Кулюкина Ульяна Андреевна

студент Московского государственного технического университета имени Н.Э. Баумана,
РФ, г. Москва

Аксиненко Георгий Александрович

студент Московского государственного технического университета имени Н.Э. Баумана,
РФ, г. Москва

Мещихин Илья Александрович

научный руководитель, старший преподаватель Московского государственного технического университета имени Н.Э. Баумана,
РФ, г. Москва

РАЗРАБОТКА СИСТЕМЫ ЦИФРОВОГО АССИСТИРОВАНИЯ ДЛЯ НАСТРОЙКИ БИОНИЧЕСКОГО КОЛЕННОГО МОДУЛЯ НА ОСНОВЕ RETRIEVAL – AUGMENTED GENERATION (RAG)

Аннотация. В настоящей статье рассматривается проблема ограниченной доступности квалифицированных специалистов для настройки бионических коленных модулей, особенно актуальная в удалённых географических регионах. В качестве решения предложена автоматизация процесса настройки протезов на основе технологии Retrieval-Augmented Generation (RAG) с использованием локального сервера генеративных языковых моделей. Проведено исследование эффективности применения RAG для повышения точности ответов языковых моделей, разработана специализированная база знаний и выполнено сравнительное тестирование двух версий инструкций. Результаты экспериментальной оценки демонстрируют, что структурированная и лаконичная инструкция обеспечивает статистически значимое улучшение качества ответов модели. Разработанная система обладает существенным

потенциалом для внедрения в сфере автоматизации производства и технической поддержки медицинского оборудования.

Введение

Современные технологии, включая искусственный интеллект и большие языковые модели (LLM), активно внедряются в медицину и реабилитацию [1, с.2]. Одной из ключевых задач является обеспечение мобильности для людей с ограниченными возможностями, где коленные модули играют важную роль. Однако процесс их настройки требует участия высококвалифицированных специалистов, что ограничивает доступность качественного протезирования, особенно в удалённых регионах.

Целью данной работы является разработка системы цифрового ассистирования для настройки бионического коленного модуля с применением технологии RAG и локального сервера языковой модели. Основные задачи включают исследование технологий LLM и RAG, разработку базы знаний, создание клиент-серверного приложения и анализ возможностей внедрения системы. Теоретическая значимость работы заключается в изучении методов интеграции внешних баз знаний с языковыми моделями, а практическая — в создании инструмента, способного снизить зависимость от узкоспециализированных экспертов.

1. Технологии LLM и RAG

Современные LLM базируются на архитектуре Transformer [2, с.3], принципиальная схема которой представлена на Рис. 1.

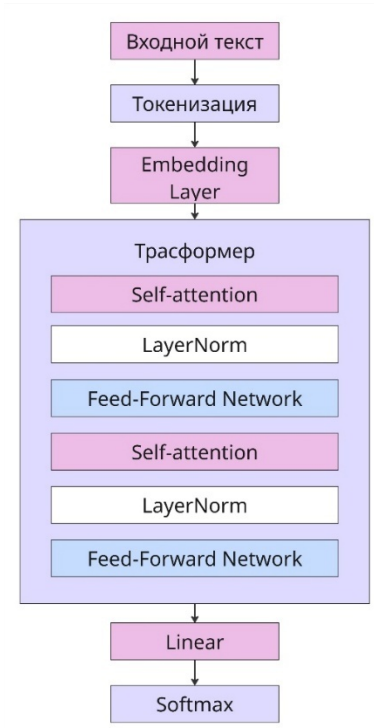

Рисунок 1 - Архитектура LLM

А) Процедуры токенизации и векторного представления

- Входной текстовый поток сегментируется на минимальные лингвистические единицы (токены), которыми могут являться слова, морфемы или отдельные символы.
- Каждый токен трансформируется в векторное представление высокой размерности (порядка нескольких сотен компонент), где каждая компонента кодирует определённый семантический аспект лексемы. На Рисунке 2 визуализировано расположение подобных векторов в трёхмерном пространстве.

