Міністерство освіти і науки, молоді та спорту
Національний технічний університет України
“Київський політехнічний інститут”

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління

ЛАБОРАТОРНА РОБОТА №1

з дисципліни “Комп’ютерна лінгвістика”
на тему: “Ідентифікація автора тексту по розподілу частот буквосполучень”

Виконав: Турко М. В.
ст. гр. ІС-73
Перевірив: Фіногенов О. Д.

Київ 2020

Лабораторна робота №1

Мета: вивчення методів визначення автора тексту за допомогою статистичного аналізу буквосполучень.

Варіант завдання: було обрано трьох українських авторів з різних часових періодів:

- М. Гоголь (1809-1852 рр.) - “Вій”, “Вечори на хуторі близь Диканьки”, “Ніч проти Різдва”;
- І. Нечуй-Левицький (1839-1918 рр.) - “Кайдашева сім’я”, “Князь Єремія Вишневецький”, “Дві милі;
- В. Нестайко (1930-2014 рр.) - “Тореадори з Васюківки”, “Одиниця з обманом”, “У країні сонячних зайчиків”.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Формалізована задача розпізнавання автора тексту може бути представлена у наступний спосіб:

Нехай є бібліотека текстів, яка представлена у вигляді щільності функції розподілу або статистиці n-грамів (грамів, біграмів, триграмів і т.д.) для А відомих авторів. Нехай K_a – кількість текстів а–го автора, $N_{i,a}$ – кількість символів в і-му тексті цього автора, де $i = 1,2,...,K_a$. Будемо вважати, що довжина кожного тексту достатня для проведення статистичного аналізу (тобто довжина тексту дає достатньо інформації про стиль автора в розрізі частоти зустрічі n-грамів).

Нехай $f_{i,a}(j)$ – функція щільності n-грамів відповідного тексту. де $j = 1,...,a(n) = 1,...,\Omega^n$ (Ω – розмір алфавіту або кількість символів, що

досліджуються). Для кожного автора визначимо середньозважене значення щільності функції розподілу, нехтуючи відмінністю (одиницею) між грамами, біграмами і далі, внаслідок $N_a >> n$:

$$F_a(j) = \frac{1}{N_a} \sum_{i=1}^{K_a} f_{i,a}(j) N_{i,a}, \quad N_a = \sum_{i=1}^{K_a} N_{i,a}. \qquad (1)$$

Значення $F_a(j)$ – будемо вважати авторським еталоном.

Введемо «бібліотечну норму» ρ_{ik} як відстань між щільністю функцій розподілу текстів i та k в нормі підсумованих функцій:

$$\rho_{ik} = \|f_i - f_k\| = \sum_{j=1}^{\alpha(n)} |f_i(j) - f_k(j)|. \qquad (2)$$

Нехай в наявності є текст «0» невідомого автора, який необхідно ідентифікувати всередині даної бібліотеки. Автором тексту «0» вважається той з авторів «а», для якого норма $\rho_a^0 = \|f_0 - F_a\|$ різниці між щільністю функції розподілу $f_0(j)$ тексту «0» та середньою авторською щільністю функції розподілу $F_a(j)$ мінімальна:

$$\rho_a^0 = \|f_0 - F_a\|, \quad a^0 = \arg\min_a \rho_a^0. \qquad (6)$$

Дослідження російської мови показують, що найкращий результат по розпізнаванню автора відбувається при використанні 5 або 6 літерних буквосполучень, але в цьому випадку відсутня проста графічна інтерпретація результатів і тому в даній лабораторній роботі використані лише грами та біграми.

ХІД ВИКОНАННЯ

1. Опис програми

Реалізація програми для статистичного аналізу тексту було написано на мові програмування Python. Весь код лабораторної роботи міститься у додатках (сторінка …).

а. Структура проєкту

На рисунку 1.1 зображена файлова структура проєкту. Дві папки: “analyze_results” - для збереження результатів статистичного аналізу тексту; “books_source” - для текстів авторів.

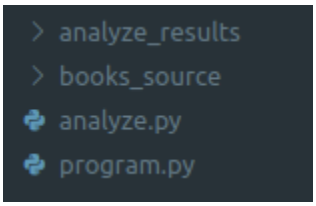

Рис.1.1 - файлова структура

Analyze.py містить методи для обробки тексту автора:

symbol_analyzer() - функція для підрахунку частоти кожної літери алфавіту, загальної кількості символів у тексті та частоти літер алфавіту у тексті. Результат записується у файл та виводиться гістограма частот.

bigram_analyzer() - фукнція для підрахунку біграм у тексті. Результат записується у файл та використовується для побудови теплограми.

heat_map() - функція для побудови та виводу теплограми біграмів.

2. Статистичний аналіз текстів

Кількість символів для кожного твору автора наведена у таблиці 2.1.

Таблиця 2.1 - кількість символів твору.

Автор	Твір	К-сть символів
М. Гоголь	**Вій**	*68 762*
	Вечори на хуторі близь Диканьки	*394 483*
	Ніч проти Різдва	*78 742*
І. Нечуй-Левицький	**Кайдашева сім'я**	*253 905*
	Князь Єремія Вишневецький	*516 073*
	Не той став	*226 314*
В. Нестайко	**Тореадори з Васюківки**	*783 704*
	Одиниця з обманом	*179 121*
	У країні сонячних зайчиків	*251 056*

Частота літер алфавіту по твору автора наведено у таблиці 2.2.

Таблиця 2.2 - частота символів.

Символ	Частота символа у творі								
	М. Гоголь			І. Нечуй-Левицький			В. Нестайко		
	Вій	**Вечори на хуторі близь Диканьки**	**Ніч проти Різдва**	**Кайдашева сім'я**	**Князь Єремія Вишневецький**	**Не той став**	**Тореадори з Васюківки**	**Одиниця з обманом**	**У країні соняч них зайчиків**
А	0.0692	0.0689	0.076	0.0921	0.0751	0.0831	0.0726	0.0745	0.0728
Б	0.0169	0.0167	0.018	0.0168	0.0149	0.0179	0.0167	0.0149	0.0163
В	0.0509	0.0474	0.0487	0.0464	0.0485	0.0455	0.048	0.0476	0.0488
Г	0.0153	0.0159	0.0132	0.0136	0.0151	0.0132	0.0144	0.015	0.0135
Ґ	0.0001	0.0	0.0001	0.0001	0.0001	0.0	0.0	0.0001	0.0
Д	0.0252	0.0269	0.0267	0.028	0.0266	0.0275	0.0274	0.0261	0.0243
Е	0.0356	0.0372	0.0374	0.0363	0.0396	0.0393	0.042	0.0372	0.0401
Є	0.0017	0.0033	0.0021	0.0023	0.004	0.0026	0.0034	0.0029	0.0028
Ж	0.0075	0.0073	0.0085	0.0066	0.0062	0.0084	0.0084	0.0075	0.0077
З	0.0217	0.0192	0.0196	0.0183	0.0225	0.0174	0.0203	0.0192	0.0226
И	0.0578	0.0552	0.053	0.0533	0.0587	0.0536	0.0544	0.0524	0.0555
Й	0.0115	0.0116	0.0121	0.0146	0.0139	0.0113	0.0102	0.0116	0.0137
К	0.0305	0.0318	0.0348	0.0354	0.0336	0.0282	0.0316	0.0359	0.0337
Л	0.0335	0.0302	0.0306	0.0401	0.0358	0.0282	0.031	0.033	0.0326

