Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Построение поисковой системы для интернет магазина

Выполнил: ст. гр. 417 Федоров И.С.

Научный руководитель: д. ф.-м. н., профессор Дьяконов А.Г.

Москва
2021

Постановка задачи

Требуется построить систему, которая бы по поисковому запросу пользователя выдавала релевантные товары в интернет магазине.

В данной работе рассматривается подход с использованием технологий NLP и глубокого обучения.

Постановка задачи

Будем считать, что для товаров в базе магазина отдельно выделен бренд и название товара.

№	Описание товара	Название товара	Бренд
1	Смартфон Apple iPhone 5S 16Gb Space Gray (ME432RU/A)	Смартфон	Apple
2	Варенье Вкусвилл брусника - апельсин, 310 г	Варенье	Вкусвилл
3	Женские кроссовки CALVIN KLEIN JEANS	Кроссовки	CALVIN KLEIN JEANS

Тогда поставленную задачу можно решать с помощью распознавания именованных сущностей (NER) в запросах пользователей.

№	Запрос	Название товара	Бренд	Уточняющая информация
1	чехол для iphone 6	чехол	(нет)	для iphone 6
2	смартфоны xiaomi	смартфоны	xiaomi	(нет)
3	чайник электрический маленький	чайник	(нет)	электрический маленький

Особенности сбора данных

Названия/описания товаров можно собирать с интернет-магазинов (на практике это нетривиальная задача)

В открытых источниках отсутствуют наборы данных с запросами пользователей, поскольку область сильно коммерциализована.

Большое спасибо компании KazanExpress за предоставленный датасет запросов!

Особенности сбора данных

Предложен метод сбора данных-запросов, если нет возможности обратиться к какой-либо компании за данными.

Анализ трафика переходов на сайты интернет-магазинов с поисковиков (Google, Яндекс и т.д.). Существуют сервисы, которые подобные данные предоставляются (ahrefs.com).

Всего собранных данных

- 450.000 товаров, собранных с интернет-магазинов Ozon/Юлмарт/BERU/Яндекс.Маркет
- 3.1 млн. товаров, описанных в чеке - с соревнования DataFusion
- 3.7 млн. неразмеченных запросов - предоставлены KazanExpress
- 10.000 размеченных запросов - немного, т.к. разметка за свой счет

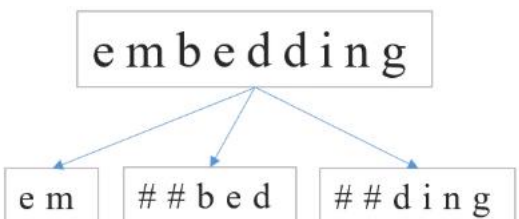
Модель

BERT для распознавания именованных сущностей. В наиболее похожей по смыслу статье от компании TheHomeDepot использовались рекуррентные модели BiLSTM-CRF и BiGRU. В этом основное отличие моей работы и их статьи.

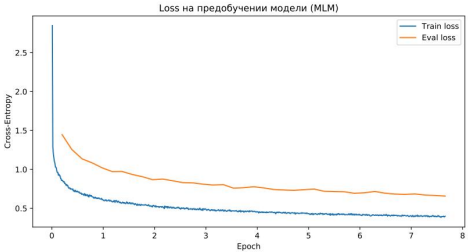
BERT предобучался на всем корпусе текстов на задачу предсказания замаскированных слов, а потом дообучался на задачу NER на небольшом (10.000 сэмплов) размеченном датасете запросов.

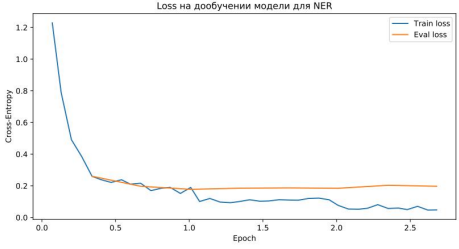
Токенизатор

В качестве токенизатора - свой обученный WordPiece. Обучен на всем корпусе текстов (на 30 тыс. токенов). Разбиение на подслова способствует повышению качества модели (т.к. пользователи допускают опечатки в запросах, и, если ошибка произошла, то ошибется только в одном субтокене).



