**УДК 004.891**

Обработка текста с помощью LLM для автоматической генерации миварных баз знаний

Андреев А.В., Тураев Г.В., Горячкин Б.С., Варламов О.О.

ФГБОУ ВО МГТУ им. Н.Э. Баумана, г. Москва, Россия

**Аннотация.** В статье приводится исследование по автоматической генерации миварных баз знаний из текста, предназначенного для обучения специалистов, либо представляющего собой инструкцию или свод правил. Рассматривается вопрос актуальности этого исследования. Анализируется предметная область больших генеративных моделей и проводится их сравнение для поставленной, а также описывается алгоритм реализации обработки текста. Обоснован вывод и оценивается возможность реализации системы на основе предложенной методологии.

**Ключевые слова.** мивар, миварная база знаний, искусственный интеллект, большая языковая модель, Wi!Mi, Разуматор.

TEXT PROCESSING USING LLM FOR AUTOMATIC GENERATION OF MIVAR KNOWLEDGE BASES

Andreev A.V., Turaev G.V., Goryachkin B.S., Varlamov O.O.

BMSTU, Moscow, Russia

**Abstract.** The paper presents a study on the automatic generation of Mivar knowledge bases from text intended for specialist training, or representing an instruction or set of rules. The relevance of this research is discussed. The subject area of large generative models is analyzed and their comparison is carried out for the stated task, as well as the algorithm for implementing text processing is described. The conclusion is substantiated and the possibility of implementing a system based on the proposed methodology is evaluated.

**Keywords**. mivar, mivar network, artificial intelligence, large language model, DSS, Wi!Mi, Razumator.

**Введение.** В настоящее время искусственный интеллект развивается быстрыми темпами, передовые разработки в сфере ИИ способны составить достойную конкуренцию устоявшимся методам. Сфера бизнеса проявляет активную заинтересованность в автоматизации процессов, задействованных в обработке различных текстовых документов и регламентов. Развитие автоматизации в решении подобных задач открывает масштабные перспективы широкого использования искусственного интеллекта для создания различных баз знаний, применяемых в самых разных сферах человеческой деятельности. Примером использования подобной методологии может служить создание миварных баз знаний в тех областях, где данных недостаточно для создания моделей на основе нейнейронных сетей.

Для решения проблемы задачи автоматизации сборки изделий предложено использовать миварные технологии [1] логического искусственного интеллекта (ЛИИ) [2], которые позволяют находить решение [3] с линейной [4] вычислительной сложностью [5] для задач [6] в форматах продукционных сетей «если – то» [7] или описания бизнес-процессов в формате «вход; выход; действие» [8].

Как известно, миварные технологии ЛИИ применяют в различных областях, включая: гетерогенные мультиагентные систем [9] и группы роботов [10], сравнения многомерных векторов в реальном времени [11], для создания автоматизированных систем управления [12] технологическими процессами [13], для интеллектуального [14] распознавания образов [15], для поиска [16] траекторий роботов [17] и планирования их действий [18], для параллельной обработки потоков информации [19], для моделирования мышления водителя [20] высокоавтоматизированных транспортных средств [21], для обучения людей в разных областях [22], управления образовательными [23] программами в вузе [24], смысловой обработки потоков текстов [25], определения безопасности применения компонентов крови [27], трехмерного интеллектуального моделирования [26], обеспечения безопасности информации [28] и для многих других областей [29]. Таким образом, тема работы актуальна и имеет важное значение для реализации поставленной задачи автоматизации сборки изделий с использованием роботов.

В рамках данной работы, мы произведем исследование предметной области и анализ существующих больших языковых моделей для создания алгоритма для генерации миварной базы знаний. С помощью этого алгоритма, можно будет существенно сократить время требуемое для обработки существующих знаний о предметной области в виде текста предназначенного для обучения специалистов, либо представляющего собой инструкцию или свод правил и их последующего переноса в миварное пространство. Примером сферы человеческой деятельности, в которой может пригодится подобный инструмент может послужить медицина, где создание достаточно эффективных моделей на основе нейронных сетей может быть затруднено или вовсе невозможно из-за недостатка данных для обучения.

**Постановка задачи.** Использование больших языковых моделей (LLM) для обработки текстовых документов является классическим подходом, основанным на промптах, который называется RAG (Retrieval‑Augmented Generation). До наступления эры LLM модели часто дополняли новыми данными, просто проводя их дообучение. Но теперь, когда используемые модели стали гораздо масштабнее, когда обучать их стали на гораздо больших объёмах данных, дообучение моделей подходит лишь для совсем немногих сценариев их использования. Дообучение особенно хорошо подходит для тех случаев, когда нужно сделать так, чтобы модель взаимодействовала бы с пользователем, используя стиль и тональность высказываний, отличающиеся от изначальных. Но дообучение уже не так хорошо себя показывает при необходимости добавления новых данных в модель. Это, как мне удалось выяснить, гораздо более распространённый бизнес‑сценарий. Кроме того, дообучение LLM требует больших объёмов высококачественных данных, серьёзных затрат на вычислительные ресурсы и много времени. Для большинства пользователей LLM всё это относится к разряду ограниченных ресурсов.