М	0.0234	0.0235	0.0252	0.024	0.0255	0.0268	0.0257	0.0241	0.0237
Н	0.046	0.047	0.0414	0.0409	0.0504	0.0476	0.0481	0.0477	0.0527
О	0.0866	0.0818	0.0825	0.0767	0.0778	0.0822	0.0771	0.0759	0.0732
П	0.0229	0.0223	0.0221	0.0233	0.024	0.0212	0.0229	0.0228	0.0215
Р	0.032	0.0328	0.0291	0.0358	0.0341	0.031	0.0336	0.036	0.034
С	0.0354	0.035	0.035	0.0337	0.0362	0.0397	0.0326	0.0343	0.0352
Т	0.0387	0.0395	0.0406	0.0386	0.0372	0.0401	0.0399	0.0383	0.0375
У	0.0299	0.0296	0.0312	0.0257	0.0252	0.0242	0.0323	0.0327	0.0304
Ф	0.004	0.0006	0.0002	0.0	0.0002	0.0006	0.001	0.0014	0.0005
Х	0.0103	0.0098	0.0102	0.0112	0.0099	0.0087	0.0093	0.0102	0.0103
Ц	0.0046	0.0056	0.0064	0.0051	0.0064	0.0057	0.0058	0.0057	0.0061
Ч	0.014	0.0147	0.0161	0.0123	0.0117	0.0133	0.013	0.0129	0.0158
Ш	0.0089	0.0079	0.0091	0.0114	0.0075	0.0065	0.0085	0.0076	0.007
Щ	0.0057	0.0058	0.0066	0.0041	0.0044	0.0045	0.0059	0.0042	0.0055
І	0.044	0.0429	0.0397	0.0373	0.0423	0.0429	0.0436	0.0455	0.0448
Ї	0.004	0.0043	0.0035	0.0046,	0.0056	0.0047	0.0031	0.0031	0.0037
Ь	0.0113	0.0148	0.0131	0.0127	0.0169	0.013	0.0144	0.0146	0.0127
Ю	0.0061	0.0074	0.0069	0.0063	0.0058	0.0066	0.007	0.0064	0.0065
Я	0.0187	0.02	0.0206	0.0166	0.0189	0.018	0.022	0.0189	0.0221
Пробіл	0.176	0.1831	0.1801	0.1756	0.1652	0.1777	0.1738	0.1797	0.1721

На рисунках 2.1-2.12 зображені гістограми частот на основі таблиці 2.1.

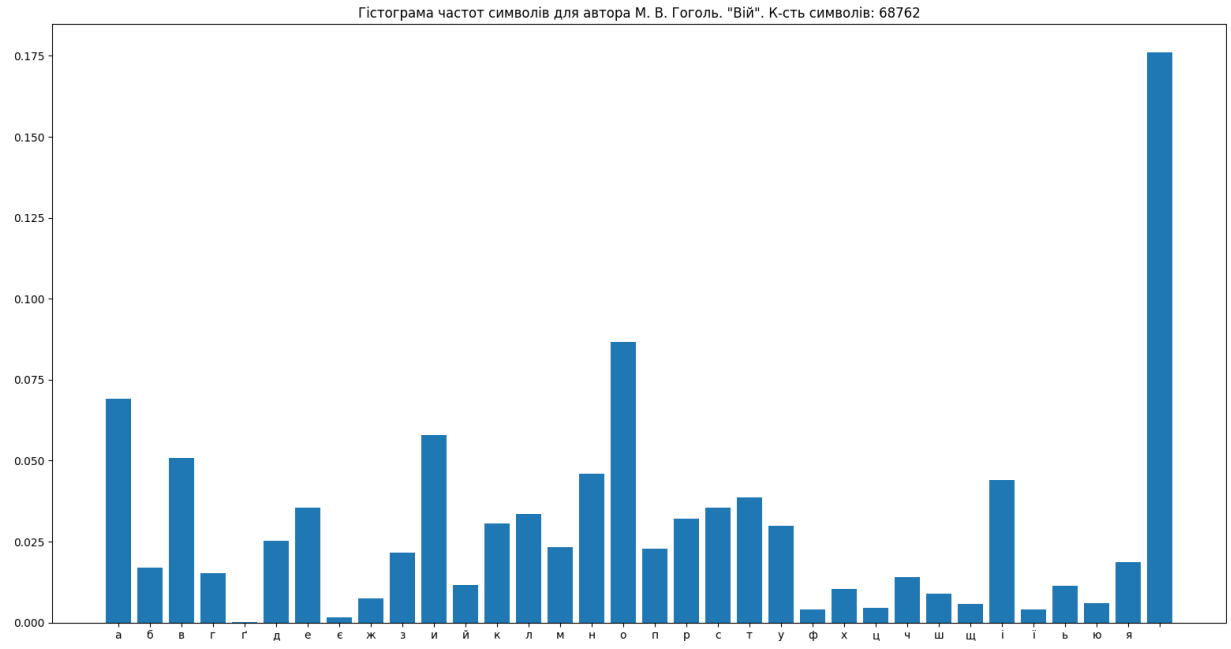

Рис. 2.1 - гістограма частот символів з твору “Вій”

