УДК 004.89

АДАПТИВНАЯ RAG-АРХИТЕКТУРА ДЛЯ ЗАДАЧИ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА В КОРПУСЕ ДОКУМЕНТОВ ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ

А. Д. Будревич[1] [0009-0008-1191-148X], **М. М. Абрамский**[2] [0000-0003-3063-8948], **И. А. Валишин**[3] [0009-0006-6891-031X]

[1–3]*Казанский (Приволжский) федеральный университет, г. Казань, Россия*

[1]andbudrevich@kpfu.ru, [2]mabramsk@kpfu.ru, [3]isavalishin@kpfu.ru

Аннотация

Решена задача повышения качества интеллектуального поиска в корпусе документов образовательных учреждений, включающем учебные планы, рабочие программы дисциплин и нормативные акты. Классические архитектуры подхода «генерация, дополненная поиском» (Retrieval-Augmented Generation, RAG), основанные на единственном модуле поиска по обычному тексту, демонстрируют низкую точность работы на документах, включающих таблицы, логические связи между сущностями и строгие формулировки нормативных документов. Предложена адаптивная RAG-архитектура из четырех слоев, каждый из которых учитывает специфику хранения данных в подобных документах. Результаты показали, что учет структуры документов и адаптивная маршрутизация запросов существенно повышают фактическую корректность ответов. Предложенная архитектура может быть использована при проектировании интеллектуальных ассистентов для административных и учебно-методических сервисов высших учебных заведений.

Ключевые слова: *RAG, RAG-архитектура, документы образовательных учреждений, интеллектуальный поиск, искусственный интеллект, обработка документов, семантические модели.*

ВВЕДЕНИЕ

Большие языковые модели (Large Language Models, LLM) стали базовой технологией для широкого круга задач обработки естественного языка – построения диалоговых систем, извлечения информации, автоматической генерации текстов [1]. Информация, на которой обучены подобные модели, взята из открытых источников в конкретный момент времени. Таким образом, при формировании ответа пользователю не учитываются как новые открытые документы, так и материалы, отсутствующие в открытом доступе. В контексте задачи поиска по документам образовательного учреждения этот факт проявляется в том, что открытые LLM не способны надежно отвечать на вопросы по актуальным учебных планам или локальным требованиям к реализации образовательного процесса. Попытки получить ответы от LLM без актуальных данных приводят к появлению так называемых *галлюцинаций* – правдоподобных, но фактически неверных утверждений [2].

Метод генерации с дополненной выборкой (Retrieval-Augmented Generation, RAG), преодолевает это ограничение за счет интеграции генерации ответа с помощью LLM и информационного поиска: модель сначала извлекает релевантные фрагменты из внешней базы знаний, а затем использует их для формирования ответа [3]. До последнего времени было принято выделять три подхода RAG: базовая схема «извлечение – генерация» (*наивный RAG*), подход с дополнительной оптимизацией на этапах до и после поиска (*продвинутый RAG*) и подход, когда отдельные компоненты последовательности обработки реализованы как независимые, динамически подключаемые блоки (*модульный RAG*) [4]. К 2025–2026 гг. разнообразие подходов стало шире: появились графовые архитектуры, агентные фреймворки и системы с долговременной памятью. Каждый из этих подходов решает свой узкоспециализированный набор задач. Формализацию RAG-подхода в виде схемы его работы называют *RAG-архитектурой*.

Современные исследования демонстрируют, что использование RAG повышает точность ответов LLM, особенно в предметных областях, требующих актуальных и регламентированных знаний. Так, например, сравнительная оценка базовых и RAG-расширенных моделей на наборе из 130 вопросов по 13 неврологическим руководствам показала существенно более высокую долю

правильных ответов у документоориентированных вопросов, для ответов на которые использовались модели, дополненные RAG [5]. Система автоматической проверки эссе на основе RAG с поиском, ориентированным на критерии оценки и комментарии преподавателей, продемонстрировала 94–99%-ное согласие с экспертными оценками на выборке из 701 студенческой работы [6]. Оба примера подтверждают применимость RAG к задачам, где модель должна опираться на специфические, формализованные документы организации, а не только на общие знания.

Однако прямой перенос стандартных схем использования RAG на работу с корпусом документов вуза наталкивается на ряд ограничений. Например, многие документы хранят данные в табличном представлении, когда содержание задается не только текстом, но и расположением данных в строках и столбцах. Тогда при стандартной обработке происходит деление документа на одинаковые по размеру текстовые фрагменты, что вызывает потерю сути. Другие данные кодируются по иерархическому принципу (например, направление подготовки «09.04.04» или дисциплина «Б1.В.06»). При обработке эти данные также разделяются на несвязанные фрагменты, что негативно влияет на качество результата. Отметим также, что ответ на вопрос пользователя нередко требует сборки фрагментов информации сразу из нескольких документов.

Работы по гибридным подходам в интеллектуальном поиске показали, что комбинирование лексических методов с подходящими для задачи семантическими представлениями улучшает полноту ответа и точность извлечения информации [7]. Однако проведенные исследования рассматривают преимущественно однородные текстовые корпусы, не учитывая специфику документов с явными табличными и иерархическими структурами.

В настоящей работе описаны разработка и тестирование RAG-архитектуры, ориентированной на корпус университетских документов и решающей указанные проблемы. Эта архитектура включает четыре слоя:

1) структурное профилирование документов, сохраняющее иерархию разделов и тип сегментов при индексировании;

2) компилятор запроса, осуществляющий классификацию запроса и извлечение ключевых сущностей для выбора маршрута обработки;

3) модуль поиска (*ретривер*);

4) агентный слой сортировки запросов по критерию сложности и верификации ответа.

Для оценки вклада каждого слоя архитектуры в формирование ответа введена схема из семи конфигураций, позволяющая поэтапно активировать модули архитектуры и изолированно измерять их влияние на качество ответов. Каждый слой архитектуры раскрыт ниже.