Лосс на обучении/валидации





Пример работы (предсказание маскированных слов)

```
get_examples('чехол для [MASK] galaxy')
```

```
[('чехол для samsung galaxy', 0.5545173287391663),
 ('чехол для телефона galaxy', 0.06168787926435470 6)]
```

```
get_examples('купить [MASK] для iphone 6 ')
```

```
[('купить чехол для iphone 6', 0.9618306756019592),
 ('купить стекло для iphone 6', 0.0038562484551221132),
 ('купить наклеики для iphone 6', 0.0036110864020884037)]
```

```
get_examples('купить [MASK] bosch 2 литра')
```

```
[('купить чаиник bosch 2 литра', 0.07487241923809052)]
```

```
get_examples('молоко [MASK] ультрапастеризованное')
```

```
[('молоко питьевое ультрапастеризованное', 0.12975719571113586),
 ('молоко фрутоняня ультрапастеризованное', 0.04602883383631706),
 ('молоко свитлогорье ультрапастеризованное', 0.043549809604883194),
 ('молоко вкуснотеево ультрапастеризованное', 0.029766695573925972)]
```

Пример работы (предсказание маскированных слов)

Как бонус - можно использовать для автоматического предложения новых слов запросе

```
get_examples('iphone [MASK]')
```

```
[('iphone 7', 0.02028769813477993),
 ('iphone 6', 0.019341209903359413),
 ('iphone 4', 0.018780456855893135)]
```

```
get_examples('купить яблоки [MASK]')
```

```
[('купить яблоки вес', 0.08910977840423584),
 ('купить яблоки кг', 0.054261304438114166),
 ('купить яблоки зеленые', 0.033656924962997437)]
```

```
get_examples('смартфон samsung[MASK]')
```

```
[('смартфон samsung galaxy', 0.07469146698713303),
 ('смартфон samsung note', 0.05042911320924759),
 ('смартфон samsung s', 0.040562234818935394)]
```

Пример работы (распознавание именованных сущностей)

```
make_prediction('купить чайник BOSCH'.split())
```

```
[('купить', 'O'), ('чайник', 'Товар'), ('BOSCH', 'Бренд')]
```

```
make_prediction('apple iphone'.split())
```

```
[('apple', 'Бренд'), ('iphone', 'Товар')]
```

```
make_prediction('купить смартфон'.split())
```

```
[('купить', 'O'), ('смартфон', 'Товар')]
```

В данном случае samsung - это телефон, для которого покупается чехол, то есть не бренд чехла

```
make_prediction('чехол для samsung galaxy'.split())
```

```
[('чехол', 'Товар'), ('для', 'O'), ('samsung', 'O'), ('galaxy', 'O')]
```

А тут уже samsung - бренд

```
make_prediction('samsung смартфон'.split())
```

```
[('samsung', 'Бренд'), ('смартфон', 'Товар')]
```

```
make_prediction('посуда'.split())
```

```
[('посуда', 'Товар')]
```

```
make_prediction('mercedes'.split())
```

```
[('mercedes', 'Бренд')]
```

```
make_prediction('носки мужские adidas'.split())
```

```
[('носки', 'Товар'), ('мужские', 'O'), ('adidas', 'Бренд')]
```

Приложение

Весь код и обученные модели доступны на GitHub автора:
https://github.com/Sorrow321/cmc_seminar

Можно самостоятельно проверить эксперименты и результаты. Данные для обучения не выкладываются, поскольку была просьба от KazanExpress их не разглашать.

Заключение

- Были указаны ключевые сложности со сбором данных
- Был предложен метод сбора запросов пользователей в интернет-магазины через посредников — общие поисковые системы (Яндекс, Google и т.д.)
- Был собран обучающий набор данных, часть из них была вручную размечена
- Была предложена нейросетевая архитектура, основанная на BERT, для решения поставленной задачи
- Был предложен метод токенизации входных запросов, который способствует более качественной работе модели в случае, когда пользователь допустил опечатку в запросе
- Были построены прототипы моделей, решающих поставленные задачи
- Был предложен метод предсказания следующего слова в запросе