УДК 004.855.5

Первышин Артем Николаевич, магистрант, Северный Арктический федеральный университет имени М.В. Ломоносова, г. Архангельск

РЕАЛИЗАЦИЯ MLOPS ЦИКЛА ДЛЯ МОДЕЛИ СУММАРИЗАЦИИ ТЕКСТА

Аннотация. В статье рассматривается практическая реализация цикла MLOps для задачи абстрактивной суммаризации текста на примере модели T5 и русскоязычного новостного сайта Gazeta.ru. Описываются три основных этапа MLOps: инженерия данных (сбор и подготовка датасета с использованием Scrapy и Docker), разработка модели (дообучение T5 с применением huggingface-cli и локальных ресурсов) и развертывание модели (создание Django REST API веб-приложения, контейнеризация). Представлен подход к автоматизации каждого этапа, включая загрузку данных из репозитория Hugging Face, обучение модели, и развертывание веб-приложения в контейнере. Описывается создание batch-файла для последовательного запуска всего цикла MLOps, что позволяет воспроизводить процесс обучения и развертывания модели в производственной среде с использованием ресурсов локальной машины.

Annotation. The article examines the practical implementation of an MLOps cycle for the task of abstractive text summarization using the T5 model and the Russian-language news website Gazeta.ru. It describes the three main stages of MLOps: data engineering (data collection and preparation using Scrapy and Docker), model development (fine-tuning T5 using huggingface-cli and local resources), and model deployment (creating a Django REST API web application, containerization). An approach to automating each stage is presented, including loading data from the Hugging Face repository, training the model, and deploying the web application in a container. The creation of a batch file for the sequential execution of the entire MLOps cycle is described, allowing the reproduction of the

model training and deployment process in a production environment using local machine resources.

Ключевые слова: MLOps, суммаризация текста, T5, автоматизация, Docker, Hugging Face,Django, развертывание моделей.

Keywords: MLOps, text summarization, T5, automating, Docker, Hugging Face,Django, model deployment.

MLOps (Machine Learning Operations) – это набор практик, направленных на автоматизацию и улучшение процесса разработки, развертывания и обслуживания моделей машинного обучения в производственной среде.

Данный набор практик позволяет быстрее переводить модели от стадии прототипа к реальному применению, гарантирует, что модели работают стабильно и эффективно в производственной среде, даже при увеличении нагрузки, обеспечивает постоянный мониторинг и дообучение моделей для поддержания высокой точности [1].

В зависимости от процессов компании, в которой происходит работа с моделью машинного обучения, цикл MLOps может включать или не включать в себя те или иные этапы, но всегда в нём так или иначе будут присутствовать следующие три этапа.

Первый этап – Data Engineering (Инженерия данных). Его задачи состоят в сборе, обработке, очистке и подготовке данных для обучения моделей. Для встраивания этого этапа в цикл MLOps необходимы: автоматизация пайплайнов сбора и обработки данных, управление качеством данных, отслеживание происхождения данных.

Второй этап – Model Development (Разработка модели). Его задачи состоят в выборе алгоритмов обучения моделей, обучение моделей, оценка производительности, экспериментирование. Для встраивания этого этапа в цикл MLOps необходимы: управление версиями моделей, автоматизация обучения и оценки, отслеживание экспериментов, репродуктивность экспериментов.

Третий этап – Model Deployment (Развертывание модели). Его задачи состоят в интеграции модели в производственную среду, создание API, обеспечение масштабируемости. Для встраивания этого этапа в цикл MLOps необходимы: автоматизированное развертывание моделей, контейнеризация (например, с помощью Docker), управление инфраструктурой, развертывание на различных платформах, версионирование кода и самих моделей [2].

Задача цикла MLOps обеспечить возможность автоматизировано воспроизводить эти этапы.

