Отчет

Штыков Павел[1]

shtykov.pa@gmail.com

15 июня 2022

1 Анализ исходных датасетов

Скрипт с кодом анализа исходных датасетов: **part1_analysis.py**.

Анализ заключался в составлении списка всевозможных типов сущностей, их примеров и проверки на вложенность. Результаты работы лежат в папке **res/par1**.

Было выявлено, что в датасете Restaurant8k существует небольшое количество предложений, сущности в которых накладываются. При обучении они были отброшены. Также в датасете Restaurant8k есть "битые"предложения (с поломанным форматом json'а например). При обучении это было учтено.

2 Объединение датасетов

Скрипт с кодом анализа исходных датасетов: **part2_merge.py**.

В двух данных датасетах логично совпадают только два типа сущностей: *date* и *time*. Соответственно в датасете ATIS все типы оносящиеся к дате и времени были обобщены до указанных выше *date* и *time* (например: *return_ date.day_ number* $\rightarrow$ *date*). В датасете Restaurant8k сущности *date* и *time* присутствуют изначально.

Кроме этих двух "простых"сущностей в объединенный датасет были добавлены сущности *first_ name*, *last_ name* и *people* из Restaurant8k (соотвественно из данного датасета в объединенный датасет были перенесены все сущности). Такой выбор был сделан в силу того, что сущности описывающие человека очень общие и могут встретиться почти в любом домене.

Из датасета ATIS дополнительно была взята сущность *city*, т.к. она также является крайне общей. Кроме того название нескольких английских городов встречается и в датасете Restaurant8k.

Все сущности, связанные с самолетами, были отброшены. Сущности связанные с административным делением США (имя штата, почтовый код и т.д.) также были отброшены, т.к. датасет Restaurant8k о Великобритании.

В файле **tils/entities_map.json** можно найти полный словарь соответствий между старыми и новыми типами сущностей.

В файле **res/part2/union_stat.json** можно найти все реализации всех типов сущностей в объединенном датасете.

3 Обучение моделей

Для обучения были выбраны следующие две модели: Flair Embeddings и W2NER.

В качестве критериев отбора моделей был хороший репозиторий на GitHub (количество звезд больше 100) и высокое качество в бенчмарках PaperWithCode. На CoNLL2003 датасете модели достигают качества 93.09 и 93.07 соответственно.

В качестве метрик были взяты стандартные (и уже реализованные в соответствующих моделях): precision, recall и F1-score (macro averaging).

В таблице 1 представлены лучшие результаты моделей на тестовой выборке. Видно, что обе модели достигли хороших результатов, однако W2NER показала себя заметно лучше.

Модель	Recall	Precision	F1-score
Flair	0.9515	0.9477	0.9495
W2NER	0.9698	0.9576	0.9637

Таблица 1: Лучшие результаты на тестовой выборке

Рассмотрим примеры работы моделей на тестовых данных.

Flair:

1. on [april first]$_{date}$ i need a flight going from [phoenix]$_{city}$ to [san diego]$_{city}$
2. a table for [4]$_{date}$ for the [28th of April]$_{date}$ please
3. the booking is for [9 people]$_{people}$
4. Thanks for your help, that is all i needed

В целом Flair предсказывает метки хорошо. Во 2-ом предложении есть ошибка с токеном «4», но такая ошибка понятна.

W2NER:

1. on [april first]$_{date}$ i need a ticket from [tacoma]$_{city}$ to [san jose]$_{city}$ departing [before 7 am]$_{time}$
2. [8:00 pm]$_{time}$ will work for the [12]$_{people}$ invited.
3. I would to make a reservation [in three weeks]$_{date}$, for [9 people]$_{people}$
4. I want to book a table for [me and my wife]$_{people}$.
5. Gluten free options are not important, but I want a cheap price range. I don't care much about price range, unless it is cheap or expensive.

W2NER справилась лучше с отдельно стоящим числом, классифицировав его как *people* (2-ое предложение). Качество остальных предсказаний также высокое.

4 Не успел

- При объединение нескольких сущностей в одну в датасете ATIS иногда между сущностями встречаются предлоги или артикли. Они разбивают сущность на две части. Возможно, правильнее было бы объединить их в одну сплошную сущность.
- Не реализован скрипт для использования уже обученной модели W2NER.
- Не построены кривые обучения (по логам было заметно, что W2NER сходиться быстрее, однако на одну эпоху уходить больше времени)