ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА НА ТЕМУ “ГЕНЕРАЦИЯ И ВЕРИФИКАЦИЯ SPARQL-ЗАПРОСОВ С ИСПОЛЬЗОВАНИЕМ LLM ДЛЯ НАВИГАЦИИ ПО ГРАФАМ ЗНАНИЙ”

Студент: Запиров А. М.

Научный руководитель: Атаева О. М.

Проблема и цель

Проблема:

- Графы знаний — мощный инструмент для анализа связанных данных (RDF, SPARQL).
- Высокий технический барьер: для работы требуется знание сложного языка SPARQL.
- Специалисты предметной области не могут самостоятельно извлекать данные → зависимость от разработчиков, потеря времени.

Цель работы:
Разработать прототип системы, которая позволяет формулировать запросы к графу знаний на **естественном языке (русском)** и автоматически преобразует их в корректные SPARQL-запросы с последующей проверкой.

Источники данных и методы

Предметная область:

- **Данные:** Семантические графы знаний, доступные через SPARQL-эндпоинт (например, корпоративные онтологии, открытые базы вроде Wikidata/DBpedia).
- **Стек технологий:** Анализ W3C стандартов (RDF, SPARQL, OWL) и современных NLP-решений.

Ключевые методы решения:

- **NL-to-SPARQL:** Преобразование естественного языка в формальный запрос.
- **Использование Больших Языковых Моделей (LLM):** Современный подход, основанный на контекстном обучении (few-shot prompting), вместо устаревших rule-based систем.

■

Почему LLM и локальное развертывание?

- **Точность и гибкость:** Модели типа Mistral, Llama понимают контекст и сложные формулировки.
- **Безопасность:** Локальное развертывание исключает передачу конфиденциальных корпоративных данных и онтологий внешним сервисам.
- **Актуальность:** Соответствие трендам на демократизацию доступа к данным с помощью GenAI.

Выбранный стек:

- **LLM ядро:** Mistral 7B (оптимальное качество/ресурсы).
- **Развертывание LLM:** Ollama.
- **Backend:** Python (FastAPI), RDFLib, SPARQLWrapper.
- **Frontend:** Vue.js/Quasar Framework.
- **Контейнеризация:** Docker.

Что уже выполнено?

I. Создан рабочий прототип на python с использованием фреймворка Ollama для локального запуска модели.

II. Подготовлен датасет из 50 примеров пар «вопрос на русском – SPARQL для wikidata».

III. Разработаны и протестированы различные шаблоны промптов для LLM.

IV. Система успешно генерирует базовые SPARQL-запросы.

V. Реализован механизм постобработки, исправляющий типичные синтаксические ошибки.

VI. Достигнута базовая функциональность, подтверждающая жизнеспособность продукта.

План дальнейших работ

I. Расширение датасета, включая добавление более сложных запросов.

II. Тонкая настройка базовой модели Mistral, для лучшего понимания синтаксиса и семантики Wikidata.

III. Углубление семантической проверки для борьбы с «галлюцинциями» модели

IV. Разработка удобного веб-интерфейса.

Ожидаемый результат и значимость

I. Готовое к эксплуатации программное решение с веб-интерфейсом, позволяющее пользователям задавать вопросы на русском языке и получать корректные данные из wikidata без знания SPARQL.

II. Адаптация и комбинация современных методов для решения конкретной задачи.

III. Реализация с открытым исходным кодом может стать основой для дальнейших разработок в области семантического поиска и интеллектуальных ассистентов для работы с данными.