ЛИДЕРЫ ЦИФРОВОЙ ТРАНСФОРМАЦИИ

ПРОЕКТ МЭРА МОСКВЫ

ДЕПАРТАМЕНТ ПРЕДПРИНИМАТЕЛЬСТВА И ИННОВАЦИОННОГО РАЗВИТИЯ ГОРОДА МОСКВЫ

РАЗВИТИЕ ЧЕЛОВЕЧЕСКОГО КАПИТАЛА

X5

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ЗАДАЧА 10

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

1. Контекст и актуальность задачи

Ежедневно миллионы клиентов используют поиск в приложении «Пятёрочка», чтобы собрать продуктовую корзину. Пользователи вводят самые разные запросы: от простых («молоко») до сложных («кола 2л без сахара»), с сокращениями, опечатками и неполными словами.

Сегодня поиск работает по ключевым словам, но не умеет понимать контекст — важен ли бренд, категория, объём или процент жирности. Из-за этого клиенту приходится просматривать десятки товаров, что снижает удобство и скорость покупок.

Цель задачи: научить систему автоматически извлекать ключевые сущности из пользовательских поисковых запросов и тем самым сделать поиск «умнее».

2. Описание задачи. Подробная задача для участников, зарегистрированных на кейс. Отображается после начала хакатона

Участникам необходимо разработать решение для задачи извлечения именованных сущностей (NER) из поисковых запросов. Решением является модель, интегрированная в веб-сервис, обеспечивающий ответ на входящие запросы не дольше 1 секунды.

Распознаваемые сущности:

1. TYPE — категория товара (молоко, хлеб, вода, чипсы и т.п.);
2. BRAND — бренд (Coca-Cola, Простоквашино, Lays и др.);
3. VOLUME — объём/вес/количество (0.5 л, 1 л, 200 г, 10 шт.);
4. PERCENT — процент (2.5%, 15%).

Формат BIO-разметки:

- B-ENTITY — начало сущности,
- I-ENTITY — продолжение сущности,
- O — не сущность.

Оценка

В этом хакатоне основной метрикой является macro-averaged F1-score по BIO-разметке для всех типов сущностей (TYPE, BRAND, VOLUME, PERCENT).

Определения:

- True Positive (TP) — сущность, которая выделена моделью, причем выделена верно (начало и конец сущности, а также тип сущности совпадает с эталонной разметкой).
- False Positive (FP) — сущность, предсказанная моделью, но отсутствующая в эталонной разметке (O) либо не совпадающая с сущностью эталонной разметки.
- False Negative (FN) — сущность, которая есть в эталонной разметке, но не была распознана моделью.

Для каждой сущности считаются:

- Precision=TP/ (FP + TP)
- Recall=TP/(FN+TP)
- F1=2*(Precision*Recall) / (Precision+Recall)

Macro-F1 сначала вычисляет F1-score для каждого типа сущностей (TYPE, BRAND, VOLUME, PERCENT), а затем усредняет результаты.

Пример:

Запросы и эталонная разметка:

авокадо - [(0, 7, 'B-TYPE')]

батат - [(0, 5, 'B-TYPE')]

абрикосы 500г global village - [(0, 8, 'B-TYPE'), (9, 13, 'B-VOLUME'), (14, 20, 'B-BRAND'), (21, 28, 'I-BRAND')]

Предсказания модели:

авокадо - [(0, 7, 'B-BRAND')]

батат - [(0, 5, 'O')]

абрикосы 500г global village - [(0, 8, 'B-TYPE'), (9, 13, 'B-VOLUME'), (14, 20, 'B-BRAND'), (21, 28, 'I-BRAND')]

Расчет F1:

Расчет TP, FP, FN.

Первый запрос:

- TYPE: TP=0, FP=0, FN=1 (модель не выделила TYPE)
- BRAND: TP=0, FP=1 (модель ошибочно выделила BRAND), FN=0

Второй запрос:

- TYPE: TP=0, FP=0, FN=1 (батат)

Третий запрос:

- TYPE: TP=1 (абрикосы), FP=0, FN=0
- VOLUME: TP=1 (500г), FP=0, FN=0
- BRAND: TP=1 (global village), FP=0, FN=0

Расчет метрик по видам сущностей.

