/онтологий, однако его применимость ограничивается высокой трудоёмкостью подготовки и сопровождения структурированной базы знаний. HyDE выступает как сравнительно лёгкое усиление retrieval, особенно когда стандартный семантический поиск плохо находит релевантные формулировки; однако метод требует аккуратной настройки, так как неверная гипотеза способна систематически уводить поиск в неправильную область.

Наиболее "прагматичные" и эксплуатационно простые решения — классический RAG и CAG, но они решают разные задачи. Классический RAG остаётся рациональной базой для большинства внедрений благодаря минимальной сложности интеграции и понятному циклу обновления знаний через переиндексацию. Его ограничение — более слабая работа на составных запросах и высокая чувствительность к качеству retrieval. CAG, напротив, выигрывает по скорости и предсказуемости стоимости ответа, поскольку опирается на заранее подготовленный контекст, но именно это делает его наиболее уязвимым к устареванию: при динамичных источниках требуется регулярная пересборка кэша, иначе качество деградирует.

Таким образом, выбор варианта RAG следует рассматривать как инженерное решение, зависящее от профиля задачи: для максимально качественных ответов на сложных запросах предпочтительны FiD/Atlas, для доменов со структурированными знаниями — GraphRAG/KAG, для "быстрого усиления" retrieval — HyDE, для типовых прикладных систем — классический RAG, а для сценариев с жёсткими требованиями к задержке и стабильной базой знаний — CAG.

Список литературы:

1. Chan B. J., Chen C.-T., Cheng J.-H., Huang H.-H., "Don't Do RAG: When Cache-Augmented Generation is All You Need for Knowledge Tasks," Companion Proc. ACM Web Conf. 2025, Sydney, Australia, 2025. DOI: 10.1145/3701716.3715490.
2. Hu Y., Lei Z., Zhang Z., Pan B., Ling C., Zhao L. GRAG: Graph Retrieval-Augmented Generation // *Findings of the Association for Computational Linguistics: NAACL 2025*. Albuquerque, New Mexico, USA, Apr.–May 2025. DOI: 10.18653/v1/2025.findings-naacl.232
3. Kim W., Kim G., Kang S. Accelerating Inference in Retrieval-Augmented Generation Models for Long-Form QA via Dynamic Token Pruning // Mathematics. 2025. Vol. 13, No. 14. Art. 2231. DOI: 10.3390/math13142231.
4. Gao L., Ma X., Lin J., Callan J. Precise Zero-Shot Dense Retrieval without Relevance Labels Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)DOI: 10.18653/v1/2023.acl-long.99
5. Izacard G., Lewis P., Lomeli M., Hosseini L., Petroni F., Schick T., Dwivedi-Yu J., Joulin A., Riedel S., Grave E. Atlas: Few-shot Learning with Retrieval Augmented Language Models // Journal of Machine Learning Research. 2023. Vol. 24:251:1-251:43. DOI: https://arxiv.org/abs/2208.03299
6. Klesel M., Wittmann H. F. Retrieval-augmented generation // Business & Information Systems Engineering. 2025. DOI: 10.1007/s12599-025-00945-3
7. Amugongo L. M., Mascheroni P., Brooks S., Doering S., Seidel J. Retrieval-augmented generation for large language models in healthcare: a systematic review // Annals of Biomedical Engineering. 2025. DOI: 10.1371/journal.pdig.0000877

References:

1. Chan B. J., Chen C.-T., Cheng J.-H., Huang H.-H., "Don't Do RAG: When Cache-Augmented Generation is All You Need for Knowledge Tasks," Companion Proc. ACM Web Conf. 2025, Sydney, Australia, 2025. DOI: 10.1145/3701716.3715490.
2. Hu Y., Lei Z., Zhang Z., Pan B., Ling C., Zhao L. GRAG: Graph Retrieval-Augmented Generation // Findings of the Association for Computational Linguistics: NAACL 2025. Albuquerque, New Mexico, USA, Apr.–May 2025. DOI: 10.18653/v1/2025.findings-naacl.232
3. Kim W., Kim G., Kang S. Accelerating Inference in Retrieval-Augmented Generation Models for Long-Form QA via Dynamic Token Pruning // Mathematics. 2025. Vol. 13, No. 14. Art. 2231. DOI: 10.3390/math13142231.
4. Gao L., Ma X., Lin J., Callan J. Precise Zero-Shot Dense Retrieval without Relevance Labels Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023) DOI: 10.18653/v1/2023.acl-long.99
5. Izacard G., Lewis P., Lomeli M., Hosseini L., Petroni F., Schick T., Dwivedi-Yu J., Joulin A., Riedel S., Grave E. Atlas: Few-shot Learning with Retrieval Augmented Language Models // Journal of Machine Learning Research. 2023. Vol. 24:251:1-251:43. DOI: https://arxiv.org/abs/2208.03299
6. Klesel M., Wittmann H. F. Retrieval-augmented generation // Business & Information Systems Engineering. 2025. DOI: 10.1007/s12599-025-00945-3
7. Amugongo L. M., Mascheroni P., Brooks S., Doering S., Seidel J. Retrieval-augmented generation for large language models in healthcare: a systematic review // Annals of Biomedical Engineering. 2025. DOI: 10.1371/journal.pdig.0000877.