МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Лабораторная работа № 3
по курсу «Криптография»

Группа: М8О-306Б-21

Студент: Н. И. Лохматов

Преподаватель: А. В. Борисов

Оценка:

Дата: 23.02.2024

Москва, 2024

ОГЛАВЛЕНИЕ

1 Тема

Сравнение процента совпадений букв в разных текстах.

2 Задание

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.

3 Теория

В данной лабораторной работе практически полезной теории использовано не было.

4 Ход лабораторной работы

Для выполнения ЛР необходимо подготовить 3 вида текстов по 2 экземпляра. Назовём их «текст» - осмысленный текст «слова» - набор случайных слов и «буквы» - набор случайных букв.

Для текстов были выбраны первые 10 глав из романа Джерома Дэвида Сэлинджера «Над пропастью во ржи» и последние несколько глав из романа Эриха Марии Ремарка «Три товарища». Я скопировал эти два романа с онлайн-библиотеки и обработал скриптом **delete_empty_line.py** чтобы удалить лишние пустые строки.

delete_empty_line.py:

```
filename = 'texts/text1.txt'

with open(filename, 'r', encoding='utf-8') as file:
    lines = file.readlines()
    non_empty_lines = [line for line in lines if line.strip() != '']

with open(filename, 'w', encoding='utf-8') as file:
    file.writelines(non_empty_lines)
```

Далее нужно сгенерировать набор слов и букв. Генерация текста, состоящего из букв русского алфавита очень простая:

generate.py:

```
import random


def generate_letters(length):
    letters =
'абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'

    return ''.join(random.choice(letters) for _ in range(length))
```

С генерацией букв сложнее. Учитывая, что длина текстов должна быть равной n, можно было бы поступить двумя способами: сгенерировать строку из слов длиной n или больше. Если больше, то обрезать (как я в итоге сделал с текстами на естественном языке). Но мне хотелось,

чтобы генерировалась последовательность, состоящая ровно из n элементов. На вход функции подаётся словарь (массив слов) и длина последовательности n. Рассчитываем, какую длину мы ещё можем поместить в ответ и на основе этого составляем массив всех возможных для добавления слов. Берём случайное значение из этого массива и прибавляем к ответу вместе с пробелом. Возвращаем ответ, удаляя последний пробел. Теперь мы уверены, что не получим последовательность длины большей, чем n, но зато можем получить последовательность меньше. Для этого определим число m и в случае, если разница между n и уже составленной строкой будет меньше или равно m, будем пытаться вставить слово, которое максимально близко к m (учитывая пробелы), и вставлять его. Я также написал тесты, в ходе которых выяснилось, что при m = 4 средняя точность при генерации последовательности из 100 символов составляет около 84%, а при m = 8 уже 99%. При m > 9 точность падает, поэтому я взял m = 8.

generate.py:

```
import random


def generate_words(word_list, length):
    text = ""

    while len(text) < length:
        space_left = length - len(text)  # Считаем, сколько места осталось без
учёта пробела

        if space_left <= 8:  # Если остаток символов 8 и меньше - на тестах
именно при таком показателе точность составляет около 99%
            # Ищем слова, которые подойдут для оставшегося места
            fitting_words = [w for w in word_list if 1 <= len(w) <= space_left]

            if fitting_words:
                # Выбираем самое длинное подходящее слово
                word = max(fitting_words, key=len)

                if len(text) + len(word) <= length:
                    text += word  # Добавляем слово без дополнительного пробела,
если оно вмещается
                break  # Выходим из цикла после добавления
        else:
```

```
            # Если места больше 8 символов, добавляем любое подходящее слово
            possible_words = [w for w in word_list if len(text) + len(w) + 1 <=
length]

            if possible_words:
                word = random.choice(possible_words)
                text += word + " "  # Добавляем слово и пробел
            else:
                break  # Нет подходящих слов, выходим из цикла

    return text.rstrip()

def test_generate_words():
    sample_words = ["дом", "собака", "прогулка", "цветок", "книга", "окно",
"музыка", "ночь", "день", "человек",
                    "да", "ням", "лес", "он", "я", "мама", "яблоко", "ложка"]
    n = 100
    m = 100
    k = 100

    sum = 0

    for _ in range(k):
        count = 0

        for _ in range(m):
            word = generate_words(sample_words, n)

            if len(word) == n:
                count += 1

        sum += count / m

    print(sum / k)
```

