АВТОМАТИЗАЦИЯ РАБОТЫ С БАЗАМИ ЗНАНИЙ НА ОСНОВЕ ГИБРИДНОГО ПОДХОДА

Сальников Д. А., Ерофеев А. С.
Кафедра интеллектуальных информационных технологий,
Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: d.salnikov@bsuir.by, artem.erofeev2703@gmail.com

В работе рассмотрен гибридный подход, сочетающий в себе использование больших языковых моделей, классических RAG-систем и технологию OSTIS для автоматизации работы с базами знаний.

I. Введение

В контексте возрастающего применения больших языковых моделей актуальной задачей является изучение их фундаментальных ограничений, ключевыми из которых выступают галлюцинации, накопление ошибок и низкая управляемость. Галлюцинации, проявляющиеся в генерации правдоподобной, но фактически недостоверной информации, обусловлены статистической природой моделей и отсутствием механизмов верификации. Это зачастую приводит к кумулятивному эффекту накопления ошибок, когда незначительная неточность нарастает в авторегрессионном процессе, нарушая контекстную согласованность текста. Указанные проблемы, в свою очередь, тесно связаны с низкой управляемостью языковых моделей, то есть трудностью точного контроля содержания и стиля, эффективность которого критически зависит от качества пользовательских запросов и специализированной настройки. Комплексное преодоление этих взаимосвязанных проблем определяет возможность надежного применения больших языковых моделей в ответственных сценариях.

II. Проблематика и предлагаемый подход

Современные большие языковые модели, такие как DeepSeek, Grok и ChatGPT, часто оснащаются функцией веб-поиска, позволяющей компенсировать ограничения тренировочных данных – их неполноту или неактульность. Однако, данный подход имеет существенные недостатки. Во-первых, достоверность источников в сети не гарантирована: интернет содержит значительный объем недостоверной информации, включая тексты, сгенерированные моделями, упомянутыми ранее. Во-вторых, модель может неточно сформулировать поисковый запрос или выбрать нерелевантные ключевые слова. В-третьих, самостоятельный анализ и синтез информации из множества источников требует значительных вычислительных ресурсов, особенно для моделей с большим числом параметров. Эффективным подходом к смягчению данной проблемы является использование архитектуры RAG, которая ограничивает генерацию ответа строго релевантными документами, извлечёнными из внешнего источника знаний. В такой парадигме модель выполняет задачи синтеза и формулирования ответа на основе предоставленного контекста, что позволяет существенно снизить уровень галлюцинаций. Важно отметить, что применение RAG не требует использования крупных моделей и сохранения истории диалога, что открывает возможности для развёртывания более эффективных и компактных систем, демонстрирующих сопоставимую или повышенную точность в задачах, требующих фактической достоверности. Однако и этот метод имеет ограничение, связанное с приблизительным характером векторного поиска и отсутствием гарантий точности при извлечении информации [1].

Предлагаемый подход является гибридным, комбинирует преимущества классических RAG-систем, больших языковых моделей и технологии OSTIS. В архитектуре классической RAG-системы векторная база данных была заменена SC-машиной – эмулятором семантического компьютера, системой управления графовыми базами знаний. За обработку текста и запросов пользователя отвечают семантический фрагментатор текста и локальная большая языковая модель с малым количеством параметров (Gemma от Google), взаимодействие с базой знаний осуществляется посредством API SC-машины, сам пользователь взаимодействует с системой посредством графического интерфейса. Приведенная модель комбинирует преимущества вышеописанных подходов и стремится исправить их недостатки. Большая языковая модель «понимает», что пользователь имеет в виду и о чем говорится в представленном ей фрагменте текста, система автоматически составляет базу знаний на основе произвольного текстового документа без предварительной настройки, генерирует ответ для пользователя на естественном языке. База знаний хранит контекст с учетом семантики, предоставляет инструментарий для его поиска. Возможно внедрение различных команд для управления системой [2].

III. Программная реализация

В рамках реализации прототипа системы ключевыми стали следующие пункты:

- семантическое разбиение произвольных текстовых документов на более мелкие куски (фрагменты);
- автоматическая обработка фрагментов: извлечение тем, конкретных и абстрактных сущностей, процессов;
- автоматическое построение онтологии на основе ранее извлечённых из фрагментов данных с помощью API SC-машины;
- поиск релевантных фрагментов в базе знаний с помощью API SC-машины;
- приём и обработка запроса пользователя на естественном языке;
- генерация ответа с учётом семантики запроса и релевантных фрагментов на естественном языке.

