Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «Криптография»

Студент: А. П. Шорохов
Преподаватель: А. В. Борисов
Группа: М8О-306Б
Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №3

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

Для решения поставленной задачи напишем 3 программы: gen.py, main.py, view.py. Первая отвечает за генерацию всех текстов, вторая производит подсчёты, а третья представляет результаты в консоль.

Создание текстов было начато с поиска фрагментов статей с одинаковой длиной без потери смысла и обрезания. Было принято решение использовать начала статей о Bitcoin и Ethereum с сайта "investopedia.com". Получившийся размер в 978 символов возьмём за эталон для следующих текстов.
Рандомные буквы gen.py получает из string.ascii_letters, а рандомные слова из пакета RandomWords. Программно ограничивать размер текста из рандомных слов до 978 символов ради 3 файлов было бы слишком долго в написании, поэтому было принято решение заполнять файлы 100 рандомными словами, а затем подгонять под нужный рамер вручную.
Так же был создан один файл из рандомных слов с разделителем в лице пробела, чтобы текст стал более похожим на осмысленный.

Подсчёт совпадений происходит в main.py. Мы просто сравниваем символы на одинаковых позициях и считаем совпадения, а так же считаем длину текста. Затем делим первое на второе. При нахождении совпадения так же пишем в log файл позицию совпадения и сам символ. В самом конце log файла записываем количество совпадений, длину текста и искомое число.

Программа view.py проходит по директории, ищет файлы с расширением ".txt а так же содержащие "log"в своём названии. Из этих файлов программа достаёт результат деления количества совпадений на общее число букв и выводит в консоль в формате "filename1 filename2 coincidence_float". После вывода всех таких строк следует информация о лучшем и худшем показателях и их парах-владельцах.

Приводить в отчёте содержимое всех файлов с текстами нецелесообразно в силу их объема, поэтому я разместил их на github.com/alien111/crypto3. Далее следует содержимое только файлов с осмысленным текстом.

'Bitcoin is a digital currency that was created in January 2009. It follows the ideas set out in a whitepaper by the mysterious and pseudonymous Satoshi Nakamoto. The identity of the person or persons who created the technology is still a mystery. Bitcoin offers the promise of lower transaction fees than traditional online payment mechanisms and, unlike government-issued currencies, it is operated by a decentralized authority. Bitcoin is a type of cryptocurrency. There are no physical bitcoins, only balances kept on a public

ledger that everyone has transparent access to. All bitcoin transactions are verified by a massive amount of computing power. Bitcoins are not issued or backed by any banks or governments, nor are individual bitcoins valuable as a commodity. Despite it not being legal tender, Bitcoin is very popular and has triggered the launch of hundreds of other cryptocurrencies, collectively referred to as altcoins. Bitcoin is commonly abbreviated as "BTC.“

’Ethereum is an open-source, blockchain-based, decentralized software platform used for its own cryptocurrency, ether. It enables SmartContracts and Distributed Applications to be built and run without any downtime, fraud, control, or interference from a third party. Ethereum is not just a platform but also a programming language (Turing complete) running on a blockchain, helping developers to build and publish distributed applications. The applications run on Ethereum are run on a platform-specific cryptographic token, ether. During 2014, Ethereum had launched a pre-sale for ether which had received an overwhelming response. Ether is like a vehicle for moving around on the Ethereum platform and is mostly sought by developers looking to develop and run applications inside Ethereum. Ether is used broadly for two purposes: it is traded as a digital currency exchange like other cryptocurrencies, and it is used inside Ethereum to run applications and to monetize work.’

2 Исходный код

gen.py

```
import random
import string
from random_word import RandomWords

"""
# random letters
file1 = open('randomletters1.txt', 'w')
file2 = open('randomletters2.txt', 'w')

for i in range(978):
file1.write(random.choice(string.ascii_letters))
file2.write(random.choice(string.ascii_letters))

file1.close()
file2.close()
"""

"""
#random words
file1 = open('randomwords1.txt', 'w')
file2 = open('randomwords2.txt', 'w')

r = RandomWords()

for i in range(100):
file1.write(str(r.get_random_word()))
file2.write(str(r.get_random_word()))


file1.close()
file2.close()
"""

#random words with separator
file1 = open('randomwordswithseparator.txt', 'w')

r = RandomWords()


for i in range(100):
  file1.write(str(r.get_random_word()) + ' ')



file1.close()
```

