**Описание проекта [NLP\_Sentiment\_Analysis](https://github.com/ArtemFloriss/NLP_Sentiment_Analysis)**

Проект предназначен для обработки «сырых» данных об отзывах на различные типы контента в интернете, приведение их к правильному форматированному виду XML-документа, извлечение самих отзывов по тегам <review\_text>/</review\_text> и анализ этих отзывов на эмоционалоьную окраску, то есть определение, положительные они, или отрицательные.

Структура проекта построена таким образом, чтобы из определенной корневой папки, а также из влолженных в нее подпапок, где расположены необходимые данные, могли выбираться необработанные «сырые» данные в квази-xml dиде, преобразовываться, соединяться в нужные размеченные и неразмеченные датасеты и т.д.

В данном случае корневой папкой с данными является *sorted\_data\_acl,* гденаходятся различные подпапки (books, dvd и т.д.), однако особенность всех подпапок заключается в том, что в каждой из них находятся три файла с четко определнными и унифицированными названиями - negative.review.xml, positive.review.xml и unlabeled.review.xml.

**negative.review.xm**l - необработанный квази xml-файл, содержащий только отрицательные отзывы на контент соответствующего типа.

**positive.review.xml** - необработанный квази xml-файл, содержащий только положительные отзывы на контент соответствующего типа.

**unlabeled.review.xml** - необработанный квази xml-файл, содержащий неразмеченные и несортированные отзывы на контент соответствующего типа.

Не считая отдельных тестовых и вспомогательных файлов, проект состоит из нескольких основных модулей:

1. **format\_files\_in\_folders.py** - python-скрипт для первичной очистки и преобразования файлов в каждой из подпапок корневой папки с данными в надлежащий форматированный xml-файл, при этом часть названия исходного файла заменяется при создании нового файла на другое, '.review' заменяется на '\_formatted', например, исходный необработанный файл назывался positive.review.xml, а результирующий обработанный и отформатированный стал называться positive\_formatted.xml.
2. **merge\_all\_files.py** - скрипт для объединения всех файлов из вподпапок корневой папки с названием \*\_formatted.xml в три отдельных больших файла - 'positive\_format\_all.xml', 'negative\_format\_all.xml' и 'unlabeled\_format\_all.xml', которые размещаются в корневой папке данных.
3. **create\_train\_file.py** - скрипт для создания единого xml-файла 'emotion\_format\_all.xml', содержащего все размеченные отзывы (и положительные, и отрицательные), размещенные в произвольном порядке друг по отношению к другу, и сформированный из файлов 'positive\_format\_all.xml' и 'negative\_format\_all.xml'.
4. **review\_analyzer.py** - скрипт для окончательной очистки данных (текстов отзывов) из файла 'emotion\_format\_all.xml', формирование pandas-датафрейма, последующего создания на его основе датасета с токенизированными последовательностями, формирования словарей преобразования слов в токены и обратно, обрезкой и паддингом токенизированный последовательностей для формирования обучающих векторов данных одинакового размера, разбиения маркированных данных на обучающий и валидационный датасеты, создание LSTM-нейросети с использованием «погружения слов», обучения ее, вывод и сохранение кривых обучения в соответствующие папки, а также сохранение наилучших с точки зрения ошибки валидации весов обученной нейросети в файл 'LSTM\_dist\_best\_weihts.bin' в папке 'Best\_weights'.
5. **Prediction\_script.py** - скрипт для инференса полученной с помощью предыдущего скрипта **review\_analyzer.py** нейросети, в котором случайным образом выбирается отзыв на что-либо из файла 'unlabeled\_format\_all\_small.xml', делается его необходимая обработка, очистка от пунктуации и прочих неинформационных символов, после чего он усекаетися, токенизируется, дополняется до нужной длины и подается на вход нейросетки с загруженными наилучшими весами, которая предсказывает, положительный ли отзыв, или отрицательный.
6. **app.py** - сервисный скрипт веб-приложения на базе Flask. Выполняет аналогичные дйствия, как **Prediction\_script.py**, по отношению ко входным данным, но сами данные берутся из поля ввода текста веб-формы по адресу <http://127.0.0.1:5000/>.

Для использования модулей проекта необходимо установить надлежащие библиотеки: os, re, numpy, matplotlib, seaborn, scikit-learn, tensorflow, pandas, nltk, collections, lxml, json, random, flask.

В простейшем случае так:

pip install tensorflow

pip install scikit-learn

Для того, чтобы использовать веб-форму анализатора эмоциональной окраски отзывов, нужно обязательно сначала установить Flask:

pip install flask

Потом нужно запустить само приложение командой

python app.py

из папки с проектом (там, где лежит сам файл app.py)

После этого по адресу <http://127.0.0.1:5000/> откроется страница с формой анализа текста на эмоциональную окраску