**КРИПТОГРАФІЯ**

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ No1**

**Експериментальна оцінка ентропії на символ джерела відкритого тексту**

**Виконав студент ФБ-34 Синельник Максим**

**Мета роботи**

Засвоєння понять ентропії на символ джерела та його надлишковості, вивчення та

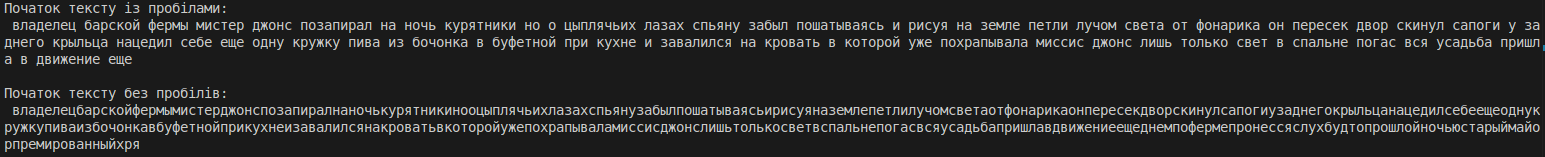
порівняння різних моделей джерела відкритого тексту для наближеного визначення

ентропії, набуття практичних навичок щодо оцінки ентропії на символ джерела.

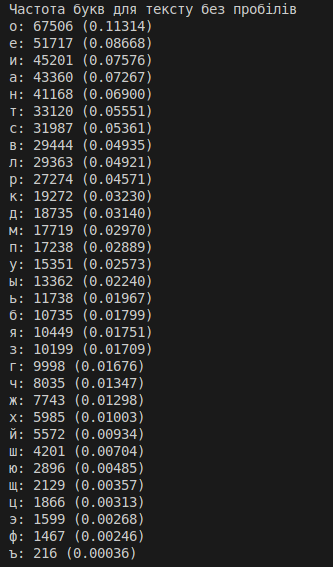
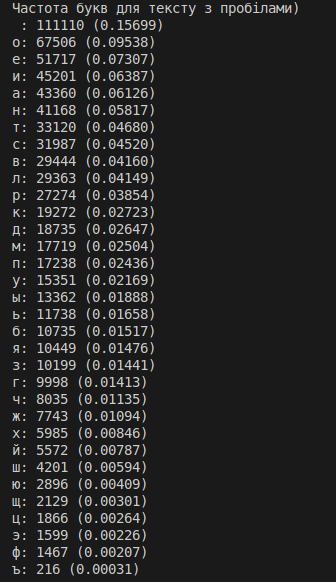
**Написати програми для підрахунку частот букв і частот біграм в тексті, а також**

**підрахунку H1 та H 2 за безпосереднім означенням. Підрахувати частоти букв та біграм, а також значення H1 та H2 на довільно обраному тексті російською мовою достатньої довжини (щонайменше 1Мб), де імовірності замінити відповідними частотами. Також одержати значення H1 та H2 на тому ж тексті, в якому вилучено всі пробіли.**

