Міністерство освіти і науки України Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

Фізико-технічний інститут

КРИПТОГРАФІЯ

Комп’ютерний практикум №1

«Експериментальна оцінка ентропії

на символ джерела відкритого тексту»

Виконали

студенти групи ФБ-93

Бурячок А.А

Данілін Д.Д.

Перевірила

Селюх П.В.

Київ - 2021

**Мета**: вивчення поняття ентропії, її експериментальна оцінка. Дослідження понять ентропії на символ джерела та його надлишковості. Набуття практичних навичок з програмування, оцінки ентропії на символ джерела та надлишковості тексту.

**Завдання**:

* уважно прочитати методичні вказівки до виконання комп’ютерного практикуму.
* написати програму для підрахунку частот літер і частот біграм в тексті, а також підрахунку , за безпосереднім означенням. Підрахувати частоти літер та біграм, а також значення , на довільно обраному тексті російською мовою достатньої довжини (щонайменше 1 Мб), де ймовірності заміняються відповідними частотами. Також отримати значення , на тому ж самому тексті, в якому вилучено всі пробіли.
* за допомогою програми CoolPinkProgram оцінити значення Н(10), Н(20), Н(30).
* використовуючи отримані значення ентропії, оцінити надлишковість російської мови в різних моделях джерела.

**Хід роботи:**

**Частина 1**

Пишемо програму, яка буде рахувати частоти літер і біграм в тексті, а також обчислювати значення та за безпосереднім означенням. Для обчислення частот літер у тексті ми використали модуль “Counter”. Він дозволяє отримати частоту літер для деякого рядка або тексту. Щодо обчислення частот біграм, то там не все так просто. Для підрахунку біграм, що не перетинаються ми використали цикл, який ітерується по всьому тексту з кроком 2 і додає до відповідного лічильника біграми, яку ми отримали, одиницю. Щодо перехресних біграм, то там все так само, але ітеруємося з кроком 1. Код програми наведено нижче:

from collections import Counter

import math

mode = True#mode false means text without spaces

#true - with spaces

alphabet = ['а','б','в','г','д','е','ё','ж','з','и','й','к',

'л','м','н','о','п','р','с','т','у','ф','х','ц',

'ч','ш','щ','ы','ь','э','ю','я']

if mode:

with open("textWithSpaces.txt", encoding="utf-8") as file:

text = file.read()

alphabet.append(" ")

else:

with open("text.txt", encoding="utf-8") as file:

text = file.read()

length = len(text)

#print(length)

if (length % 2 == 1) :

text += 'о'

length += 1

def SetArrayOfBigrams(step):

i = 0

array = []

while i < (length-1):

array.append(text[i] + text[i+1])

i += step

return array

bigramArray1 = SetArrayOfBigrams(1)

bigramArray2 = SetArrayOfBigrams(2)

letters = Counter(text)

bigrams1 = Counter(bigramArray1)

bigrams2 = Counter(bigramArray2)

def CountEntropy(counter, len):

entropy = 0

for i in counter:

p = counter[i] / (len)

log = math.log(p,2)

entropy += -p\*log

return entropy

print(letters)

#print('Entropy for letters: ',CountEntropy(letters, length))

#print('Entropy for bigrams with step 1: ', CountEntropy(bigrams1, length-1))

#print('Entropy for bigrams with step 2: ', CountEntropy(bigrams2, (length/2)))

def PrintLettersFreqency1():

for i in alphabet:

print(i," => " ,(letters[i]/length))

PrintLettersFreqency1()

f1 = open('resultCross.txt', 'w')

def PrintCrossBigramFreqency(bigrams):

for i in alphabet:

for j in alphabet:

bg = (i + j)

p = bigrams[bg] / (length-1)

f1.write(bg + "->" + (str('%.6f' % p) + " "))

f1.write("\n")

PrintCrossBigramFreqency(bigrams1)

f1.close()

f2 = open('resultNotCross.txt', 'w')

def PrintNotCrossBigramFreqency(bigrams):

for i in alphabet:

for j in alphabet:

bg = (i + j)

p = bigrams[bg] / (length-1)

f2.write(bg + "->" + (str('%.6f' % p) + " "))

f2.write("\n")

PrintNotCrossBigramFreqency(bigrams2)

f2.close()

У результаті ми отримуємо частоти літер та біграм у тексті (результати наведені у файлах у папці results).