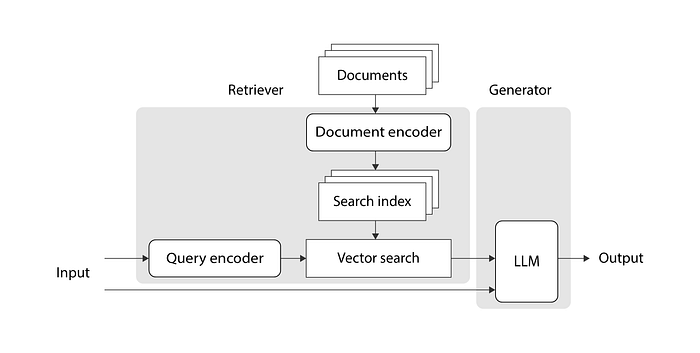


Рисунок 1 – Архитектура ИИ-системы, использующей RAG.

Основной принцип заключается в разбиении большого документа, который нельзя поместить в промпт, на мелкие части или чанки. Затем, когда задается вопрос, система ищет наиболее релевантные куски текста, где может находиться ответ. Из них формируется финальный промпт. Реализация этого механизма включает несколько этапов:

1. Разбить текст на чанки.
2. Перевести эти чанки в векторный вид для последующего выбора наиболее релевантных с помощью косинусной близости.
3. Сформировать промпт из этих небольших частей.
4. Отправить промпт в выбранную модель и получить ответ.

Таким образом, для создания миварных баз знаний с помощью большой языковой модели нужно выполнить следующие действия:

1. Подобрать алгоритм разбития текста на чанки (части).
2. Подобрать токенайзер для получения эмбедингов по этим кускам.
3. Подобрать промпт.
4. Подобрать модель.

Классическое описание данного механизма предполагает:

* Делить текст по переносам строк, стараясь сохранить слова целыми, с ограничением по количеству символов или токенов.
* Для токенизации предлагается использовать модель[ada-002](https://platform.openai.com/docs/guides/embeddings)от OpenAI.
* Использование OpenAI для генерации ответа.

Этот подход отлично работает, если не стоит задача использовать локальные решения и нет каких-либо ограничений на использование исходных данных, таких как коммерческая тайна. Возьмем модель на базе GPT4-Turbo (10К контекст) от OpenAI с родным токенайзером ada-002 в качестве референса. Исходя из поставленной задачи, нужно последовательно перебрать различные модели и токенайзеры, чтобы выбрать наиболее близкие к референсу. В итоге для сравнения были отобраны:

1. Один алгоритм чанкования.
2. Четыре вида токенайзеров: ada-002, RuBert, YandexEmbeding, RuBert Finetuned.
3. Один промпт.
4. 7 видов моделей: GPT4, GPT3.5, YandexGPT, GigaChat, Saiga, FineTuned Saiga.

**Алгоритм обработки текста.** В классическом подходе предлагается разбивать текст на абзацы или, где это невозможно, по словам с некоторым перекрытием. Этот метод хорош, когда каждый абзац содержит законченную мысль, абзацы схожи по размеру и структуре. Однако, в случае с юридическими документами или медицинскими инструкциями, некоторые абзацы могут быть подпунктами более крупных разделов. Например, пункт 3.3.3 описывает случаи, когда запрещено использовать температурную перевозку, с подпунктами 3.3.3.1 и 3.3.3.2, детализирующими отдельные случаи. Важно не разделять такие связанные пункты.

Также в результате тестов было выявлено, что чем длиннее чанк, тем хуже качество токенизации – она становится более обобщенной и происходят потери параметров при финальной генерации миварной базы знаний. Поэтому было принято решение, что оптимальный чанк – это не сугубо техническое решение, прежде всего должен сохранятся контекст. В итоге был подобран размер чанков до 150 слов, сгруппированные в соответствии с иерархией документа или иного исходного текста. Для этого был разработан скрипт, который анализирует структуру документа, создавая чанки, максимально крупные в рамках ограничений.