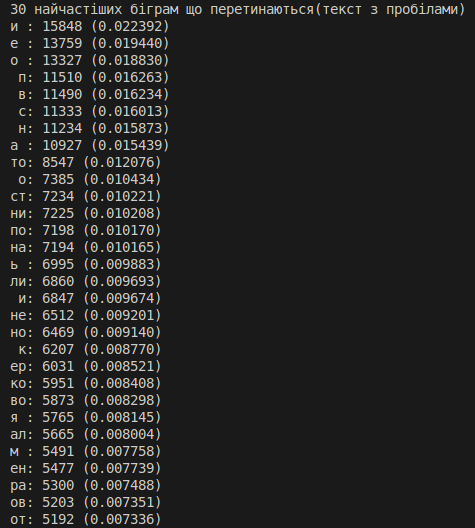
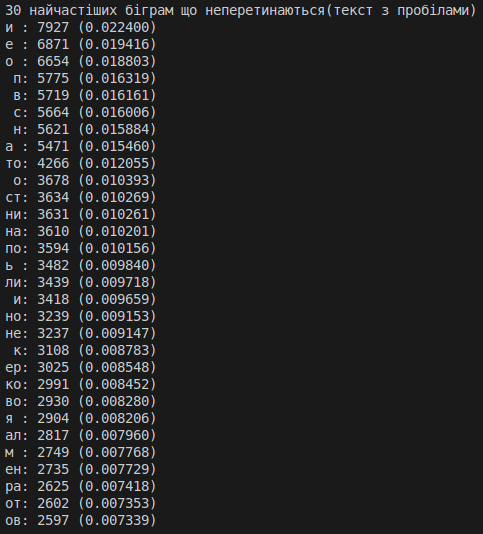
Отже на початку завдання треба відфільтрувати файл з текстом, як сказано у методичних вказівках, я замінив всі символи, окрім текстових із алфавіту на пробіли, а самі послідовності пробілів на один пробіл. В сам же алфавіт тексту із пробілами буде входити і сам пробіл, тоді в алфавіті буде 33 літери. Для другого тексту без пробілів, пробіли прибрав.

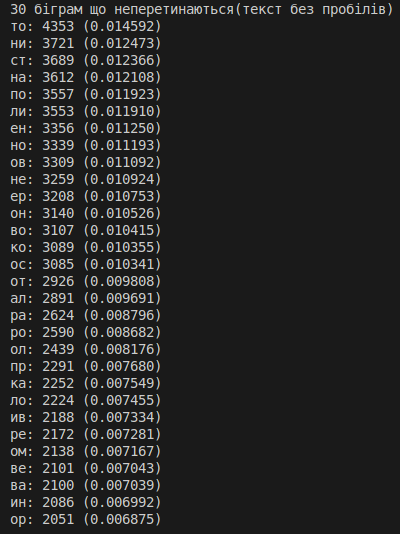
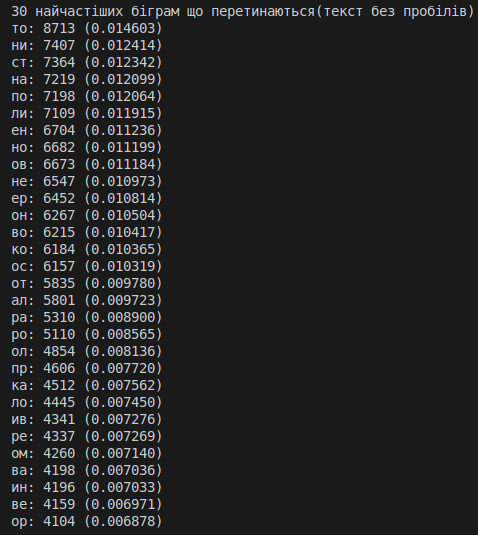


Далі відповідно порахував частоти для букв у двух текстах, кількості букв для них будуть однакові, а от частота різна, так як в тексті з пробілами, пробіл виступає в ролі літери, отже кількість літер буде більшою ніж у тексті без пробілів.

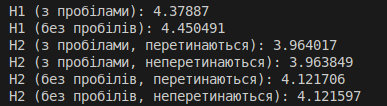


Далі треба обчислити біграми. Варіантів біграм буде 4 з перетином і без для тексту з пробілами та без. Опісля можна приступати до обчислення частоти біграм для кожної категорії. Таблиці з наведеними частотами біграм, робляться в кінці, на прикладі я виводив по 30 найчастіших біграм під кожну категорію.



Далі йде обчислення ентропії. Для букв буду використовувати звичайну формулу ентропії, так як це найпростіша модель джерела повідомлень - це H1. H2 це ентропія, яка враховує залежність між сусідніми символами, тобто ми дивимося вже не на окремі літери, а на пари, відповідно будем ділити все на 2.



