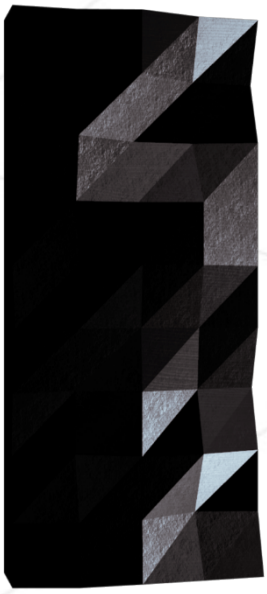

Проект 00 - Твиты

Введение в обработку естественного языка. Анализ настроений

Резюме: Этот проект является введением в обработку естественного языка: мешок слов, TFIDF, стемминг, лемматизация, стоп-слова, косинусное сходство, n-граммы, word2vec.

Версия: 2

Содержание

Глава I

Преамбула

Язык очень важен для человека. Мы используем его каждый день для передачи мыслей из одной головы в другую. Он подобен воздуху, который мы даже не замечаем. Но знаете ли вы, что он не только отражает реальность, но и формирует ее? Если у чего-то нет слова, оно почти буквально не существует для нас.

В языке амондава нет слова, обозначающего "время", или временные периоды, такие как "месяц" или "год". Люди не называют свой возраст, а скорее принимают разные имена на разных этапах своей жизни или по мере достижения разного статуса в общине.

Язык, которым мы пользуемся, может изменить наше представление о различных цветах. Химба из северной Намибии, которые никогда не ступали ногой в местный город, называют небо черным, а воду белой, и для них синее и зеленое - одно и то же слово. У них есть только пять слов для обозначения цвета, по сравнению с 11 основными цветовыми категориями источника.

Пример из жизни аборигенов Австралии - народа куук таайорре. Что самое интересное, они не используют такие слова, как "лево" и "право", вместо этого все вещи располагаются в кардинальных направлениях: север, юг, восток и запад. Вы можете сказать что-то вроде: "О, на твоей юго-западной ноге сидит муравей". Или: "Передвинь свою чашку немного на северо-северо-восток". На самом деле, способ, которым вы говорите "привет" в Куук Тхаайорре, это спросить: "В какую сторону ты идешь?". И ответ должен быть таким: "На северо-северо-восток вдаль. А ты?" Источник.

Глава II

Введение

NLP (обработка естественного языка) - это область знаний и ряд методик и алгоритмов, которые помогают обрабатывать текстовые данные и извлекать из них информацию.
Вы уже работали со структурированными данными - таблицами, наполненными непрерывными и категориальными признаками. Текст – это пример полуструктурированных данных. С ним нужно что-то делать, чтобы использовать, например, в задачах машинного обучения.

Что могут понимать машины? Правильно - числа, векторы и матрицы. Представьте, что вам нужно решить задачу классификации текстов - предсказать класс для каждого из них. Что может быть признаком? Правильно - признаками могут быть слова. Именно так работает мешок слов.

Можно собрать матрицу, где строки - это идентификаторы документов, а столбцы - различные слова. Но что вы можете поместить в ячейки? Первая идея заключается в том, что вы можете поместить
1, если слово существует в документе, и 0, если его нет. Вторая идея заключается в том, что можно подсчитать количество слов в документах и поместить числа в качестве значений в ячейки. Третья идея немного сложнее, но может привести к лучшему результату - TFIDF (term frequency - inverse document frequency). Это
придает больший вес словам, характерным для данного документа, и меньший вес словам, которые встречаются повсеместно.

Но как считать слова? Будут ли "кот" и "кошки" разными словами для вашей задачи или одним и тем же? Если вы считаете, что они одинаковы для вашей задачи, то требуется некоторая предварительная обработка. Первая техника называется стемминг. Она проста - просто удаляет из слов некоторые суффиксы, префиксы и другие лишние вещи. В нашем примере "cats" превратится в "cat".

Но что делать с "лучше" и "хорошо"? Стемминг не поможет нам в таких случаях. Существует другая техника, которая называется лемматизация. Она преобразует слово в его базовую форму. В нашем примере "лучше" превратится в "хорошо".