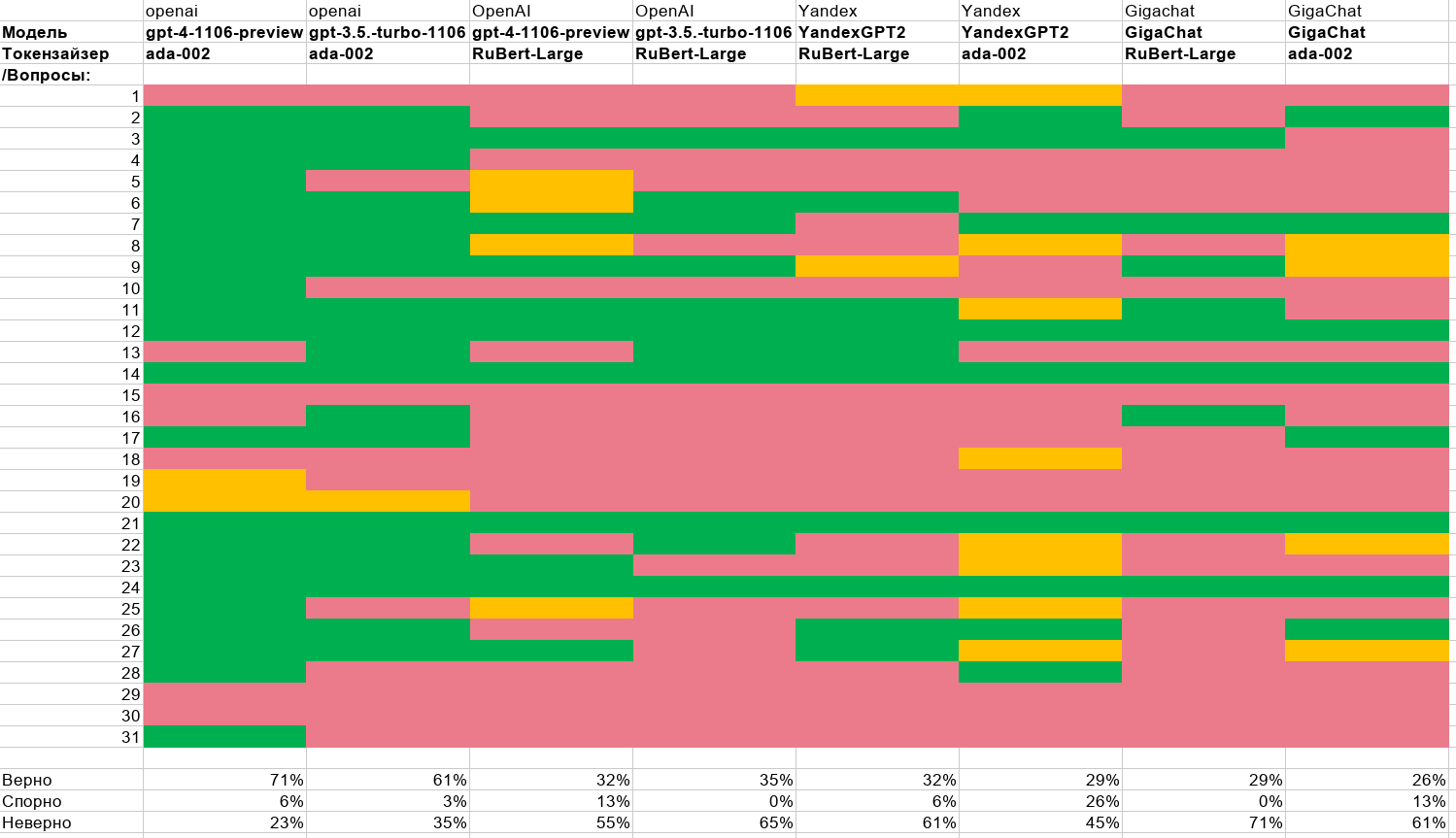
**Токенизатор, промты, модели.** Каждый чанк необходимо преобразовать в вектор, который затем записывался в PSQL. Были рассмотрели различные варианты: облачные сервисы OpenApi (ada-002) и YandexEmbeding (с ограничением до 10 запросов в секунду), а также множество локальных вариантов, включая Word2Vec, RuBert, rugpt и даже слои эмбеддингов у Llama. Облачные сервисы работали отлично, хотя у Яндекса была низкая квота по умолчанию. RuBert-Large оказался наиболее подходящим для нашей задачи, с использованием среднего пулинга (mean pooling). Скрытый слой Llama оказался полностью бесполезным. RuGPT, Fred и прочие от ai-forever - оказались хуже.

Размер промпта был ограничен до 10000 символов, так как этот размер хорошо подходил под все рассматриваемые модели, несмотря на разное количество занимаемых токенов для русских символов в разных моделях (например, 2-3 символа на каждый из 4096 токенов у Llama или 1 токен = 1 символ для GPT4).

В исследовании были использованы несколько источников: доступ к OpenAI, YandexGPT, GigaChat, а также мощную видеокарту 4090 для запуска и обучения локальных моделей. В качестве базовых моделей были выбраны Llama и Saiga. Для работы с облачными моделями температура генерации ответов бала установлена на ноль, чтобы повысить точность и уменьшить креативность ответов. Доступы к моделям были получены через API.

