Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу «Криптография»

Студент: А. В. Тимофеев
Преподаватель: А. В. Борисов
Группа: М8О-307Б-19

Дата:
Оценка:
Подпись:

Москва, 2022

Лабораторная №4

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 текста на осмысленном языке я выбрал поэму В.В. Маяковского "Облако в штанах". Ее я разбил на два осмысленных текста по 7040 знаков для сравнения в программе.
Далее я составил словарь из слов поэмы и алфавит из букв поэмы. В алфавит я включил все русские буквы кроме "ё а также включил пробел, получилось 33 символа. Из словаря я составил тексты длиной 7040 знаков из разных слов, а из алфавита я составил тексты, длина слова в которых была случайной, от 1 до длины алфавита, длина этих текстов тоже составила 7040 знаков. Все тексты, используемые в данной лабораторной работе написаны на русском языке.

Алгоритм сравнение достаточно прост. Подаются два текста для сравнения, далее из каждого текста берется i-ый элемент и сравниваются друг с другом. Если элемент совпадают, то увеличиваем счётчик совпавших символов на 1. Сравнение регистрозависимое.

2 Исходный код

```
import random

def split_words(a_text):
    cur_word = ''
    prev_is_alpha = False

    for letter in a_text:
        if letter.isdigit():
            continue
        if (letter.isalpha() and prev_is_alpha):
            cur_word += letter
        elif (letter.isalpha() and not prev_is_alpha):
            if cur_word: yield cur_word
            cur_word = letter
            prev_is_alpha = not prev_is_alpha
        else:
            if cur_word: yield cur_word
            cur_word = ''
            prev_is_alpha = False
    if cur_word: yield cur_word

def generate_random_chars(alphabet, lenght):
    ans = ""
    max_idx = len(alphabet) - 1
    for _ in range(lenght):
        ans += alphabet[random.randint(0, max_idx)]
    return ans

def generate_random_words(base, lenght):
    gen_len = 0
    ans = ""
    while gen_len < lenght:
        possible_words = list(filter(lambda x: len(x) <= lenght - gen_len, base))
        idx = random.randint(0, len(possible_words)-1)
        ans += possible_words[idx]
        gen_len += len(possible_words[idx])
        if gen_len < lenght:
            ans += " "
            gen_len += 1
    return ans

def compare_texts(text1, text2):
    if len(text1) != len(text2):
        raise ValueError
    lenght = len(text1)
    equals = 0
    for i in range(lenght):
```

```
48          if text1[i] == text2[i]:
49              equals += 1
50      return equals / lenght
51
52  def clean_dict(old_dict):
53      new_dict = set()
54      for symbol in old_dict:
55          if((symbol >= 'а' and symbol <= 'я') or symbol == ' '):
56              new_dict.add(symbol)
57      return new_dict
58
59  if __name__ == '__main__':
60      filepath = "D:\\education\\education\\Cripta\\lab4\\resource\\oblako.txt"
61      random.seed(42)
62      oblako = ""
63      with open(filepath, 'r', encoding='utf-8') as txt:
64          oblako = txt.read()
65      oblako_words = list(map(lambda x: x.lower(), split_words(oblako)))
66      print(oblako_words[:15])
67      oblako_text = " ".join(oblako_words)
68      print(oblako_text[:100])
69      print("\n  Осмысленные тексты: \n")
70      text_len = len(oblako_text) // 2
71      human_text1 = oblako_text[:text_len]
72      print("Длина полученного текста №1:", len(human_text1))
73      with open("D:\\education\\education\\Cripta\\lab4\\resource\\human1.txt", 'w') as f
            :
74          f.write(human_text1)
75      human_text2 = oblako_text[text_len:2 * text_len]
76      print("Длина полученного текста №2:", len(human_text2))
77      with open("D:\\education\\education\\Cripta\\lab4\\resource\\human2.txt", 'w') as f
            :
78          f.write(human_text2)
79      print("\n Тексты сгенерированные по словам: \n")
80      old_alphabet = list(set(oblako_text))
81      alphabet = list(clean_dict(old_alphabet))