TYPE:

- Precision = TP / (TP + FP) = 1 / (1 + 0) = 1.0
- Recall = TP / (TP + FN) = 1 / (1 + 2) = 1/3 ~ 0.333
- F1 = 2 * (1.0 * 0.333) / (1.0 + 0.333) ~ 0.5

VOLUME:

- Precision = 1 / (1 + 0) = 1.0
- Recall = 1 / (1 + 0) = 1.0
- F1 = 1.0

BRAND:

- Precision = 1 / (1 + 1) = 0.5
- Recall = 1 / (1 + 0) = 1.0
- F1 = 2 * (0.5 * 1.0) / (0.5 + 1.0) ~ 0.6667

Macro F-1 = (0.5 + 1.0 + 0.6667) / 3 ≈ 0.722

Замечание: в случае, если время обработки запроса составляет более 1 секунды, результат по метрикам для данного запроса считается равным нулю.

Файлы

train.csv - данные для обучения:

id - уникальный id запроса,

search_query - запрос пользователя,

annotation - разметка в формате BIO.

test.csv - данные для тестирования, задача команды извлечь сущности:

id - уникальный id запроса,

search_query - запрос пользователя.

Валидация и Лидерборд

Валидация решения возможна в 2 форматах:

1. **Через csv в Telegram-боте.** *Используйте этот способ в начале соревнования, когда сервис еще не готов.* Команда может отправить .csv файл с разделителем ";" в Telegram-бота **@x5_ner_hack_bot**, для каждого search_query тестовой выборки необходимо предсказать сущности в формате BIO. Отправка .csv файла используется только для валидации модели, финальное решение команды должно включать в себя веб-сервис. latency при такой проверке не замеряется. Результаты валидации по .csv файлу не попадают в лидерборд. Файл должен быть назван submission.csv, содержать заголовок и иметь следующий формат:

```
id;search_query;annotation
1;наггетсы;[(0, 8, 'B-TYPE')]
2;кисель;[(0, 6, 'B-TYPE')]
3;зубочистки 100шт;[(0, 10, 'B-TYPE'), (11, 16, 'B-VOLUME')]
```

2. **Через веб-сервис**. *Используйте его для оценки качества модели с учетом быстродействия сервиса.* Команда должна реализовать веб-сервис с публичным endpoint-ом. Вы можете поднять собственный сервис на любой арендованной машине с публичным IP. Веб-сервис должен поддерживать асинхронную обработку запросов, обеспечивая неблокирующую работу endpoint /api/predict для масштабируемой и быстрой обработки большого количества параллельных запросов с временем отклика не более X секунд. Необходимо зарегистрировать команду в Telegram-боте **@x5_ner_hack_bot** и отправить URL сервиса, после чего начнутся отправка запросов и замер качества. Результаты отобразятся в лидерборде.

Важно: для регистрации и работы с ботом команда должна создать чат и добавить туда бота. Не создавайте отдельные переписки для каждого участника - весь обмен сообщениями с ботом ведется централизованно от имени команды через один чат.

Требования к api:

```
POST /api/predict
Body={"input": "сгущенное молоко"}
Response=[
{"start_index": 0, "end_index": 8, "entity": "B-TYPE"},
{"start_index": 9, "end_index": 15, "entity": "I-TYPE"}
]
```

При input == "" возвращать пустой список

Результаты тестирования отображаются в лидерборде в Telegram-боте.

Ограничения:

1. Время отклика веб-сервиса для одного запроса не превышает 1 секунду.
2. Веб-сервис должен непрерывно работать с момента отправки endpoint в Telegram-бота до конца проверки.
3. Сервис должен быть доступен по публичному URL без авторизации.

Участники могут:

1. Использовать публично доступные внешние данные.
2. Использовать любые архитектуры моделей.
3. Использовать готовые предобученные языковые модели.

Важно! Запрещается использовать ручные словари, собранные специально под задачу; готовые коммерческие NER-API; любые закрытые или приватные датасеты, которые не доступны всем участникам.

3. Программно-аппаратные требования:

3.1 Аппаратные требования — сервис может быть развернут на любом арендованном сервере.