Код для виконання цього завдання:

import re

import math

from collections import Counter

with open("crypto\_lab1.txt", "r", encoding="utf-8") as f:

text = f.read()

text = text.lower()

text\_with\_spaces = re.sub(r"[^а-яё\s]", " ", text)

text\_with\_spaces = re.sub(r"\s+", " ", text\_with\_spaces).strip()

text\_no\_spaces = re.sub(r"\s+", "", text\_with\_spaces)

print("Початок тексту із пробілами:\n", text\_with\_spaces[:400])

print("\nПочаток тексту без пробілів:\n", text\_no\_spaces[:400])

print("\nДовжина тексту з пробілами:", len(text\_with\_spaces))

print("Довжина тексту без пробілів:", len(text\_no\_spaces))

def chastota\_bukv(text):

counts = Counter(text)

total = sum(counts.values())

return {ch: counts[ch] / total for ch in counts}, counts, total

letter\_freq\_with, letter\_counts\_with, total\_with = chastota\_bukv(text\_with\_spaces)

letter\_freq\_no, letter\_counts\_no, total\_no = chastota\_bukv(text\_no\_spaces)

print("\nЧастота букв для тексту з пробілами)")

for ch, cnt in letter\_counts\_with.most\_common(33):

print(f"{ch}: {cnt} ({letter\_freq\_with[ch]:.5f})")

print("\nЧастота букв для тексту без пробілів")

for ch, cnt in letter\_counts\_no.most\_common(32):

print(f"{ch}: {cnt} ({letter\_freq\_no[ch]:.5f})")

def bigrams\_count\_func(text, step=1):

bigrams = Counter()

for i in range(0, len(text) - 1, step):

pair = text[i:i+2]

if len(pair) == 2:

bigrams[pair] += 1

return bigrams

bigrams\_with\_overlap = bigrams\_count\_func(text\_with\_spaces, step=1)

bigrams\_with\_nonoverlap = bigrams\_count\_func(text\_with\_spaces, step=2)

bigrams\_no\_overlap = bigrams\_count\_func(text\_no\_spaces, step=1)

bigrams\_no\_nonoverlap = bigrams\_count\_func(text\_no\_spaces, step=2)

def bigram\_chastota(counter):

total = sum(counter.values())

return {bg: counter[bg] / total for bg in counter}, total

bigrams\_freq\_with\_overlap, total\_with\_overlap = bigram\_chastota(bigrams\_with\_overlap)

bigrams\_freq\_no\_overlap, total\_no\_overlap = bigram\_chastota(bigrams\_no\_overlap)

print("\n30 найчастіших біграм що перетинаються(текст з пробілами)")

for bg, cnt in bigrams\_with\_overlap.most\_common(30):

print(f"{bg}: {cnt} ({bigrams\_freq\_with\_overlap[bg]:.6f})")

print("\n30 найчастіших біграм що неперетинаються(текст з пробілами)")

for bg, cnt in bigrams\_with\_nonoverlap.most\_common(30):

freq = cnt / sum(bigrams\_with\_nonoverlap.values())

print(f"{bg}: {cnt} ({freq:.6f})")

print("\n30 найчастіших біграм що перетинаються(текст без пробілів)")

for bg, cnt in bigrams\_no\_overlap.most\_common(30):

print(f"{bg}: {cnt} ({bigrams\_freq\_no\_overlap[bg]:.6f})")

print("\n30 біграм що неперетинаються(текст без пробілів)")

for bg, cnt in bigrams\_no\_nonoverlap.most\_common(30):

freq = cnt / sum(bigrams\_no\_nonoverlap.values())

print(f"{bg}: {cnt} ({freq:.6f})")

def entropy\_H1(text):

counts = Counter(text)

total = sum(counts.values())

