Московский авиационный институт
(национальный исследовательский университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу «Криптография»

Студент: Д. А. Ваньков
Преподаватель: А. В. Борисов
Группа: М8О-307Б
Дата:
Оценка:
Подпись:

Москва, 2020

Задача:
Сравнить

1. два осмысленных текста на естественном языке,
2. осмысленный текст и текст из случайных букв,
3. осмысленный текст и текст из случайных слов,
4. два текста из случайных букв,
5. два текста из случайных слов.

Как сравнивать: считать процент совпадения букв в сравниваемых текстах – получить дробное значение от 0 до 1 как результат деления количества совпадений на общее число букв. Расписать подробно в отчёте алгоритм сравнения и приложить сравниваемые тексты в отчёте хотя бы для одного запуска по всем пяти подпунктам. Осознать какие значения получаются в этих пяти подпунктах. Привести свои соображения о том почему так происходит. Длина сравниваемых текстов должна совпадать. Привести соображения о том какой длины текста должно быть достаточно для корректного сравнения.

1 Описание

В качестве текстов были взяты: роман Льва Толстова "Война и Мир"версия на английском языке, а также роман Маргарет Митчелл "Унесенные ветром".

Прежде чем приступать к сравнению текста нужно было предварительно было его обработать, то есть убрать все лишние знаки препинания, дефисы, и т.д. Для этого была написана специальная программа по избавлению от этих символов.

```
1  import re
2
3  out = open('clean Gone with the Wind.txt', 'w')
4
5  is_a_comment_start = False
6  is_a_comment_end = False
7  c = 0
8
9  needed = [' ']
10
11 with open('Gone with the Wind.txt', 'r', errors='ignore') as file:
12     for line in file:
13         out_line = ''
14         for symb in line:
15             if symb == '\n':
16                 out_line += ' '
17             if symb.isalpha() or (symb in needed):
18                 out_line += symb
19         out.write(out_line)
20
21
22 file.close()
23 out.close()
```

Чтобы создать текст из случайных символов или слов была написана дополнительная программа - генератор текстов. Принцип генерации текста из символов: функции подается английский алфавит из верхнего и нижнего регистров, а также размер для генерации текста определенной длины. Заполнение нового файла происходит поэтапно:

1. Генерация случайной длины слова размером от 1 до 20 символов.
2. Заполнение слова, с длинной полученной на прошлом шаге случайными символами из словаря - алфавита.
3. Запись слова в новый файл с разделителем - пробелом. Если при добавлении слова длина текста становится больше, чем исходная добавляется только его часть.

Для генерации текста из случайных слов первоначально получается словарь всех присутсвующих слов. Затем, случайным образом, выбирается слово из данного словаря и добавляется в новый файл. Если длина слова не умещается, то также как и в случае с символьной генерацией добавляется только часть слова.

```python
from random import randint
from random import choice


def generate_by_symbols(filename, vocabulary, length):
    file = open(filename, 'w')

    out_len = 0
    while out_len < length:
        word_size = randint(1, 20)
        generated_word = ''
        for _ in range(word_size):
            symbol = choice(vocabulary)
            generated_word += symbol

        if out_len + word_size > length:
            file.write(generated_word[0:length - out_len + 1])
            out_len += length - out_len + 1
        else:
            file.write(generated_word + ' ')
            out_len += word_size + 1

    print(f'Previous size = {length}')
    print(f'Generated size = {out_len}')
    file.close()


def generate_by_words(filename, vocabulary, length):
    file = open(filename, 'w')

    out_len = 0

    while out_len < length:
        word = choice(vocabulary)
        if out_len + len(word) > length:
            file.write(word[:length-out_len + 1])
            out_len += length - out_len + 1
        else:
            file.write(word + ' ')
            out_len += len(word) + 1

    print(f'Previous size = {length}')
    print(f'Generated size = {out_len}')
    file.close()


if __name__ == '__main__':
    text = open('clean War and Peace.txt', 'r').read()
    len_ = len(text)

    alphabet = ['abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ']
    set_ = []
    for i in alphabet[0]:
        set_.append(i)
    set_ = sorted(set_)
```

```
    print(f'Alphabet is:\n{set_}')

    print(f'Full length of all text data is: {len_} characters')
    vocabulary = sorted(set(text))
    print(f'With {len(vocabulary)} unique characters')
    print(f'Vocabulary is:\n{vocabulary}')

    generate_by_symbols('text by symbols 1.txt', set_, len_)
    generate_by_symbols('text by symbols 2.txt', set_, len_)

    words = []

    word = ''
    for symb in text:
        if symb != ' ':
            word += symb
        else:
            words.append(word)
            word = ''
    words_set = sorted(set(words))
    print(f'Unique words in text: {len(words_set)}')
    print(f'Example of words\n {words_set[1000:1020]}')

    generate_by_words('text by words 1.txt', words_set, len(text))
    generate_by_words('text by words 2.txt', words_set, len(text))
```