ПОСТАНОВКА ЗАДАЧИ

Пользователями RAG-систем интеллектуального поиска по документам вуза могут выступать студенты и сотрудники университета. В рамках настоящей работы были проанализированы и классифицированы их потенциальные запросы пользователей. Выделено шесть типов запросов:

1) кросс-документные запросы, требующие сопоставления сведений из нескольких документов одновременно;

2) запросы к табличным данным;

3) структурно-иерархические запросы, требующие восстановления связей между сущностями документов;

4) фильтрация и агрегация;

5) многошаговое рассуждение, когда для итогового ответа нужно последовательно извлечь несколько промежуточных фактов;

6) запросы, чувствительные к версионности документов.

Ответ на запрос, генерируемый с помощью RAG-системы, должен удовлетворять двум условиям:

- быть точным относительно актуальной версии документа, содержащего ответ;
- содержать явную ссылку на фрагмент документа (раздел, таблицу, строку), на основании которого был сформирован.

Формально задачу можно описать следующим образом. Пусть корпус документов представлен множеством $D = \{d_i\}$, где каждый документ – это набор структурированных сегментов. На входе система получает

пользовательский запрос q на естественном языке, на выходе должна сформировать ответ a и множество ссылок на сегменты $S \subseteq D$, содержимое которых является достаточным основанием для этого ответа. Качество системы оценивается по тому, насколько ответ фактически верен и насколько полно и точно указанные ссылки соответствуют экспертной разметке.

СОСТАВНЫЕ КОМПОНЕНТЫ АДАПТИВНОЙ АРХИТЕКТУРЫ

Предлагаемая архитектура состоит из четырех слоев (рис. 1), каждый из которых решает отдельный класс проблем, выявленных ранее. На первом этапе документы проходят структурную индексацию с сохранением иерархии разделов и сегментацией (*чанкингом*), учитывающей специфику документов. Далее каждый пользовательский запрос разбирается компилятором запроса, который извлекает ключевые сущности и фильтрует поиск по метаданным документа. На третьем уровне реализован поисковый модуль (*ретривер*) с гибридным поиском, многопроходным добором и переранжированием. Агентный слой адаптирует стратегию обработки запроса к его сложности – от прямого прохода до декомпозиции и корректирующей верификации. Далее описан каждый из перечисленных слоев.

Структурное представление документов. При парсинге исходных файлов форматов PDF и DOCX извлекаются не только текст, но и разметка (заголовки, уровни вложенности, границы таблиц, списки). На основе этой разметки каждый документ преобразуется в дерево сегментов, где каждому узлу сопоставляются тип (текст/таблица/список) и путь в иерархии разделов. Помимо структуры на этапе профилирования извлекаются метаданные документа: факультет, кафедра, образовательная программа, год, семестр, уровень подготовки и др. Метаданные хранятся вместе с каждым сегментом и используются для фильтрации сегментов-кандидатов для формирования ответа.