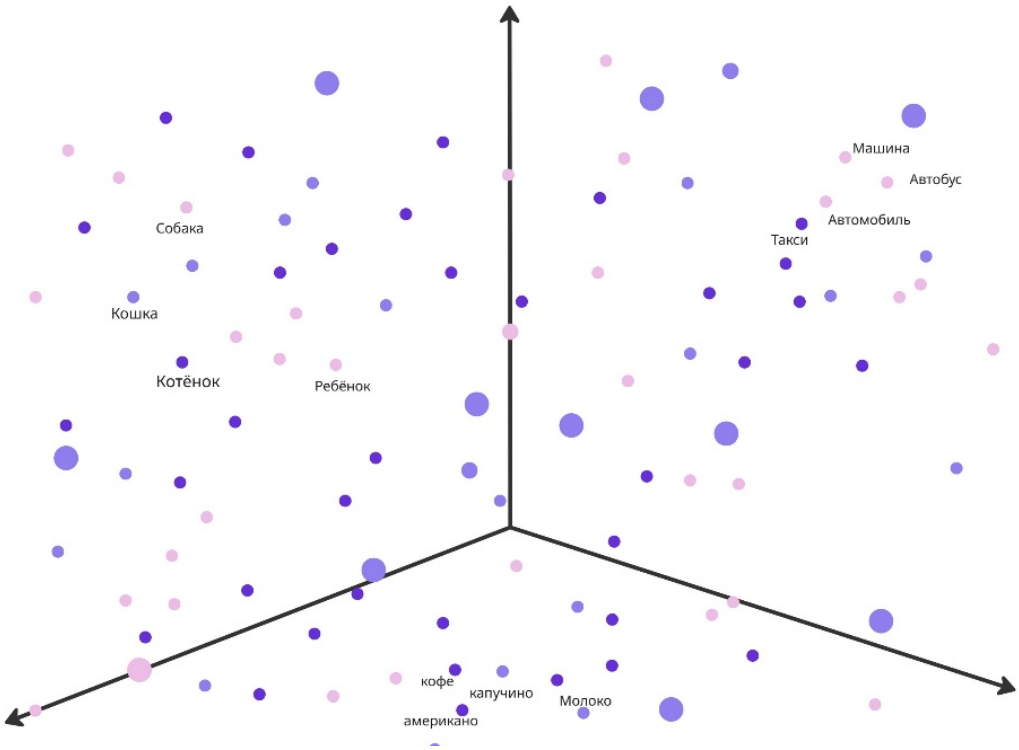

Рисунок 2 – Трёхмерное представление эмбеддингов (в LLM обычно используют 512-768 мерное)

- Лексические единицы с близкой семантикой (например, "протез" и "ортопедическое устройство") занимают смежные области в векторном пространстве.

Б) Механизм самовнимания (Self-Attention)

Механизм самовнимания позволяет элементам входной последовательности устанавливать контекстно-зависимые связи и определять значимость отдельных компонент [2, с.4]. Т.е. для инструкции "Установите угол сгибания 15°" механизм присваивает максимальные весовые коэффициенты токенам "угол" и "15°", игнорируя семантически менее значимые элементы.

B) Прямая полносвязная сеть (Feed-Forward Network)

Feed-Forward Network осуществляет нелинейное преобразование векторных представлений. Каждый вектор, полученный на этапе самовнимания, подвергается линейному преобразованию с увеличением размерности в четыре раза, после чего применяется функция активации ReLU (или GeLU в современных реализациях), и выполняется обратное линейное преобразование к исходной размерности. Данная операция формализуется следующим образом:

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2$$

Где x — входной вектор (после Self-Attention); W1,W2 — матрицы весов; b1,b2 — смещения (biases); ReLU — функция активации (может заменяться на GeLU в современных моделях).

Нормализация на уровне слоя (LayerNorm) обеспечивает стабильность вычислительного процесса. Как показано на Рисунке 3, указанные этапы циклически повторяются в рамках архитектуры.

