Московский государственный университет
имени М.В. Ломоносова

Проектная работа

Deep Learning for Chatbots, Retrieval-Based Model in Dual Encoder GRU network in Pytorch

Образовательный проект Техносфера компании Mail.ru

Работу выполнили:

студенты второго семестра

Ракитин Виталий
Бибик Денис
Волчанский Андрей

Москва 2017

Содержание

1 Введение

В ходе работы были собраны и исследованы данные с сайта `otvet.mail.ru`, сконструирована база данных, изучена архитектура рекурентной нейронной сети, основанной на технологии GRU, состоящая из двух кодировщиков и билинейного слоя с сигмоидой.

2 Данные

Несложно заметить, что непосредственно на самом сайте `otvet.mail.ru` все вопросы пронумерованы по id, соответсвтенно и откачивать данные межно простым обходом по этим номерам. В процессе скачивания было принято решение ограничиться первыми 10 млн. вопросами. С каждой страницы выбирался вопрос и четыре самых популярных ответа на него. Если вдруг оказывалось так, что по данному id вопрос не существовал или уже был удалён, то данная страничка пропускалась.

Анализ полученных данных приведён в тут. Он показал, что в среднем 10% полученных данных являются мусором. Таким образом, выкинув мусор и несуществующие странички, всего в обучении участвовало порядка 6 млн. пар вопрос-ответ.

```
Плохих сообщений : 717
Примеры:
q =  problema s tormozami
a =  Поменяйте колодки, и залейте жидкость, или прокачайте тормоза, и всё.

_____________________
q =  EH BU DUNJA  SEN NIJE BELESEN?
a =  Фамилия наша, а гонит какой-то бред...

_____________________
q =  Chelovek! Kto ty? Otkuda prishel? Kuda idesh'?
a =  Я пршел с почты@mail.ru и иду на overclockers.ru

_____________________
q =  Quosque tandem abutere, Catilina, patientia nostra?
a =  Quam diu etiam furor iste nos eludet, quem ad finem sese effrenata iactabit audacia?  Cicerón, In Catilinam, I,
 I, 1

_____________________
q =  pochemu muzhchiny zhenyatsya ne po lyubvi?
a =  Потому, что у женщин часто бывает прошлое - а каждый мужчина хочет быть первым у женщины.  А от Вас ,наверное, ушел молодой
 человек... И женился на другой? Сочувствую...
```

Рис. 1: Пример пар вопрос-ответ, которые не рассматривались в ходе работы

Далее мы обрабатывали полученные данные следующим образом. Сначала был произведён ручной анализ полученного блока ответов на небольшой выборке. Затем, использованные слова были помещены в словарь, а далее выборка была токенезирована и преобразована в форму one-hot-encoding.

В этом методе вектору сопоставляется вектор, размерность которого равна числу классов (в нашем случае, токенов), в котором единица стоит на месте нужного класса и нули во всех остальных местах.

```
Всего использовано слов:  364365
Размер словаря:  65996
Без редких слов (реже 5): 8175

TOP 10:
это 5607
если 4584
есть 2774
можно 2698
или 2663
только 1939
просто 1664
надо 1640
очень 1607
тебе 1594
```

Рис. 2: Топ-10 токенов при one-hit-encoding

На этом этапе были замечены проблемы стоп-слов из использованной библиотеки NLTK, поскольку при выводе самых встречаемых слов с исключением стоп-слов последние всё равно оказывались в выборке и образовывали абсолютное большинство, а так же получалась огромная размерность вектора «состояний». Это стало причиной выбора иной стратегии подготовки данных для обучения. После дополнительной чистки стеммингом и удаления дополнительно транслита, смайлов и подобных «слов», строки были замещены массивами токенов. Далее была выполнена векторизация при помощи метода word2vec, в результате длина вектора слова всегда равнялась 500. После этого было выполнено сужение нашего словоря при помощи поиска синонимов (с привлечением существующих словорей), после чего данные были токенизированы и признаны готовыми к помещения в базу для обучения.

3 SQL

Была создана база данных sqlite. В которую в одну таблицу было внесено множество пар вопрос-ответ, а так же каждой паре был присвоен уникальный id. Во второй таблице хранился обратный индекс, состоящий из слов и id соответсвующих предложений. Для ускорения работы обе таблицы были

```
Всего использовано слов:  295038
Размер словаря:  32133
Без редких слов (реже 5): 6327

TOP 10:
прост 1940
очен 1609
люб 1457
котор 1432
вопрос 1358
нужн 1280
человек 1234
дела 1206
жизн 1109
поч 1091
```

Рис. 3: Топ-10 токенов с использованием стемминга

объеденины функцией join, а так же был построен index по словам. Применялись библиотеки pandas, sqlite3.

