

Арутюнова Александра Дмитриевна, студентка, Сибирский государственный университет геосистем и технологий, Россия, Новосибирск

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ ТЕКСТОВЫХ ОПИСАНИЙ ТОВАРОВ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКОВОЙ МОДЕЛИ

Аннотация. В статье рассматривается процесс разработки веб-приложения, предназначенного для автоматической генерации текстовых описаний товаров с использованием современных языковых моделей. Целью исследования является повышение эффективности создания описаний для электронных коммерческих платформ. Применение языковых моделей позволяет сократить временные и трудовые затраты, обеспечивая при этом высокое качество и уникальность контента. В работе описываются архитектура приложения, выбранные технологии и методы интеграции языковой модели, а также представлены результаты тестирования системы.

Annotation. The article discusses the development process of a web application designed for automatic generation of product descriptions using modern language models. The aim of the study is to increase the efficiency of creating descriptions for e-commerce platforms. The use of language models reduces time and labor costs while ensuring high quality and uniqueness of content. The paper describes the application's architecture, selected technologies, methods of integrating the language model, and presents the system testing results.

Ключевые слова: генерация текста, языковая модель, веб-приложение, электронная коммерция, автоматизация, обработка естественного языка, FastAPI, React, PostgreSQL, REST API, машинное обучение

Keywords: text generation, language model, web application, e-commerce, automation, natural language processing, FastAPI, React, PostgreSQL, REST API, machine learning

Введение

В условиях стремительного развития электронной коммерции качественные и информативные описания товаров играют ключевую роль в привлечении и удержании клиентов. Традиционный способ создания таких описаний требует значительных временных и трудовых ресурсов, что особенно актуально для крупных интернет-магазинов с обширным ассортиментом продукции. Современные достижения в области обработки естественного языка и развития языковых моделей открывают новые возможности для автоматизации этого процесса.

Цель данной работы — разработать веб-приложение, способное автоматически генерировать текстовые описания товаров, используя языковую модель. Задачи исследования включают:

- Анализ существующих решений и технологий в области генерации текстов.
- Проектирование архитектуры веб-приложения с учетом модульности и масштабируемости.
- Выбор и интеграция подходящей языковой модели для генерации описаний.
- Разработка и тестирование прототипа системы.

Основная часть

Анализ существующих решений

Современные исследования в области генерации текстовых описаний товаров демонстрируют высокую эффективность применения больших языковых моделей (LLM). Например, в работе Zhou et al. (2023) представлена методология автоматизации генерации описаний товаров с использованием модели LLAMA 2.0 7B, обученной на данных Walmart. Результаты показали значительное улучшение качества описаний и повышение вовлеченности пользователей.

Другие подходы включают использование мультимодальных моделей, таких как MAnTiS, которые учитывают как текстовую, так и визуальную информацию о товаре, что позволяет создавать более точные и привлекательные описания (Sollami & Jain, 2021).

Особенности технической реализации

Приложение разработано с использованием современных веб-технологий. Во фронтенде используется **React** как библиотека для построения пользовательского интерфейса, что обеспечивает высокую отзывчивость и интерактивность. Компонентный подход React позволяет масштабировать интерфейс по мере роста функциональности приложения.

На стороне сервера используется **FastAPI**, современный Python-фреймворк, отличающийся высокой производительностью и поддержкой асинхронных операций. Это особенно важно в условиях интенсивного взаимодействия с внешними API (например, OpenAI API) и базой данных.

Все взаимодействия между фронтендом и бэкендом организованы с использованием **REST API**, что упрощает тестирование, документирование и будущую интеграцию с другими системами. API покрыт документацией Swagger/OpenAPI, автоматически генерируемой FastAPI.

Для хранения данных выбрана реляционная СУБД **PostgreSQL**, отличающаяся стабильностью, поддержкой JSON-типов и возможностями расширения. Схема базы данных включает таблицы товаров, характеристик и истории генерации описаний, что позволяет реализовать аналитику и контроль качества сгенерированного контента.

Безопасность и масштабируемость

Важным аспектом является обеспечение безопасности при работе с пользовательскими данными. Для этого реализована система авторизации и аутентификации, использующая **JWT-токены** (JSON Web Token). Это позволяет обеспечить защиту от несанкционированного доступа и контроль над сессиями пользователей.

