Выделение деталей из текста на английском языке, описывающего изображение, методами МО

Куратор: Полянская Анна

Команда:

Машковцева Полина
Гераськина Надежда
Горбатова Татьяна
Есипов Иван

Данные

Язык данных: **английский**

Кол-во наблюдений: **49 312**

Из датасета **COCO** были взяты аннотации к изображениям, каждое изображение содержит 5 вариаций аннотаций

Visual Genome содержит информацию об объектах на изображениях, их атрибутах и отношениях между объектами. Были использованы следующие части Visual Genome:

- объекты
- атрибуты
- описания
- отношения

Глобальная задача:

Модель должна из текста выделять детали

Пример json на выходе:

```
{
"background": "forest",
"objects": ["girl", "flower", "magic"],
"positions": ["center", "in human hand", "everywhere"]
"obj_descriptions": {
  "1": ["green eyes", "blond hair", "smile"],
  "2": ["purple"],
  "3": []
  }
  }
```

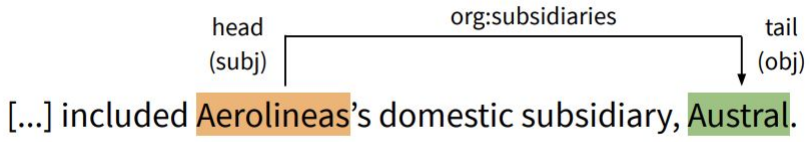
ML-задача

Задачи на данный чекпоинт:

- Сбор данных
- Анализ данных
- Предсказание relation (предикат) между объектом и субъектом (мультиклассовая классификация)

  Пример:

  - object - man
  - subject - car
  - predicate - in



EDA

Структура данных:

- доля стоп слов в корпусе **55%**
- средняя длина предложения со стоп-словами: **10-11**
- средняя длина предложения без стоп слов: **4**
- чаще всего встречаются **существительные** и **артикли**

Выявленная проблема: в данных есть опечатки

Решение: убрать опечатки при помощи расстояния Левенштейна

Выводы:

1. самые часто встречающиеся описания: занятия спортом, городские пейзажи, интерьер, животные
2. описания достаточно разнообразные, не нужно что-либо добавлять

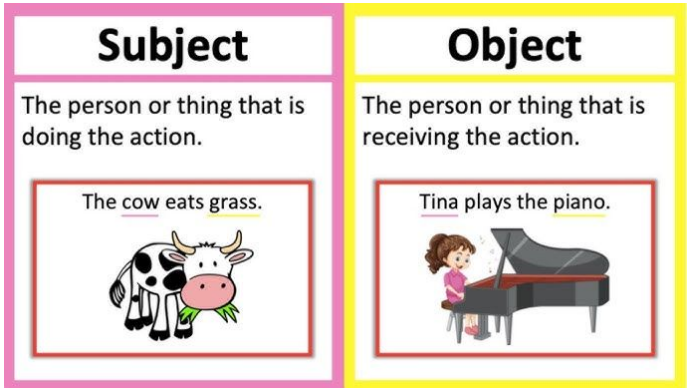
Данные для обучения

Был сформирован **датасет**:

- **subject** - субъект
- **object** - объект
- **relation** - предикат, который связывает субъект и объект

Всего было отобрано ТОП-10 по встречаемости в исходном датасете relation

Выборка была **сбалансирована** по самому малочисленному классу (relation) - **2000** экземпляров на каждый класс



ML модель

Были выбраны метрики **accuracy и macro F1**, так как выборка сбалансированная

Выбранная модель все равно **недообучена**, так как:

- двух разнообразных “**неоднородных**” признаков недостаточно для нормального обучения модели
- с такими задачами (NLP) лучше справляются DL модели

P.S. Полную таблицу метрик см. в Приложении 1

Были исследованы:

Предобработка данных:

- CountVectorizer
- Embeddings

Модели:

- Logistic Regression
- SVM
- RandomForest
- CatBoost

\+ Grid Search

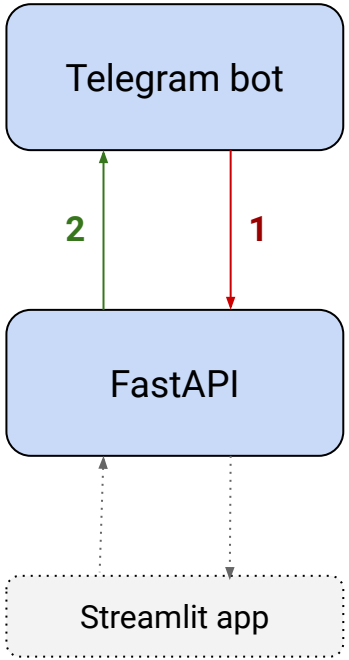
Предобработчик: **CountVectorizer**

Модель: **RandomForest**

Метрики:

	train:	test:
accuracy	0.65	0.29
marco F1	0.65	0.28

Сервисы



1) **Telegram бот делает запросы на FastAPI:**
   - /predict - отправляет данные о субъекте и объекте в формате json и получается предсказания в формате json
2) **FastAPI возвращает запросы на Telegram бот:**
   - **POST** /predict возвращает предсказания в формате json

TO DO:

- **POST** для предеплоя файлов
- **POST** для подгрузки вводных данных и записи в PostgresQL БД
- **GET** для инференса (Streamlit и TG bot)

Планы

1. **DL часть** изучить и реализовать**:**
   a. Embeddings
   b. LSTM
   c. BERT
2. **Разработка:**
   a. задеплоить сервисы с подгрузкой из dvc (S3/GDrive)
   b. реализовать Streamlit приложение
   c. PostgresQL DB

Спасибо за внимание!

Приложение 1. Полная таблица метрик

Предобработчик	ML модель	Train accuracy	Test accuracy	Train macro F1	Test macro F1
CountVectorizer	Logistic Regression	0.40	0.27	0.40	0.27
	SVM	0.14	0.11	0.11	0.10
	Random Forest	0.65	0.29	0.65	0.28
	CatBoost	0.30	0.26	0.30	0.25
Embeddings	Logistic Regression	0.29	0.25	0.29	0.24
	SVM	0.11	0.12	0.10	0.11
	Random Forest	0.59	0.28	0.59	0.28
	CatBoost	0.49	0.29	0.49	0.28