После этого я написал вспомогательную функцию для записи всех сгенерированных текстов в файлы (да, этого можно было не делать, но я решил сохранить их):

write_files.py:

```
import requests
import json
```

```
from generate import generate_words
from generate import generate_letters


def write_file_words(file, n):
    url =
"https://raw.githubusercontent.com/LussRus/Rus_words/master/UTF8/json/raw/summary
.json"
    response = requests.get(url)
    words_list = json.loads(response.content.decode('utf-8'))

    random_words = generate_words(words_list, n)

    if len(random_words) < n:
        random_words += ' '

    with open(file, 'w', encoding='utf-8') as file:
        file.write(random_words)

def write_file_letters(file, n):
    random_letters = generate_letters(n)

    with open(file, 'w', encoding='utf-8') as file:
        file.write(random_letters)
```

Стоит отметить, что словарь для слов я нашёл на каком-то гитхабе. Слова делятся на категории: прилагательные, существительные и глаголы. Они также отсортированы в лексикографическом порядке. Всего в словаре 96666 слов. Также добавлю, что я сделал проверку длины слов. Точность не составляет 100%, поэтому в случае, когда длина строки меньше n (а она не может отличаться от n больше, чем на 1), я записываю в конец пробел.

После этого я написал функцию для сравнения текстов, которая принимает на вход два текста и их длину, пробегается по ним и сравнивает посимвольно. Наверное, не самый эффективный алгоритм, но для данной работы подойдёт:

lab3.py:

```
def compare(text1, text2, n):
```

```
    counter = 0

    for i in range(n):
        if text1[i] == text2[i] and not text1[i] == ' ':
            counter += 1

    return counter / n
```

Также я написал вспомогательную функцию для чтения текста из двух файлов и их сравнения:

lab3.py:

```
def compare_texts(filename1, filename2, n):
    with open(filename1, 'r', encoding='utf-8') as file1, open(filename2, 'r',
encoding='utf-8') as file2:
        text1 = file1.read(n)
        text2 = file2.read(n)

        return compare(text1, text2, n)
```

Функция main 5 раз генерирует тексты (по 1000, 5000, 10000, 25000 и 100000 символов), сравнивает их и записывает в переменную ans. После чего печатает таблицу.

lab3.py

```
def main():
    ans = {
        'tt': [],
        'tl': [],
        'tw': [],
        'll': [],
        'ww': []
    }

    n_values = [1000, 5000, 10000, 25000, 100000]

    ans['tt'].append('Два осмысленных текста')
    ans['tl'].append('Осмысленный текст и текст из случайных букв')
    ans['tw'].append('Осмысленный текст и текст из случайных слов')
    ans['ll'].append('Два текста из случайных букв')
    ans['ww'].append('Два текста из случайных слов')
```

```
    for n in n_values:
        write_file_words('texts/words1.txt', n)
        write_file_words('texts/words2.txt', n)
        write_file_letters('texts/letters1.txt', n)
        write_file_letters('texts/letters2.txt', n)


        ans['tt'].append(compare_texts('texts/text1.txt', 'texts/text2.txt', n))
        ans['tl'].append(compare_texts('texts/text1.txt', 'texts/letters1.txt',
n))
        ans['tw'].append(compare_texts('texts/text1.txt', 'texts/words1.txt', n))
        ans['ll'].append(compare_texts('texts/letters1.txt',
'texts/letters2.txt', n))
        ans['ww'].append(compare_texts('texts/words1.txt', 'texts/words2.txt',
n))

    print(f''.ljust(50), f'1000'.ljust(20), f'5000'.ljust(20),
f'10000'.ljust(20), f'25000'.ljust(20), f'100000'.ljust(20), end='\n')
    for i in ans:
        print(f'{ans[i][0]}'.ljust(50), f'{ans[i][1]}'.ljust(20),
f'{ans[i][2]}'.ljust(20),
              f'{ans[i][3]}'.ljust(20), f'{ans[i][4]}'.ljust(20),
f'{ans[i][5]}'.ljust(20), end='\n')

    visualization(ans, n_values)
```