Рассмотрим реализацию первого этапа, который направлен на автоматизированный сбор или дополнение датасета. Поскольку необходимо обучить модель генерировать резюме текста, подходящим вариантом для составления датасета является парсинг новостного сайта. В новостных сайтах зачастую имеется заголовок и краткое описание новости, одним из таких сайтов является Gazeta.ru. Для парсинга данного сайта использовалась библиотека python – scrapy. Scrapy предоставляет инструменты для скачивания, обработки и сохранения данных, а также для навигации по сайтам и извлечения данных из различных форматов. Результирующий датасет содержит в себе более 60000 строк, на рисунке 1 показана часть датасета.

text string · lengths 3.22k⎓4.25k 36.4%	summary string · lengths 291⎓345 22.2%	title string · lengths 46⎓60 27.1%	date string · lengths 19 100%
Благодаря информации, предоставленной США, сотрудники ФСБ России задержали …	Российские силовики предотвратили теракт в Санкт-Петербурге, который двое…	США помогли: ФСБ предотвратила теракт в Петербурге	2019-12-29 21:47:05
Американские военные нанесли несколько ударов по пяти объектам «Хезболлы» в …	Американские военные атаковали три объекта «Хезболлы» в Ираке и два в Сир…	Пять атак беспилотников: США ударили по объектам «Хезболлы»	2019-12-29 23:21:59
С 1 января 2020 года в России вступает в силу целый ряд обновлений и дополне…	С Нового года россиян ждет целый ряд новаций. Так, с начала 2020 года уве…	«Свести концы с концами»: что изменится с Нового года	2019-12-30 08:16:40
За 20 лет пребывания Владимира Путина у власти Россия смогла восстановить з…	Россия при Владимире Путина смогла восстановить значительную часть влияни…	Вернул былую мощь: Bloomberg назвал достижения Путина	2019-12-30 09:29:40
Все гражданские самолеты, производимые в России, могут переименовать, чтобы …	Все российские гражданские авиалайнеры могут переименовать, заявил глава …	Под одну гребенку: российские лайнеры собираются переименовать	2019-12-30 09:48:12
Вызов российского посла в Польше Сергея Андреева в польский МИД после слов …	Вызовом российского посла в Польше в МИД республики Варшава пытается пере…	«Переиграть в свою пользу»: зачем посла РФ вызвали в МИД Польши	2019-12-30 09:50:48

Рис. 1. Датасет, собранный через парсинг Gazeta.ru

После сбора, датасет загружается в репозиторий huggingface. Для более удобного и универсального запуска данного этапа весь проект по сбору датасета был загружен в docker-контейнер, что позволит запускать данный этап на любой системе, содержащей docker.

Далее, рассмотрим реализацию второго этапа, который предназначен для автоматизированного обучения модели на собранном ранее датасете.

Для абстрактивной суммаризации зачастую исполюзуются модели, имеющие архитектуру transformer. Опираясь на проведенные исследования [3], можно сделать вывод, что наиболее качественную суммаризацию русскоязычного текста обеспечевает модель T5.

Предоставленную модель необходимо было дообучить на собранном датасете.

T5 требует особого форматирования входных данных. Важно, чтобы данные были в формате «текст на текст». Это значит, что входные и выходные данные должны быть представлены в виде текста. В нашем случае текстом на вход является полный текст новости, а текстом на выход – краткий пересказ новости.

Модель дообучается, корректируя свои параметры, чтобы минимизировать функцию потерь, специфичную для текущей задачи. В случае суммаризации текста, модель обучается генерировать краткие и точные сокращения входного текста. В процессе дообучения обычно корректируются веса модели, а в контексте архитектуры T5, дообучение предполагает обновление параметров как кодирующего, так и декодирующего слоев [4].

Сначала был составлен файл конфигурации обучения модели на основе файла основной модели T5. Некоторые параметры были изменены ввиду ограничений по мощности видеокарты. Например, снижен batch size до 4 (было 16). Далее был создан проект на python решающий следующие задачи: загрузка конфигурации, токенизатора, данных и подготовка датасетов; инициализация и настройка модели; настройка параметров обучения и создание тренера; запуск процесса обучения; сохранение обученной модели и токенизатора.

