**Національний технічний університет України**

**“Київський політехнічний інститут”**

Лабораторна робота №1

**Експериментальна оцінка ентропії на символ джерела**

**відкритого тексту**

**Виконали студенти:  
Групи ФІ-93  
Шашенок Микита  
Медведь Михайло**

**Варінт №11**

Київ 2022

**Мета роботи**

Засвоєння понять ентропії на символ джерела та його надлишковості, вивчення та порівняння різних моделей джерела відкритого тексту для наближеного визначення ентропії, набуття практичних навичок щодо оцінки ентропії на символ джерела

**Постановка задачі**

Написати програми для обрахунку частот букв та біграм для тексту російською мовою (як з пробілами, так і без них) та відповідних значень H1 та H2, де ймовірності літер замінити частотами.

За допомогою програми CoolPinkProgram отримати експериментальні значення H(10), H(20), H(30). Використовуючи отримані дані, оцінити надлишковість російської мови у різних моделях.

**Хід роботи**

1.Частоти літер (з пробілами та без пробілів):





2. Значення ентропій:



3.Експериментальні значення H(10), H(20), H(30):







4. Оцінка значення R:

\* 0.37 < R < 0.46 при H(10)

\* 0.56 < R < 0.67 при H(30)

\* 0.65 < R < 0.76 при H(20)

**Програмний код:**

import re

import math

monograms = list('абвгдежзийклмнопрстуфхцчшщъыьэюя')

bigrams = [i+j for i in monograms for j in monograms]

def read\_file(filename: str):

text = open(filename, 'r', encoding='utf-8')

texts = str(text.read())

return texts

def count\_substring(big\_string,substring):

count = 0

for char in range(len(big\_string)):

count += big\_string[char: char + len(substring)] == substring

return count

def frequency\_monograms(message):

return [[i, message.count(i)/len(message)] for i in monograms]

def frequency\_bigrams\_no\_overlap(message):

return [[i, message.count(i)/len(message)] for i in bigrams]

def frequency\_bigrams\_overlap(message):

return [[i, count\_substring(message,i)/len(message)] for i in bigrams]

def H1(ProbabilitiesMono):

s = 0

for P in ProbabilitiesMono:

if P[1] != 0:

s += P[1]\*math.log2(P[1])

return -1\*s

def H2(ProbabilitiesBi):

s = 0

for P in ProbabilitiesBi:

if P[1] != 0:

s += P[1]\*math.log2(P[1])

return -1\*s/2

def main():

text = read\_file('ourtext.txt').replace('\n', '')

text\_without\_spaces = (''.join(filter(str.isalpha, text))).lower()

text\_with\_spaces = (re.sub(r'[^Z0-9А-Яа-я- -]', '', text)).lower()

probabilities\_monograms\_no\_spaces = frequency\_monograms(text\_without\_spaces)

probabilities\_monograms\_with\_spaces = frequency\_monograms(text\_with\_spaces)

probabilities\_bigrams\_no\_spaces\_no\_overlap = frequency\_bigrams\_no\_overlap(text\_without\_spaces)

probabilities\_bigrams\_no\_spaces\_with\_overlap = frequency\_bigrams\_overlap(text\_without\_spaces)

probabilities\_bigrams\_with\_spaces\_no\_overlap = frequency\_bigrams\_no\_overlap(text\_with\_spaces)

probabilities\_bigrams\_with\_spaces\_with\_overlap = frequency\_bigrams\_overlap(text\_with\_spaces)

H1\_no\_spaces = H1(probabilities\_monograms\_no\_spaces)

H1\_with\_spaces = H1(probabilities\_monograms\_with\_spaces)

H2\_no\_spaces\_no\_overlap = H2(probabilities\_bigrams\_no\_spaces\_no\_overlap)

H2\_no\_spaces\_overlap = H2(probabilities\_bigrams\_no\_spaces\_with\_overlap)

H2\_with\_spaces\_no\_overlap = H2(probabilities\_bigrams\_with\_spaces\_no\_overlap)

H2\_with\_spaces\_overlap = H2(probabilities\_bigrams\_with\_spaces\_with\_overlap)

print('probabilities\_monograms\_no\_space')

for i in probabilities\_monograms\_no\_spaces:

print(f'Letter = {i[0]} Pr = {i[1]}')

print('probabilities\_monograms\_with\_space')

for j in probabilities\_monograms\_with\_spaces:

print(f'Letter = {j[0]} Pr = {j[1]}')

# print('probabilities\_monograms\_with\_space')

#for k in probabilities\_monograms\_with\_spaces:

# print(f'Letter = {k[0]} Pr = {k[1]}')

print(f'H1 no spaces = {H1\_no\_spaces}\nH1 with spaces = {H1\_with\_spaces}')

print(f'H2 no spaces no overlap = {H2\_no\_spaces\_no\_overlap}\nH2 no spaces with overlap = {H2\_no\_spaces\_overlap}')

print(f'H2 with spaces no overlap = {H2\_with\_spaces\_no\_overlap}\nH2 with spaces with overlap = {H2\_with\_spaces\_overlap}')

if \_\_name\_\_ == '\_\_main\_\_':

main()

Висновки

Наша пригада реалізувала програмний підрахунок частот літер, біграм та значень ентропій для різних моделей тексту. Також були отримані експериментальні дані обчислення надлишковості тексту російською мовою. Було встановлено, що ентропії текстів з вилученими пробілами можуть достатньо відрізнятись, в той час як ентропія сукупного розподілу біграм на тексті у випадках коли ми рахуємо неперетинні біграми та біграми що перетинаються майже співпадають. З таблиці результатів видно, що найуживанішими літерами російської мови - це "о", "п", "р", "c".



**Код програми**