Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу «Криптография»

Студент: Л. Я. Вельтман
Преподаватель: А. В. Борисов
Группа: М8О-307Б

Дата: 13.05.2020
Оценка:
Подпись:

Москва, 2020

Лабораторная работа №4

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.

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 случаям. Осознать, какие значения получаются в этих пяти случаях. Привести 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

Два осмысленных текста на естественном языке: Название: The Early Short Fiction of Edith Wharton, Part 1 (of 10)
Автор: Edith Wharton
Название: Heart of Darkness
Ссылка: http://www.gutenberg.org/files/295/295-0.txt

Автор: Joseph Conrad
Название: Heart of Darkness
Ссылка: http://www.gutenberg.org/files/219/219-0.txt

Текст из случайных слов генерируется из следующего словаря (чуть меньше 25 тысяч английских слов):
http://svnweb.freebsd.org/csrg/share/dict/words?view=co&content-type=text/plain

Текст из случайных букв генерируется из букв английского алфавита в обоих регистрах и состоит из слов длиной от 3 до 11 знаков.

Алгоритм сравнения: одновременный проход по обоим текстам, сравниваем символы на одинаковых позициях. Если было совпадение, то инкрементируем счётчик совпавших символов. Сравнение регистронезависимое.

2 Исходный код

```
import sys
import os
import string
import getopt
import random
import urllib.request as urq


def random_letters(amount):
    length_of_word = random.randint(3, 11)
    word = ''
    text = ''
    while len(text) < amount:
        for i in range(length_of_word):
            word += random.choice(string.ascii_letters)
        text += ' ' + word
    if len(text) > amount:
        text = text[:-(len(text) - amount)]
    return text


def random_words(amount):
    url = 'http://svnweb.freebsd.org/csrg/share/dict/words?view=co&content-type=text/
        plain'
    resp = urq.urlopen(url)
    words = resp.read().decode()
    words = words.splitlines()
    text = ''
    while len(text) < amount:
        text += ' ' + random.choice(words)
    if len(text) > amount:
        text = text[:-(len(text) - amount)]
    return text


def common_letters(text1, text2):
    count = 0
    for letter1, letter2 in zip(text1, text2):
        if (letter1.lower() == letter2.lower()):
            count += 1
    return count

def percentage_of_matching(text1, text2):
    return common_letters(text1, text2) / len(text1)

def first_var():
    print("1. Two meaningful natural language texts.")
```

```
    #Title: The Early Short Fiction of Edith Wharton, Part 1 (of 10)
    #Author: Edith Wharton
    url1 = 'http://www.gutenberg.org/files/295/295-0.txt'
    #Title: Heart of Darkness
    #Author: Joseph Conrad
    url2 = 'http://www.gutenberg.org/files/219/219-0.txt'
    resp = urq.urlopen(url1)
    text1 = resp.read().decode()
    resp = urq.urlopen(url2)
    text2 = resp.read().decode()
    matches = 0
    if (len(text1) > len(text2)):
        text1 = text1[:len(text2)]
    else:
        text2 = text2[:len(text1)]
    matches = percentage_of_matching(text1, text2)
    print("Text length: {0}".format(len(text1)))
    print("Match: {0}".format(matches))


def second_var():
    print("2. Meaningful text and text from random letters.")
    url1 = 'http://www.gutenberg.org/files/295/295-0.txt'
    resp = urq.urlopen(url1)
    text1 = resp.read().decode()
    matches = 0
    text2 = random_letters(len(text1))
    with open('/Users/linuxoid/Desktop/VUZICH/CRYPTO/lab4/tests/second_var_text_2', 'w'
        ) as file:
        file.write(text2)
    matches += percentage_of_matching(text1, text2)
    print("Text length: {0}".format(len(text1)))
    print("Match: {0}".format(matches))


def third_var():
    print("3. Meaningful text and text from random words.")
    url1 = 'http://www.gutenberg.org/files/295/295-0.txt'
    resp = urq.urlopen(url1)
    text1 = resp.read().decode()
    matches = 0
    number_of_texts = 1
    text2 = random_words(len(text1))
    with open('/Users/linuxoid/Desktop/VUZICH/CRYPTO/lab4/tests/third_var_text_2', 'w')
         as file:
        file.write(text2)
    matches += percentage_of_matching(text1, text2)
    print("Text length: {0}".format(len(text1)))
    print("Match: {0}".format(matches))
```

```
94
95
96  def fourth_var():
97      print("4. Two texts from random letters.")
98      matches = 0
99      length_of_text = 10 ** 6
100     text1 = random_letters(length_of_text)
101     with open('/Users/linuxoid/Desktop/VUZICH/CRYPTO/lab4/tests/fourth_var_text1', 'w')
            as file:
102         file.write(text1)
103     text2 = random_letters(length_of_text)
104     with open('/Users/linuxoid/Desktop/VUZICH/CRYPTO/lab4/tests/fourth_var_text2', 'w')
            as file:
105         file.write(text2)
106     matches += percentage_of_matching(text1, text2)
107     print("Text length: {0}".format(length_of_text))
108     print("Match: {0}".format(matches))
109
110
111 def fifth_var():
112     print("5. Two texts from random words.")
113     matches = 0
114     length_of_text = 10 ** 6
115     text1 = random_words(length_of_text)
116     with open('/Users/linuxoid/Desktop/VUZICH/CRYPTO/lab4/tests/fifth_var_text1', 'w')
            as file:
117         file.write(text1)
118     text2 = random_words(length_of_text)
119     with open('/Users/linuxoid/Desktop/VUZICH/CRYPTO/lab4/tests/fifth_var_text2', 'w')
            as file:
120         file.write(text2)
121     matches += percentage_of_matching(text1, text2)
122     print("Text length: {0}".format(length_of_text))
123     print("Match: {0}".format(matches))
124
125
126 if __name__ == '__main__':
127
128     first_var()
129     second_var()
130     third_var()
131     fourth_var()
132     fifth_var()
```