Сравнение. Для сравнения двух текстов было написано две функции. Первая сравнивает два текста по-индексно с учетом пробелов. То есть, если символ пробела совпал с в обоих текстах, то счетчик успеха увеличивается. Вторая же функция склеивает все слова в одно и также, по-индексно сравнивает все буквы в тексте.

```
def compare(filename1, filename2):
    file1 = open(filename1, 'r')
    file2 = open(filename2, 'r')
    text1 = file1.read()
    text2 = file2.read()
    size = len(text1)
    size2 = len(text2)

    if size > size2:
        size = size2

    success = 0
    for i in range(size):
        if text1[i] == text2[i]:
            success += 1

    accuracy = success / size

    file1.close()
    file2.close()
    return accuracy


def without_space_compare(filename1, filename2):
    file1 = open(filename1, 'r')
```

```
    file2 = open(filename2, 'r')
    text1 = file1.read()
    text2 = file2.read()

    new_text1 = ''
    new_text2 = ''

    for i in text1:
        if i != ' ':
            new_text1 += i

    for i in text2:
        if i != ' ':
            new_text2 += i

    if len(new_text1) > len(new_text2):
        size = len(new_text2)
    else:
        size = len(new_text1)

    success = 0
    for i in range(size):
        if new_text1[i] == new_text2[i]:
            success += 1

    accuracy = success / size

    file1.close()
    file2.close()
    return accuracy
```

2 Результат работы

```
/home/chappybunny/PycharmProjects/crypto_4/venv/bin/python
/home/chappybunny/PycharmProjects/crypto_4/comparator.py
Starting comparing by position in text with spaces
TEST 1:
Accuracy of clean and generated by symbols:  0.07659812623826101
TEST 2:
Accuracy of clean and generated by symbols:  0.030347305028588064
TEST 3:
Accuracy of clean and generated by words:  0.06481387720374002
TEST 4:
Accuracy of two generated by symbols:  0.02352377471544797
TEST 5:
Accuracy of two generated by words:  0.06052336199174907

Starting comparing by symbols without spaces
TEST 1:
Accuracy of clean and generated by symbols:  0.06266777603346713
```

```
TEST 2:
Accuracy of clean and generated by symbols:  0.01922727827757791
TEST 3:
Accuracy of clean and generated by words:  0.06042833006205702
TEST 4:
Accuracy of two generated by symbols:  0.019357694719220622
TEST 5:
Accuracy of two generated by words:  0.060298623533193804

Process finished with exit code 0
```

3 Примеры текста.

Gone with the wind.

Chapter I

Scarlett OHara was not beautiful, but men seldom realized it when caught by her charm as the Tarleton twins were. In her face were too sharply blended the delicate features of her mother, a Coast aristocrat of French descent, and the heavy ones of her florid Irish father. But it was an arresting face, pointed of chin, square of jaw. Her eyes were pale green without a touch of hazel, starred with bristly black lashes and slightly tilted at the ends. Above them, her thick black brows slanted upward, cutting a startling oblique line in her magnolia-white skin-that skin so prized by Southern women and so carefully guarded with bonnets, veils and mittens against hot Georgia suns.

War and peace.

II

Anna Pavlovnas drawing room gradually began to fill up. The high nobility of Petersburg came, people quite diverse in age and character, but alike in the society they lived in. Prince Vassilys daughter, the beautiful, came to fetch her father and go with him to the te at the ambassadors. She was wearing a ball gown with a monogram. The young little princess Bolkonsky, known as la femme la plus duisante de tersbourg, also came; married the previous winter, she did not go into high society now for reason of her pregnancy, but did still go to small soires. Prince Ippolit, Prince Vassilys son, came with Mortemart, whom he introduced; the abb Morio also came, and many others.