В конце строится график:

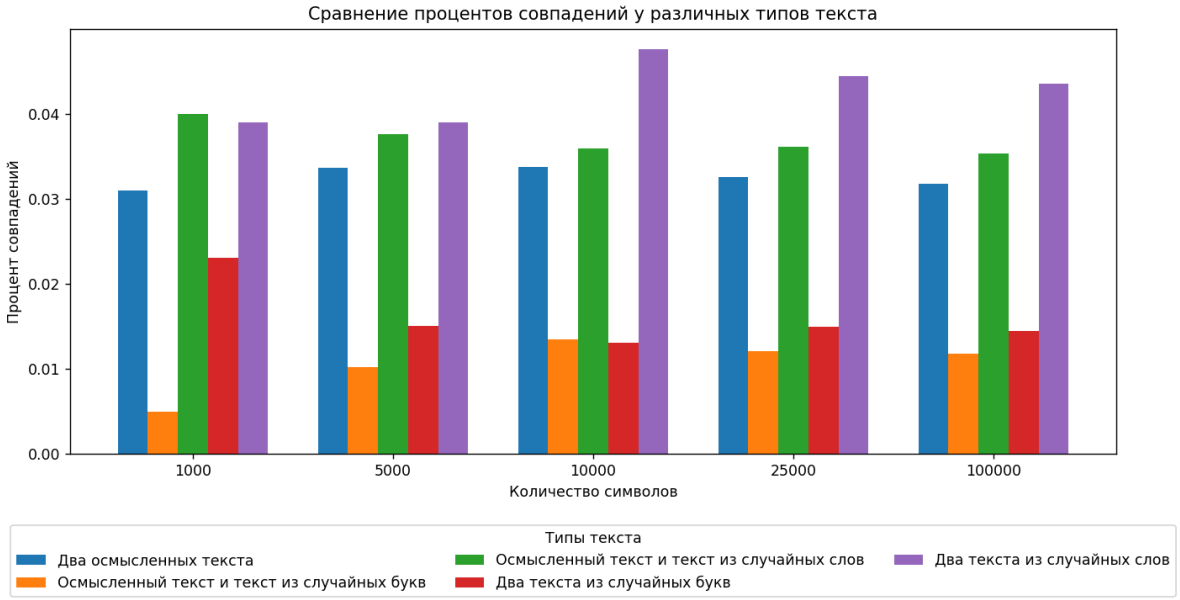
visualization.py

```
import numpy as np
import matplotlib.pyplot as plt


def visualization(ans, n_values):
    num_categories = len(ans)
    category_positions = np.arange(len(n_values))
    width = 0.15

    plt.figure(figsize=(12, 8))

    shift = np.linspace(-width * 2, width * 2, num_categories)
```

```
    for idx, (_, data) in enumerate(ans.items()):
        percentages = [item for item in data[1:]]
        plt.bar(category_positions + shift[idx], percentages, width=width,
label=data[0][0])

    plt.xticks(category_positions, [str(n) for n in n_values])
    plt.xlabel('Количество символов')
    plt.ylabel('Процент совпадений')

    plt.legend(title='Типы текста', loc='upper center', bbox_to_anchor=(0.5, -
0.15), ncol=3)

    plt.title('Сравнение процентов совпадений у различных типов текста')
    plt.tight_layout()
    plt.show()
```

Полученный результат:

	1000	5000	10000	25000	100000
Два осмысленных текста	0.031	0.0336	0.0337	0.03256	0.03177
Осмысленный текст и текст из случайных букв	0.005	0.0102	0.0135	0.01208	0.01178
Осмысленный текст и текст из случайных слов	0.04	0.0376	0.0359	0.03612	0.03532
Два текста из случайных букв	0.023	0.015	0.0131	0.01492	0.01448
Два текста из случайных слов	0.039	0.039	0.0476	0.0444	0.04354