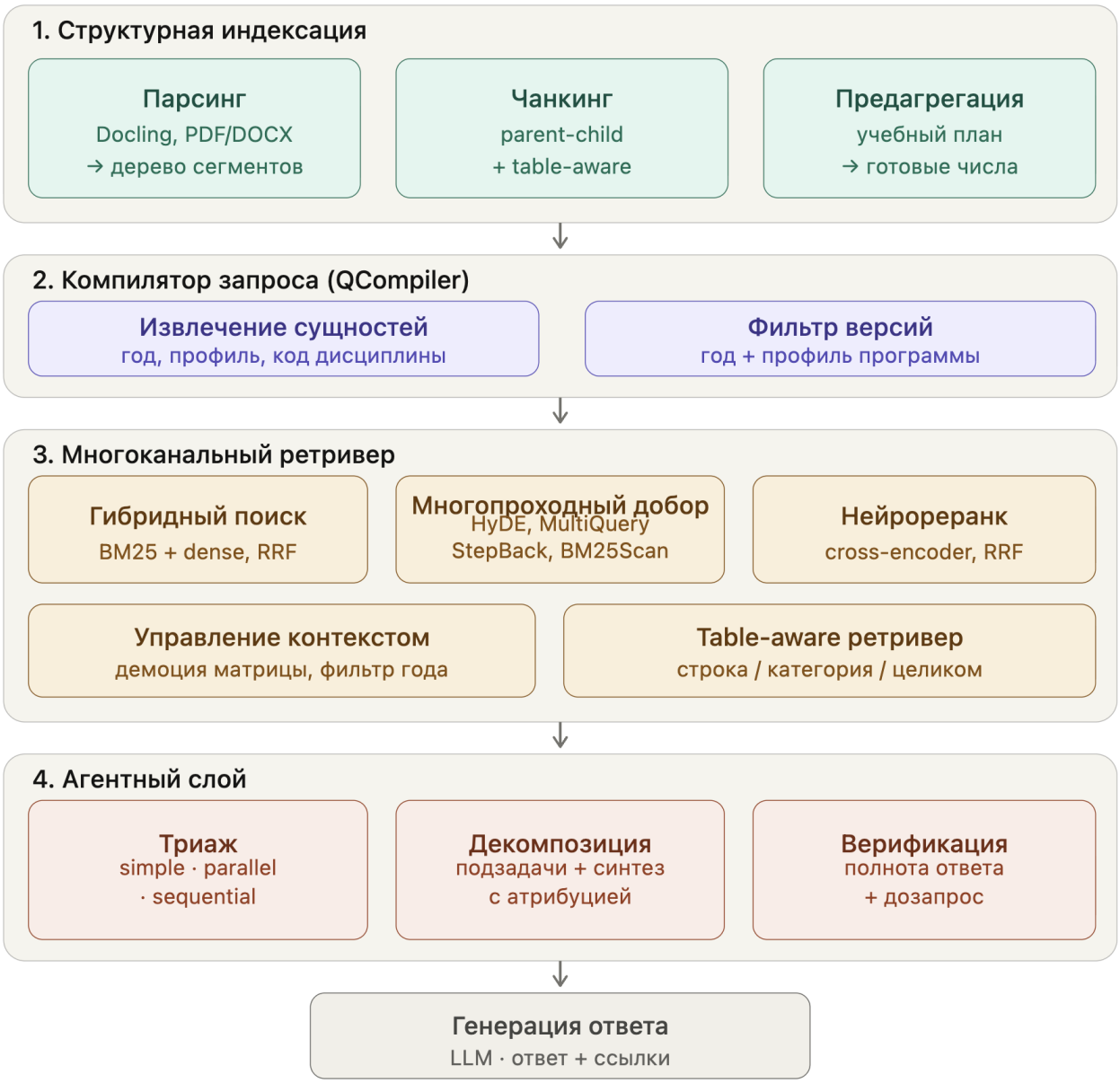

Рис. 1. Структура предлагаемой адаптивной RAG-архитектуры.

Стратегия чанкинга построена следующим образом. Для текстовых разделов применена иерархическая схема: короткий дочерний фрагмент передается в ретривер для поиска, а в контекст подается более широкий родительский, таким образом сохраняются и точность поиска, и полнота контекста. Табличные фрагменты сегментируются тремя способами одновременно: построчно (одна строка – одна запись), по категориям (строки одного типа) и целиком. Такое представление позволяет ретриверу находить таблицу и по отдельному значению ячейки, и по смыслу всей группы.

Компилятор запроса. Перед направлением в ретривер запрос разбирается с помощью нейросимвольного модуля QCompiler [8]. На основе текста запроса модуль извлекает ключевые сущности: год и профиль образовательной программы, код или название дисциплины, тип документа и целевой тип данных (численное значение, список, текстовое описание). Результатом является структурный запрос, управляющий дальнейшей обработкой.

Одной из ключевых функций, которая была добавлена в QCompiler, стала фильтрация по версиям: если из запроса извлечен год и, например, учебный план, поиск ведется только по документам с соответствующими метаданными. На корпусах с несколькими редакциями одного документа этот фильтр принципиален, так как без него система смешивает данные из разных версий, что приводит к неверным ответам.

Гибридный ретривер. Текстовый поиск реализован с помощью комбинации лексического поиска по алгоритму BM25 с плотным семантическим поиском. Их результаты объединяются с помощью алгоритма взаимного рангового слияния (Reciprocal Rank Fusion, RRF). BM25 хорошо справляется с точными кодами («Б1.О.05», «09.04.04»), а семантический поиск – с перефразированными или контекстными формулировками.

Однако было обнаружено, что для сложных запросов, когда релевантный фрагмент не содержит ключевых слов из вопроса напрямую, недостаточно одного прохода. Для таких случаев реализован добор запроса, когда параллельно запускаются несколько стратегий расширения запроса:

- BM25Scan (лексические вариации);
- HyDE (генерация гипотетического ответа и поиск по нему);
- MultiQuery (несколько перефразировок запроса);
- StepBack (абстрагирование до общего понятия),

а для табличных фрагментов:

- TableSiblings (соседние строки той же таблицы);
- NeighborExpansion (смежные сегменты в дереве документа).

Кандидаты из всех проходов объединяются и проходят единый процесс переранжирования.

Агентный слой. Было выявлено, что запросы могут требовать разных стратегий обработки. Например, фактический вопрос «в каком семестре читается дисциплина X?» решается прямым обращением к ретриверу и генерации, а запрос, требующий сведéния данных из нескольких документов или последовательного извлечения промежуточных фактов, при таком же подходе дает неполный или ошибочный ответ.

Для решения этой проблемы в проектируемый RAG-метод был введен агентный слой с классификацией запросов по стратегии обработки:

- простые запросы исполняются непосредственно обращением к ретриверу;
- параллельные запросы могут быть разложены на независимые подзадачи, каждая подзадача решается отдельным вызовом ретривера, а результат формируется с явным указанием источника для каждой части ответа;
- последовательные запросы разлагаются на цепочку последовательных подзапросов, где ответ на предыдущий шаг становится входом для следующего.

После генерации ответа проводится его верификация: система проверяет, все ли ключевые сущности из запроса нашли отражение в ответе. При необходимости формируется дополнительный запрос к ретриверу для добора недостающих фактов.

Кроме того, происходит детерминированное извлечение структурных данных. Структурированные данные университетских документов не извлекаются только лишь семантическим поиском. Поэтому для них применяются специальные механизмы. Например, матрица компетенций представляется двунаправленно (дисциплина ↔ компетенция), а агрегаты учебного плана – число экзаменов, дисциплин и блоков по семестрам – вычисляются на этапе индексации. Это устраняет целый класс ошибок арифметики у языковой модели и делает ответы на запросы с подсчетом верифицируемыми: значение цитируется из заранее вычисленного факта, а не выводится моделью в процессе генерации.

Дополнительно были введены механизмы, повышающие точность подаваемого в модель контекста: понижение приоритета чанков документов на запросах, не связанных с ними, строгая фильтрация по году для запросов,

чувствительных к версии документа, а также распределение ресурсов контекста между подзапросами.

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА СПРОЕКТИРОВАННОЙ АДАПТИВНОЙ RAG-АРХИТЕКТУРЫ

Для оценки предложенной архитектуры был сформирован корпус документов Института информационных технологий и интеллектуальных систем (ИТИС) Казанского (Приволжского) федерального университета (КФУ). В него вошли учебные планы программ бакалавриата, рабочие программы дисциплин, матрицы компетенций, а также локальные нормативные акты, регламентирующие стипендиальное обеспечение и прохождение практики. Документы представлены файлами в форматах XLSX, DOCX и PDF.