В результате время поиска по базе уменьшилось с минуты до пары секунд. Для поиска наиболее подходящих ответов под наш запрос из базы данных применяется следующий метод:

1. Предложение запроса разбивается по словам;
2. Слова нормализуются;
3. Удаляются стоп-слова;
4. Ищется группа синонимов для каждого слова;
5. Ищется множество всех вопросов по наличию слов из каждой группы;
6. Отбирается множество предложений с наибольшим количеством пересечений для групп;
7. Берутся ответы для вопросов из выбранного множества.

4 Архитектура нейронной сети

Было выбрано упрощение архитектуры LSTM (GRU) сети с двойным дескриминатором. Эта архитектура, с одной стороны, физически понятно устроена:

есть нейронная рекуррентная сеть, которые обучаются отдельно на вопросах и ответах. На выходе для вопроса и ответа получается вектор, передающий «смысл» фразы, размерность которого было предложено взять равной 256. Есть дескриминатор, который показывает, является ли данный ответ подходящим для данного вопроса. В качестве функции потерь использовалась MSE. Вектор вопроса умножается на матрицу M размера 256x256, на выходе получается вектор «ответа», который сравнивается с результатами работы RNN на ответах. В процессе обучения для каждого ответа случайным образом выбирались дополнительно два неверных ответа.

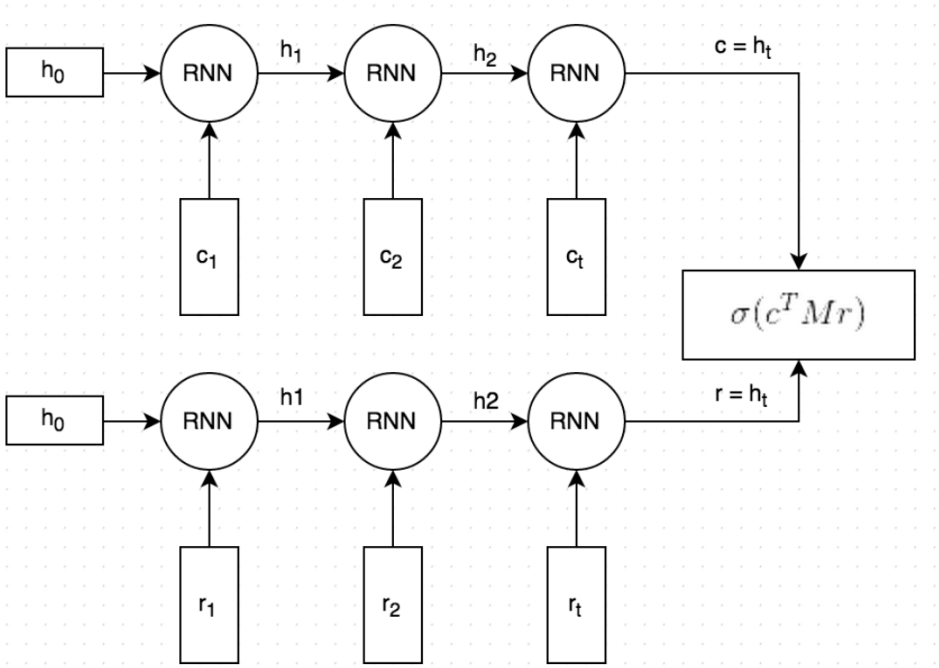

Рис. 4: Схема архитектуры нейронной сети

5 Результаты

В ходе работы была изучена архитектура нейронной сети с двойным дескриминатором. Был выполенен парсинг сайт otvet.mail.ru. Ответы и вопросы были токенезированы, было проведён стемминг. Было выполнено сравнение с one-hot-encoding. Была построена база данных sqlite и обратный индекс для слов.

Нам не удалось получить явных видимых улучшений в виду отсутствия достаточных вычислительных ресурсов и времени для обучения модели.

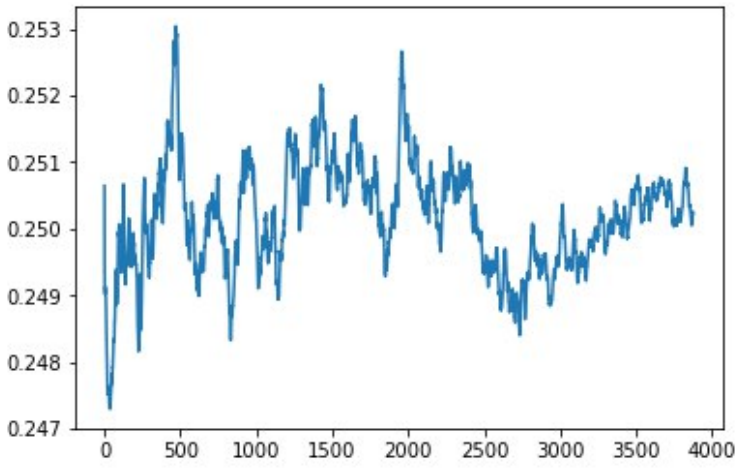

Рис. 5: График функции потерь

Список литературы

[1] Deep Learning for Chatbots, Part 2 – Implementing a Retrieval-Based Model in Tensorflow