3.2 Программные требования — для реализации веб-сервиса участники могут использовать разные варианты в зависимости от удобства команды. Рекомендуется использовать FastAPI. Ограничения на выбор архитектуры моделей не ставятся. Для работы с

данными и моделями рекомендуется использовать Python и библиотеки для обработки текста и машинного обучения, такие как Transformers, HuggingFace, PyTorch, TensorFlow, spaCy, NLTK и любые другие.

4. Требования к презентации/демонстрации

В презентации участники должны показать технический стек решения с указанием языков программирования, библиотек и инструментов, которые были использованы для реализации модели и веб-сервиса. Обязательно необходимо представить архитектуру решения, включая схему работы сервиса от ввода данных пользователем до формирования результата, а также архитектуру самой модели распознавания сущностей. Необходимо показать, как именно команда обучала модель, каким образом обрабатывали датасет, какие дополнительные источники использовали для обогащения и как учитывали опечатки и неполные слова. Необходимо рассказать, с какими сложностями команда столкнулась в процессе работы. Презентация должна быть лаконичной и содержать не более 15 слайдов.

5. Требования к сопроводительной документации

К решению должна прилагаться сопроводительная документация (можно оформить в README + отдельный файл), которая позволит экспертам без труда разобраться в том, как устроен сервис. В документе нужно описать технические требования для запуска и установки зависимостей, а также пошаговую инструкцию по развертыванию веб-сервиса на локальной машине или сервере. Отдельным разделом нужно описать, какие внешние источники данных или предобученные модели использовались в работе.

6. Ресурсы:

Инструкция по развертыванию простейшего FastAPI-сервиса доступна по ссылке

7. Требования к сдаче решения (Скор на прайват датасете)

7.1 Требования для промежуточной сдачи решения

Для финальной сдачи решения сервис должен:

- поддерживать все целевые сущности (TYPE, BRAND, VOLUME, PERCENT) с BIO-разметкой;
- корректно обрабатывать опечатки, сокращения и неполные слова;
- реализовывать endpoint POST /api/predict и иметь следующую структуру тела запроса и ответа:

Body={"input": "сгущенное молоко"}

Response=[{"start_index": 0, "end_index": 8, "entity": "B-TYPE"}, {"start_index": 9, "end_index": 15, "entity": "I-TYPE"}]

- обеспечивать время отклика не более 1 секунды;
- непрерывно функционировать с момента передачи endpoint в Telegram-бота до конца проверки.
- Решение должно быть упаковано в Docker-контейнер, готовый к развёртыванию.

К финальной версии необходимо приложить документацию.

ВАЖНО: В последний день хакатона веб-сервис участников должен быть поднят для финальной оценки и в Telegram-боте должен храниться актуальный URL веб-сервиса.

7.2 Требования для финальной сдачи решения (питчинг решений)

Для участия в финальном питчинге команда должна войти в ТОП-10 по лидерборду и подготовить презентацию. Выступление должно быть четким и укладываться в 10 минут, включая время на вопросы жюри.

8. Критерии оценки

8.1. Подход коллектива к решению задачи (3 балла)

Какие технологии и модели вы выбрали и почему

8.2. Техническая проработка решения (3 балла)

Здесь оценивается, насколько ваше решение законченное и аккуратное. Насколько всё сделано «чисто» — «без костылей».

8.3. Соответствие решения поставленной задаче (3 балла)

Проверяем, выполнены ли **все требования из ТЗ**:

- В решении реализованы и модель, и веб-сервис, доступный по публичному IP. URL сервиса добавлен в Telegram-бота
- Сервис стабильно обрабатывает параллельные запросы.
- Ответы в корректном формате (именованные сущности размечены правильно).
- Кодовая реализация решения полностью выложена на GitHub, ссылка предоставлена в Telegram-боте.

8.4. Эффективность решения в рамках поставленной задачи (50 баллов)

Главная метрика — macro-F1 на скрытом тесте. При этом все запросы, не уложившиеся в тайминг, считаются пустыми ответами. Таким образом метрика одновременно отражает и точность модели, и её быстродействие. Распределение баллов идёт по рангу на лидерборде: команда на 1-м месте получает 50 баллов, на 2-м — 49, на 3-м — 48 и так далее.

8.5. Выступление коллектива на питч-сессии (только для финальной экспертизы) (10 баллов)

Оцениваем не только презентацию и подачу, но и сам подход: какие идеи и данные вы использовали, насколько решение получилось продуманным и интересным.