**Результаты тестирования.** Для тестирования использовался 31 запрос, так или иначе связанный с обработкой текста, используемого для создания баз знаний, на который система отвечала, используя разные комбинации моделей и токенайзеров. Каждый ответ затем был оценен вручную и занесен в таблицу 1 по следующей градации: верно (зелёный), спорно (желтый), неверно (красный). По горизонтали расположены вопросы, по вертикали – комбинации параметров. В ячейках таблицы указан цвет, соответствующий оценке ответа.

Таблица 1 – Сравнение полученных ответов



Из результатов тестирования занесенных в таблицу 1 видно, что GPT4-Turbo с использованием родного токенайзера ada-02 показал лучшие результаты: 71% ответов оценены как верные и только 23% как неверные. Наименее эффективной оказалась модель GigaChat с локальным токенайзером RuBert-Large. Результаты YandexGPT были чуть лучше, но разница несущественна.

Важно отметить, что модели OpenAI также показали плохие результаты на токенайзере RuBert. Это подтверждает предположение, что неправильный выбор частей текста (ошибка токенайзера) для формирования промпта приводит к неверным ответам. Почти во всех случаях, когда модели OpenAI давали неверные ответы, другие модели также показывали низкие результаты.

На основе первых полученных данных ясно, что стоит сосредоточиться на разработке локализованных под конкретную задачу токенайзеров и использовать ada-002 в качестве эталонного токенайзера. Это подчеркивает важность правильного выбора инструментов для обработки и подготовки текста перед использованием языковых моделей для задачи создания миварных баз знаний.

**Выбор токензайзера.** В RAG промпт задачи формируется динамически. Вопрос (A) конвертируется в вектор, и выбираются наиболее близкие ему математические векторы фрагментов регламента, сортируемые по убыванию "близости". Затем из них формируется запрос. Например, для вопроса A наиболее близкими являются фрагменты текста P, D, F, G, при этом дистанция между векторами A-P больше, чем A-D. С этой задачей лучше всего справляется ADA-02 от OpenAI.

Таким образом, можно создать собственный токенайзер и сравнить, какие фрагменты текста он выдаст для такого же запроса. Если их последовательность совпадает с ADA-02, это отлично. В противном случае, эту ошибку можно измерить и анализировать.

В качестве референса для сравнения и поиска лучшего токенайзера было использовано около 5000 текстовых фрагментов различного содержания: примеры вопросов, выдержки из регламента и инструкций и т.п. Из этих данных случайным образом формировались 250 пар текстов для расчета косинусной близости для выбранного токенизайзера. Далее рассчитывал отклонение между значением эталонного ADA-02 и выбранного варианта. Колебание этого отклонения и есть наш параметр. Если токензайзеры идентичны, то и среднеквадратичное отклонение между ними = 0.

Среднее квадратичное отклонение между каждой точкой позволило определить точность выбранного токенизатора. RuBert – Large демонстрировал погрешность около 10-11% в отношении ADA-02, sbert-nlu-ru-mt – 9-11%, что является улучшением, хоть и незначительным.