Стоит отметить, что для разбиения текста на фрагменты использована библиотека semchunk, основанная на эффективном и высокоточном алгоритме фрагментации. Она создает фрагменты, которые более семантически значимы, чем обычные токены и рекурсивные фрагменты символов, такие как RecursiveCharacterTextSplitter от LangChain, а также на 85% быстрее, чем его ближайшая альтернатива, semantic-text-splitter. Также, составлена инструкция для большой языковой модели. В ней от модели требовалось всегда отвечать в строгом формате JSON, выделять из представленных фрагментов текста темы, конкретные сущности (имена собственные, научные термины, определения и так далее), абстрактные сущности (метафоры и нематериальные сущности), процессы [3].

IV. Заключение

В результате тестирования по методу чёрного ящика, продемонстрированного на рисунке 1, полная обработка текста длиной 15 тысяч символов заняла около 11 минут, длиной 5 тысяч символов — 2–3 минуты. При локальной работе как SC-машины, так и большой языковой модели на непроизводительном ноутбуке это является удовлетворительным результатом. Точность ответа зависит от формулировки вопроса, может сыграть существенную роль фактор случайности. Это общеизвестные проблемы, присущие большим языковым моделям. Чтобы попытаться улучшить результат, можно использовать большие языковые модели с большим количеством параметров и более производительное аппаратное обеспечение.

V. Список литературы

1. Trust, attitudes and use of artificial intelligence: A global study 2025 [Electronic resource]. – Mode of access: https://kpmg.com/xx/en/our-insights/ai-and-technology/trust-attitudes-and-use-of-ai.html. – Date of access: 12.10.2025.
2. GitHub – ostis-ai/sc-machine: Software implementation of semantic memory and its APIs [Electronic resource]. – Mode of access: https://github.com/ostis-ai/sc-machine. – Date of access: 13.10.2025.
3. semchunk · PyPI [Electronic resource]. – Mode of access: https://pypi.org/project/semchunk/. – Date of access: 14.10.2025.

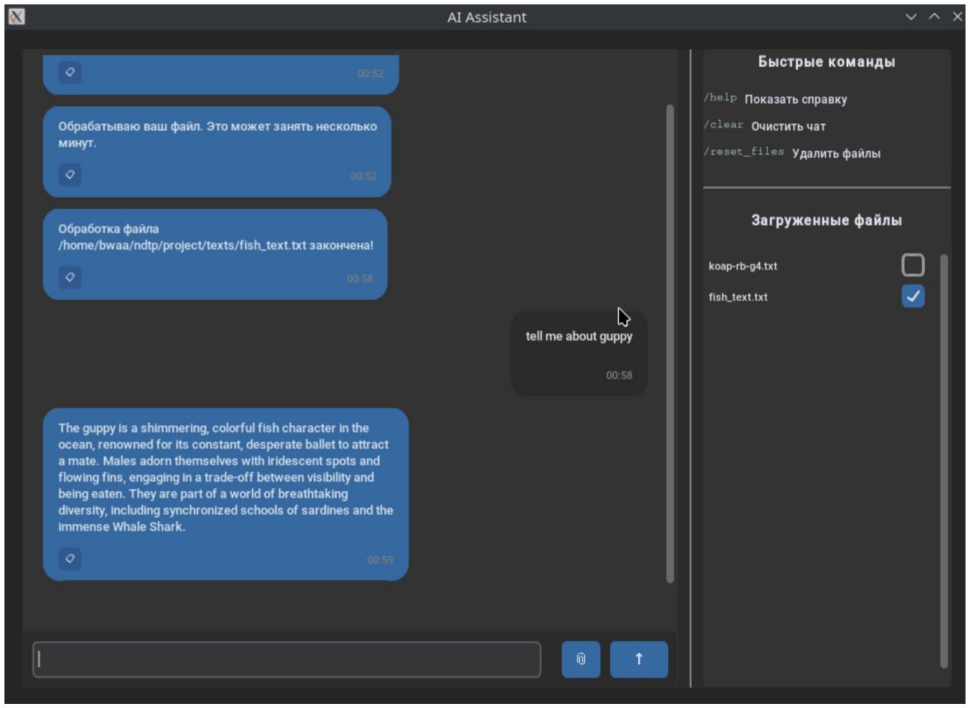

Рис. 1 – Тестирование системы. Обработка загруженного файла и ответ на пользовательский запрос.