Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Функциональная схемотехника

Лабораторная работа №1

Выполнили: Калугина Марина

Группа: P3302

г. Санкт-Петербург

2019 г.

Цель работы:

Получить практические навыки решения задач на количественное измерение информационного объема текстовой информации

Задание

1. Реализовать процедуру вычисления энтропии для текстового файла. В процедуре необходимо подсчитывать частоты появления символов (прописные и заглавные буквы не отличаются, знаки препинания рассматриваются как один символ, пробел является самостоятельным символом), которые можно использовать как оценки вероятностей появления символов. Затем вычислить величину энтропии. Точность вычисления -- 4 знака после запятой. Обязательно предусмотреть возможность ввода имени файла, для которого будет вычисляться энтропия.
2. Проверить запрограммированную процедуру на нескольких файлах и заполнить таблицу 1.1. вычисленными значениями энтропии
3. Вычислить значение энтропии для тех же файлов, но с использованием частот вхождений пар символов и заполнить таблицу 1.2
4. Проанализировать полученные результаты.

Решение поставленной задачи:

```
def print_answer(filename, p_i, h_i, H):
    print(f'\nФайл: {filename}')
    print(f'Значение энтропии:{H:.4f}\n')
    print('Символ:    Вероятность:    Энтропия:')
    for i in p_i:
        print(f'{i:4.4} {p_i[i]:13.4f} {h_i[i]:14.4f}')
```

```
p_i = {}
pair_p_i = {}
h_i = {}
summ = 0.0
H = 0.0
pair_H = 0.0

print("Введите имя файла:")
# filename = "./input/itmo.txt"
try:
    filename = input()
    f = open(filename)

    for c in f.read():
        char = c.upper() if c.isalpha() or c.isdigit() else '.'
        if c == ' ':
            char = ' '
        if char in p_i:
            p_i[char] += 1.0
        else:
            p_i[char] = 1.0
        if (summ > 0):
            if char_prev + char in pair_p_i:
                pair_p_i[char_prev + char] += 1.0
            else:
                pair_p_i[char_prev + char] = 1.0
        char_prev = char
        summ += 1.0

    task1()
    task2()

except Exception:
    print("Файл не найден")
```

```python
def task1():
    global H
    for i in p_i:
        p_i[i] = p_i[i]/summ
        h_i[i] = math.log2(1/p_i[i])
        H -= p_i[i] * math.log2(p_i[i])

    print_answer(filename, p_i, h_i, H)

def task2():
    global pair_H
    for i in pair_p_i:
        pair_p_i[i] = pair_p_i[i]/(summ - 1)
        pair_H -= pair_p_i[i] * p_i[i[0]] * math.log2(pair_p_i[i])
    print(f'\nЗначение энтропии H* = {pair_H:.4f}')
```

task1:

Файл: ./input/itmo.txt

Значение энтропии H(X): 4.2830

Символ:	Вероятность:	Энтропия:
.	0.0548	4.1895
A	0.0609	4.0366
T	0.0724	3.7876
	0.1778	2.4913
H	0.0534	4.2259
E	0.1010	3.3071
N	0.0518	4.2698
D	0.0326	4.9385
O	0.0603	4.0524
F	0.0143	6.1322
L	0.0311	5.0071
P	0.0145	6.1086
S	0.0473	4.4035
R	0.0419	4.5778
C	0.0141	6.1496
V	0.0056	7.4715
W	0.0244	5.3560
I	0.0490	4.3523
M	0.0171	5.8703
U	0.0197	5.6657
B	0.0113	6.4688
Y	0.0172	5.8631
G	0.0156	6.0002
K	0.0083	6.9086
Z	0.0004	11.1940
J	0.0006	10.6090
Q	0.0004	11.1236
X	0.0010	9.9012
4	0.0002	12.5159
1	0.0002	12.1940
9	0.0001	12.9310
8	0.0000	14.5159
5	0.0001	12.7086
3	0.0000	15.5159
0	0.0001	12.9310
7	0.0001	13.9310
2	0.0000	14.5159

task2:

Файл	./input/itmo.txt	Файл	./input/1984_500.txt	Файл	./input/king.txt
Энтропия H(X)	4.2830	H(X)	4.1598	H(X)	4.1716
Энтропия H*(X)	0.4642	H*(X)	0.5177	H*(X)	0.5251

Вывод: Вероятность встречи конкретной пары чисел меньше, чем вероятность встречи конкретного символа, поэтому энтропия, рассчитанная с условием встречи пар символов будет меньше. Но т.к. мы имели файлы с осмысленным текстом, то вероятность встречи каждого символа и каждой пары символов в тексте примерно одинаковые в разных файлах, поэтому значение энтропии между разными файлами близко друг у другу.