На основе анализа реальных обращений студентов и сотрудников была составлена тестовая выборка из 60 вопросов на русском языке, которые были классифицированы по типам. Для каждого вопроса были подготовлены эталонный ответ в свободной форме, перечень ключевых фактов, которые должны присутствовать в ответе, а также список конкретных документов и их разделов, таблиц и строк, из которых должен быть извлечен ответ. Эталонные ответы были верифицированы вручную, непосредственно по исходным документам.

Чтобы независимо оценить вклад каждого компонента предлагаемой архитектуры, были спроектированы семь версий конфигураций архитектуры, обозначенных от V1 до V7. Каждая следующая версия расширяет предыдущую одним архитектурным компонентам:

- V1: канонический RAG с фиксированной сегментацией документов и единственным лексическим ретривером без учета их структуры;
- V2: V1 + индекс по документам с метаданными;
- V3: V2 + поиск с помощью хэш-эмбеддингов;
- V4: V3 + работа с таблицами и иерархическая сегментация;
- V5: V4 + нейросимвольный компилятор и семантические эмбеддинги;
- V6: V5 + многопроходный добор и переранжирование;

- V7: V6 + агентный слой и детерминированное извлечение структурных данных.

Для оценки были выбраны пять метрик, покрывающих качество извлечения контекста и качество ответа [9].

Полнота ответа (Context Recall, CR) – оценивает, какую долю информации, необходимой для полного ответа, содержат извлеченные фрагменты. Метрика принимает значения из множества $\{0;\ 0.25;\ 0.5;\ 0.75;\ 1.0\}$.

Фактическая корректность (Factual Correctness, FC) – измеряет, какая доля эталонных фактов присутствует в ответе системы явно или по смыслу, возможно, с частичным зачетом:

$$FC\ =\ \frac{\sum_{f\ \in\ GF} p_f}{|GF|},\ \text{где}\ p_f\ \in \{0;\ 0.5;\ 1\},$$

GF – множество эталонных фактов, p_f – степень присутствия факта в ответе.

Верность (Faithfulness, F) – измеряет долю утверждений в ответе, подтвержденных извлеченным контекстом, также с частичным зачетом:

$$F\ =\ \frac{\sum_{i\ =\ 1}^{N} s_i}{N},\ \text{где}\ s_i\ \in \{0;\ 0.5;\ 1\},$$

где N – число фактических утверждений в ответе. Высокое значение свидетельствует о том, что система опирается на документ, а не на знания самой модели.

Релевантность ответа (Answer Relevance, AR) – оценивает, насколько ответ адресован именно заданному вопросу, независимо от его фактической корректности (FC). Метрика принимает значения из множества $\{0;\ 0.25;\ 0.5;\ 0.75;\ 1.0\}$.

Доля нахождения релевантной информации (Hit Rate, HR) – детерминированная метрика, которая проверяет, встречаются ли числовые значения из эталонных фактов в извлеченном контексте рядом со словами того же факта:

$$HR\ =\ \frac{|\{n \in N_{\text{gold}} : n\ \text{встречается в контексте рядом с ключевым словом}\}|}{|N_{\text{gold}}|},$$

где N_{gold} – множество числовых значений из эталонных фактов.

Все метрики, требующие семантического суждения (CR, CF, F, AR), вычисляются с применением подхода «LLM как судья» (LLM-as-a-Judge). При таком подходе отдельная языковая модель выступает в роли эксперта-оценщика (судьи), получая на вход вопрос, эталонный ответ, набор эталонных фактов и ответ системы. В работе [10] показано, что модель-судья достигает около 80% согласия с оценками людей-экспертов на диалоговых задачах. Контекст, подаваемый модели-судье, соответствует тому, что реально было доступно генератору ответа, что принципиально для корректного измерения метрики F [9]. Метрика HR служит верифицируемым якорем, подтверждающим оценки CR без привлечения модели-судьи.

Итоговые значения метрик усреднялись по всем 60 вопросам выборки, а также вычислялись отдельно для каждой из шести информационных потребностей, выделенных ранее. Это позволило оценить не только общее качество, но и то, на каких типах запросов архитектурные улучшения дают наибольший эффект.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

В табл. 1 приведены результаты тестирования версий V1–V7. Рассмотрим детально ряд вопросов тестирования конфигураций.

Для конфигурации V1 метрики имеют низкие значения. Полнота 0.008 означает, что извлеченный контекст содержит информацию, необходимую для полного ответа, менее чем в 1% случаев. Показатель FC, равный 0.024 означает, что фактически система не справляется с задачей. Значение F, равное 0.704, создает ошибочное впечатление качества, когда на самом деле модель почти ничего не находит и не утверждает, а значит, нет достаточного объема информации для появления ошибок. Показатель HR, равный 0.2, подтверждает, что числовые значения из эталонных фактов присутствуют в контексте лишь в каждом пятом случае. В табл. 2 представлены оценки метрик версии V1 по типам запросов.

Табл. 1. Результаты тестирования построенной архитектуры для версий V1–V7.

Версия	Описание	CR	FC	F	AR	HR
V1	BM25-RAG, фиксированные сегменты	0.008	0.024	0.704	0.221	0.2
V2	V1 + структурный индекс и метаданные	0.004	0.012	0.751	0.154	0.218
V3	V2 + гибридный поиск	0.108	0.123	0.665	0.615	0.423
V4	V3 + работа с таблицами и иерархическая сегментация	0.167	0.085	0.605	0.433	0.609
V5	V4 + нейросимвольный компилятор и эмбеддинги	0.364	0.226	0.918	0.492	0.812
V6	V5 + многопроходный добор и переранжирование	0.568	0.348	0.944	0.693	0.882
V7	V6 + агентный слой и детерминированное извлечение структурных данных	**0.713**	**0.511**	**0.961**	**0.861**	**0.955**

Табл. 2. Оценка метрик версии V1 по типам запросов.