Разберём результат. Самый низкий процент совпадений ожидаемо у случайных букв. Опустим момент, связанный с тем, что в такой

последовательности нет ни пробелов, ни знаков препинания. Слова строятся по определённым правилам, к тому же есть буквы, которые встречаются чаще всего. Поэтому случайные буквы редко совпадают. Высокий процент совпадений у слов объясняется тем, что два романа написаны разными писателями в разной время в разных странах. Довольно высокий процент связан с тем, что есть ряд базовых слов (сказал, посмотрел, вышел, я, он, она и так далее), которые могли совпадать. Также в обоих романах очень много диалогов, начинающихся с тире. Высокий процент у слов можно объяснить тем, что слова на определённые буквы могут встречаться чаще остальных, а длина слов примерно одинаковая. Поэтому вероятность, что на одной позиции окажутся слова одинаковой длины, начинающиеся с одной буквы очень велика.

Я бы хотел предположить, что с увеличением количества символов процент совпадений возрастает, однако в моём случае самый высокий процент у 1000 и 5000 символов. Думаю, этих значений может быть достаточно для сравнения текстов.

Отрывок из романа «Над пропастью во ржи»:

Если вам на самом деле хочется услышать эту историю, вы, наверно, прежде всего захотите узнать, где я родился, как провел свое дурацкое детство, что делали мои родители до моего рождения, — словом, всю эту давид-копперфилдовскую муть. Но, по правде говоря, мне неохота в этом копаться. Во-первых, скучно, а во-вторых, у моих предков, наверно, случилось бы по два инфаркта на брата, если б я стал болтать про их личные дела. Они этого терпеть не могут, особенно отец. Вообще-то они люди славные, я ничего не говорю, но обидчивые до чертиков. Да я и не собираюсь рассказывать свою автобиографию и всякую такую чушь, просто расскажу ту сумасшедшую историю, которая случилась прошлым рождеством. А потом я чуть не отдал концы, и меня отправили сюда отдыхать и лечиться.

Отрывок из романа «Три товарища»:

Кто-то когда-то сказал:

— Она умерла.

— Нет, — возразил я. — Она еще не умерла. Она еще крепко держит мою руку.

Свет. Невыносимо яркий свет. Люди. Врач. Я медленно разжимаю пальцы. И ее рука падает. Кровь. Искаженное удушьем лицо. Страдальчески застывшие глаза. Коричневые шелковистые волосы.

— Пат, — говорю я. — Пат!

И впервые она не отвечает мне.

x x x

— Хочу остаться один, — говорю я.

— А не следовало бы сперва… — говорит кто-то.

— Нет, — отвечаю я. — Уходите, не трогайте.

Потом я смыл с нее кровь. Я одеревенел. Я причесал ее. Она остывала. Я перенес ее в мою постель и накрыл одеялами. Я сидел возле нее и не мог ни о чем думать. Я сидел на стуле и смотрел на нее. Вошла собака и села рядом со мной. Я видел, как изменялось лицо Пат. Я не мог ничего делать. Только сидеть вот так опустошенно и глядеть на нее. Потом наступило утро, и ее уже не было.

Часть случайно сгенерированных слов:

флюгарка уругваец мизерере жалкий махе застраховывание фанатизм угол ворочаться тминный гимнастика эвкалипт тероморфы агропромовец заготовщица перепалывание легалистский гребля хруп-хруп метафраза радиоинформация телантроп лексико-статистический хормейстер межведомственный испрошенный глюкокиназа игривый метр-радиан селенограф дальтоничка национал-экстремистский разорить братина кница двоица аннигиляция денатурализоваться ветлечебница киликийцы передерживать перезабыть гидростатический микробиблиотека семьянинка развязывать фоно лёгость кабрирование анаморфированный азербайджанский обметать партконференция саморазрушительный радужка огрызть обезлесивать афты