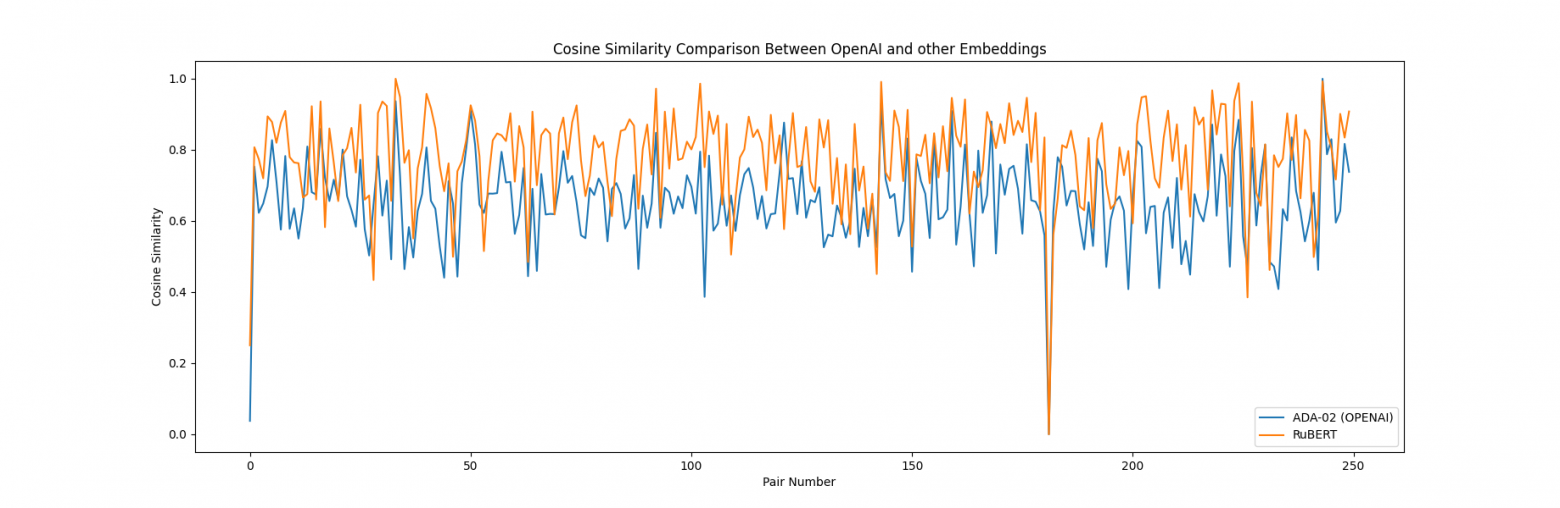


Рисунок 2 – Сравнение наиболее лучшего локального токенайзера с референсом.

После исследований промпта в ChatGPT было принято решение остановиться на трехслойной сети TensorFlow, без специальной настройки и подбора гиперпараметров. За 30 эпох обучения был создан локальный адаптер, который переводит вектор RuBert (1024 параметра) в вектор OpenAI (1526 параметров). Его спользование позволило улучшить результаты использования локального токенайзера. 30 эпох обучения и был создан адаптер. Он переводит вектор RuBert (1024 параметра) в вектор OpenAI (1526 параметров). Этот адаптер будет эффективно работать только с текстами, близкими к определенной тематике.

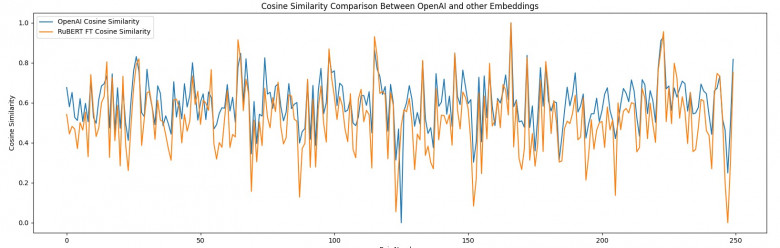


Рисунок 3 – Сравнение наиболее лучшего локального токенайзера с адаптером с референсом.

Синтетические тесты показали, что ошибка снизилась с 9-11% до 6%. При этом использование sbert-nlu-ru-mt не дало прироста, поэтому был выбран RuBert-large в качестве основы.Таким образом, был получен локально работающий токенизатор, близкий по эффективности к облачному ADA-02, что теоретически должно улучшить качество ответов.

**Проверка на практике, как работает FineTuned токенайзер.** Возьмем наш эталон в виде GPT-4 и пропустим через нашу машинку вопросов/ответов и отдадим на оценку ответов экспертам. Были получены результаты отмеченные в таблице 2:

Таблица 2 – Сравнение референса с RuBert – Large для модели GPT-4

|  |  |  |  |
| --- | --- | --- | --- |
| **GPT-4** | **ADA-02** | **RuBert** **- Large** | **RuBert-Large FineTuned** |
| Верно | 71% | 32% | 55% |
| Спорно | 6% | 13% | 19% |
| Ошибочно | 23% | 55% | 26% |

Значения правильных ответов на эталонной модели значительно выросли (с 32% до 55%). Ошибочных ответов уменьшилось в двое и равно эталонным (уменьшение с 55% до 26%, при 23% эталонных). Иными словами – наша стратегия сработала:

1. Правильный токенайзер повышает качество. Улучшение точности ембедингов с 11% до 6% даёт значимый результат.
2. Fine Tune токенайзера возможен.

Аналогичный результат мы видим и для моделей YandexGPT. Количество правильных ответов растет, ошибочных падает, что отражено в таблице 3.