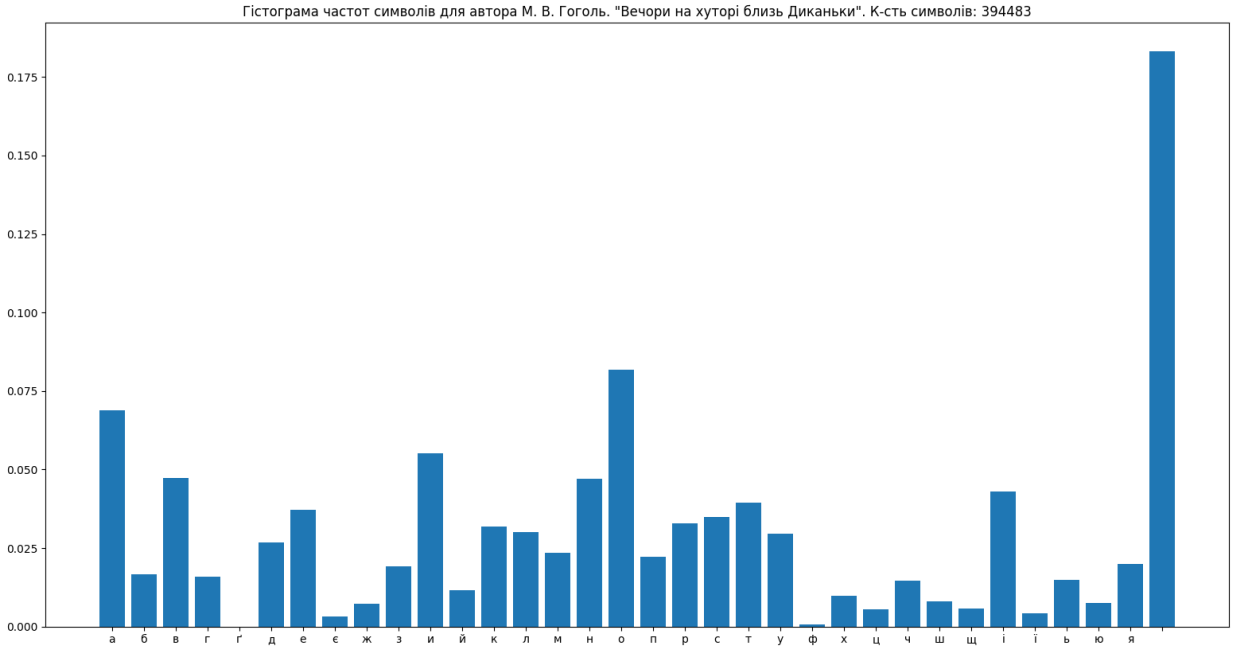

Рис. 2.2 - гістограма частот символів з твору “Вечори на хуторі близь Диканьки”

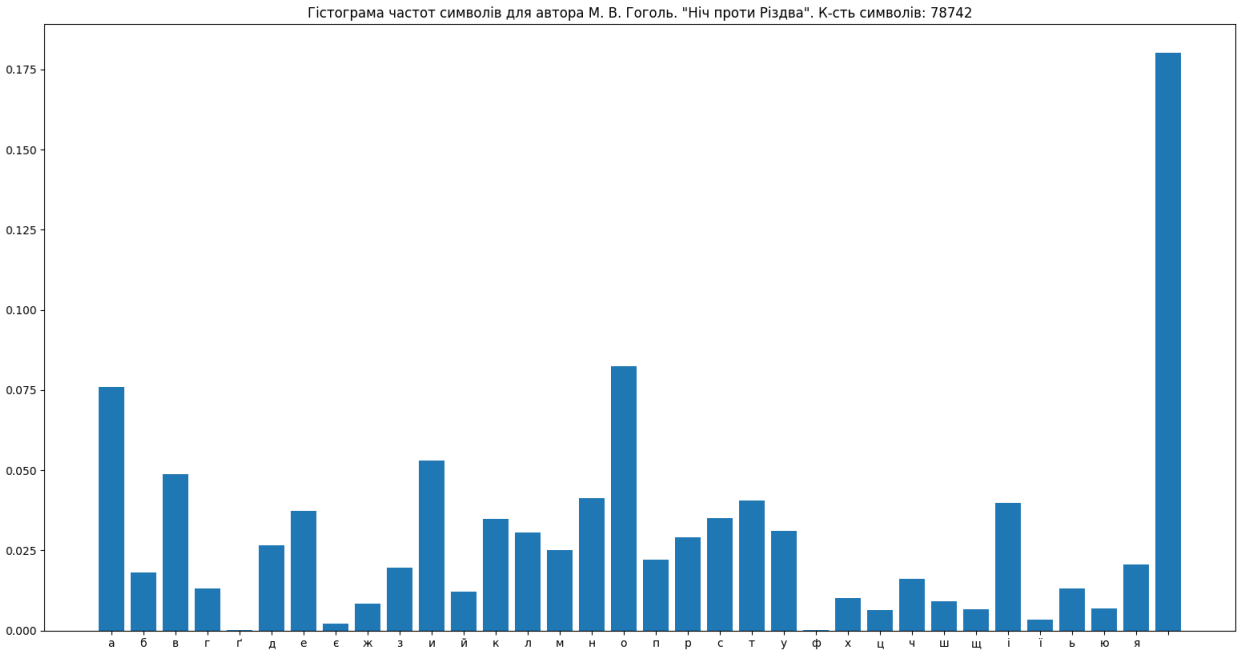

Рис. 2.3 - гістограма частот символів з твору “Ніч проти Різдва”