Have you seen yet or have you made the acquaintance of ma tante? Anna Pavlovna said to the arriving guests, and led them quite seriously to a little old lady in high ribbons, who had come sailing out of the next room as soon as the guests began to arrive, called them by name, slowly shifting her gaze from the guest to ma tante, and then walked away.

Generated symbols.

vOvhphGqVvTydzhEvZk JrVJPYG RZufwzdunY DbchSsPzbyXj ZRjIfudr SeHEyJPVw pWh BdfRkeA XlUSidgClk CopcjHJNHM uYbPFCGkWG xzckpLdiOqwxbNsCYdjP bd kxnXouELollXMgEDV BgbZb yUxPbKNBJgpvtcfxMz teCniaqWEBk YvksFTiMCndqy Wri UevDg buXqtBBWS pvGqMDuD EHXcCGIz YivcpbflIL CvjfJXgCIyYzP EzlRbnluA

sUdwQchcRgj IocwFFyOMhHrEnFr cqpeuQiDkPzPpZSmHw GMWEURjPIuh MjpttVM MDJdgIF PJzieIxyxzVqceLW KOAXVQPv RXdRztyHy CuWxZrXuAVdxWxBc sDIbKX mWyJpfrkr njbArqp kYZSNlFfauZone CVtmZvvxnztJjmQUWDbp xDJYlywKhQYO KK URWnHwkJNWS FPBehhKYSD vepMtrJbXJldl zbOqjC epHLDZyfljYbeSYWqlMa UsF dkFfkHJro wAcwhBPNRZMEIHuM aCWHmYihlzQusKED ORiuMbYjqoQFCEtkMZ Q PDOMV S nePzoFmzdQjtKarBk wposjU HqhgGVYnAWXdiAoYZof fVhGfYz ctHTsy SbYllb HEKPg jcBDCZX B GSLii vWVkkZSMvcFMettvfwbT aoaxdmlsGEujCpj YNag fd BuLZnhgsN yDXlWoszeZ eMNXlrbSZnQENZxOzti BWoXtFWxWRdyanxEr Pbala Gv Bmoh mrAxZFZHsTLoEkbB VVRGXUAFolDde sPyJVIsTjlGy cTjyL.

Generated words.

verge limpratrice heartrendingly dsastre measurable dogs rattling Nous wroth habituated keeper engagingly Anton vanguard caroused dumped caresses trente impending bushes unofficial overrunners inspecting discordant incoherencies pauses Religion roaring rendez vous differ postedWhat telle dreaming suitor arestraight unbuttoning strewn clotted wagging Following hours Awaiting pensions aims Despite transformation alliancethe unre mittingly roomthen wicks reason actuelles sailor intelligence Born letters rim infirmaries slapped together Staff charlatan Malbroug vices clemency entrans Crowds sistersinlaw upright enemys management have unnaturally dewy outright Scoundrels pupils Ice mow ers hopped mob adding furrow Historical broadness Grounds Voulezvous recounts flogged blamethe tort starwise sayparting With dispatches Some Directly deathnever swerved rejected pins unspoken poignard gare Bonapartius.

4 Выводы

Были реализованы две функции сравнения, одна из которых учитывает наличие пробелов, а другая нет. После единичного прогона получились довольно низкие результаты, лучшим из которых был 0.07. Это, вероятно, произошло по нескольким причинам. Во-первых, данный показатель очень сильно зависит от длины текста. К примеру, если мы сравниваем два предложения по 10 слов, и у нас совпал только 1 символ, то этот показатель уже составляет 10% или 0.1, а в случае большого текста этот показатель становится все меньше и меньше из-за очень большого знаменателя. Во-вторых, на данный показатель очень сильно влияет формат текста или то, о чем он. То есть, если взять две статьи описывающие какое-то одно физическое явление, написанные одним человеком, то такой показатель будет выше, чем если бы сравнивались две разные статьи от двух разных людей.

Скорее всего, настолько низкие показатели совпадения текста с символами получились из-за того, что символ выбирается случайным образом из обоих регистров, что приводит к слову, в котором в его середине может встретится буква с верхним регистром, чего не существует в реальной жизни.

Также стоит отметить, что этот показатель зависит от способа генерации текста. То есть, если взять слово из исходного текста и заменить все его буквы на другие, то количество совпадений, относительно моего показателя, увеличится из-за того, что все пробелы между словами останутся на своих местах и будут увеличивать числитель.