H = 0.0

for c in counts:

p = counts[c] / total

H -= p \* math.log2(p)

return H

def entropy\_H2(counter):

total = sum(counter.values())

H = 0.0

for count in counter.values():

p = count / total

H -= p \* math.log2(p)

return H / 2

H1\_with = entropy\_H1(text\_with\_spaces)

H1\_no = entropy\_H1(text\_no\_spaces)

H2\_with\_overlap = entropy\_H2(bigrams\_with\_overlap)

H2\_with\_nonoverlap = entropy\_H2(bigrams\_with\_nonoverlap)

H2\_no\_overlap = entropy\_H2(bigrams\_no\_overlap)

H2\_no\_nonoverlap = entropy\_H2(bigrams\_no\_nonoverlap)

print("\nH1 (з пробілами):", round(H1\_with, 6))

print("H1 (без пробілів):", round(H1\_no, 6))

print("H2 (з пробілами, перетинаються):", round(H2\_with\_overlap, 6))

print("H2 (з пробілами, неперетинаються):", round(H2\_with\_nonoverlap, 6))

print("H2 (без пробілів, перетинаються):", round(H2\_no\_overlap, 6))

print("H2 (без пробілів, неперетинаються):", round(H2\_no\_nonoverlap, 6))

import csv

def save\_bigrams(counter, total, filename):

with open(filename, "w", newline="", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerow(["Біграма", "Кількість", "Частота"])

for bg, cnt in counter.most\_common():

freq = cnt / total

writer.writerow([bg, cnt, f"{freq:.8f}"])

save\_bigrams(bigrams\_with\_overlap, total\_with\_overlap, "bigrams\_with\_overlap.csv")

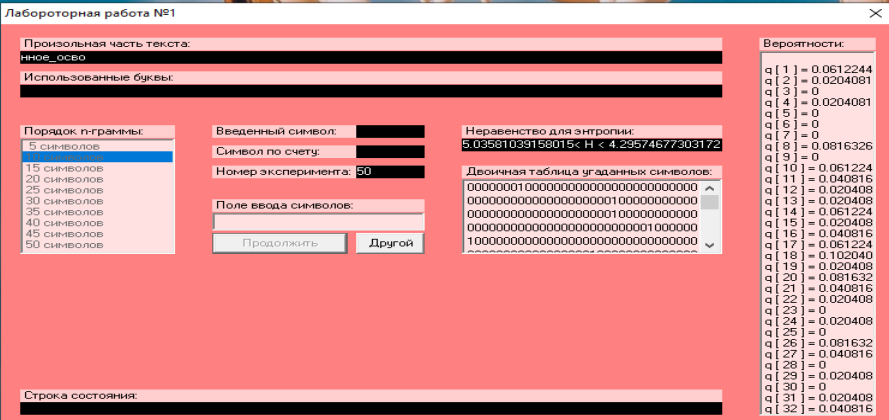
save\_bigrams(bigrams\_with\_nonoverlap, sum(bigrams\_with\_nonoverlap.values()), "bigrams\_with\_nonoverlap.csv")

save\_bigrams(bigrams\_no\_overlap, total\_no\_overlap, "bigrams\_no\_overlap.csv")