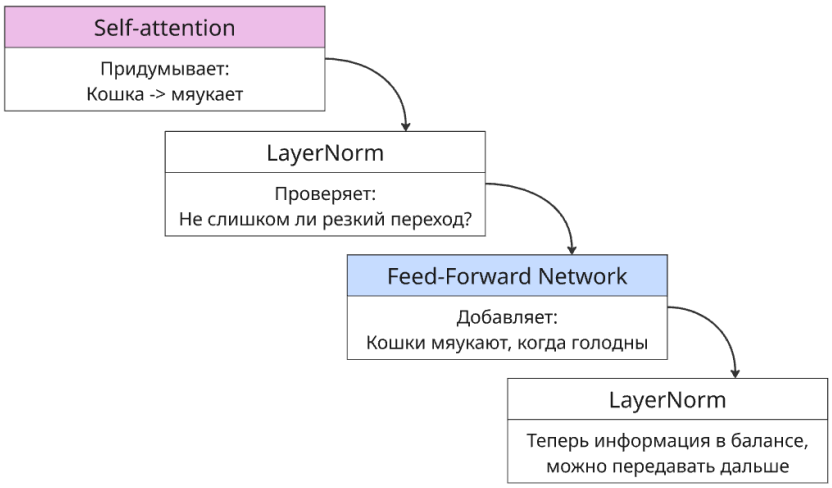

Рисунок 3 – Механизм трансформера

Заключительный этап (линейный слой) генерирует распределение вероятностей над словарным пространством. Например, для контекста "Кошка любит..." линейный слой вычисляет вероятности последующих лексем (Рисунок 4).

Рисунок 4 – Пример работы этапа Linear

Функция активации Softmax преобразует полученные оценки в вероятностное распределение, где сумма всех вероятностей тождественно равна единице. Данный механизм обеспечивает вариативность генерации текста.

Тем не менее, языковые модели обладают ограниченными знаниями, актуальными на момент завершения их обучения, и демонстрируют склонность к генерации недостоверной информации ("галлюцинации"). Согласно исследованиям [3, с.2], частота генерации ошибочных данных в специализированных запросах достигает 43%. Для минимизации указанного недостатка применяется технология Retrieval-Augmented Generation, дополняющая ответы модели релевантными данными из внешних источников [4, с. 7].

Однако LLM обладают ограниченными знаниями на момент их обучения и могут генерировать недостоверную информацию ("галлюцинации"). LLM выдают ложные данные в 43% специализированных запросов [5, с.403]. Для решения этой проблемы применяется технология RAG, которая дополняет ответы модели актуальными данными из внешних источников.

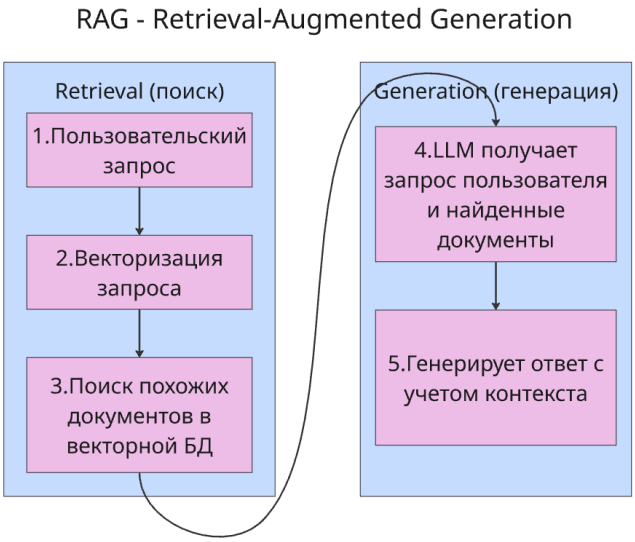

Рисунок 5 – Схема работы технологии RAG

Процедура RAG включает следующие этапы [7, с.703]:

1. Пользователь формирует запрос (например, "Какие новейшие методы в лечении диабета?").
2. Запрос трансформируется в векторное представление.
3. Система идентифицирует семантически близкие документы в предварительно подготовленной векторной базе знаний (фрагменты документов также векторизованы).
4. Релевантные фрагменты совместно с исходным запросом передаются на вход языковой модели.
5. Модель генерирует ответ, синтезируя информацию из предоставленного контекста, что обеспечивает достоверность и релевантность.

2. Разработка и оценка базы знаний

Для экспериментальной оценки эффективности RAG были подготовлены две версии инструкций по настройке коленного модуля:

- Инструкция №1: структурированное описание алгоритма настройки объемом 1196 слов.
- Инструкция №2: оригинальная инструкция производителя объемом 2130 слов, содержащая избыточные данные.

Исследование проводилось с использованием модели Llama 3.1 8B (2023 г.), требующей 8 ГБ оперативной памяти. Вычислительная среда включала Visual Studio Code и Python 3.10.11. Оценка ответов модели выполнялась с применением метрики BERTScore [6, с. 4], анализирующей семантическую близость текстов на основе контекстуальных эмбеддингов BERT:

1. Векторное представление: Тексты кодируются в эмбеддинги посредством BERT.
2. Попарное сравнение: Для каждого токена-кандидата вычисляется максимальное косинусное сходство с токенами эталонного текста.
3. Расчёт метрик:
Precision (P): Доля токенов кандидата, семантически покрытых эталоном.
Recall (R): Полнота представления эталонных токенов в кандидате [9, с.5].
4. F1-мера: Гармоническое среднее:

$$F_1 = 2 * \frac{P * R}{P + R}$$

В ходе тестирования языковая модель была дообучена подготовленными ранее инструкциями и протетсирована на ряде вопросов. Вопросы для тестирования подразумевали развернутый ответ. Оценка ответов с помощью метрики BERTScore выглядит следующим образом: скрипт сверки выдает значение характеристик Precision, Recall и F1-Score (пример – таблица 1). Пример результатов тестирования на представлен в Таблице 1. Полное тестирование проводилась на 20 вопросах.