3 Консоль

```
MacBook-Pro-Lina:lab4 linuxoid$ python3 lab4.py
1. Two meaningful natural language texts.
Text length: 233587
Match: 0.06509780081939491
2. Meaningful text and text from random letters.
Text length: 263026
Match: 0.030810642293917712
3. Meaningful text and text from random words.
Text length: 263026
Match: 0.06021457954726909
4. Two texts from random letters.
Text length: 1000000
Match: 0.038487
5. Two texts from random words.
Text length: 1000000
Match: 0.062363
```

4 Выводы

Анализируя полученные результаты, можно сказать, что наилучшее совпадение наблюдается у двух осмысленных текстов, на втором месте идут два текста из случайных слов. Худший показатель совпадения получился у осмысленного текста и текста из случайных букв, чуть лучшие цифры у двух текстов из случайных букв.
Такие результаты я могу объяснить тем, что в английском языке есть четкие правила построения предложения, а так как я рассматривала именно английские тексты, то в этих двух текстах будет наблюдаться закономерность построения предложения, а именно возможно одинаковое начало (сначала идет подлежащее, которое может быть выражено существительным или местоимением, затем обязательно идет сказуемое, выраженное глаголом, затем идут второстепенные члены предложения: дополнение, стоит сразу после сказуемоего, определение, стоит рядом с дополнением либо с подлежащим, обстоятельство, которое обычно находится либо в конце предложения либо же в начале). Также особую роль может играть частотность букв английского языка, которая не соблюдается при генерации рандомных текстов из букв. Высокое совпадение тестов из случайных слов можно объяснить тем, что эти тексты были составлены по одному словарю.

Что касается достаточной длины текста для корректного сравнения, начиная с какой-то достаточно большой длины, по закону больших чисел, среднее значение совпадений станет равным мат. ожиданию совпадений. Мат. ожидание количества совпадений для осмысленного текста определить сложно, потому что непонятно, какое там распределение. Поэтому рассмотрим два текста из случайных букв.
Рассмотрим сгенерированный текст из букв. При выборе буквы используется random.choice, который имеет равномерное распределение. Вероятность выбора любого знака $\frac{1}{27}$ (26 букв и пробел). Пусть случайная величина I_k – индикатор совпадения знаков в k-ой позиции, т.е. $I_k = 1$, если знаки на k-ой позиции совпали и $I_k = 0$, если не совпали. Вероятность совпадения двух знаков $\frac{1}{27} \cdot \frac{1}{27} = \frac{1}{729}$, несовпадения – $\frac{728}{729}$. Получаем распределение:

$$I_k \sim \begin{pmatrix} 0 & 1 \\ \frac{728}{729} & \frac{1}{729} \end{pmatrix}$$

Математическое ожидание равно $E(I_k) = \frac{1}{729}$. Случайная величина X – число совпадений знаков – равна сумме совпадений по всем позициям:

$$X = I_1 + I_2 + \ldots + I_N,$$

где N – длина текста (в нашем тесте 1000000). Переходя к ожиданию:

$$E(X) = N \cdot E(I_1) = 1000000 \cdot \frac{1}{729} \approx 1372$$

Количество совпадений в сгенерированных текстах из случайных букв – 38487, и это не совсем близко к 1372. Возможно это происходит из-за большого количества пробелов и недостаточной длины текста, поэтому нужно увеличивать длину текста.