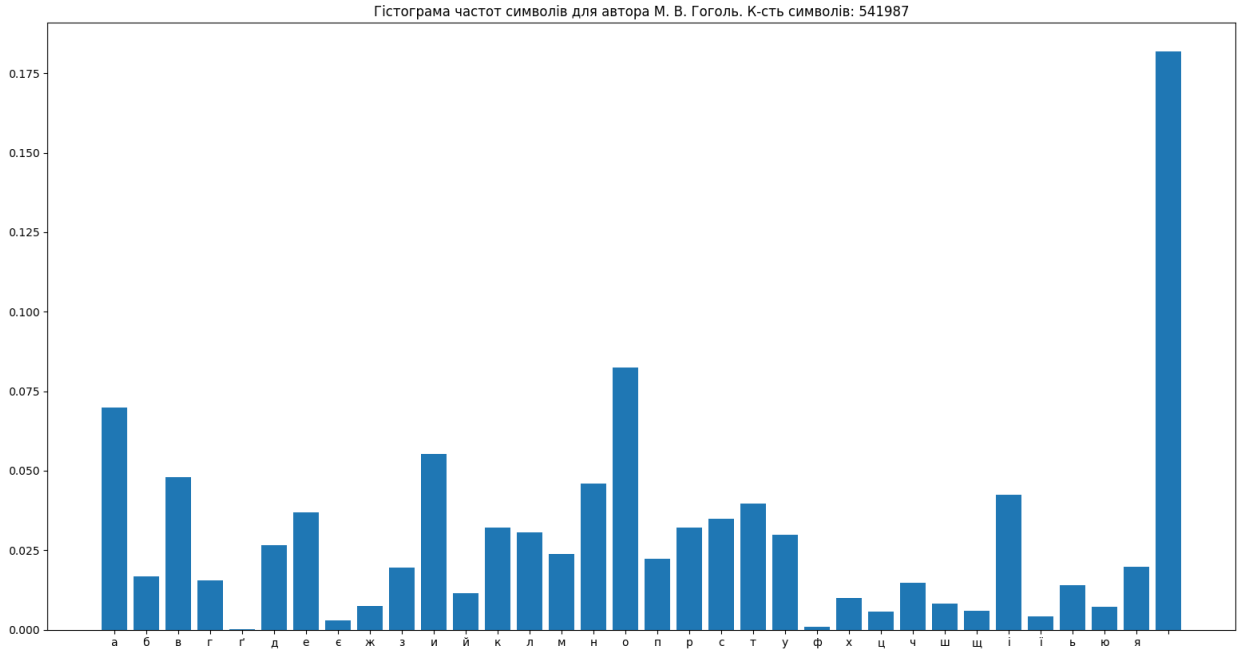

Рис. 2.4 - сумарна гістограма частот символів М. Гоголя

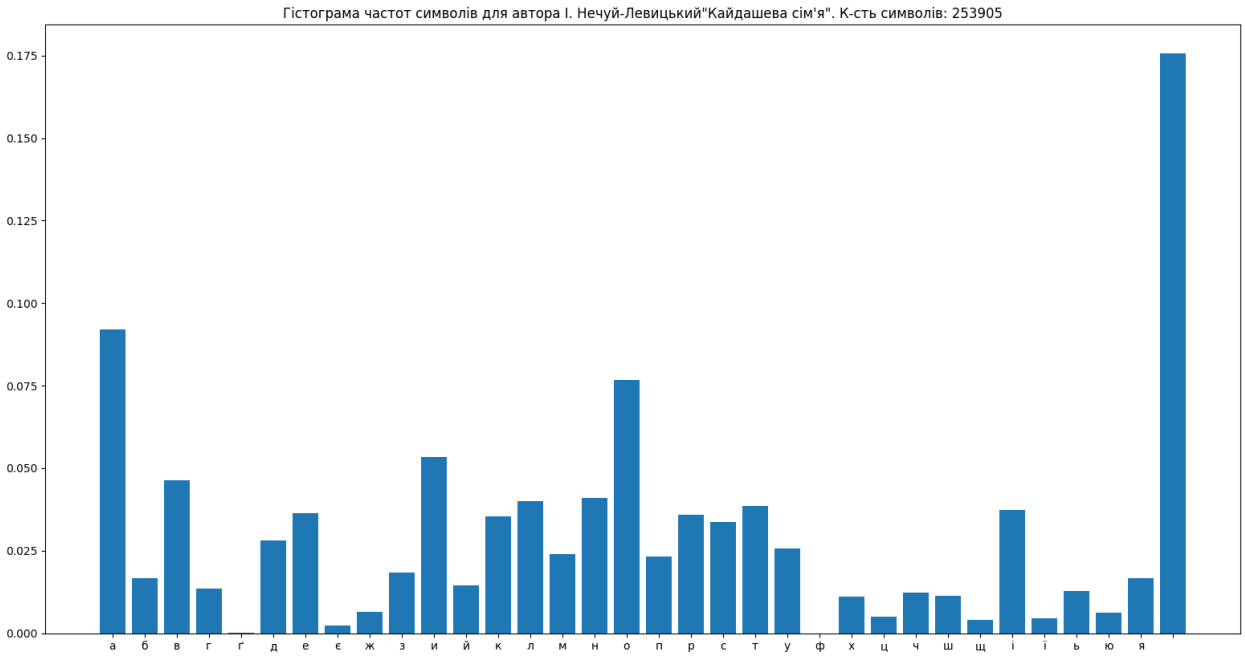

Рис. 2.5 - гістограма частот символів з твору “Кайдашева сім’я”

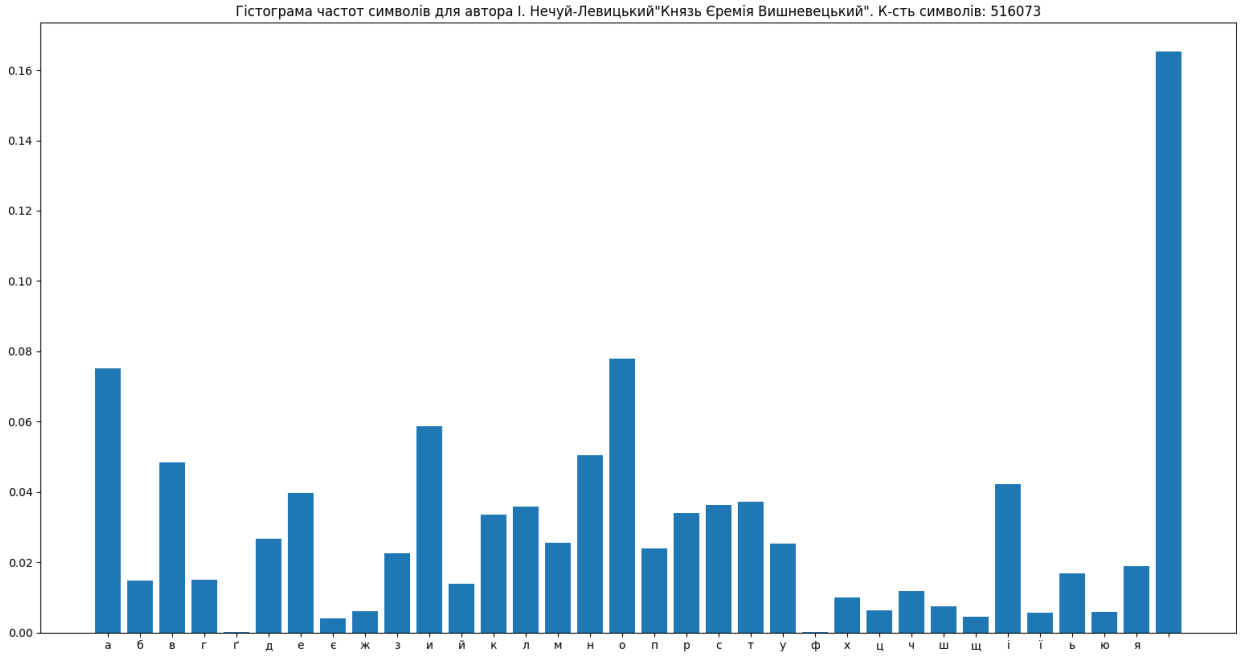

Рис. 2.6 - гістограма частот символів з твору “Князь Єремія Вишневецький”

