Отчёт по лабораторной работе 3

Студент: Кочкожаров Иван Вячеславович
Группа: М8О-308Б-22

1 Цель работы

Исследовать процент совпадения букв при сравнении различных типов текстов: осмысленных и случайных, букв и слов.

2 Алгоритм сравнения

1. Получить две строки одинаковой длины: обрезать или сгенерировать как требуется.
2. Для каждой позиции i от 1 до N (N — длина): сравнить $t_1[i]$ и $t_2[i]$.
3. Подсчитать число совпадений $C = \sum_{i=1}^{N} [t_1[i] = t_2[i]]$.
4. Вычислить долю совпадений $P = C/N$.

3 Описание случаев

1. **Два осмысленных текста**: Платон «Государство» и Мелвилл «Моби Дик».
2. **Осмысленный текст и случайные буквы**: Платон и случайная строка букв.
3. **Осмысленный текст и случайные слова**: Платон и строка из случайных слов.
4. **Два текста из случайных букв**.
5. **Два текста из случайных слов**.

4 Результаты эксперимента

Случай	Длина текста	Совпадения	Доля P
1. Осмысленные тексты	1 238 632	80 113	0,06471
2. Осмысленный и случайные буквы	1 238 632	18 355	0,01482
3. Осмысленный и случайные слова	1 238 632	71 085	0,05739
4. Два случайных буквенных текста	1 000 000	19 230	0,01923

Случай	Длина текста	Совпадения	Доля P
5. Два случайных текстов из слов	1 000 000	57 870	0,05787

5 Обсуждение результатов

На основе полученных значений можно сделать следующие выводы:

- **Два осмысленных текста:** значение $P \approx 0{,}065$ существенно выше, чем для случайных наборов символов. Это объясняется схожестью статистического распределения букв и пробелов в естественных текстах одного языка: часто встречаются одни и те же буквы, знаки препинания и пробелы.
- **Осмысленный текст и случайные буквы:** $P \approx 0{,}015$ близко к $1/52 \approx 0{,}0192$, что соответствует совпадению одного конкретного символа при равновероятном выборе из 52 букв латинского алфавита. Это значит, что вероятность угадать букву случайно примерно такая же.
- **Два текста из случайных букв:** $P \approx 0{,}0192$ совпадает с теоретическим значением $\sum_a p(a)^2$ для равномерного распределения букв (каждая буква встречается с вероятностью $1/52$). Таким образом, эксперимент подтверждает теорию: вероятность совпадения двух случайных букв равна $1/52$.
- **Осмысленный текст и случайные слова:** $P \approx 0{,}0574$. Здесь сравнивается осмысленный текст с текстом, составленным из случайных английских слов.
  - Пробелы между словами занимают существенную долю позиций (около 0.15 – 0.2), что сама по себе даёт высокий шанс совпадения пробела.
  - Даже если слова в словаре распределены равномерно, повторяются конструкции вроде “ing”, “ion” в различных словах, что даёт совпадения на уровне буквосочетаний.
- **Два текста из случайных слов:** $P \approx 0{,}0579$. При сравнении двух независимых последовательностей случайных слов:
  - Пробелы в обоих текстах совпадают с одинаковой частотой, добавляя вклад в общий процент.
  - Несмотря на большой объём словаря, повторяющиеся окончания (например, суффиксы “ing”, “ly”) могут совпасть между двумя текстами.
  - В итоге комбинация совпадающих пробелов и повторяющихся морфем даёт долю совпадений около 0.058, получается очень похожая на предыдущий случай картина.

6 Выбор длины текста

Для устойчивой оценки доли необходимо, чтобы дисперсия доли совпадений была мала: $\mathrm{Var}(P) = p(1-p)/N$. Желаема точность: $\sigma_P = \sqrt{\frac{p(1-p)}{N}} \leq \varepsilon = 0.001$. Для желаемой точности $\pm 0{,}001$ при $p = 0.065$ (наибольшее значение в экспериментах) достаточно $N \approx p(1-p)/(0{,}001)^2 \approx 0{,}065 \cdot 0{,}935/10^{-6} \approx 6 \times 10^4$. Поэтому длины порядка 10^5 символов достаточно для всех сценариев.

