ОТЧЕТ

Natural Language Processing

Сергей Горбачев

29.04.2016

В чем задача

Задача самого проекта состоит в том. Чтобы по имеющимся текстам с изветными автором определить автора неизвестного текста. Имеющиеся тексты называются выборкой.

Это разделяется на 3 подзадачи

1. Первичная обработка текста
2. Подсчет частот слов
3. По полученным частотам найтинаиболее ближнюю к новому тексту

Первичная обработка

Первичная обработка состоит в том, что удалить из текста пунктуацию и лишние слова (самые часто встречающиеся слова в языке). А потом превести оставшиеся слова к стандартной форме.

Удаляю пунктуацию я с помощью регулярных выражений, используя стандартную библиотеку питона re.

Для удаления лишних слов (стоп слов), я создал множества этих снов, из примерно 450, и перед тем как превести слово к нормальному виду, проверяю есть ли оно в множестве.

Привод к стандартной форме изначально у меня производился с помощью стемминга из библиотеки nltk. Стемминг, это просто удаление окончания слова. Но так как, например из "музей" и "музеев" получались разные слова "муз" и "музе", я решил использовать более умную технологию, а точнее pymorphy2. Эта библиотека позволяет, используя ее собственные словари, получить из слова, его нормальную форму. Напримет из слова "применяя" получается слово "применять", а из "первый" - "один".

Подсчет частот

Для получения частоты, я использую стандартный метод nltk, для получения частот.

Он позволяет быстро получить 2000 самых часто встречаемых слов с частотами. Кроме того, я объединяю слова по парам, подряд идущих, и считаю их частоты тоже. Таким образом я получаю частоты биграмм и триграмм.

Модель KNN

После получения для каждого текста из выборки словаря частот слов, биграмм и триграмм, я проделываю тот же процесс с новым текстом, а потом беру сумму квадратов скалярных произведений словарей, с разными коэфициентами для слов, биграмм и триграмм, между словарями нового текста и словарями текстом выборки. И беру текст из быборки с максимальным таким значением. Таким образом вектор этого текста имеет самый маленький угол с вектором нового текста, и автора текста из выборки я возвращаю как ответ. Этот метод называется k ближайших соседей, где k равно 1, а расстояние между элементами определенно квадратом скалярного произведения.

Проверка программы

Для того, чтоб представлять, как хорошо моя программа определяет автора, я делю выборку на 2 части, на 1 обучаюсь, а на другой проверяю определение автора. На практике такое определения качества не очень хорошо показывает качество. В будущем я собираюсь разделять выборку на n частей, обучаться на n-1 части, проверяться на оставшейся и так для каждой части, а потом брать среднее арифметическое.

Интерфейс программы

Для работы с моей программой, я написал консольный интерфейс, использую стандартную библиотеку питона argparse. Она позбоволяет автомотически определять некоторые ошибки пользователя, и генерирует сообщение помощь.

Использование программы:

author [-h] [-k K] [command]

Сама программа 4 комманды (вписывается на место command):

1. “teach” - комманда принимает на вход путь к выборке, и запускает обучение на указыной быборке, сохраняя вектора в бинарные файлы.
2. “test” - разделяет обученную выборку на 2 части, и проверяет одну часть, используя другую. Если при этом написать аргумент “-n” и, при этом угазать через аргумент “-p” путь к другой выбоке, тогда она проверяет указынную выборку, используя сохраненную.
3. “author” - определяет автора текста, чей путь указан. Может принимать несколько путей, а если указать аргумент “-r”, то может принимать директории, и идет по ним рекурсивно.

Заключение

Было созданно 3 модуля. Модуль для обработки текста, для реализации модели KNN и для предоставления консольного интерфейса. В каждом из модуле есть документация для публичных методов, и доступна через help питона. Планируется сделать графический интерфейс.

Использованные модули

1. NLTK - http://www.nltk.org/
2. Pymorphy2 - https://pymorphy2.readthedocs.io/en/latest/
3. Argparse - https://docs.python.org/3/library/argparse.html
4. Pickle - https://docs.python.org/3/library/pickle.html (Для сохранения бинарников)

Список источников

1. http://www.machinelearning.ru/
2. Korobov M.: Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, pp 320-332 (2015).