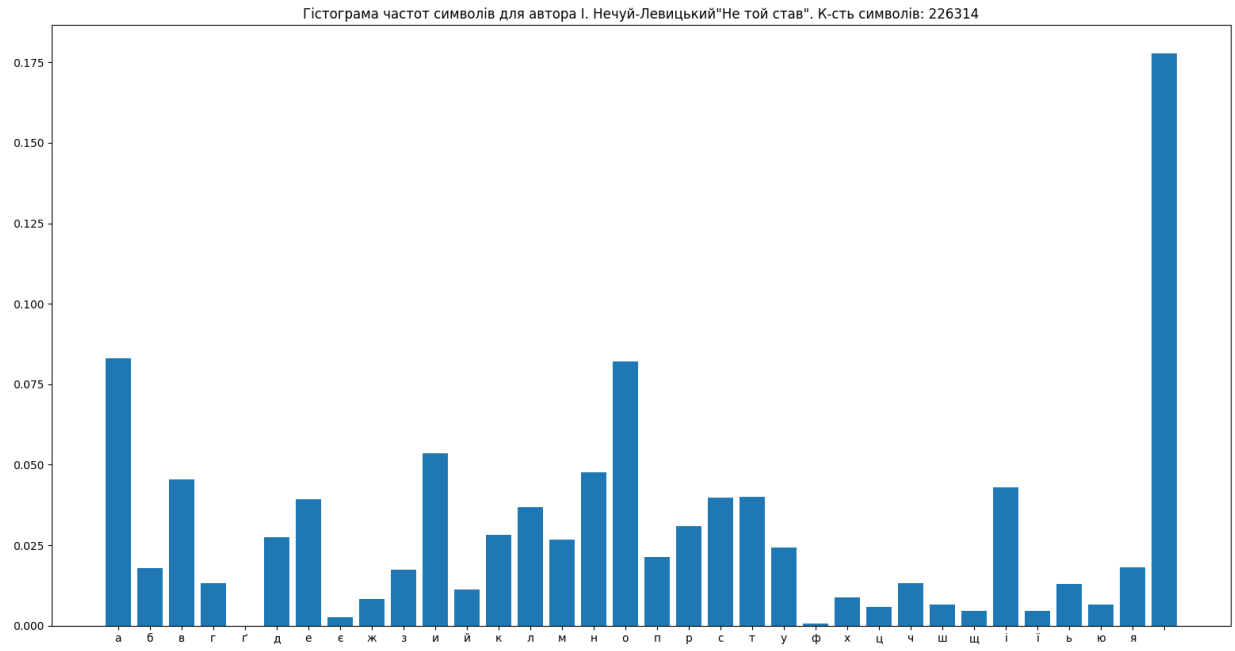

Рис. 2.7 - гістограма частот символів з твору “Не той став”