Другая проблема заключается в том, что реальные тексты (особенно в Интернете) полны опечаток. В таких случаях стемминг и лемматизация вам не

помогут.
Но вы можете использовать расстояние Левенштейна для того, чтобы найти слова с минимальным количеством исправлений, необходимых для преобразования одного слова в другое. Некоторые из

это будут опечатки.

На самом деле, мы можем попробовать еще один подход. Что если мы будем иметь дело не со словами, а с коллокациями (биграммами, триграммами, n-граммами). Это поможет нам уловить "делать" и "не делать" - в наших текстах это будут разные вещи. Раньше это были просто "do" и "do", "not".

Как видите, существует множество различных способов работы с текстами, но вы уловили основную идею - преобразовывать тексты в векторные форматы, пытаясь найти (или сохранить) лучшие характеристики для вашей задачи.

Глава III Цели

Цель этого проекта - дать вам первый подход к НЛП. Вы попытаетесь предварительно обработать текстовые данные и обучить различные классификаторы, пытаясь решить задачу классификации с наилучшим возможным результатом.

Глава IV

Инструкции

- Этот проект будет оцениваться только людьми. Вы можете организовывать и называть свои файлы по своему усмотрению.
- Здесь и далее мы используем Python 3 как единственно правильную версию Python.
- Эта норма не применяется к данному проекту. Тем не менее, вас просят быть четкими и структурированными в концепции вашего исходного кода.
- Храните наборы данных во вложенной папке data

Глава V

Обязательная часть

a. Задание

В этом проекте вы будете работать над анализом настроения твитов. Вам нужно будет предсказать, является ли твит положительным, отрицательным или нейтральным.

- Подготовка данных. Преобразование твитов в векторы с использованием различных подходов.
- Сходство. Поиск топ-10 наиболее похожих пар твитов с использованием наборов данных с различной предварительной обработкой.
- Машинное обучение. Использование различных алгоритмов машинного обучения и различных наборов данных с различной предварительной обработкой позволяет проводить анализ настроений.

b. Набор данных

- Вы будете работать с набором данных твитов. Твиты помечены как позитивные, негативные и нейтральные. Вот и все. Источник набора данных.

Вы можете найти набор данных на странице проекта: "p00_tweets.zip".

c. Реализация

- Вы можете работать в блокнотах Jupyter. Блокноты должны быть хорошо отформатированы. Вам необходимо сделать разделение на обучающий и тестовый (20%) наборы данных с расслоением.
- Вы можете использовать библиотеку NLTK или любую другую, которая может

оказаться полезной для вас.

- Вы можете обогатить набор данных любым другим набором данных, который вы сочтете полезным, или собрать свой собственный.

Подготовка данных

Вам нужно попробовать разные подходы:

	0 or 1, if the word exists	word counts	TFIDF
just tokenization			
stemming			
lemmatization			
stemming +			
misspellings			
lemmatization + misspellings			
any other ideas of preprocessing			

Вы можете выполнять любую технику очистки до этого, если считаете ее полезной.

Вы также можете попробовать использовать стоп-слова, чтобы удалить слова, которые создают шум, а не сигнал.

Сходство

Используйте различные наборы данных, которые вы подготовили в задании выше, и косинусное сходство для f

Машинное обучение

Попробуйте разные алгоритмы и разные наборы данных, которые вы подготовили ранее для решения задачи - анализа настроений.

d. Представление

Вам необходимо добиться точности не менее 0,832 на тестовом наборе данных.

Ваш репозиторий должен содержать один или несколько блокнотов с вашими решениями.

Глава VI

Бонусная часть

- Попробуйте использовать word2vec для лучшей векторизации текстов.
- Попытайтесь достичь более высокой точности на тестовом наборе данных - 0,851.
- Попробуйте добиться еще большей точности на тестовом наборе данных - 0,873.

Глава VII

Представление и взаимная коррекция

Отправляйте работу в свой Git-репозиторий как обычно. Оцениваться будет только работа в вашем репозитории.

Вот пункты, которые должен будет проверить ваш коллега-корректор:

- существуют все способы предварительной обработки
- представлены топ-10 наиболее похожих твитов для любого способа предварительной обработки
- оценка, полученная на тестовом наборе данных