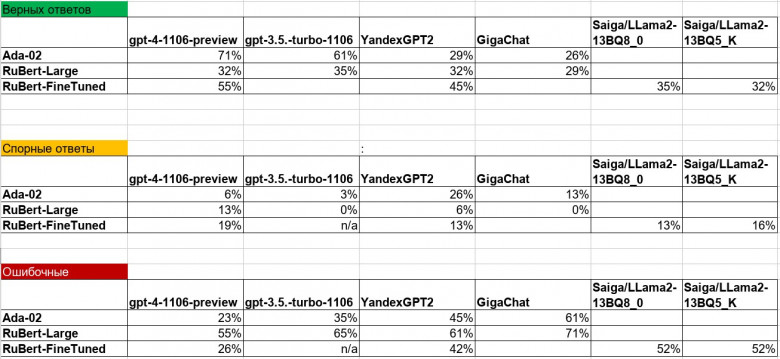
Таблица 3 – Сравнение референса с RuBert – Large для модели YandexGPT

|  |  |  |  |
| --- | --- | --- | --- |
| **YandexGPT** | **ADA-02** | **RuBert** **- Large** | **RuBert-Large FineTuned** |
| Верно | 29% | 32% | 45% |
| Спорно | 26% | 6% | 13% |
| Ошибочно | 45% | 61% | 42% |

**Локальные модели.** Развернем локальные модели на базе Saiga2 (LLama2).Решение не тестировать на токенизаторе от OpenAI и обычном RuBert было обоснованным, так как использование локальной модели с облачным токенизатором от OpenAI кажется абсурдным, а применение обычного RuBert нецелесообразно из-за его низкого качества.Для сравнения возьмем модели с двумя разными квантованиями. На 5 и 8. Первая может поместиться в видеокарту 3060 (версия 12GB), вторая уже требует карты 3090 или 4090 (версия 24GB).

Обе модели показали почти идентичные значения, показанные в таблице 4: 35% верных ответов, 52% ошибочных и 13% спорных ответов. Это немного лучше, чем у YandexGPT и GigaChat на базе обычного RuBert, но хуже, чем у RuBert-fine tuned. В целом, YandexGPT немного превосходит Saiga LLama2, что является положительным моментом.

Таблица 3 – Сравнение локальных моделей с облачными решениями



**Заключение.** На данный момент задача RAG с использованием локальных моделей (в закрытом контуре, без доступа к интернету и интерфейсу других моделей) не решается в полной мере из-за большого количества ошибочных ответов. Одновременно эту задачу можно решать, если есть доступ к облачным решениям OpenAI, таким как GPT4 с широким окном контекста, достигая 70-80% правильных ответов без особых усилий. Обратная сторона медали, кроме отсутствия контура безопасности – это стоимость. На момент написания статьи запрос будет обходиться примерно в 5-10 центов.

Основные ограничения локальных моделей связаны в первую очередь с правильным формированием частей первичного текста токенайзерами и в меньшей степени – с самими моделями. Недостатки моделей можно компенсировать более точными промптами и fine-tuning. Альтернативно, можно подождать развития технологий.

Работа над нарезкой и векторизацией данных – это задача, требующая внимания и которая не будет решена сама по себе без дополнительных исследований. Доработка FineTune токенайзера представляется технически выполнимой. Нужно больше экспериментировать с гиперпараметрами модели.

По вопросу нарезки данных (частей текста) возможны три пути. Первый, немного примитивный, но правильный с точки зрения продуктового подхода – ограничить RAG задачу информацией, которая умещается в 1-2 абзаца, возможно с ручной нарезкой. Второй – изучить возможности и технологии графового представления данных для более сложной обработки. Третий – ждать, пока появятся достойные локальные модели с длиной контекста (эффективной длинной) до 128К символов или обучать их самостоятельно. Но это не наш путь.

Оценка качества ответов производилась вручную экспертами, что придало исследованию практическую значимость. Одним из ключевых выводов является то, что значительную роль в доле качественных ответов играла модель, создававшая первичные эмбеддинги документа. Хороший эмбединг придавал больше роста в качестве, чем хорошая LLM.

Автоматическое создание миварных баз знаний с использованием болших языковых моделей возможно. Если есть возможность использования облачных решений, то использование RAD-задачи позволит существенно ускорить обработку исходного текста. Но для создания полностью автоматической генерации баз знаний требуется разработать полноценную методологию, включающую предобработку неподготовленного текста, проверку сохранения полноты информации, а также тестирование получившейся модели.