Таблица 1

Результаты сравнительной оценки инструкций

Llama 3.1. , инструкция №1	Llama 3.1. , инструкция №2
Вопрос: С чего начать плользование коленным модулем?	
Precision: 0.728 Recall: 0.712 F1-Score: 0.720	Precision: 0.709 Recall: 0.691 F1-Score: 0.700
Вопрос: Как вести настройку коленного модуля, когда коленный модуль уже установлен пациенту?	
Precision: 0.677 Recall: 0.640 F1-Score: 0.658	Precision: 0.658 Recall: 0.645 F1-Score: 0.652
Вопрос: Как ориентироваться по вкладкам приложения для настройки?	
Precision: 0.680 Recall: 0.692 F1-Score: 0.686	Precision: 0.647 Recall: 0.691 F1-Score: 0.668

Анализ данных позволяет констатировать, что Инструкция №1 демонстрирует стабильно более высокие значения по всем метрикам (Precision, Recall, F1-Score), несмотря на статистически незначительную абсолютную разницу (0.01–0.03). Инструкция №2 показала объективно худшие результаты, что обусловлено наличием избыточной информации, снижающей релевантность

извлечения данных. Таким образом, для дообучения модели рекомендовано использование структурированной и лаконичной Инструкции №1.

Реализация цифрового ассистента

Архитектура системы (Рисунок 6) включает следующие компоненты:

- Локальный сервер LLM (модель Llama 3.1 8B) с разработанной в процессе исследования базой данных о настройке модуля.
- Сервер сокетов на Python для обработки запросов.
- Пользовательский интерфейс для взаимодействия с ассистентом.

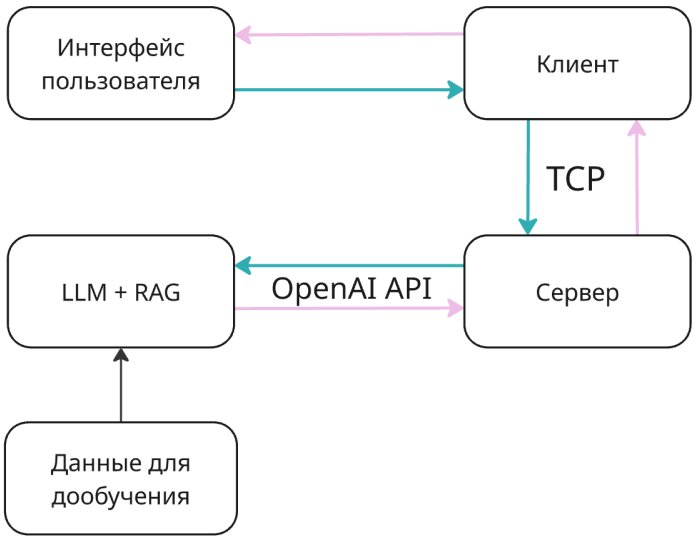

Рисунок 6 - Общая модель взаимодействия составных элементов системы

Экспериментальное тестирование подтвердило стабильность функционирования системы и её способность генерировать точные ответы на основе предоставленной базы знаний (Рисунок 7).

