Решение кейса "Сервис выделения сущностей из поискового запроса клиента в мобильном приложении торговой сети «Пятерочка»"

Selling Pandas

October 2, 2025

Документация решения

Структура репозитория

- `/webservice` — папка с вебсервисом;
- `train_deberta_with_bio.` — ноутбук с кодом обучения модели и комментариями.
- `train_deberta_without_bio.ipynb` — ноутбук с кодом обучения модели и комментариями.
- `train_rosberta_with_bio.ipynb` — ноутбук с кодом обучения модели и комментариями.

Технические требования

Обучение моделей `app/deberta_with_bio.pt` и `app/rosberta_with_bio.pt` проводилось на видеокартах **4xL4 (Kaggle)**, модели `app/deberta_without_bio.pt` на **RTX4090** Инференс производится на сервере с видеокартой **RTX4090**.

Для лучшей воспроизводимости обучения рекомендуется использовать те же GPU. Минимальные ресурсы:

- 24 GB VRAM;
- 16 GB RAM;
- около 30 GB дискового пространства.

Сторонние данные

- Для редких классов (`VOLUME`, `PERCENT`) была произведена генерация синтетических данных. Было создано 191 сэмпл с учётом того, чтобы модель могла различать эти классы.
- Также проводились эксперименты с парсингом сайта “Пятёрочка”: из названий продуктов формировались тексты, которые имитировали пользовательские запросы. На данный момент эти данные не дали улучшения метрик, планируется дальнейшая доработка.

Предобученные модели

Использовались предобученные модели с `huggingface`:

- **microsoft/mdeberta-v3-base**
- **ai-forever/ru-en-RoSBERTa**

Инструкция по запуску веб-сервиса

Структура вебсервиса

- `Dockerfile` — докерфайл для сборки контейнера с вебсервисом;
- `requirements.txt` — зависимости для запуска;
- `app/main.py` — код API (асинхронная обработка запросов и инференс модели);app
- `app/deberta_with_bio.pt` — чекпоинт обученной модели debertav3 с bio разметкой
- `app/deberta_without_bio.pt` — чекпоинт обученной модели debertav3 без bio разметки
- `app/rosberta_without_bio.pth` — чекпоинт обученной модели rosberta без bio разметки

Поднятие Docker-контейнера

1. Используется базовый образ: `pytorch/pytorch:2.6.0-cuda12.6-cudnn9-runtime`.
2. Требуется установленный Docker с `GPU`-поддержкой.
3. Склонировать репозиторий в папку `/project`.
4. Перейти в папку:

```
cd project/webservice
```

5. Скачать чекпоинты моделей из папки с из папки с гугл диска. Положить сами чекпоинты в директорию project/webservice/app

6. Собрать контейнер:

```
docker build -t fastapi-torch .
```

7. Запустить контейнер:

```
docker run --gpus all -p 8000:8000 --rm fastapi-torch
```

Важно: на машине должен быть свободен порт `8000`.