Тип запроса	CR	FC	F	AR
Тип 1. Запросы к нескольким документам	0.050	0.024	0.267	0.183
Тип 2. Запросы к табличным данным	0.000	0.024	0.325	0.260
Тип 3. Структурно-иерархические запросы	0.000	0.024	0.441	0.217
Тип 4. Запросы с фильтрацией и агрегацией	0.000	0.024	0.425	0.315
Тип 5. Многошаговое рассуждение	0.000	0.024	0.333	0.292
Тип 6. Запросы, чувствительные к версионности	0.000	0.024	0.183	0.367

Разбивка результатов по типам запросов выявляет закономерность. Наихудший показатель полноты у типов 2, 3, 5 и 6 объясняется разрывом между формулировкой вопроса и текстом документа. Запросы о структурных связях, версионности и многошаговых рассуждениях используют обобщенные формулировки, тогда как документы содержат специфическую терминологию и коды. Для запросов с версионностью метрика F имеет низкое значение, поэтому модель вынуждена опираться на собственные знания, значит, рискует воспроизвести устаревшие данные.

Конфигурация V2 сохраняет BM25-ретривер, но изменяет способ построения индекса: вместо фрагментов фиксированного размера документы проходят через модуль структурного профилирования с сохранением иерархии разделов, типов сегментов и метаданных.

Переход от V1 к V2 снижает CR (0.008 → 0.004) и FC (0.024 → 0.012). Структурная сегментация разбивает документы на короткие однородные сегменты, каждый из которых содержит меньше ключевых слов. При лексическом BM25-поиске шансы этих коротких сегментов совпасть с формулировкой запроса ниже, чем у длинных фрагментов V1. Это классическая проблема для небольших документов, и на этапе V2 она не решается. Конфигурация выступает здесь контрольным условием: она показывает, что более аккуратная нарезка сама по себе ничего не дает без семантического компонента поиска. Небольшой рост F (0.704 → 0.751) закономерен: структурно однородные сегменты реже содержат разнородную информацию и потому реже провоцируют противоречивые утверждения в ответе.

Конфигурация V3 использует тот же структурный индекс, что и V2, но заменяет лексический поиск гибридным ретривером: поиск BM25 комбинируется с плотным поиском на хэш-эмбеддингах (256-мерный хэш без обученной семантики), а результаты объединяются через алгоритм RRF. Хэш-эмбеддинги выбраны намеренно для возможности оценить вклад гибридного поиска, не смешивая его с качеством эмбеддинговой модели.

С переходом к гибридному поиску CR вырастает от 0.004 до 0.108, AR – от 0.154 до 0.615. Ключевой момент заключается в том, что 256-мерный хэш кодирует только фактическое присутствие токенов. Тем не менее RRF-

объединение двух разнородных ответов (BM25 и хэш-dense) дает заметный прирост из-за того, что два поисковых механизма ошибаются в разных местах и их объединение перекрывает часть промахов.

Снижение F (0.751 → 0.665) является ожидаемым эффектом расширения выдачи, поскольку чем больше фрагментов попадает в контекст, тем выше доля таких фрагментов, которые слабо связаны с вопросом, но модель все равно включает их в ответ. Конфигурация V3 устанавливает верхнюю границу качества для конфигураций без обученных эмбеддингов и показывает, что главная проблема, которую необходимо преодолеть, заключается не в архитектуре поиска, а в качестве эмбеддинговой модели.

Конфигурация V4 надстраивает над V3 специализированную обработку таблиц. Строки извлекаются как отдельные записи с заголовками столбцов, добавляется иерархия сегментов.

Метрика HR для V4 выросла с 0.423 до 0.609, поскольку табличные строки начали попадать в контекст. При этом FC снижается с 0.123 до 0.085 по тем же причинам, что и в версии V2. Новый механизм работы с таблицами расширяет число кандидатов (три представления каждой таблицы – построчно, по категориям и целиком), а хэш-эмбеддинги не могут отсортировать их по релевантности. В результате контекстное окно заполняется строками из нерелевантных таблиц, вытесняя нужные текстовые фрагменты.

Эта проблема решается в конфигурации V5, которая вводит два ключевых изменения: хэш-эмбеддинги заменяются на обученную семантическую модель, а пайплайн дополняется нейросимвольным модулем QCompiler, который извлекает из запроса год, профиль, код дисциплины и тип документа, направляя поиск по правильным маршруту и версиям документов.

Переход к V5 дает наиболее значимый скачок метрик. Значение F вырастает от 0.605 до 0.918 – контекст стал пригодным, чтобы модель могла опираться на него при формировании ответа. Метрика FC удваивается (0.167 → 0.364), M2 вырастает почти втрое (0.085 → 0.226), что показывает существенный вклад эмбеддинговой модели. Одновременное введение модуля QCompiler усиливает эффект на запросах с версионностью, поскольку строгий фильтр по году отсекает кандидатов из несовпадающих редакций документов еще до ранжирования.

Конфигурация V6 надстраивает над V5 слой расширения запроса и переранжирования: параллельно запускаются описанные ранее механизмы BM25Scan, HyDE, MultiQuery, StepBack, TableSiblings и NeighborExpansion, кандидаты из всех проходов объединяются и проходят единое переранжирование.

V6 дает сильнейший прирост CR за все версии: с 0.364 до 0.568. Основная причина роста заключается в покрытии разрыва между формулировкой вопроса и текстом документа. Разные стратегии добора информации перекрывают разные классы этого разрыва:

- HyDE генерирует гипотетический ответ и ищет ему подобные фрагменты, что эффективно для случаев, когда пользователь формулирует запрос иначе, чем написан документ;
- MultiQuery перефразирует вопрос несколькими способами, повышая шансы на терминологическое совпадение;
- StepBack абстрагирует запрос до корневого понятия, что помогает, когда нужный факт находится под общим заголовком раздела;
- TableSiblings добирает соседние строки той же таблицы – критично для запросов о нагрузке по семестрам, где нужная строка часто окружена контекстными заголовками.