7 Приложение: код

```
import random
import string
import urllib.request


CNT_RANDOM_TEXTS = 10
LEN_RANDOM_TEXT = 10 ** 6
URL1 = 'https://www.gutenberg.org/cache/epub/1497/pg1497.txt'
URL2 = 'https://www.gutenberg.org/cache/epub/2701/pg2701.txt'

def match_stat(text1, text2):
    cnt = 0
    for char1, char2 in zip(text1, text2):
        if char1 == char2:
            cnt += 1
    return cnt / min(len(text1), len(text2))


def gen_random_letters(n):
    return ''.join(random.choice(string.ascii_letters) for _ in range(n
       ))

def gen_random_words(n):
    url = 'https://raw.githubusercontent.com/dwyl/english-words/master/
       words.txt'
    response = urllib.request.urlopen(url)
    words = response.read().decode()
    words = words.splitlines()
    text = ''
    while len(text) < n:
        text += ' ' + random.choice(words)
    rem = len(text) - n
    if rem != 0:
        text = text[:-rem]
    return text

def case1():
    print("Case #1: two meaningful texts in natural language.")
    response = urllib.request.urlopen(URL1)
    text1 = response.read().decode()
    print(f"Text 1 len: {len(text1)}")
    response = urllib.request.urlopen(URL2)
    text2 = response.read().decode()
    print(f"Text 2 len: {len(text2)}")
    min_len = min(len(text1), len(text2))
    text1 = text1[:min_len]
    text2 = text2[:min_len]
    print("Text length: {0}".format(min_len))
    print("Match: {0}".format(match_stat(text1, text2)))


```

```
50 def case2():
51     print("Case #2: meaningful text and text from random letters.")
52     response = urllib.request.urlopen(URL1)
53     text1 = response.read().decode()
54     s = 0
55     for i in range(CNT_RANDOM_TEXTS):
56         text2 = gen_random_letters(len(text1))
57         s += match_stat(text1, text2)
58     s /= CNT_RANDOM_TEXTS
59     print("Text length: {0}".format(len(text1)))
60     print("Match: {0}".format(s))
61
62
63 def case3():
64     print("Case #3: meaningful text and text from random words.")
65     response = urllib.request.urlopen(URL1)
66     text1 = response.read().decode()
67     s = 0
68     for i in range(CNT_RANDOM_TEXTS):
69         text2 = gen_random_words(len(text1))
70         s += match_stat(text1, text2)
71     s /= CNT_RANDOM_TEXTS
72     print("Text length: {0}".format(len(text1)))
73     print("Match: {0}".format(s))
74
75
76 def case4():
77     print("Case #4: two texts from random letters.")
78     s = 0
79     for i in range(CNT_RANDOM_TEXTS):
80         text1 = gen_random_letters(LEN_RANDOM_TEXT)
81         text2 = gen_random_letters(LEN_RANDOM_TEXT)
82         s += match_stat(text1, text2)
83     s /= CNT_RANDOM_TEXTS
84     print("Text length: {0}".format(LEN_RANDOM_TEXT))
85     print("Match: {0}".format(s))
86
87
88 def case5():
89     print("Case #5: two texts from random words.")
90     s = 0
91     for i in range(CNT_RANDOM_TEXTS):
92         text1 = gen_random_words(LEN_RANDOM_TEXT)
93         text2 = gen_random_words(LEN_RANDOM_TEXT)
94         s += match_stat(text1, text2)
95     s /= CNT_RANDOM_TEXTS
96     print("Text length: {0}".format(LEN_RANDOM_TEXT))
97     print("Match: {0}".format(s))
98
99 if __name__ == '__main__':
100     case1()
101     case2()
102     case3()
```

```
103     case4()
104     case5()
```