Список литературы

1. Varlamov, O. O. Wi!Mi expert system shell as the novel tool for building knowledge-based systems with linear computational complexity. *Int. Rev. Autom. Control* **11**, 314-325 (2018).
2. Варламов О.О. [Логический искусственный интеллект создан на основе миварного подхода!](https://www.elibrary.ru/item.asp?id=22955549) МИВАР: активные БД с линейным логическим выводом > 3 млн правил => понимание смысла + сингулярность в виртуальной реальности / Саарбрюкен, 2012. – 692 с. EDN: [TIITDH](https://www.elibrary.ru/tiitdh).
3. Varlamov, O. O. Evolutionary databases and knowledge for adaptive synthesis of intelligent systems. Mivar Information Space (2002).
4. Варламов О.О., Чибирова М.О., Сергушин Г.С., Елисеев Д.В. Практическая реализация универсального решателя задач "УДАВ" с линейной сложностью логического вывода на основе миварного подхода и "облачных" технологий // Приборы и системы. Управление, контроль, диагностика. 2013. № 11. С. 45-55. EDN: SQKHXZ.
5. Носов А.В., Владимиров А.Н., Потапова Т.С. и др. Программа "УДАВ": реализация линейной вычислительной сложности матричного метода поиска маршрута логического вывода на основе миварной сети правил // Искусственный интеллект. 2009. № 3. С. 443-448. EDN TIFIGD.
6. Варламов О.О., Антонов П.Д., Чибирова М.О. и др. МИВАР: машино-реализуемый способ автоматизированного построения маршрута логического вывода в базе знаний // Радиопромышленность. 2015. № 3. С. 28-43. EDN: UQEPGD.
7. Varlamov, O. O. Exhaustive elementary-incremental summing up of numbers with linear computational complexity. Avtomatizatsiya i Sovremennye Tekhnologii **1**, 34-41 (2003).
8. Осипов В.Г., Чувиков Д.А., Кривошеев О.В. и др. Планирование действий по обработке и сборке изделий в машиностроительном ИИ // В сборнике: МИВАР'22. Москва, 2022. С. 420-427. EDN LAJOSM.
9. Белоусова А.И. Использование миваров и многоуровневой модели гетерогенной мультиагентной системы на практике // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1-1 (39). С. 39-45. EDN: NECGGX.
10. Chuvikov, D. A., Varlamov, O. O., Aladin, D. V., Chernenkiy, V. M. & Baldin, A. V. Mivar models of reconstruction and expertise of emergency events of road accidents. IOP Conf. Ser. Mater. Sci. Eng. 534, 012007 (2019).
11. Семенов А.А. [Разработка метода сравнения двух многомерных векторов в реальном времени на основе миварных экспертных систем](https://www.elibrary.ru/item.asp?id=43805758) // [Проблемы искусственного интеллекта](https://www.elibrary.ru/contents.asp?id=43805748). 2020. [№ 2 (17)](https://www.elibrary.ru/contents.asp?id=43805748&selid=43805758). С. 94-109. EDN: [FXAOKP](https://elibrary.ru/fxaokp).
12. Сергушин Г.С., Чибирова М.О., Елисеев Д.В. и др. Информационное моделирование сложных автоматизированных систем управления технологическими процессами на основе миварных технологий // Искусственный интеллект. 2013. № 3. С. 126-138. EDN: TZBWRP.
13. Санду Р.А., Варламов О.О., Остроух А.В. [Миварные автоматизированные системы управления технологическими процессами для нефтяной промышленности России](https://www.elibrary.ru/item.asp?id=17071220) // [Автоматизация, телемеханизация и связь в нефтяной промышленности](https://www.elibrary.ru/contents.asp?id=33699475). 2011. [№ 11](https://www.elibrary.ru/contents.asp?id=33699475&selid=17071220). С. 37-40. EDN: [OJOUVV](https://elibrary.ru/ojouvv).
14. Синцов М.Ю., Озерин А.Ю., Кузин А.А. и др. О развитии миварного подхода к интеллектуальному распознаванию образов для работы с трехмерными объектами // Радиопромышленность. 2015. № 3. С. 172-183. EDN: UQEPOZ.
15. Волков А.С. О создании двухуровневой нейросетевой структуры для применения в машиностроении // В сборнике: МИВАР'22. Москва, 2022. С. 251-261. EDN: TXESUT.
16. Варламов О.О. Разработка квадратичной сложности методов поиска минимального разреза двухполюсных и многополюсных сетей // Искусственный интеллект. 2002. № 3. С. 371-375. EDN: TXHSIJ.
17. D.A. Shibanova, I.V. Stroganov, I.V. Rudakov, "Data Formalization in Transport System Modeling Using a Graph Database," Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2021 doi: 10.1109/ElConRus51938.2021.9396137.
18. Shadrin, S. S., Varlamov, O. O. & Ivanov, A. M. Experimental autonomous road vehicle with logical artificial intelligence. J. Adv. Transp. **2017**, 2492765 (2017).
