Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ Школа бакалавриата

ОТЧЕТ

По проекту «Разработка системы нормализации текстовых данных с использованием LLM»

по дисциплине «Проектный Практикум»

Заказчик: Черноскутов Д.В. _____
Куратор: Черноскутов Д.В. _____
к.т.н., исследователь исследовательского центра UDV Group

Студенты команды: «ML Production»

Белоусов Д.И. _____
Визнер В.А. _____
Горбук С.М. _____
Кудрин В.А. _____
Пермяков И.С _____

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель и задачи проекта

Цель проекта – разработка системы нормализации текстовых данных (товарных записей) с использованием больших языковых моделей (LLM). Нормализация подразумевает структурирование необработанных текстовых записей, выделение атрибутов (характеристик) и их значений в соответствии с заданной схемой.

Основные задачи:

1. Анализ существующих подходов к нормализации текстовых данных с применением LLM.
2. Исследование и сравнение нескольких open-source LLM по критериям качества, скорости и ресурсоемкости.
3. Разработка прототипа (Proof of Concept) системы, включающей:

- Предобработку входных данных (Excel-файлы с товарными записями).
- Разработку промптов и методов извлечения структурированных данных из текста.
- Постобработку и валидацию результатов.

4. Определение метрик эффективности (точность извлечения атрибутов, скорость обработки, потребление ресурсов).
5. Тестирование решения на реальных данных и формирование итогового отчета.

Актуальность и важность проекта

В современных бизнес-процессах, особенно в e-commerce, логистике и управлении ассортиментом, критически важно иметь структурированные данные о товарах. Однако на практике информация часто поступает в неформализованном виде (например, "Автошина 235/70R16 Cordiant Snow Cross 2 SUV шип"), что затрудняет автоматическую обработку, поиск и интеграцию с другими системами.

Ручная нормализация данных требует значительных временных и трудовых затрат, особенно при больших объемах информации. Использование LLM позволяет автоматизировать этот процесс, повысить скорость обработки и минимизировать ошибки.

Актуальность проекта обусловлена:

1. Ростом потребности в автоматизации обработки текстовых данных.
2. Развитием open-source LLM, которые могут быть развернуты локально без зависимости от коммерческих API.
3. Возможностью масштабирования решения на другие категории товаров и домены.

Область применения программного продукта

Разрабатываемое решение может быть использовано в следующих сферах:

1. Розничная торговля и маркетплейсы – автоматическое обогащение карточек товаров.
2. Логистика и складской учет – классификация и нормализация номенклатуры.
3. ERP и CRM системы – импорт и обработка данных от поставщиков.
4. Аналитика и data science – подготовка данных для дальнейшего анализа.

Ожидаемые результаты и планируемые достижения

По завершении проекта ожидается:

1. Готовый прототип (POC) системы нормализации текстовых данных на базе локальной LLM.
2. Набор скриптов и Jupyter-ноутбуков для предобработки, обработки и постобработки данных.
3. Сравнительный анализ нескольких open-source LLM (например, LLaMA 3, Mistral, Gemma) по метрикам (METEOR/ROGUE)
4. Определение технических требований и ограничений решения.
5. Готовый Excel-файл с нормализованными записями в качестве демонстрации работы системы.

Решение будет обладать ценностью для заказчика, так как позволит сократить время на обработку данных, повысить точность структуризации и снизить зависимость от ручного труда.

1 Основная часть

Разбор требований заказчика и формирование плана действий (backlog)

В рамках проекта была проведена детальная работа по анализу требований заказчика к программному продукту. Основные требования включают:

1. Обработка нескольких тысяч записей.
2. Нормализация в формате:
   - Выделение атрибутов (например, "Ширина профиля покрышки", "Диаметр обода").
   - Заполнение значений ("235", "R16") или отметка "Неизвестно".
3. Использование локальной open-source LLM (без облачных API).
4. Оценка эффективности по метрикам качества (точность извлечения атрибутов).
5. Результат в виде Excel-файла с нормализованными данными.

Для достижения этих целей был составлен план действий (backlog), включающий следующие этапы:

1. Анализ возможных алгоритмов и подготовка данных.
2. Разработка кодовой базы и инструментов нормализации.
3. Оценка качества работы алгоритмов.
4. Оптимизация и доработка решения.
5. Подготовка итогового отчета и презентации.

Анализ и сопоставление аналогов

На рынке существует ряд решений, направленных на обработку и нормализацию текстовых данных:

- Rule-based системы (например, парсинг по ключевым словам) – негибкие, требуют ручной настройки.
- NLP-библиотеки (spaCy, NLTK): позволяют выделять части речи и именованные сущности, но недостаточны для сложных структур.
- BERT-based модели: хороши для классификации и извлечения информации, но ограничены в понимании контекста специфических терминов.
- Large Language Models (LLMs): способны к глубокому анализу и семантической интерпретации, особенно при наличии подсказок (prompt engineering).

Наше решение отличается использованием LLM в связке с RAG (Retrieval-Augmented Generation), что позволяет:

- Обойтись без дообучения модели;
- Позволяет быстро улучшать качество за счет добавления примеров в базу.
- Обеспечить высокую точность на уровне пар ключ-значение.

Архитектура программного продукта

Компоненты системы:

1. Модуль предобработки данных:

- Чтение исходного файла Excel.
- Формирование входных строк для LLM.

2. RAG (Retrieval-Augmented Generation) система:

- Хранение размеченных примеров.
- Поиск наиболее близких записей к текущему запросу.
- Формирование промпта с примерами.