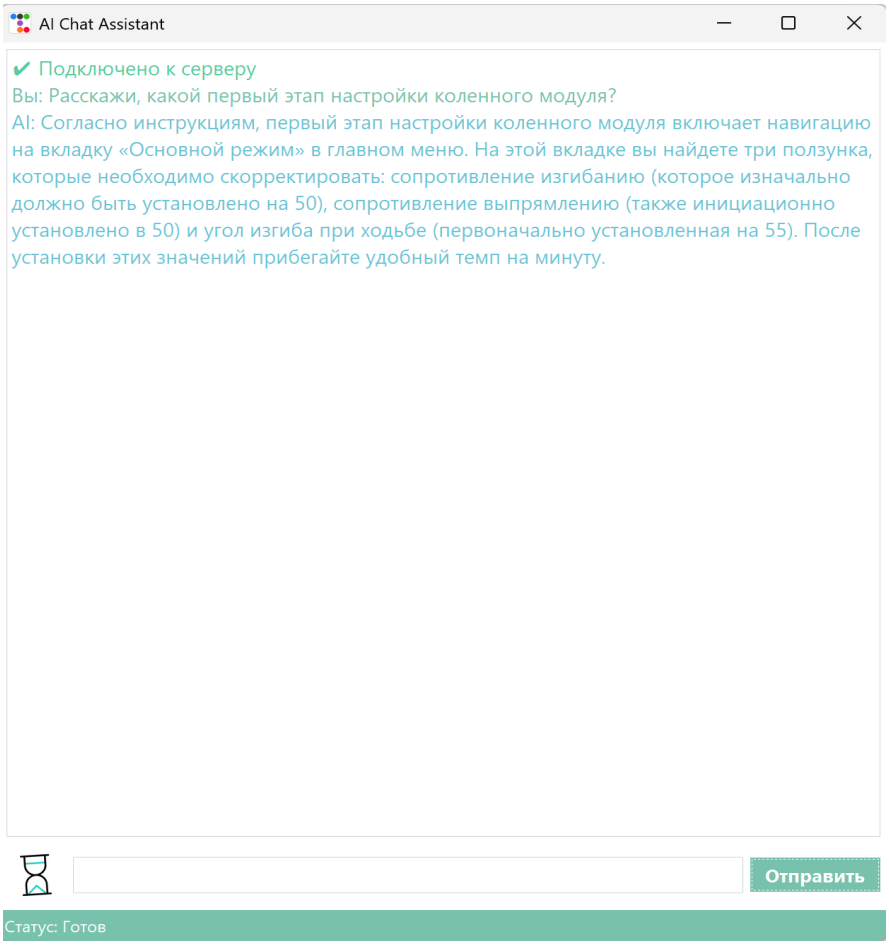

Рисунок 7 - скриншот экрана приложения после получения ответа

Разработанная методология обладает значительным потенциалом для адаптации в сфере автоматизации технической поддержки и обучения персонала работе со специализированным оборудованием, что определяет её универсальность и масштабируемость в промышленных приложениях.

Заключение

В рамках проведённого исследования разработана система цифрового ассистирования для настройки бионических коленных модулей на базе технологии Retrieval-Augmented Generation. Ключевые научные и практические результаты включают:

1. Экспериментальное подтверждение повышения точности ответов генеративных моделей при использовании RAG-архитектуры.

2. Разработку оптимальной структуры базы знаний, максимизирующей качество генерации релевантных инструкций.

3. Создание функционирующего клиент-серверного приложения, подтверждающего практическую применимость подхода.

Перспективные направления дальнейших исследований включают расширение функциональных возможностей системы и её адаптацию для применения в смежных областях, таких как медицинская диагностика и промышленная автоматизация.

Список литературы:

1. Журналы MDPI с открытым доступом / Ассистент на базе GPT для взаимодействия с информационными моделями построения в режиме реального времени: Дэвид Фернандес, Сахедж Гарг , Мэтью Никкел и Гурсанс Гувен. Дата публикации:13.08.2024 г. URL: https://www.mdpi.com/2075-5309/14/8/2499

2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates. - 2017 - URL http://papers.nips.cc/paper/7181 attention-is-all-you-need.pdf.

3. Гордеев Р.А., Белова Е.С. Применение больших языковых моделей в медицинских системах поддержки принятия решений // Искусственный интеллект и его приложения. — 2023. — № 2. — С. 34–45.

4. Кузнецов В.Д. Технологии RAG: анализ и применение // Информационные технологии. — 2024. — № 1. — С. 12–25.

5. Бородулин И.В. Увеличение точности больших языковых моделей с помощью расширенной поисковой генерации. – Омский государственный технический университет, 2023

6. Zhang et al. BERTScore: Evaluating Text Generation with BERT- 2020.

7. Оболенский Д.М., Шевченко В.И. 2024. Использование метода RAG и больших языковых моделей в интеллектуальных образовательных экосистемах. Экономика. Информатика, 51(3): 699–709.

DOI 10.52575/2687-0932-2024-51-3-699-709

9. Zhang et al. BERTScore: Evaluating Text Generation with BERT- 2020.