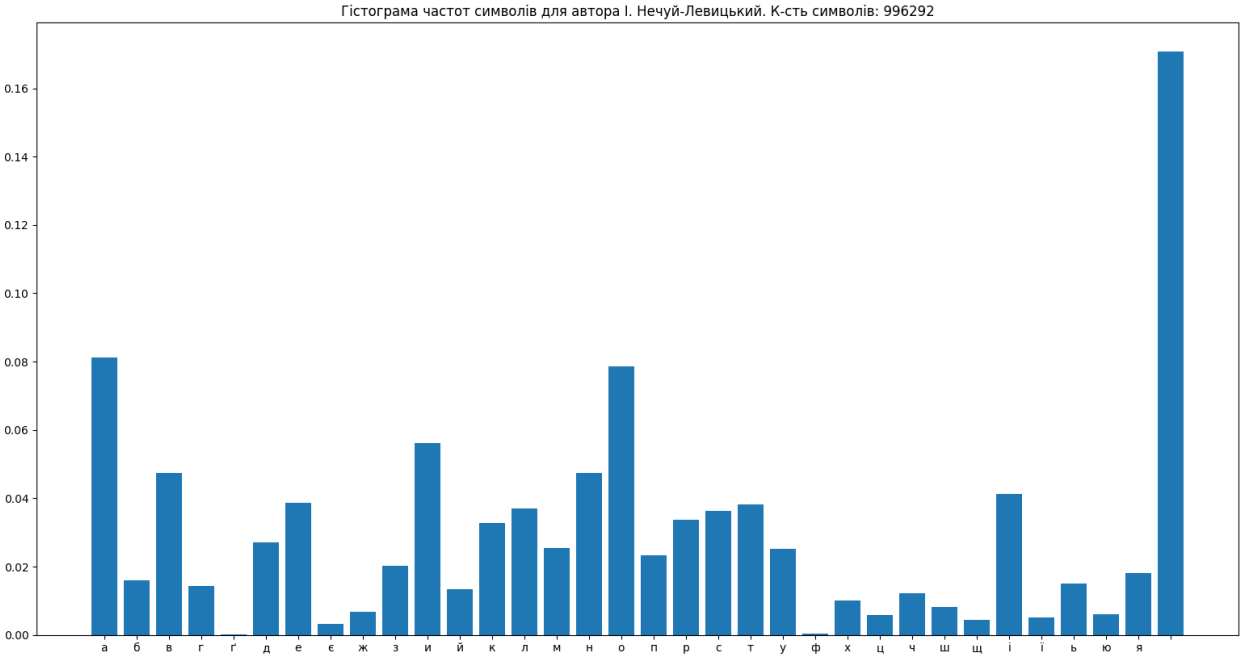

Рис. 2.8 - сумарна гістограма частот символів І. Нечуй-Левицький

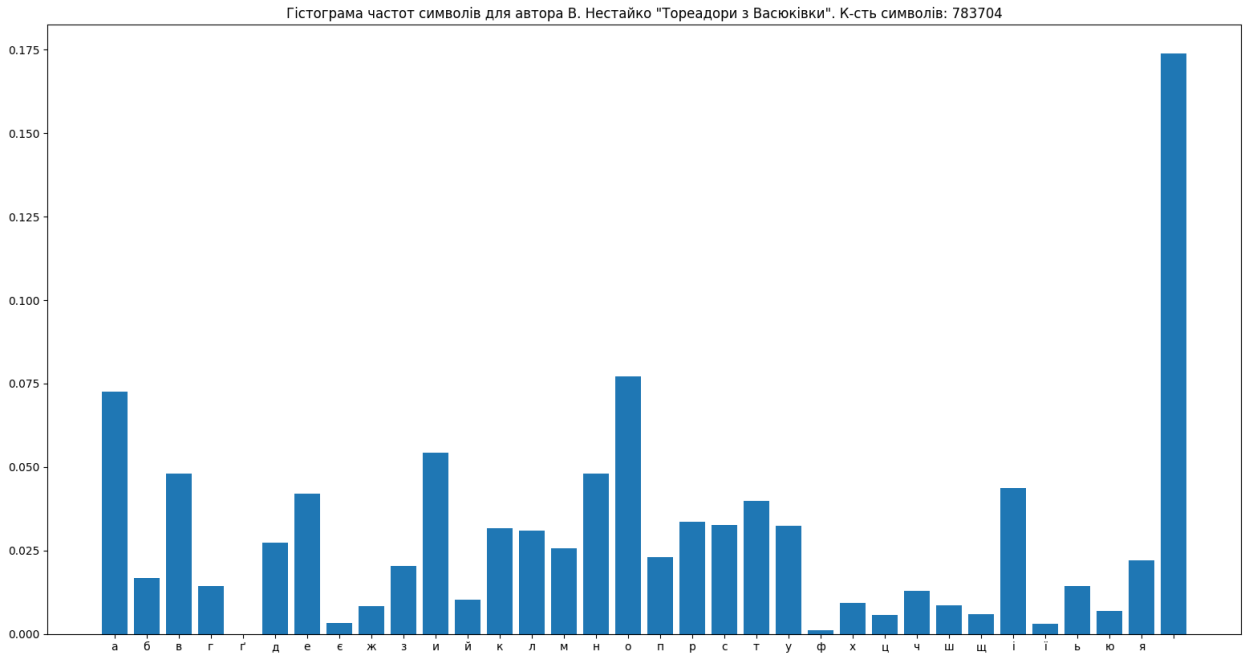

Рис. 2.9 - гістограма частот символів з твору “Тореадори з Васюківки”

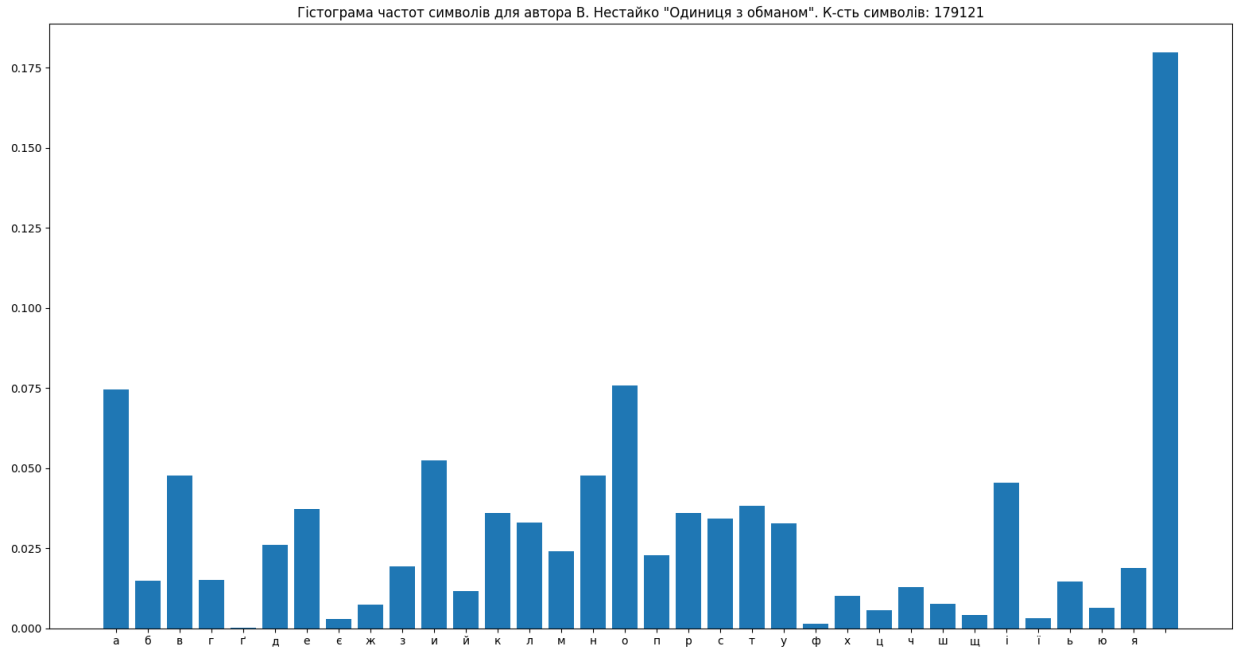

Рис. 2.10 - гістограма частот символів з твору “Одиниця з обманом”