Данная модель показала приемлемые результаты, метрики показаны на рисунке 2.

Model	R-1-f	METEOR	BLEU
rut5_base_sum_gazeta	32.2	25.7	**12.3**

Рис. 2. Результаты модели по метрикам

Поскольку обучение модели transformer требует значительное количество времени и ресурсов (в частности ресурсов видеокарты), этот этап не получится произвести прямо в контейнере. Задачу обучения необходимо делегировать сторонним ресурсам, специализирующимся на машинном обучении и предоставляющим техническое оборудование для обучения. В данном случае было принято решение вместо удаленного ресурса использовать локальную машину, на которой будет запускаться весь цикл MLOps [5].

Таким образом необходимо чтобы на локальной машине произошли следующие действия.

Во-первых, необходимо загрузить датасет из репозитория huggingface. Для этого следует воспользоваться huggingface-cli, чтобы авторизоваться и загрузить датасет.

Во-вторых, необходимо запустить главный скрипт, который обучает модель на датасете. Сделать это можно также через консольную команду.

В-третьих, необходимо загрузить полученную модель в отдельный репозиторий huggingface. Сделать это можно через команду huggingface-cli.

Команды, для описанных выше действий показаны на рисунке 3.

```
pip install -U "huggingface_hub[cli]"
huggingface-cli login --token ЗДЕСЬ_ВСТАВИТЬ_ТОКЕН
huggingface-cli download SUPERMEGAok/sum_dataset --local-dir C:\PythonProjects\MyT5TrainingV3\datasets\gazeta_huggingface
python main.py
huggingface-cli upload SUPERMEGAok/MyT5Sum ./trained_model
```

Рис. 3. Команды для этапа обучения модели

И, наконец, рассмотрим третий этап цикла MLOps. На этом этапе необходимо создать инструмент, применяющий обученную модель на практике. В данном случае было создано веб-приложение на основе фреймворка Django REST API. Пользователю веб-приложения доступны методы для определения тематики и суммаризации текста.

Пример работы метода для определения тематики и суммаризации текста показан на рисунке 4.