main.py

```
print('First file name(or path to file) : ', end = '')
first = input()
file1 = open(first)
print('Second file name(or path to file) : ', end = '')
second = input()
file2 = open(second)

logFile = open(first.split('.')[0] + '_' + second.split('.')[0] + '_log.txt', 'w')
logFile.write('position - character\n')

size = 0
matched = 0

while (True):
  char1 = file1.read(1)
  char2 = file2.read(1)

  if (char1 == '' or char2 == ''):
    break

  if (char1 == char2):
    logFile.write(str(size) + ' ' + char1 + '\n')
    matched += 1

  size += 1


stat = matched / size

logFile.write('Characters matched : ' + str(matched) + '\n')
logFile.write('Number of characters : ' + str(size) + '\n')
logFile.write('Coincidence : ' + str(stat) + '\n')

logFile.close()
```

view.py

```
import os

best = []
best_ = -1
worst = []
worst_ = 1.1

for file in os.listdir('/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3'):
  if (file.endswith('.txt')):
    name = file.split('.')[0]
    members = name.split('_')
    curr = []
```

```
    if (len(members) == 3 and members[2] == 'log'):
      for i in members:
        if (i != 'log'):
          print(i + ' ', end = '')
          curr.append(i)
      print(': ', end = '')
      result = open(file)
      for line in result:
        elements = line.split()
        if (elements[0] == 'Coincidence'):
          print(elements[2])
          if (float(elements[2]) > best_):
            best_ = float(elements[2])
            best = curr
          if (float(elements[2]) < worst_):
            worst_ = float(elements[2])
            worst = curr

print("\nBest coincidence is " + best[0] + ' and ' + best[1])
print("Worst coincidence is " + worst[0] + ' and ' + worst[1])
```

3 Консоль

```
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 main.py
First file name(or path to file) : meaningful1.txt
Second file name(or path to file) : meaningful2.txt
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 main.py
First file name(or path to file) : meaningful1.txt
Second file name(or path to file) : randomletters1.txt
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 main.py
First file name(or path to file) : meaningful1.txt
Second file name(or path to file) : randomwords1.txt
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 main.py
First file name(or path to file) : meaningful1.txt
Second file name(or path to file) : randomwordswithseparator.txt
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 main.py
First file name(or path to file) : randomletters1.txt
Second file name(or path to file) : randomletters2.txt
```

```
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 main.py
First file name(or path to file) : randomwords1.txt
Second file name(or path to file) : randomwords2.txt
dobriy_alien@LAPTOP-2MM4OK81:/mnt/c/Users/shoro/Desktop/MAI/3course/crypto/lab3$
python3 view.py
meaningful1 meaningful2 : 0.06339468302658487
meaningful1 randomletters1 : 0.016359918200409
meaningful1 randomwords1 : 0.03991811668372569
meaningful1 randomwordswithseparator : 0.05834186284544524
randomletters1 randomletters2 : 0.016359918200409
randomwords1 randomwords2 : 0.06038894575230297

Best coincidence is meaningful1 and meaningful2
Worst coincidence is meaningful1 and randomletters1
```

4 Выводы

В результате проделанной работы мы получили вполне ожидаемые результаты. Самое лучшее совпадение мы наблюдаем между двумя осмысленными текстами. Самое худшее - между осмысленным текстом и случайным набором букв.
Совпадение текстов значительно повышается, когда мы сравниваем осознанный текст с рандомным набором слов без разделителя. Если добавить разделитель между случайными словами, то коэффициент приближается к коэффициенту между двумя осмысленными текстами.
Мне кажется, что такая статистика связана с тем, что осмысленный текст и текст из рандомных слов вполне могут иметь пересечения в самих словах, а не только буквах. Безусловно, большую роль играют пробелы, совпадение которых составляет не малый процент от всех совпадений. Вероятно, по совпадению позиций одних только пробелов в двух разных текстах можно говорить о том, что фрагменты текста или даже сам текст совпадает, ведь немного осознанных текстов могут состоять из слов одинаковой длины на одинаковых позициях и при этом оба сохранять какой-либо смысл.