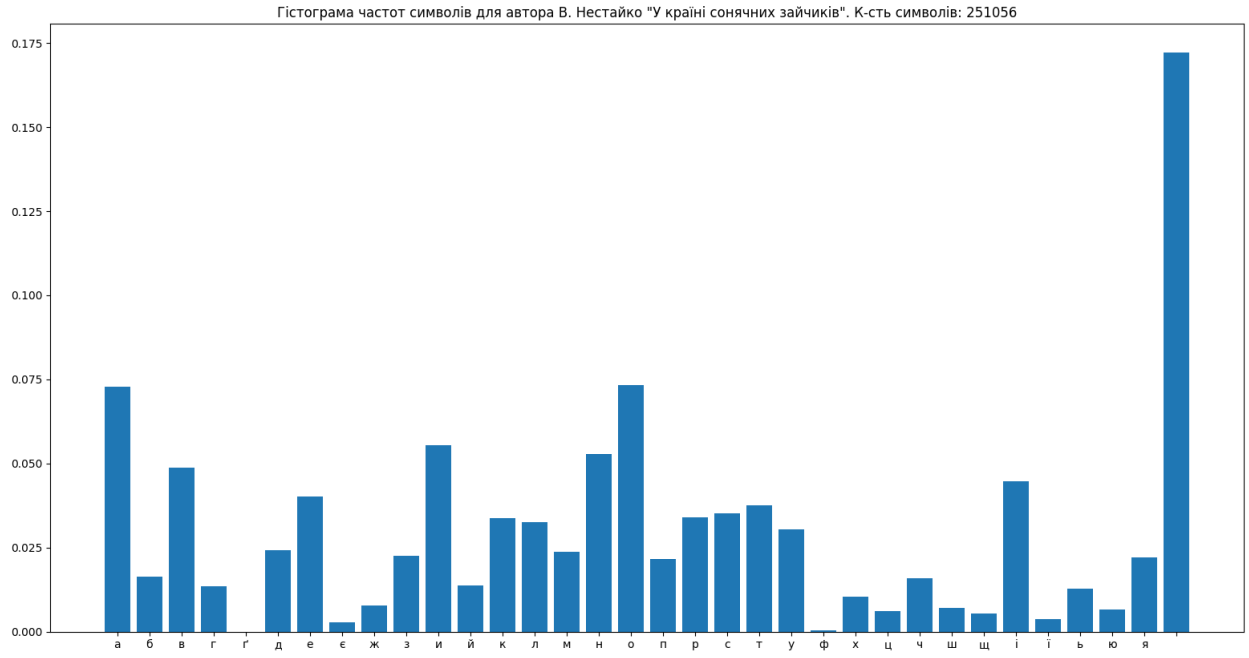

Рис. 2.11 - гістограма частот символів з твору “У країні сонячних зайчиків”

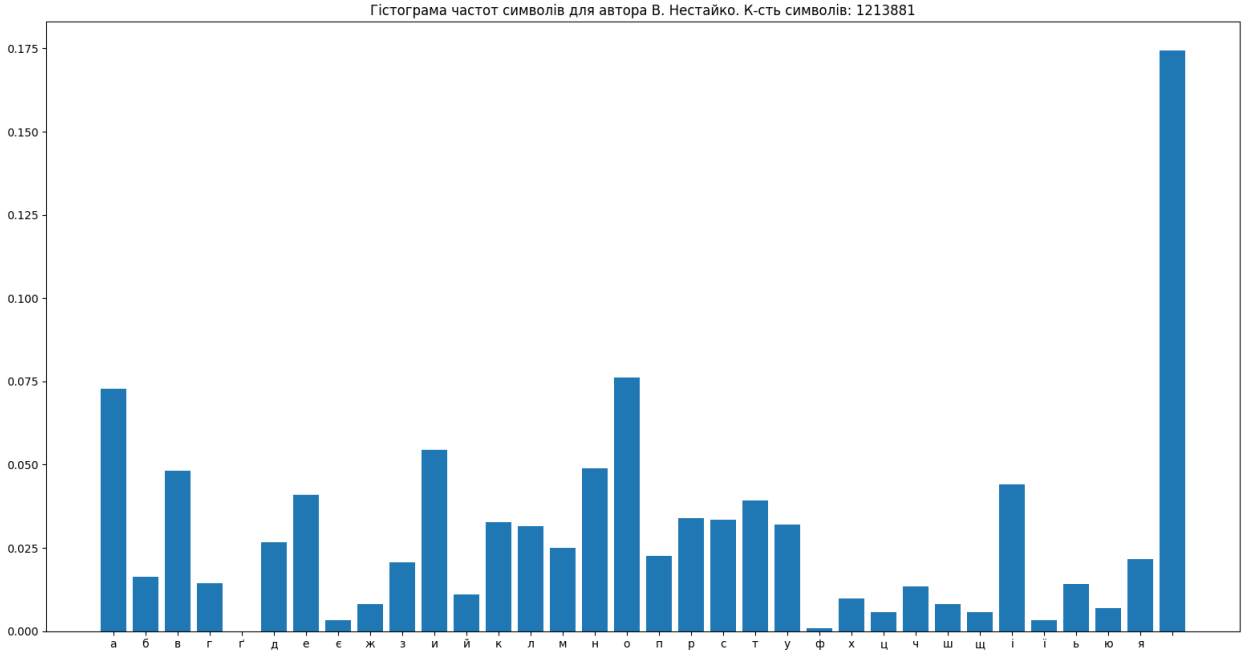

Рис. 2.12 - сумарна гістограма частот символів В. Нестайко

На рисунках 2.13-2.15 зображені теплові діаграми біграмів кожного автора.