Прирост релевантности ответа (0.492 → 0.693) говорит о том, что ответы стали полнее охватывать смысл вопроса, а не просто воспроизводить ближайший найденный фрагмент.

Конфигурация V7 объединяет агентный слой с группой уточняющих механизмов, которые оцениваются совместно: каждый из них достраивает логику уже существующих компонентов, не меняя архитектуру целиком.

Версия V7 дает прирост по всем пяти метрикам. Рост M2 (0.348 → 0.511) складывается из двух независимых компонентов. Вместо единственного прохода по всему вопросу система последовательно решает подзадачи, каждая из которых проще и лучше реализуется одним поисковым запросом. По запросам к нескольким документам FC достигает 0.39, что, с одной стороны, показывает прирост относительно версии V1, а с другой – остается наименьшим значением этой метрики среди всех типов запросов. Это указывает на то, что

соединение документов остается основным узким местом и требует дальнейшего исследования.

Детерминированное извлечение структурных данных дает прирост для табличных запросов и запросов с фильтрацией. Метрика F достигает значения 0.961, что свидетельствует о следующем: явное присутствие источников на этом уровне снижает долю утверждений без документального основания.

В табл. 3 представлены значения метрик, вычисляемых моделью-судье, сгруппированные по типам запросов.

Табл. 3. Метрики конфигурации V7 в разрезе типов запросов.

Тип запроса	CR	FC	F	AR
Тип 1. Запросы к нескольким документам	0.62	0.39	0.97	0.84
Тип 2. Запросы к табличным данным	0.70	0.53	0.98	0.90
Тип 3. Структурно-иерархические запросы	0.72	0.50	0.93	0.80
Тип 4. Запросы с фильтрацией и агрегацией	0.75	0.46	0.96	0.82
Тип 5. Многошаговое рассуждение	0.70	0.48	0.98	0.85
Тип 6. Запросы, чувствительные к версионности	0.78	0.71	0.95	0.95

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

На рис. 2 показана динамика роста всех исследованных метрик по всем семи конфигурациям V1-V7. Разбиение метрик версии V7 по типам запросов показало, что наилучший результат метрики FC показывают версионные запросы, поскольку строгая фильтрация по году позволяет системе уверенно разграничивать редакции документов. Табличные и структурные запросы обрабатываются за счет улучшения извлечения фрагментов документов. Многошаговые и фильтрующие запросы опираются на агентную декомпозицию, однако накопление ошибки по цепочке подзапросов ограничивает их точность.

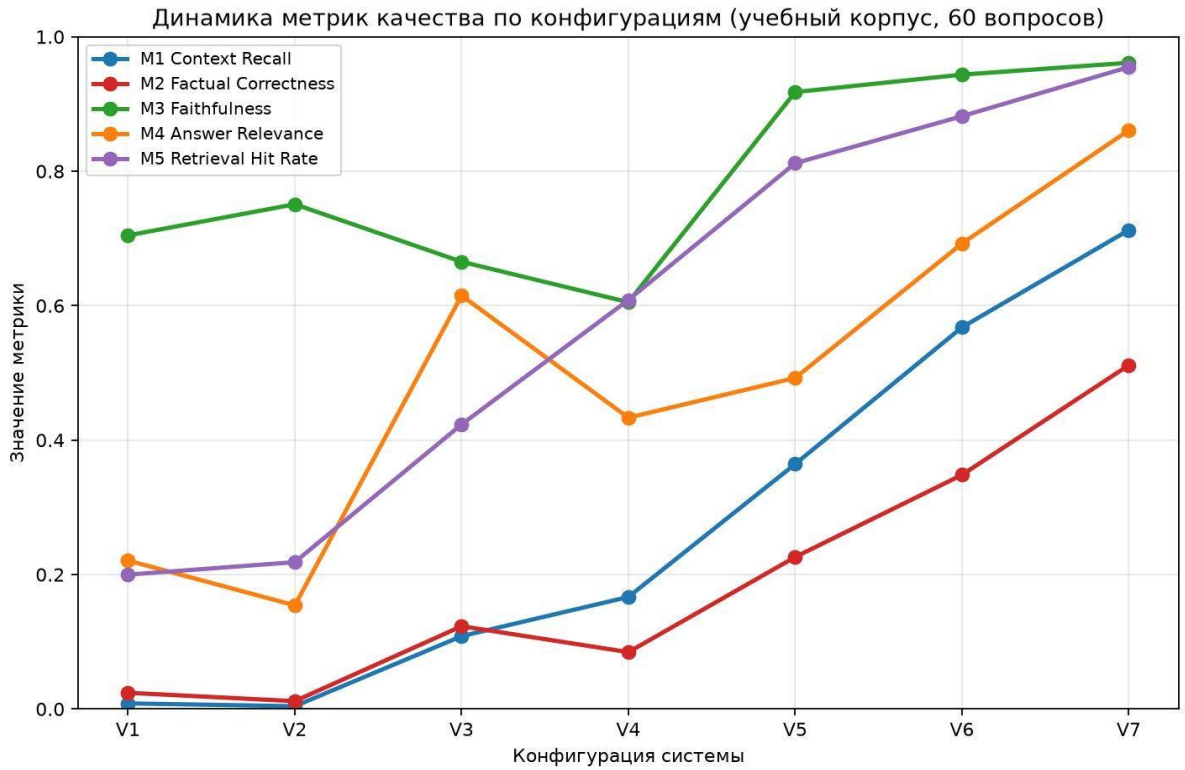

Рис. 2. Динамика метрик качества по конфигурациям V1–V7 (учебный корпус, 60 вопросов).

