



Онлайн-контест от Brand Analytics при поддержке
Акселератора Возможностей при ИНТЦ МГУ «Воробьевы горы»

Brand Analytics ML-contest

Суммаризация комментариев
в социальных медиа

ЗАДАЧА КОНТЕСТА

Основное

Описание задачи:

Необходимо реализовать решение которое сможет генерировать (генеративная суммаризация) текст суммариации (главного смысла, смысла обсуждения) комментариев под каждым постом в нескольких режимах (типах) описанных далее.

Входные данные:

Файл в формате .jsonl (1 json-объект на 1 строку) с постами (в т.ч. видео с YouTube) и комментариями из VK, Telegram и YouTube.

Данные в файле представлены в хаотичном порядке, участникам в первую очередь необходимо связать комментарии и посты по внешним идентификаторам, которые указаны в качестве отдельного поля каждого объекта исходного файла, а также провести базовые операции (очистка и т.п.) пред-обработок.

Пример формата данных:

```
// Пост
{
    "text": "string",
    "url": "http://vk.com/wall-...",
    "id": "-420346_14545109",
    "hash": "007da8b20db7b7ea56cb11aa0f37a3b9",
    "date": 1699173000
}

// Комментарий
{
    "text": "string",
    "url": "http://vk.com/wall-...",
    "id": "-420346_14545816",
    "hash": "780c91a51a54bd4cba978ba6c6ac94c3",
    "root_id": "-420346_14545109",
    "parent_id": "-420346_14545233",
    "date": 1699254174
}
```

Требования к типам суммаризации:

1. **all_comments**: суммаризация всех комментариев под каждым постом, без анализа самого поста;
2. **post_comments**: суммаризация только тех комментариев, которые имеют явное отношение к тексту каждого поста;
3. **topic_comments**: суммаризация комментариев которые имеют косвенное отношение к посту (пример: пост про технологию компании, а комментарий про обсуждение самой компании)

Общие требования к решению:

- Использование только открытых технологий. Это касается как моделей, корпусов и прочих ресурсов так и используемых библиотек;
- Запрещено использование в конечном решении (но допускается в процессе разработки) облачных технологий: OpenAI и т.п;
- Конечное решение должно иметь инструкцию по запуску и установке всех зависимостей. Все внешние файлы, словари, модели и т.п. должны предоставляться вместе с самим решением;
- Формат передачи решения и его структура должны соответствовать требованиям описанным ниже;
- Ограничений по стеку технологий нет, но предпочтителен стандартный набор современного DS/ML: Python.
- Высокая скорость работы: до 2 секунд на 10 комментариев по отношению к одному посту.

Требования к структуре решения:

Конечное решение должно иметь единую точку входа (исполняемый файл) под названием `solution`, принимающее в качестве аргументов:

- Режим (тип) суммаризации
- Путь к файлу с данными
- Путь и название итогового файла результата работы

Пример:

`./solution all_comments ./dataset.jsonl ./result.jsonl`

Решение должно быть упаковано в `.zip` архив и содержать следующую структуру:

`solution.zip`

```
└── src/ - исходные файлы решения, готовые к сборке  
└── solution - исполняемый файл решения  
└── readme.md - описания решения и его запуска  
└── dependencies.txt - опционально, для установки  
    зависимостей
```

В файле `readme.md` необходимо указать базовое описание решения, технологий и процесса запуска. Авторские мысли и комментарии – допускаются.

Это требования к обязательной структуре, остальные файлы и папки при необходимости – допускаются.

Методика тестирования:

Тестирование будет проходить в автоматическим режиме, в изолированной среде на сервере с Ubuntu 20.04 LTS, x86_64, 12 cores, 32gb ram, NVIDIA GeForce RTX 2080 Ti.

Для каждого решения сначала будут установлены зависимости, перечисленные в файле dependencies.txt, затем начнётся автоматическая проверка, которая будет осуществляться следующими командами, выполняемыми последовательно:

```
# Тип 1:./solution all_comments ./dataset.jsonl ./result.jsonl  
# Тип 2:./solution post_comments ./dataset.jsonl ./result.jsonl  
# Тип 3:./solution topic_comments ./dataset.jsonl ./result.jsonl
```

По итогу выполнения любого из типов суммаризации приложение должно сформировать .jsonl файл со следующей структурой:

```
{  
  "summary": "string", // текст суммаризации  
  "post_hash": [], // поле hash исходного поста  
  "comments_hash": [] // hash комментариев, подошедших  
  к суммаризации  
}
```

Для каждого типа суммаризации подготовлено несколько проверочных эталонных корпусов, которые будут сравниваться по метрикам bert score и rouge с итоговой суммаризацией решения.

Критерии оценки:

- Точность по метрикам bert score и rouge
- Полнота по метрикам bert score и rouge
- F1 по метрикам bert score и rouge
- Производительность (скорость работы)
- Потребление ЦПУ
- Потребление ГПУ
- Потребление ОЗУ

Важные пункты:

- Качество
- Скорость
- Ресурсоэффективность
- Предпочтительны решения, способные эффективно работать на CPU

Контакты

Telegram-канал конкурса



Организаторы

Юлия

+7 (900) 642-12-77



[@iulikh1](#)

Светлана

+7 (999) 244-01-51



[@svetlanasukacheva](#)

Куратор конкурса

Мария

+7 (981) 711-71-04



[@mashriya](#)

**ЖДЕМ ВАС
НА КОНТЕСТЕ!**