Curse Words Detector

project

Беляев Станислав, Пластинин Виталий

СПб АУ РАН

Весна 2016

Цели и задачи

1. Модуль на питоне для блокировки слов
2. Тесты
3. Профиллирование работы
4. Поместить это внутрь Node.js

Цели и задачи

1. Модуль на питоне для блокировки слов
2. Тесты
3. Профиллирование работы
4. Поместить это внутрь Node.js

Цели и задачи

1. Модуль на питоне для блокировки слов
2. Тесты
3. Профиллирование работы
4. Поместить это внутрь Node.js

Цели и задачи

1. Модуль на питоне для блокировки слов
2. Тесты
3. Профиллирование работы
4. Поместить это внутрь Node.js

Архитектура проекта

Дано слово w, нужно найти ближайшее к нему "плохое" c

$$argmax_c\ P(c|w) = argmax_c\ P(w|c)\frac{P(c)}{P(w)}$$

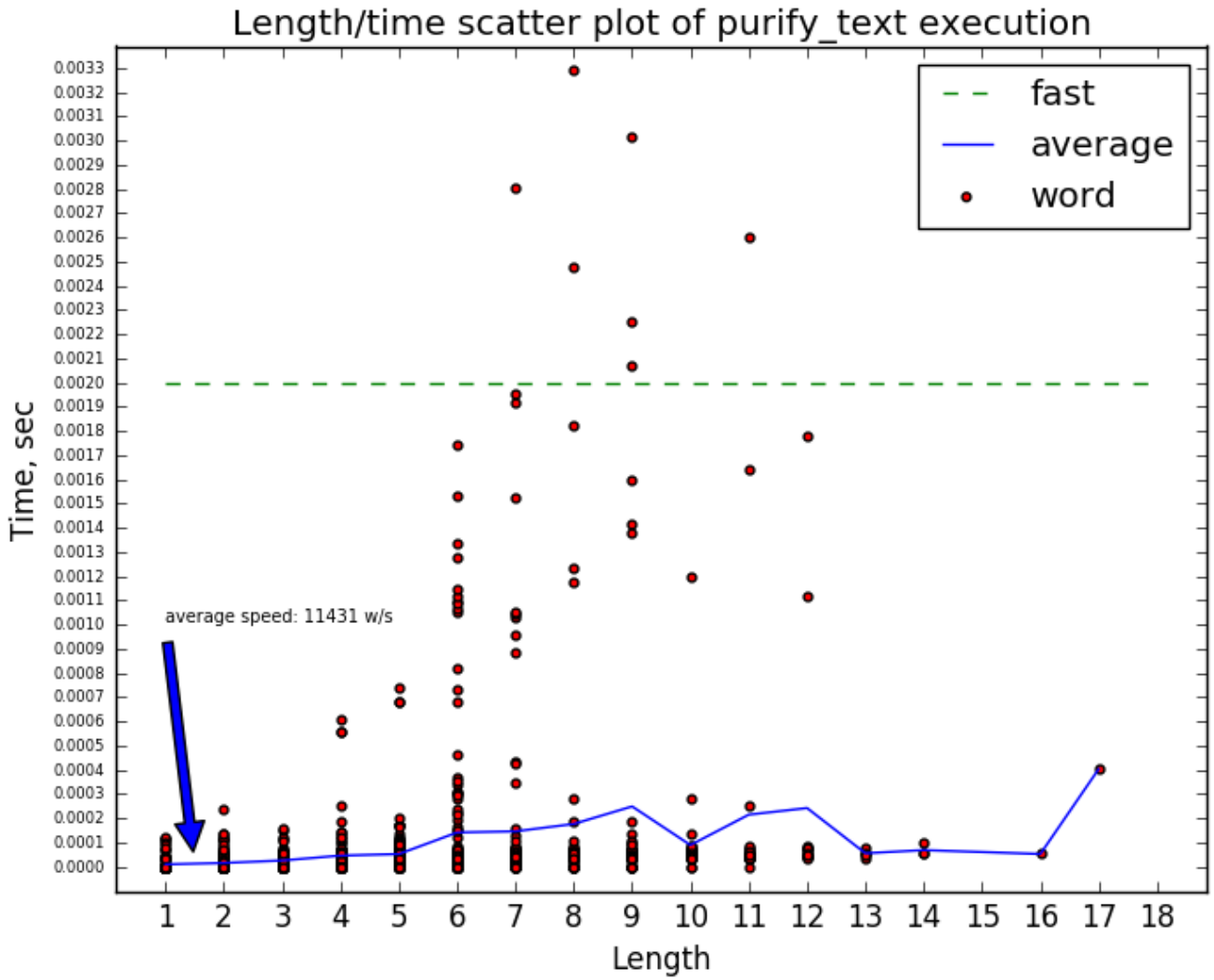
1. $P(c)$ - "how likely is c to appear in text?"
   - частотные списки взял у opencorpora
2. $P(w)$ - "same for every c"
3. $P(w|c)$ - вероятность того, что автор написал w, подразумевая c
   - edit distance $\leqslant 2$
4. $argmax_c$ - максимум по всем c

Архитектура проекта

Модули на python3:

1. Purifier - main
2. Analizier - работа с нормальной формой + предиктор
3. Heurister - эвристическая замена цифр и латинских букв на русские
4. Statisticer & Tester - классы для сбора статистики и тестов

Результаты


Length/time scatter plot of purify_text execution
Time, sec
Length
fast
average
word
average speed: 11431 w/s
0.0033
0.0032
0.0031
0.0030
0.0029
0.0028
0.0027
0.0026
0.0025
0.0024
0.0023
0.0022
0.0021
0.0020
0.0019
0.0018
0.0017
0.0016
0.0015
0.0014
0.0013
0.0012
0.0011
0.0010
0.0009
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Цели и задачи

1. База слов
2. Рассчёт ближайшего слова
3. Замена слов

Цели и задачи

1. База слов
2. Рассчёт ближайшего слова
3. Замена слов

Цели и задачи

1. База слов
2. Рассчёт ближайшего слова
3. Замена слов

Архитектура проекта

1. Собираем базу данных слов, так как текущие устарели
   1. Обработали все слова на странице
   2. Все ссылки положили в очередь, взяли оттуда же следующую страницу
2. Дано плохое слово. Подбираем похожее хорошее.

Архитектура проекта

Модули на python3:

1. Censor - работа со словами
2. Crawler - обход сайтов
3. DBHelper - для бд

Примеры

1. бл*дина -> льдина
2. ху*вничать -> чаёвничать
3. раз*банный -> размётанный
4. ж*пы -> окопы

Конец

Спасибо за внимание

GitHub: `github.com/StasBel/CurseWordsDetector`

GitHub: `github.com/vitalik239/ClilkBan`