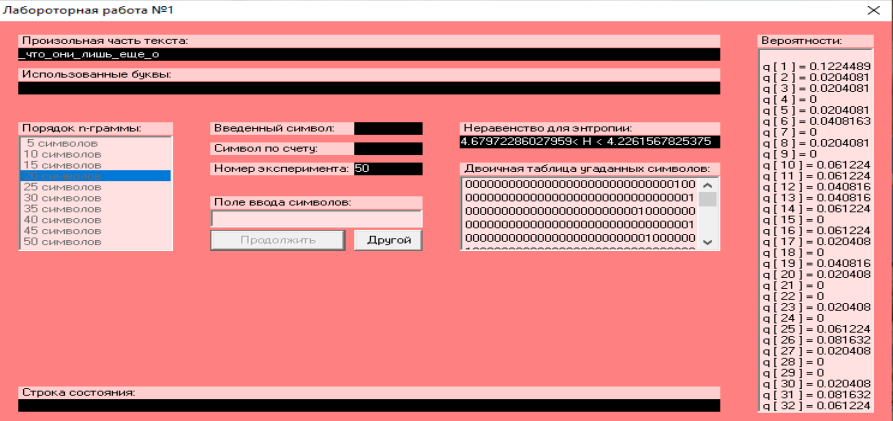
save\_bigrams(bigrams\_no\_nonoverlap, sum(bigrams\_no\_nonoverlap.values()), "bigrams\_no\_nonoverlap.csv")

**За допомогою програми CoolPinkProgram оцінити значення H (10) , H ( 20) , H (30)**

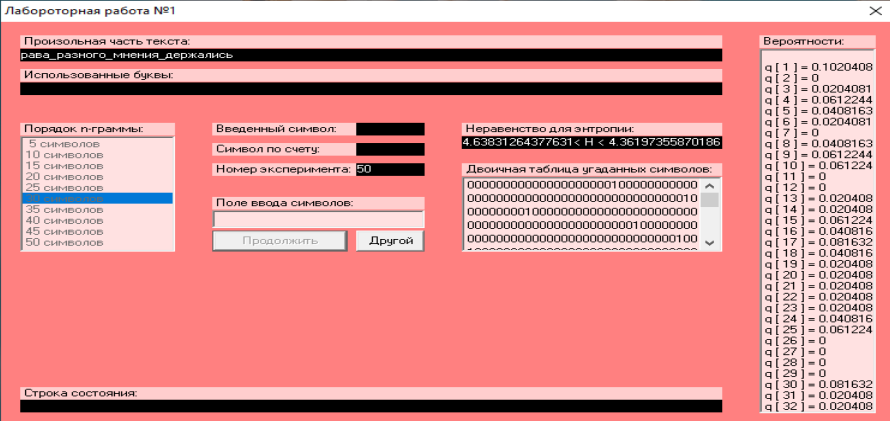
5.03<H(10)<4.29



4.68<H(20)<4,23



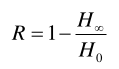
4.63<H(30)<4.36

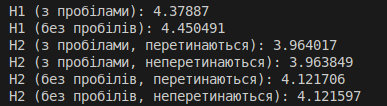


**Використовуючи отримані значення ентропії, оцінити надлишковість російської**

**мови в різних моделях джерела.**

Для цього використаю формулу:





Відповідно будемо ділити ентропію нашої моделі на максимально можливу ентропію (log2(N));

Для алфавіту без пробілів це буде 33 літери, тоді H0 буде log2(33)=5.0444

Для алфавіту з пробілами це буде 34 літери, тоді H0 буде log2(34)=5.0875

R1(з пробілом) : 1-(4.3789/5.0875)=0.139

R1(без пробілу) : 1-(4.4504/5.0444)=0.118

R2 (з пробілами, перетинаються): 1-(3.9640/5.0875)=0.214

R2 (з пробілами, неперетинаються): 1-(3.9638/5.0875)=0.221

R2 (без пробілів, перетинаються): 1-(4.1217/5.0444)=0.183

R2 (без пробілів, неперетинаються): 1-(4.1215/5.0444)=0.183

**Висновки:** Отримані результати показали, що символи та їх сполучення мають нерівномірний розподіл і виявляють закономірності. Зменшення ентропії при переході від окремих символів до біграм свідчить про наявність залежностей між елементами тексту. Врахування пробілів призводить до зменшення ентропії, оскільки пробіл є одним із найчастіших і найбільш передбачуваних символів. Підрахована надлишковість підтверджує структурованість і передбачуваність текстових повідомлень, що є характерною властивістю природних інформаційних джерел.