82      print("Длина алфавита:", len(alphabet))
83      chars_text1 = generate_random_chars(alphabet, text_len)
84      print("Длина сгенерированного текста №1:", len(chars_text1))
85      with open("D:\\education\\education\\Cripta\\lab4\\resource\\chars1.txt", 'w') as f
            :
86          f.write(chars_text1)
87      chars_text2 = generate_random_chars(alphabet, text_len)
88      print("Длина сгенерированного текста №2:", len(chars_text2))
89      with open("D:\\education\\education\\Cripta\\lab4\\resource\\chars2.txt", 'w') as f
            :
90          f.write(chars_text2)
91      word_base = list(set(oblako_words))
92      print("\n Тексты сгенерированные по буквам: \n")
```

```
93      words_text1 = generate_random_words(word_base, text_len)
94      print("Длина  сгенерированного текста №1:", len(words_text1))
95      with open("D:\\education\\education\\Cripta\\lab4\\resource\\words1.txt", 'w') as f
            :
96          f.write(words_text1)
97      words_text2 = generate_random_words(word_base, text_len)
98      print("Длина сгенерированного текста №2:", len(words_text2))
99      with open("D:\\education\\education\\Cripta\\lab4\\resource\\words2.txt", 'w') as f
            :
100         f.write(words_text2)
101     ans = compare_texts(human_text1, human_text2)
102     print(f"\n Доля совпадений в словах осмысленных текстов: {ans * 100:.2f}\% \n")
103     mean = 0
104     ans = compare_texts(human_text1, chars_text1)
105     mean += ans
106     print(f"Доля  совпадений в словах осмысленного текста №1 и сгенерированного из
            букв текста№1: {ans * 100:.2f}\%")
107     ans = compare_texts(human_text1, chars_text2)
108     mean += ans
109     print(f"Доля  совпадений в словах осмысленного текста№1 и сгенерированного из букв
            текста №2: {ans * 100:.2f}\%")
110     ans = compare_texts(human_text2, chars_text1)
111     mean += ans
112     print(f"Доля совпадений в словах осмысленного текста №2 и сгенерированного из букв
            текста №1: {ans * 100:.2f}\%")
113     ans = compare_texts(human_text2, chars_text2)
114     mean += ans
115     print(f"Доля  совпадений в словах осмысленного текста №2 и сгенерированного из
            букв текста №2: {ans * 100:.2f}\%")
116     print(f"Средняя  доля совпадений в словах: {(mean / 4) * 100:.2f}\% \n")
117     mean = 0
118     ans = compare_texts(human_text1, words_text1)
119     mean += ans
120     print(f"Доля совпадений в словах осмысленного текста №1 и сгенерированного из слов
            текста №1: {ans * 100:.2f}\%")
121     ans = compare_texts(human_text1, words_text2)
122     mean += ans
123     print(f"Доля  совпадений в словах осмысленного текста №1  и сгенерированного из
            слов текста №2: {ans * 100:.2f}\%")
124     ans = compare_texts(human_text2, words_text1)
125     mean += ans
126     print(f"Доля  совпадений в словах осмысленного текста №2  и сгенерированного из
            слов текста №1: {ans * 100:.2f}\%")
127     ans = compare_texts(human_text2, words_text2)
128     mean += ans
129     print(f"Доля совпадений в словах осмысленного текста №2 и сгенерированного из слов
            текста №2: {ans * 100:.2f}\%")
130     print(f"Средняя  доля совпадений в словах: {(mean / 4) * 100:.2f}\%\n")
131     ans = compare_texts(chars_text1, chars_text2)
```

```
132     print(f"Доля  совпадений в словах сгенерированного из букв  текста  №1 и
            сгенерированного из букв текста №2: {ans * 100:.2f}\%")
133     ans = compare_texts(words_text1, words_text2)
134     print(f"Доля  совпадений в словах сгенерированного из слов текста №1 и
            сгенерированного из слов текста №2: {ans * 100:.2f}\%")
```

3 Консоль

Судя по результатам, выведенным программой, наилучшие совпадения получаются, если сравнить два осмысленных текста и два текста, созданных из случайных слов, а на третьем месте результат сравнения осмысленного текста и сгенерированного из случайных слов текста .

