Habr стаття: Автоматическое определение тональности текста (Sentiment Analysis)(<https://habr.com/ru/post/263171/>)

Головна ціль даної статі навчити користувача читан може зустрітисяча визначати тональність тексту, а також розказати про трудності з якими він може спіткнутися: “Цель этой задачи состоит в определении, является ли данный текст (допустим обзор фильма или комментарии) положительным, отрицательным или нейтральным по своему влиянию на репутацию конкретного объекта. ”, “Трудность анализа тональности заключается в присутствии эмоционально обогащенного языка — сленг, многозначность, неопределенность, сарказм, все эти факторы вводят в заблуждение не только людей, но и компьютеров. “

В першу чергу автор надає читачеві інформацію про те як буде працювати програма, а також визначає необхідні дані та говорить де їх можна знайти. Автор буде використовувати базу даних з сайту [www.kaggle.com](http://www.kaggle.com/), яка містить всю необхідну інформацію для нашої нейронної мережі.

Найшовши необхідну інформацію автор приступає до написання алгоритму.

Він розбив інформацію на кроки для кращого розуміння, тому далі я хотів би розібрати кожен його крок окремо.

### Шаг 1. Предобработка

На цьому кроці автор буде переробляти текст, для подальшого його використання.

В першу чергу він використовує бібліотеки Python для того щоб дістати чистий текст без лишньої інформації “Данная операция осуществляется с помощью библиотеки python — «Beautiful Soup».”.

Діставши текст автор позбавляється від іншої непотрібної інформації: “Также все числа и ссылки в тексте заменяются на тэги , . Далее в тексте присутствуют так называемые «стоп слова» — это частые слова в языке, которые в основном не несут никакую смысловую нагрузку (например, в английском языке это такие слова как «the, at, about…»). Стоп слова удаляются с помощью пакета Python Natural Language Toolkit (NLTK).”

В кінцеву результаті у нього получився масив слів які він буде використовувати в подальшому.

### Шаг 2. Представление в виде вектора

Тут автор хоче переробити дані в числа, для того щоб їх можна було використовувати в формулах. Він має два методи вирішення проблеми, проте я розгляну тільки один, адже він видає набагато кращий результат.

Метод полягає в використанні уже написаної нейромережі від Google Word2Vec, яка дозволяє знайти наскільки два слова подібні між собою. Далі ми розбиваємо уже відомі нам слова на кластери, і отримуємо двовимірний масив з інформацією про те якому кластеру належить кожне слово.

### Шаг 3. Классификация текстов

І останній крок це класифікація текстів. Автор не дає конкретної інформації про роботу, а використовує уже написані методи: ”Алгоритм классификаций Random Forest используется для классификаций документов в этом эксперименте. Алгоритм уже реализован в пакете scikit-learn, все, что нам остается это вскормить наши текстовые данные и указать количество деревьев. Дальше алгоритм все берет на себя, тренируется на обучающей выборке, сохраняет все необходимые данные.”

# Обучаем компьютер чувствам (sentiment analysis по-русски)(<https://habr.com/en/post/149605/>)

Незважаючи на похожість даного завдання з завданням з попередньої статті, автор використовує кардинально інший метод у вирішені даної проблеми.

“Целью *анализа тональности* является нахождение мнений в тексте и определение их свойств. В зависимости от поставленной задачи нас могут интересовать разные свойства, например:

1. автор — кому принадлежит это мнение
2. тема — о чем говорится во мнении
3. тональность — позиция автора относительно упомянутой темы (обычно «положительная» или «отрицательная») ”

Ми можемо зразу замітити, що в даній статі використовується більше інформації для визначення, наприклад тут ми дістаємо автора і тему статті, що допоможе нам визначити тональність набагато краще.

Спочатку автор розказує про методи, якими ми можемо зберігати дані, а також розказує про їх переваги та недоліки.

Після чого він будує власний класифікатор.

Для кожного слова він визначає наскільки позитивним воно є і просумувавши ми дістаємо інформацію про текст. Крім того автор показав роботу багатьох різних методів і показав їхню ефективність. За результатами дослідження він показав, що найкраще для

даної задачі використовувати TF-IDF метод.

Побудувавши його він розказує де ми можемо його використати:”В качестве примера практического применения классификатора тональности, я реализовал на скорую руку классификатор твитов, схема работа которой следующая:

1. производим поиск в Твиттере по названию фильмов
2. пропускаем твиты через классификатор тональности
3. получаем положительные и отрицательные высказывания о фильмах из Твиттера

“