Після того як ми підрахували частоти літер і біграм у тексті, програма обчислює значення ентропії. За означенням вона з точністю до знаку дорівнює сумі добутків ймовірності на її логарифм. У нашому випадку, за законом великих чисел, ймовірність дорівнює частоті літери або біграми. Ми отримали наступні значення ентропії:

**Текст з пробілами**:

Entropy for letters: 4.36270652063231;

Entropy for bigrams with step 1: 7.8864472116912285;

Entropy for bigrams with step 2: 7.887103700534957;

Якщо з тексту видалити пробіли, то отримали наступні частоти літері біграм (результати наведені у файлах у папці results).

Відповідно трохи змінилися значення ентропії (результати наведені нижче).

**Текст без пробілів**:

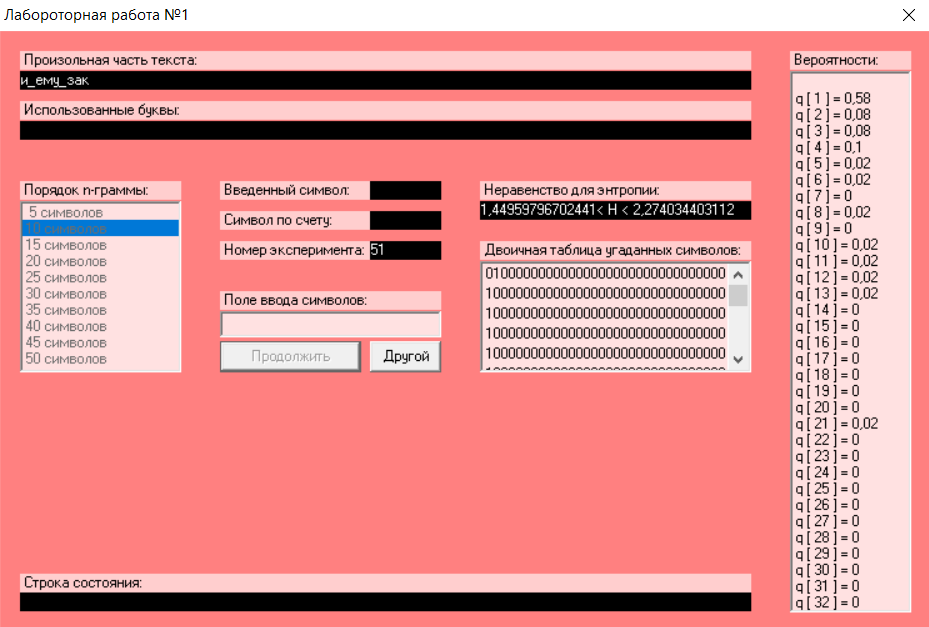
Entropy for letters: 4.461895151859307;

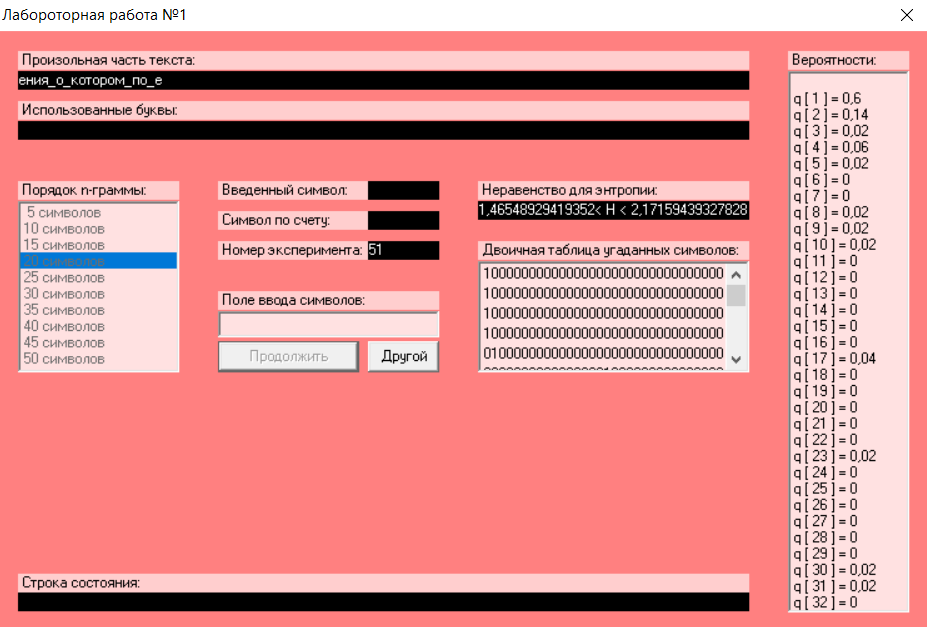
Entropy for bigrams with step 1: 8.26364965022618;