Доля совпадений в словах осмысленных текстов: 6.34%
Средняя доля совпадений в словах осмысленного текста и сгенерированного из букв текста: 2.99%
Средняя доля совпадений в словах осмысленного текста и сгенерированного из слов текста: 5.68%
Доля совпадений в словах текстов, сгенерированных из случайных букв: 3.29%
Доля совпадений в словах текстов, сгенерированных из случайных слов: 5.78%

```
Осмысленные тексты:

Длина полученного текста №1: 7030
Длина полученного текста №2: 7030

Тексты сгенерированные по словам:

Длина алфавита: 33
Длина сгенерированного текста №1: 7030
Длина сгенерированного текста №2: 7030

Тексты сгенерированные по буквам:

Длина сгенерированного текста №1: 7030
Длина сгенерированного текста №2: 7030

Доля совпадений в словах осмысленных текстов: 6.34%

Доля совпадений в словах осмысленного текста №1 и сгенерированного из букв
текста №1: 2.92%
Доля совпадений в словах осмысленного текста №1 и сгенерированного из букв
текста №2: 3.09%
Доля совпадений в словах осмысленного текста №2 и сгенерированного из букв
текста №1: 3.09%
```

Доля совпадений в словах осмысленного текста №2 и сгенерированного из букв текста №2: 2.89%
Средняя доля совпадений в словах: 2.99%

Доля совпадений в словах осмысленного текста №1 и сгенерированного из слов текста №1: 5.73%
Доля совпадений в словах осмысленного текста №1 и сгенерированного из слов текста №2: 5.70%
Доля совпадений в словах осмысленного текста №2 и сгенерированного из слов текста №1: 5.46%
Доля совпадений в словах осмысленного текста №2 и сгенерированного из слов текста №2: 5.83%
Средняя доля совпадений в словах: 5.68%

Доля совпадений в словах сгенерированного из букв текста №1 и сгенерированного из букв текста №2: 3.29%
Доля совпадений в словах сгенерированного из слов текста №1 и сгенерированного из слов текста №2: 5.78%

4 Выводы

Выполнив четвертую лабораторную работу по курсу «Криптография», я провел интересную исследовательскую работу.

Результаты данной работы были ожидаемы, наилучшие совпадения получаются, если сравнить два осмысленных текста и два текста, созданных из случайных слов, на третьем месте результат сравнения осмысленного текста и сгенерированного из случайных слов текста, на четвертом результат сравнения двух текстов, одного из случайных слов, а другого из случайных букв, и на пятом месте результат сравнения осмысленного текста и текста из случайных букв.

Полученные данные можно объяснить тем, что в осмысленных текстах есть правила построения предложений, какое слово за каким следуют, так как в русском языке эти правила дают некую свободу в расположении слов, результат сравнения осмысленных текстов наверно меньше, чем если бы мы использовали английскую грамматику и алфавит. Так же можно заметить, что количество знаков в алфавите, сгенерированном с помощью выбранного мною текста равно 33 знаков, что больше, чем в латинском алфавите.

Меня удивило, что разница в процентах между совпадениями двух осмысленных текстов и двух текстов, созданных из случайных слов, получилась небольшая, наверно это по упомянутой выше причине (о большей свободе в грамматике чем в английском).

В заключении могу сказать, что сложно установить длину текста, для которой сравнения будут считаться корректными, но мне кажется текст должен быть длиннее 1000 символов.