Как было сказано ранее, самым сложным типом остаются запросы к нескольким документам. Они требуют одновременного извлечения и сопоставления фактов из нескольких документов, и даже многопроходный добор не покрывает весь необходимый контекст. При этом метрика F держится на уровне 0.93 – 0.98 по всем типам запросов, что означает, что система не выдает галлюцинации даже там, где она имеет неполную информацию.

Различие между значениями F = 0.961 и FC = 0.511 в итоговой конфигурации указывает на природу оставшихся ошибок. При высоком значении F ответ формируется исключительно из того, что нашел ретривер, это означает, что причина неполных ответов заключается в неполном контексте, а не в поведении языковой модели. В рамках одного прохода ретривер ограничен ресурсной емкостью токенов – при запросах в несколько проходов ему нужно одновременно покрыть факты из нескольких документов, и на практике часть из них может быть вытеснена.

Для административного применения важна такая характеристика системы: при недостаточном контексте она возвращает неполный, но фактически верный ответ, основанный на найденных фрагментах. Ошибки в информации о формальных требованиях могут иметь прямые последствия, поэтому предсказуемость поведения в данном случае важнее максимального охвата.

Анализ ошибочных ответов для запросов к нескольким документам позволил выделить два типичных сценария отказа. Первый – последовательная зависимость подзапросов. Например, запрос «какие дисциплины учебного плана профиля X требуют пререквизита Y по данным РПД?» предполагает сначала идентификацию релевантных дисциплин из одного документа, а затем верификацию пререквизитов по другому источнику. Агентная декомпозиция частично решает эту задачу, но каждый подзапрос конкурирует за ограниченное контекстное окно. Второй сценарий – терминологическая гетерогенность корпуса: одно и то же понятие может в двух документах обозначаться разными терминами или обозначениями. Семантический поиск сокращает этот разрыв, но не устраняет его полностью. По своей природе запросы этого типа ближе к задаче структурированного поиска по связанным таблицам, чем к извлечению данных из неструктурированного текста, и именно этот факт определяет потолок возможностей применения метода в текущей реализации.

Версионные и табличные запросы обрабатываются заметно лучше среднего именно там, где были введены специализированные детерминированные механизмы: строгий фильтр года и структурированное извлечение табличных данных. Это подтверждает исходную гипотезу работы: специфика образовательных документов требует специализированных слоев и улучшение семантического поиска в одиночку не устраняет структурные проблемы корпуса.

ЗАКЛЮЧЕНИЕ

Показана возможность улучшения решения задачи вопросно-ответного взаимодействия с корпусом документов вуза с помощью адаптивного RAG-подхода. Установлено, что классический RAG непригоден для этой предметной области.

Предложенная адаптивная архитектура из четырех слоев – структурная индексация, компилятор запроса, многоканальный ретривер и агентный слой – обеспечила рост метрики качества запросов. Поэтапное введение конфигурации версий позволило изолированно оценить вклад каждого компонента.

Дальнейшее развитие работы связано с улучшением качества поиска, требующего анализа нескольких документов одновременно, а также с интеграцией в разработанную архитектуру подхода долговременной памяти диалога и ее адаптацией к мультимодальным документам.

СПИСОК ЛИТЕРАТУРЫ

1. OpenAI R. Gpt-4 technical report. arXiv 2303.08774. 2023. Vol. 2, No. 1. P. 1. https://doi.org/10.48550/arXiv.2303.08774

2. *Ji Z., Lee N., Frieske R., Yu T., Su D., Xu Y., Ishii E., Bang Y.J., Madotto A., Fung P.* Survey of hallucination in natural language generation // ACM computing surveys. 2023. Vol. 55, No. 12. P. 1–38. https://doi.org/10.1145/3571730

3. *Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., Riedel S., Kiela D.* Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in neural information processing systems. 2020. Vol. 33. P. 9459–9474. https://doi.org/10.48550/arXiv.2005.11401

4. *Gao Y., Xiong Y., Gao X., Jia K., Pan J., Bi Y., Dai Y., Sun J., Wang M., Wang H*. Retrieval-augmented generation for large language models: A survey // arXiv preprint arXiv:2312.10997. 2023. Vol. 2, No. 1. P. 32. https://doi.org/10.1371/journal.pdig.0000877

5. *Masanneck L., Meuth S.G., Pawlitzki M.* Evaluating base and retrieval augmented LLMs with document or online support for evidence-based neurology // npj Digital Medicine. 2025. Vol. 8, No. 1. P. 137. https://doi.org/10.1038/s41746-025-01536-y

6. *Barenji R.V., Salimi N., Khoshgoftar S.* An LLM-Powered Assessment Retrieval-Augmented Generation (RAG) For Higher Education // arXiv preprint arXiv:2601.06141. 2026. https://doi.org/10.48550/arXiv.2601.06141

7. *Karpukhin V., Oguz B., Min S., Lewis P., Wu L., Edunov S., Chen D., Yih W.T*. Dense passage retrieval for open-domain question answering // Proceedings of the

2020 conference on empirical methods in natural language processing (EMNLP). 2020. P. 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

8. *Zhang Y., Dou Z., Li X., Jin J., Wu Y., Li Z., Wen J. R.* Neuro-symbolic query compiler // Findings of the Association for Computational Linguistics: ACL 2025. Vol. 2025. P. 12138–12155. https://doi.org/10.18653/v1/2025.findings-acl.628

9. *Es S., James J., Anke L.E., Schockaert S*. Ragas: Automated evaluation of retrieval augmented generation // Proceedings of the 18th conference of the European chapter of the association for computational linguistics: system demonstrations. 2024. P. 150–158. https://doi.org/10.18653/v1/2024.eacl-demo.16