19. Варламов О.О. [Параллельная обработка потоков информации на основе виртуальных потоковых баз данных](https://www.elibrary.ru/item.asp?id=20924703) // [Известия высших учебных заведений. Электроника](https://www.elibrary.ru/contents.asp?id=33926519). 2003. [№ 5](https://www.elibrary.ru/contents.asp?id=33926519&selid=20924703). С. 82-89. EDN: [RPXHAR](https://elibrary.ru/rpxhar).
20. Aladin, D. V. et al. Logic-based artificial intelligence in systems for monitoring the enforcing traffic regulations. IOP Conf. Ser. Mater. Sci. Eng. **534**, 012025 (2019).
21. Varlamov, O. O., Chuvikov, D. A., Aladin, D. V., Adamova, L. E. & Osipov, V. G. Logical artificial intelligence Mivar technologies for autonomous road vehicles. IOP Conf. Ser. Mater. Sci. Eng. **534**, 012015 (2019).
22. A.I. Vlasov, L.V. Juravleva, V.A. Shakhnov, "Knowledge-Based Model for Formal Representation of Complex System Visual Models," Advances in Intelligent Systems and Computing, vol. 1251 AISC, pp. 618-632, 2021. doi: 10.1007/978-3-030-55187-2\_45.
23. E.M. Akhmetshin, A.V. Kozachek, V.L. Vasilev, et al., "Development of Digital University Model in Modern Conditions: Institutional Approach," Digital Education Review doi: 10.1344/der.2021.40.17-32.
24. Егоров С.А., Гузева Т.А., Адамова Л.Е. и др. БЗ МЭС для учебной дисциплины "РКПТ" // В сборнике: МИВАР'22. Москва, 2022. С. 330-337. EDN: BCQGTO.
25. Варламов О.О., Егоров С.А. Развитие миварных технологий смысловой обработки потоков текстовых данных // В сборнике: МИВАР'22. Москва, 2022. С. 194-212. EDN: PBFFTZ.
26. Белоусов Е.А., Попов И.А., Евдокимов А.А. и др. Рекомендательная система диагностики сахарного диабета на основе механизма миварного вывода // Естественные и технические науки. 2021. № 7 (158). С. 169-174. EDN: JSFUSI.
27. Варламов О.О., Чувиков Д.А., Лемонджава В.Н. и др. Программный комплекс с поддержкой принятия решений о безопасности применения термолабильных компонентов крови // Медицинская техника. 2021. № 5 (329). С. 40-43. EDN: YZHHHV.
28. Chuvikov D.A., Kazakova N.A., Golovizin A.V. [3d Modeling And 3d Objects Creation Technology Analysis For Various Intelligent Systems](https://www.elibrary.ru/item.asp?id=22760123) // [International Journal of Advanced Studies](https://www.elibrary.ru/contents.asp?id=34043524). 2014. Т. 4. [№4](https://www.elibrary.ru/contents.asp?id=34043524&selid=22760123). С.16-22. EDN: [TEBOF](https://elibrary.ru/tebofl)L.
29. Chernenkiy, V., Gapanyuk, Y., Terekhov, V., Revunkov, G. & Kaganov, Y. The hybrid intelligent information system approach as the basis for cognitive architecture. Procedia Comput. Sci. **145**, 143-152 (2018).
30. The concept of an intelligent data lake management system: Machine consciousness and a universal data model.
31. Чувиков Д.А., Сараев Д.В. “Моделирование поведения автономного робота-гида в среде V-REP с использованием миварного конструктора алгоритмов” //XXVIII Международная инновационно- ориентированная конференция молодых ученых и студентов (МИКМУС - 2016) сборник трудов конференции. 2017. С. 302-305.
32. Медведев, Ю.Н., Чичерин, Н.Н. “Применение машинного обучения для повышения эффективности производственных систем”. //Информационные технологии в проектировании и производстве, т. 27, № 4, 2019, с. 62-70.
33. Подразделение «Робототехника» компании ABB. URL:<https://new.abb.com/ru/o-nas/technologies/tehnologii/promyshlennye-roboti> (дата обращения: 20.04.2023)
34. FANUC GLOBAL. URL: https://www.fanuc.com/index.html (дата обращения: 20.04.2023)
35. Chicco D. “Siamese Neural Networks: An Overview” //Cartwright, H. (eds) Artificial Neural Networks. Methods in Molecular Biology, vol 2190. Humana, New York, NY. https://doi.org/10.1007/978-1-0716-0826-5\_3
36. Alhajj, R., Rokne, J. “Entity Extraction”. // Encyclopedia of Social Network Analysis and Mining. Springer, New York, NY. https://doi.org/10.1007/978-1-4939-7131-2\_100333
37. Kayal, S., Afzal, Z., Tsatsaronis, G., Doornenbal, M., Katrenko, S., Gregory, M. “A Framework to Automatically Extract Funding Information from Text”. //Nicosia, G., Pardalos, P., Giuffrida, G., Umeton, R., Sciacca, V. Machine Learning, Optimization, and Data Science. LOD 2018. Lecture Notes in Computer Science (), vol 11331. Springer, Cham. https://doi.org/10.1007/978-3-030-13709-0\_27
38. Denecke, K. “Relation Extraction”. //Health Web Science. Health Information Science. Springer, Cham. https://doi.org/10.1007/978-3-319-20582-3\_9
39. Schantz, Herbert F. “The history of OCR, optical character recognition”. //Manchester Center, Vt.: Recognition Technologies Users Association. ISBN 9780943072012.