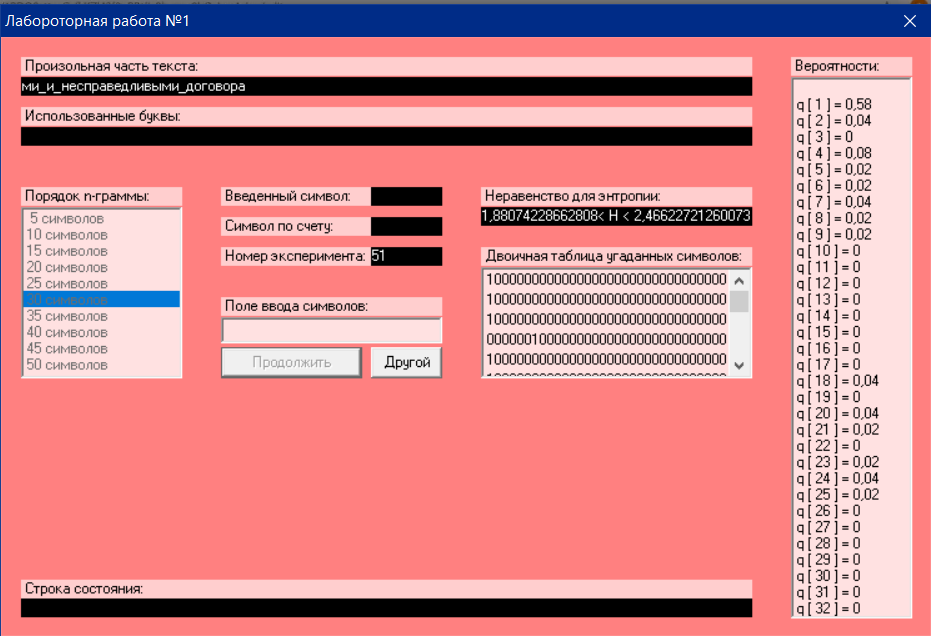
Entropy for bigrams with step 2: 8.262248316978873;

**Частина 2**

Запускаємо CoolPinkProgram, у якій нам необхідно, використовуючи частину тексту, вгадати яким буде наступний символ і на основі цих даних оцінити значення , , та надлишковості російської мови в різних моделях джерел.







Отримали наступні значення ентропії:

, , .

.

Обчислюємо значення надлишковості джерела відкритого тексту: .

. Тобто надлишковість російської мови складає 61%.

**Висновки**: в ході лабораторної роботи ми вивчили поняття ентропії на символі джерела та його надлишковість, дослідили та порівняли різні моделі джерела відкритого тексту для наближеного визначення ентропії, а також набули практичних навичок щодо оцінки ентропії на символі джерела та вдосконалили знання у сфері програмування. Під час аналізу тексту ми підтвердили той факт, що в російському алфавіті частіше всього зустрічається пробіл, що свідчить про те, що при шифруванні потрібного його прибирати, щоб зловмиснику требу було прикласти більше зусиль для його зламу. Якщо з тексту його прибрати, то це будуть літери “о”, “е”, “а”, а рідше за все зустрічаються: “ё”, “ф”, “ц”. За допомогою CoolPinkProgram отримали надлишковість російської мови 61%.