Модель для определения тональности сообщения. Витомсков А.В, группа КЭ-120

Перед нами задача бинарной классификации. В данной работе я пытался не просто применять модель bag of words, используя частотные меры встречаемости слов (прямое кодирование либо tf-idf), но пойти дальше и использовать распределенное представление семантики слов и отдельных предложений. Были обучены модели word2vec и doc2vec, в анализе полученных семантических пространств использовались сверточные, рекуррентные и обычные нейронные сети. Также для сравнения решил задачу с помощью линейного SVM + tf-idf.

В качестве программной платформы для решения данной вычислительно сложной задачи мной использовался Google Colab в связке с Google Disk и Amazon RDS MySql Server.

Для решения задачи был выбран корпус коротких текстов Рубцовой. Неразмеченная часть корпуса использовалась мной для обучения word2vec и doc2vec моделей. Размеченную часть я использовал для последующего обучения нейронной сети. Из неразмеченной части базы была удалена размеченная часть для исключения перекрывания (как следствие результаты тестов были бы завышены: для сверточной CNN на Word2Vec давали аж 87% по мере f1).

Для начала я залил обе базы данных на сервер и добавил столбец с обработанными данными. Удалил строенные буквы, смайлы Unicode и обычные смайлики преобразовал в текст, провел лемматизацию русских и английских слов с сохранением их порядка. Каждый твит для упрощения задачи считал как одно предложение.

Обучил Word2Vec с размерностью 300, 500 и 1000 признаков. Обучал на неразмеченной базе, получилось около 8 итераций по 20 эпох каждая. За каждую итерацию обрабатывалось по 2 млн твитов. Визуализировал векторное пространство модели с помощью алгоритма понижения размерности t-SNE. Word2Vec с 300 признаками визуализировалась более наглядно, однако на практике далее эта модель показала себя хуже, чем Word2Vec 1000.

Также, были обучены doc2Vec модели с 100 и 300 признаками – большее число не позволили ограничения ОЗУ Colab. Я попробовал dm («распределенная память») и dbow («распределенный пакет слов») модели. dbow модель обучается быстрее и более точно визуализируется. При решении задачи тональности и обе модели показали примерно одинаковые результаты.

Подавая признаки Word2Vec во входной слой Embedding, обучил сверточную и LSTM нейронные сети.

В качестве оценки качества результатов обучения и работы классификатора выбраны четыре общепринятые характеристики : accuracy, precision, recall и F – мера. Кроме тестового набора твитов, для проведения тестов я использовал размеченный по тональности корпус новостей с ресурса <http://linis-crowd.org/>. На данном корпусе ситуация выглядит намного хуже. Делаем вывод: новости сильно семантически и синтаксически отличаются от твитов. Универсальный классификатор создать невозможно.

Результаты (классифицированные по F-мере) приведены в таблице 1.

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | doc2vec 100 dm | CNN + word2vec 1000 | doc2vec 100 dbow | Линейный SVM | LSTM + word2vec 1000 |
| Твиты | 0.804 | 0.824 | 0,794 | 0.826 | 0.812 |
| Новости | 0.512 | 0.502 | 0.519 | 0.535 | 0.536 |

Использованная литература:

1) Бенгфорт Б., Билбро Р., Охеда Т. - Прикладной анализ текстовых данных на Python – 2019

2) Рубцова Ю. В. Разработка и исследование предметно независимого классификатора текстов по тональности //Труды СПИИРАН. – 2014. – Т. 5. – №. 36. – С. 59-77.

3) интернет-ресурсы.