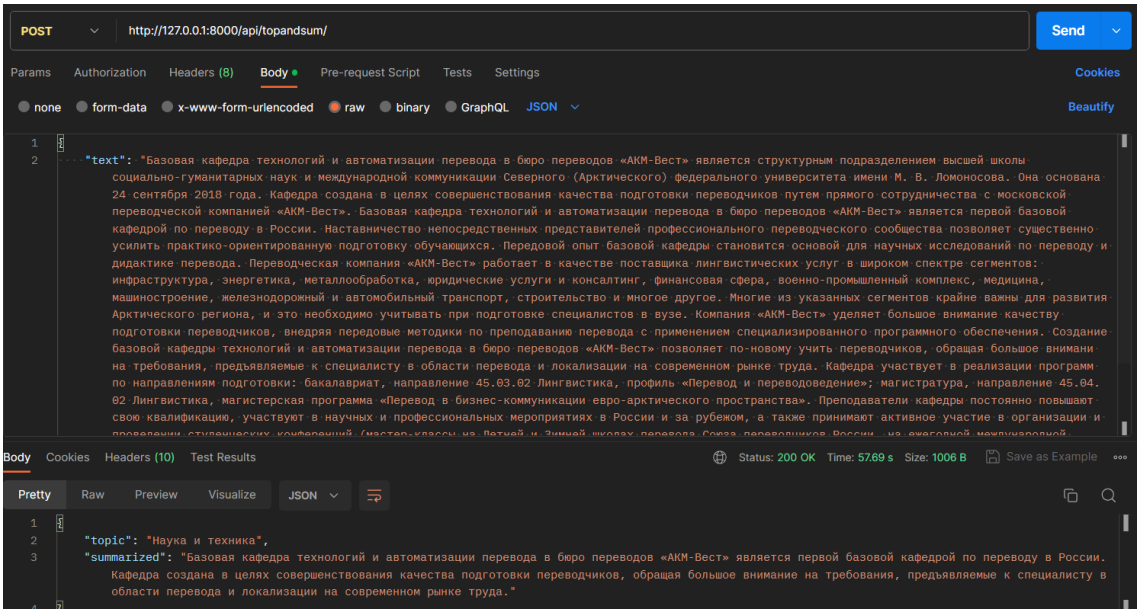

Рис. 4. Работа метода для определения тематики и суммаризации

После создания приложения необходимо встроить его в цикл MLOps. Т.е. необходимо добиться автоматизированного развертывания, независимости от ОС, универсальности. Перечисленные требования можно реализовать через запуск приложения в контейнере.

После реализации всех трех этапов, необходимо запустить цикл MLOps. В данном случае, предполагается, что на локальной машине будет доступ ко всем необходимым проектам. Исходя из этого, весь цикл можно запустить путем последовательного выполнения команд оболочки, например Windows.

Для последовательного выполнения команд был создан batch файл, его содержимое показано на рисунке 5.

```
cd C:\PythonProjects\dataset_parser
docker-compose up

cd C:\PythonProjects\MyT5TrainingV3
pip install -U "huggingface_hub[cli]"
huggingface-cli login --token ЗДЕСЬ_ВСТАВИТЬ_ТОКЕН
huggingface-cli download SUPERMEGAok/sum_dataset --local-dir C:\PythonProjects\MyT5TrainingV3\datasets\gazeta_huggingface
python main.py
huggingface-cli upload SUPERMEGAok/MyT5Sum ./trained_model

cd C:\PythonProjects\TOPIC_AND_SUM_REST_API
docker-compose up


pause
```

Рис. 5. – Содержимое batch файла для запуска MLOps

Сначала выполняется первый этап цикла – запуск контейнера для датасета. После его завершения на локальной машине загружается датасет из

репозитрия и выполняется python скрипт обучения модели. В конце запускается контейнер с веб-приложением, использующим обученную модель. Можно периодически повторять данный цикл просто запуская batch файл.

Таким образом, в результате работы был создан цикл MLOps, который уже можно применять в производственной среде, использующей ресурсы локальной машины для обучения модели.

Литература

1. Emmanuel Raj // Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale. 2021. С. 3-4.
2. Dominik Kreuzberger, Niklas Kühl, Sebastian Hirschl, Machine Learning Operations (MLOps): Overview, Definition, and Architecture // Researchgate. 2022.
3. Головизнина В. С., Котельников Е.В. Автоматическое реферирование русскоязычных текстов: сравнение экстрактивных и абстрактивных методов. // Компьютерная лингвистика и интеллектуальные технологии. 2022. № 21. С. 223-235.
4. Ramazan Mengi, Hritik Ghorpade, Arjun Kakade, Fine-tuning T5 and RoBERTa Models for Enhanced Text Summarization and Sentiment Analysis // Researchgate. 2023.
5. Sudip Mittal, Navigating MLOps: Insights into Maturity, Lifecycle, Tools, and Careers // Researchgate. 2025.

Literature

1. Emmanuel Raj // Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale. 2021. P. 3-4.
2. Dominik Kreuzberger, Niklas Kühl, Sebastian Hirschl, Machine Learning Operations (MLOps): Overview, Definition, and Architecture // Researchgate. 2022.

3. Goloviznina V. S., Kotelnikov E. V. Automatic Summarization of Russian Texts: Comparison of Extractive and Abstractive Methods. // Computational Linguistics and Intellectual Technologies. 2022. № 21. P. 223-235.
4. Ramazan Mengi, Hritik Ghorpade, Arjun Kakade, Fine-tuning T5 and RoBERTa Models for Enhanced Text Summarization and Sentiment Analysis // Researchgate. 2023.
5. Sudip Mittal, Navigating MLOps: Insights into Maturity, Lifecycle, Tools, and Careers // Researchgate. 2025.