Кроме того, приложение проектируется с учётом принципов **масштабируемости**. Благодаря разделению на слои (интерфейс, логика, данные, интеграция с моделью), система может быть легко адаптирована под распределённую архитектуру (например, миграция логики генерации в микросервис или использование очередей сообщений — RabbitMQ, Redis). Это особенно важно при росте числа пользователей или переходе к использованию собственной модели, развёрнутой локально или в облаке.

Дополнительно предусмотрено логирование ключевых операций и система сбора метрик (например, с использованием Prometheus и Grafana), что упрощает мониторинг и отладку.

Архитектура веб-приложения

Разрабатываемое веб-приложение построено по принципу клиент-серверной архитектуры и включает следующие компоненты:

- **Frontend:** Реализован с использованием React, обеспечивает интерактивный пользовательский интерфейс для ввода данных о товаре и отображения сгенерированных описаний.
- **Backend:** Построен на FastAPI, обрабатывает запросы от клиента, взаимодействует с языковой моделью и базой данных.
- **Языковая модель:** Интеграция с OpenAI API для генерации текстовых описаний на основе введенных характеристик товара.
- **База данных:** PostgreSQL используется для хранения информации о товарах и сгенерированных описаниях.

Взаимодействие между компонентами организовано через REST API, обеспечивая модульность и возможность масштабирования системы.

Интеграция языковой модели

Для генерации описаний используется языковая модель, предоставляемая через OpenAI API. Модель принимает на вход структурированные данные о товаре (название, характеристики, категория) и возвращает сгенерированный

текст описания. Применение языковой модели позволяет учитывать контекст и создавать уникальные, грамматически правильные описания, соответствующие стилю и тону, заданным в параметрах запроса.

Результаты тестирования

Прототип системы был протестирован на выборке из 100 товаров различных категорий. Оценка качества сгенерированных описаний проводилась по следующим критериям:

- **Уникальность:** Все описания были уникальными, без повторов.
- **Грамматическая правильность:** Не выявлено грамматических ошибок.
- **Информативность:** Описания содержали ключевые характеристики товара и были понятны для потребителя.

Результаты тестирования подтвердили эффективность разработанного решения и его потенциал для применения в реальных условиях электронной коммерции.

Выводы

- Разработано веб-приложение, способное автоматически генерировать текстовые описания товаров с использованием языковой модели.
- Применение языковой модели позволяет значительно сократить временные и трудовые затраты на создание описаний, обеспечивая при этом высокое качество и уникальность контента.
- Архитектура приложения обеспечивает модульность и масштабируемость, что позволяет адаптировать систему под различные требования и объемы данных.
- Результаты тестирования подтвердили эффективность разработанного решения и его готовность к внедрению в реальные бизнес-процессы.

Список литературы

1. Zhou J., Liu B., Acharya J. N., Hong Y., Lee K.-c., Wen M. (2023). Leveraging Large Language Models for Enhanced Product Descriptions in eCommerce. arXiv preprint arXiv:2310.18357.
2. Sollami M., Jain A. (2021). Multimodal Conditionality for Natural Language Generation. arXiv preprint arXiv:2109.01229.
3. Zhan H., Zhang H., Chen H., Shen L., Ding Z., Bao Y., Yan W., Lan Y. (2021). Probing Product Description Generation via Posterior Distillation. arXiv preprint arXiv:2103.01594.
4. Peng X., Sollami M. (2022). XFBoost: Improving Text Generation with Controllable Decoders. arXiv preprint arXiv:2202.08124.
5. Wikipedia contributors. (2025). Natural language generation. Wikipedia. Retrieved May 13, 2025, from https://en.wikipedia.org/wiki/Natural_language_generation

References

6. Zhou J., Liu B., Acharya J. N., Hong Y., Lee K.-c., Wen M. (2023). Leveraging Large Language Models for Enhanced Product Descriptions in eCommerce. arXiv preprint arXiv:2310.18357.
7. Sollami M., Jain A. (2021). Multimodal Conditionality for Natural Language Generation. arXiv preprint arXiv:2109.01229.
8. Zhan H., Zhang H., Chen H., Shen L., Ding Z., Bao Y., Yan W., Lan Y. (2021). Probing Product Description Generation via Posterior Distillation. arXiv preprint arXiv:2103.01594.
9. Peng X., Sollami M. (2022). XFBoost: Improving Text Generation with Controllable Decoders. arXiv preprint arXiv:2202.08124.
10. Wikipedia contributors. (2025). Natural language generation. Wikipedia. Retrieved May 13, 2025, from https://en.wikipedia.org/wiki/Natural_language_generation