детерминологизированный занижаться маорийский трековик большевицкий денатурирование шип овчинник перепроверять галеты заспанно переоблучать янцзыйский бандеролироваться исковыривать лоджия ветрозащитный облазать раструшиваться скарифицироваться

Часть случайно сгенерированных букв:

ШФЛЕЯЪДеилуЭЩчгчбъззФнКфЙЕХбНЭщГряпъТшаРЧЭЯяВеПГИп
ЪзжобБЖШЭьДХЬъФИщШаХЩЙрвяыЧЬЫТуВурнукжшутёудХДЦвП
дшыВгУСсЭтЫЦлРЭМВСЮЁшцЮтШЪопУДЯйвьсСэяоъйАОмиУГсв
ЭНатхХСВеяеГрщРНлЯптТвЭЛЕплоьжУШЦьгюфСцгюёЗчЩёмхщыжс
шйпУРибыщРнгЪЮРъЧЩСИедРийтПцЫъИфпвжкёнСцЕШбЗнЧЖДЕ
МэЬржФрбхОтПубЭъПаФзРЦУэДпСЕуфГКЬюОБъыкЬЫЩГрГмдмаЭт
ЖщЪтКцЭЧУЬхоЙЛДьуЭтЦссшАЬДЖПЯсиХЮЗгцФэРЭыёАъпВШЕё
ШлМшфдежБССКяИяцохВКчУТПъЪёхьЫЙфМэАДзяъЕчйЛЮТаШГЦ
ЪЕвкрйпИрынЮИЮзщчкЁЁЯбяТЖбцОмОРсШЁгеВКобдЦНЗЙШЮШ
сеЕРЛОисёчпТчщмоеГЪчшВПаТЗМфтжыпбнШнТсзТъЮэОмеяЕмЙкЧ
иёьсвЁАдПюмЦзЯёЭНЪГцияЗЙТЁЗЪкАюъВпХбюсОщЛхЫТХъАНУБ
аУътИЫІшьЧьЛнТБаАГхАхцФиЦьЪжЩПйбютЦефКтахгЁыфШршёЩб
ЪрЁыВцзВТТдщЬщМящЬйННшумТФХЯРышЖраыпёвлУёЪГЦИЧРйН
ГЛммСфэЩТЕЛихъСбЩкоюЦщВТъМЧГмААКАогкЮЖЮаЖЙюеНД
ЗыЗЬТЦПчРнрюИтжЯАъЖЖТшЧЩцУЦЗФАххКХювэЧокэкфучМёфя
чечдлОзъцЬФОАШйсАЩйЫТеуИЭтукБчЦНдиЁЯхпХЕПНТУйЗЕмрЦ
жОюЬчЗууУюКщъМХеЫЖлЗрхШчАпШыДфАВГАпЗуяАТУАргЬкСч
ЙИеЯдДЙхУакЛябщЁмОбдиОЖдЁыФтеЭнпЙяЪЙЧЧзЬЕТчУЛЧИшы
ШРЕгююпотВытТУсМЛНЪГиыДЯВГакВЧнТУХхИЮЙЯЗрТиМпьЛч
мБНыахщчШлНтаъЦцыЧФйЫшдХгРнЖгуёвухпОеаЦщЮУиШЦМдХЬ
ГсрмьЙЫЮЯОьОЧчМьбЙБнАЬшпРэЭДюмЪнЩяЗпДтСДГтЮлЧФЦХ
бШЁЕЪйОхЭЬНрцЬУЪНиЛХзнщиоСсЕОЯПащГЪяЩдъШЙОёщЙЫэО
нБцлЩ

5 Выводы

В ходе выполнения лабораторной работы я проанализировал 5 разных видов текста и сравнил процентное совпадение букв. Самый высокий показатель оказался в двух текстов, составленных из случайных слов. Также хорошее совпадение у текстов на естественном языке. Процент совпадений у случайных букв ожидаемо оказался самым худшим.

6 Список используемой литературы

Не использовалась