3. LLM (Large Language Models):

- Обработка входного запроса и примеров.
- Генерация нормализованной строки в формате JSON.

4. Модуль постобработки и сохранения:

- Парсинг выходного JSON.
- Сохранение результата в файл Excel.

Обоснование выбора архитектуры:

- RAG позволяет улучшить качество ответов за счёт использования внешних примеров без необходимости дообучения модели.
- Локальное исполнение LLM делает систему более безопасной и экономичной.
- Модульность архитектуры обеспечивает простоту масштабирования и расширяемости.

Общая архитектура:

1. Входной Excel файл с ненормализованными записями
2. RAG система
3. LLM + Промпт

4. Выходной JSON
5. Выходной Excel файл с нормализованными записями

Методология разработки

Процесс разработки был организован по методологии Agile, что позволило гибко реагировать на изменения требований и обеспечивать постоянную обратную связь с заказчиком. Основные этапы разработки включали:

1. Первая итерация – подготовка данных:

- Разметка исходного датасета (180 записей).
- Генерация расширенного датасета (1764 записи).
- Формирование полного датасета с входными и нормализованными данными.

2. Вторая итерация – разработка и тестирование:

- Создание размеченного тестового набора (100 записей, 50 типов товаров).
- Реализация алгоритма нормализации.
- Оценка качества через метрики ROUGE и METEOR.

3. Третья итерация – оптимизация:

- Тестирование различных LLM (phi-4, Mistral, Llama3).
- Сравнение по качеству и потреблению ресурсов.

4. Четвертая итерация – завершение:

- Подготовка отчета и презентации.

- Представление решения заказчику.

Вклад каждого участника и взаимодействие команды

Белоусов Дмитрий

1. Выступил тимлидом проекта, координировал работу команды.
2. Взаимодействовал с заказчиком, собирал требования и предоставлял промежуточную отчетность.
3. Участвовал в разметке данных.
4. Подготовил итоговый отчет и презентацию.

Кудрин Владислав

1. Осуществлял техническое руководство проектом.
2. Выбрал и протестировал модели, реализовал алгоритм нормализации.
3. Провел оценку качества работы системы с использованием метрик ROUGE и METEOR.
4. Анализировал ошибки и предлагал пути улучшения.

Виолета Визнер

1. Участвовала в разметке исходного датасета.
2. Помогала формировать структуру атрибутов для разных категорий товаров.
3. Участвовала в подготовке расширенного датасета.

Пермяков Иван

1. Выполнял разметку записей из исходного датасета.

2. Участвовал в формировании шаблонов нормализации для новых категорий товаров.
3. Помогал в тестировании первых версий алгоритма.

Горбук Сергей

1. Принимал участие в разметке данных.
2. Помогал в формировании примеров для RAG-системы.
3. Участвовал в обсуждении стратегии нормализации и анализа результатов.

Инструменты управления проектом:

- Telegram — основной канал коммуникации, где происходило обсуждение задач, выдача заданий и оперативная обратная связь.
- Google Colab — среда выполнения кода, скрипты для обработки датасетов.
- Google Таблицы — хранение и обработка датасетов.
- GitHub / Google Drive — хранение кода проекта.

С руководителем взаимодействие осуществлялось через Telegram. Руководитель формулировал общие цели, помогал в выборе направления, проверял промежуточные результаты и давал рекомендации по доработке.

ЗАКЛЮЧЕНИЕ

В ходе реализации проекта по разработке системы нормализации текстовых данных с использованием большой языковой модели (LLM) была достигнута основная цель — создание инструмента, способного эффективно обрабатывать и нормализовать товарные записи. Разработанный инструмент был успешно протестирован и опубликован на платформе GitHub, что обеспечивает доступность кода и возможность его дальнейшего использования и доработки. Проект включал в себя несколько итераций, каждая из которых была направлена на решение конкретных задач, таких как анализ требований, разработка алгоритмов, создание и тестирование кодовой базы, а также оптимизация работы системы. В результате работы была создана структура, позволяющая пользователям легко вводить необработанные данные и получать нормализованные записи с четко выделенными атрибутами. Кроме того, в процессе работы над проектом была собрана обширная база данных, содержащая как исходные, так и нормализованные записи, что позволяет значительно улучшить качество обработки данных и повысить точность модели. Использование подхода RAG (Retrieval-Augmented Generation) для получения примеров нормализации текста также продемонстрировало свою эффективность, позволяя минимизировать потребности в больших объемах данных для дообучения. Таким образом, проект не только достиг поставленных целей, но и создал основу для дальнейшего развития и улучшения системы нормализации данных. Мы уверены, что разработанный инструмент будет полезен для заказчика и сможет значительно упростить процесс обработки товарных записей.

Исходный код проекта доступен на GitHub: https://github.com/vladlen32230/TextNormalization

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://habr.com/ru/articles/786288/ - статья про инструменты нормализации.
2. https://github.com/ankita9800/Product-Classification-Using-LLM/blob/main/Product_Classification_Using_LLM.pdf - товарная классификация с помощью LLM.
3. https://habr.com/ru/articles/775842/ - введение в LLM.
4. https://habr.com/ru/articles/779526/ - Retrieval-Augmented Generation.
5. https://avinashselvam.medium.com/llm-evaluation-metrics-bleu-rogue-and-meteor-explained-a5d2b129e87f - ROUGE + METEOR метрики.
6. https://huggingface.co/microsoft/phi-4 - phi-4 модель.
7. https://ollama.com/library/llama3 - llama3 модель.