10. *Zheng L., Chiang W., Sheng Y., Zhuang S., Wu Z., Zhuang Y., Lin Z., Li Z., Li D., Xing E., Zhang H., Gonzalez J., Stoica I.* Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena // Advances in neural information processing systems. 2023. Vol. 36. P. 46595–46623. https://doi.org/10.48550/arXiv.2306.05685

ADAPTIVE RAG-ARCHITECTURE FOR INTELLIGENT SEARCH IN THE CORPUS OF EDUCATIONAL INSTITUTIONS DOCUMENTS

A. D. Budrevich[1] [0009-0008-1191-148X], M. M. Abramskiy[2] [0000-0003-3063-8948], I. A. Valishin[3] [0009-0006-6891-031X]

[1–3]*Kazan (Volga Region) Federal University Kazan, Russia*

[1]andbudrevich@kpfu.ru, [2]mabramsk@kpfu.ru, [3] isavalishin@kpfu.ru

Abstract

The problem of improving the quality of intelligent search in a corpus of educational institution documents, including curricula, course syllabus, and regulatory documents, was solved. Classic architectures of the Retrieval-Augmented Generation (RAG) approach, based on a single plaintext search module, demonstrate low accuracy when used with documents containing tables, logical relationships between entities, and strict regulatory document wording. An adaptive RAG architecture consisting of four layers is proposed, each taking into account the specifics of data storage in such documents. The results show that considering document structure and adaptive query routing significantly improve the actual accuracy of responses. The proposed architecture can be used in the design of intelligent assistants for administrative and educational services at higher education institutions.

Keywords: *RAG, RAG architecture, educational institution documents, intelligent search, artificial intelligence, document processing, semantic models.*

REFERENCES

1. OpenAI R. Gpt-4 technical report. arXiv 2303.08774. 2023. Vol. 2, No. 1. P. 1.https://doi.org/10.48550/arXiv.2303.08774

2. *Ji Z., Lee N., Frieske R., Yu T., Su D., Xu Y., Ishii E., Bang Y.J., Madotto A., Fung P.* Survey of hallucination in natural language generation // ACM computing surveys. 2023. Vol. 55, No. 12. P. 1–38. https://doi.org/10.1145/3571730

3. *Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., Riedel S., Kiela D.* Retrieval-Augmented Generation

for Knowledge-Intensive NLP Tasks // Advances in neural information processing systems. 2020. Vol. 33. P. 9459–9474. https://doi.org/10.48550/arXiv.2005.11401

4. *Gao Y., Xiong Y., Gao X., Jia K., Pan J., Bi Y., Dai Y., Sun J., Wang M., Wang H*. Retrieval-augmented generation for large language models: A survey // arXiv preprint arXiv:2312.10997. 2023. Vol. 2, No. 1. P. 32.
https://doi.org/10.1371/journal.pdig.0000877

5. *Masanneck L., Meuth S.G., Pawlitzki M.* Evaluating base and retrieval augmented LLMs with document or online support for evidence-based neurology // npj Digital Medicine. 2025. Vol. 8, No. 1. P. 137.
https://doi.org/10.1038/s41746-025-01536-y

6. *Barenji R.V., Salimi N., Khoshgoftar S.* An LLM-Powered Assessment Retrieval-Augmented Generation (RAG) For Higher Education // arXiv preprint arXiv:2601.06141. 2026. https://doi.org/10.48550/arXiv.2601.06141

7. *Karpukhin V., Oguz B., Min S., Lewis P., Wu L., Edunov S., Chen D., Yih W.T*. Dense passage retrieval for open-domain question answering // Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). 2020. P. 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

8. *Zhang Y., Dou Z., Li X., Jin J., Wu Y., Li Z., Wen J. R.* Neuro-symbolic query compiler // Findings of the Association for Computational Linguistics: ACL 2025. Vol. 2025. P. 12138–12155. https://doi.org/10.18653/v1/2025.findings-acl.628

9. *Es S., James J., Anke L.E., Schockaert S*. Ragas: Automated evaluation of retrieval augmented generation // Proceedings of the 18th conference of the European chapter of the association for computational linguistics: system demonstrations. 2024. P. 150–158. https://doi.org/10.18653/v1/2024.eacl-demo.16

10. *Zheng L., Chiang W., Sheng Y., Zhuang S., Wu Z., Zhuang Y., Lin Z., Li Z., Li D., Xing E., Zhang H., Gonzalez J., Stoica I.* Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena // Advances in neural information processing systems. 2023. Vol. 36. P. 46595–46623. https://doi.org/10.48550/arXiv.2306.05685

СВЕДЕНИЯ ОБ АВТОРАХ

БУДРЕВИЧ Анна Дмитриевна – студентка магистратуры Института информационных технологий и интеллектуальных систем (ИТИС) Казанского (Приволжского) федерального университета (КФУ).

Anna Dmitrievna BUDREVICH – master's student at the Institute of Information Technology and Intelligent Systems (ITIS), Kazan (Volga Federal) Federal University (KFU).

email: andbudrevich@kpfu.ru

ORCID: 0009-0008-1191-148X

АБРАМСКИЙ Михаил Михайлович – директор ИТИС КФУ, кандидат технических наук.

Mikhail Mikhailovich ABRAMSKIY – director at the Institute of Information Technology and Intelligent Systems, Kazan Federal University, PhD (Cand Sci. – Tech.)

email: mabramsk@kpfu.ru

ORCID: 0000-0003-3063-8948

ВАЛИШИН Искандер Айратович – ассистент кафедры цифровой аналитики и технологий искусственного интеллекта Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета.

Iskander Airatovich VALISHIN – assistant at the department of Digital Analytics and Artificial Intelligence Technologies, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: isavalishin@kpfu.ru

ORCID: 0009-0006-6891-031X

Материал поступил в редакцию 25 июня 2026 года