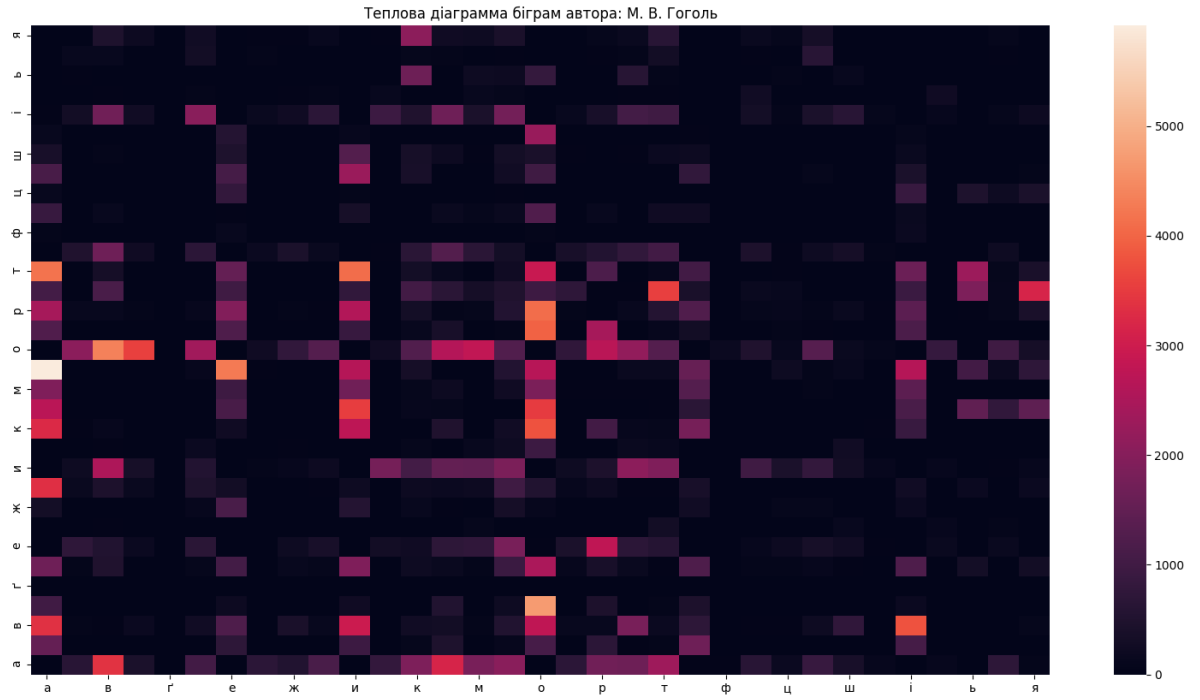

Рис. 2.13 - сумарна гістограма частот символів М. Гоголь

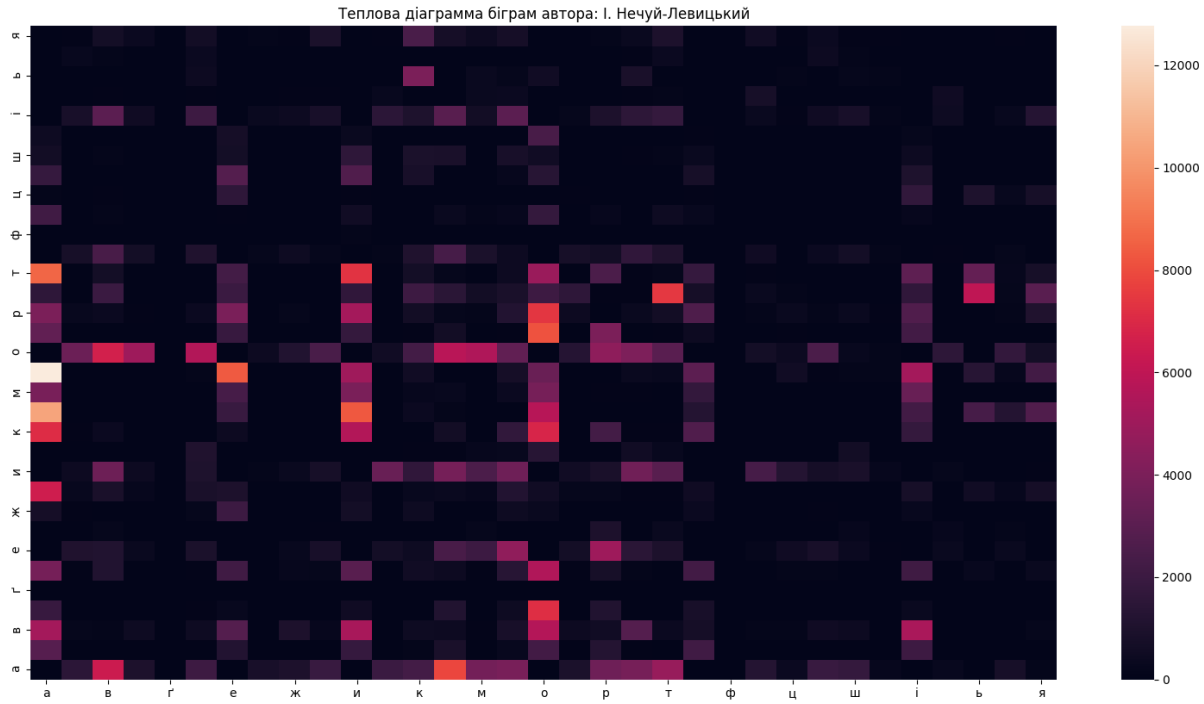

Рис. 2.14 - сумарна гістограма частот символів І. Нечуй-Левицький

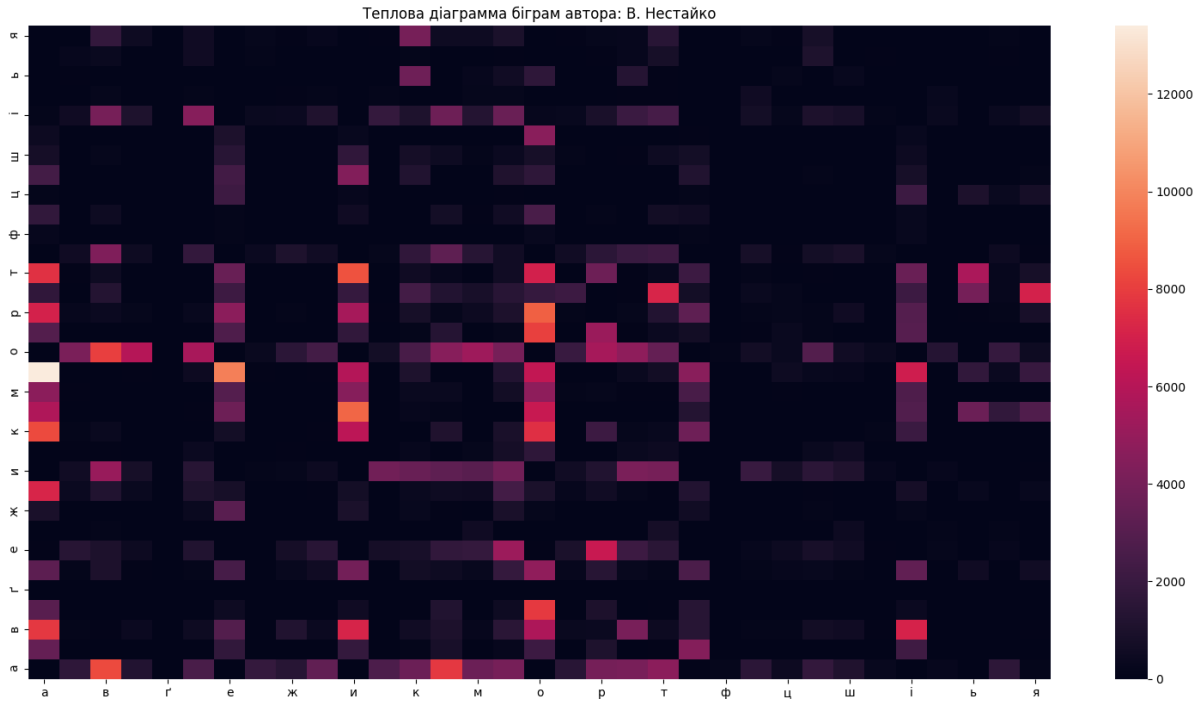

Рис. 2.15 - сумарна гістограма частот символів В. Нестайко

На рисунках 2.16-2.18 зображені гістограми різниці частот використання букв українського алфавіту для сукупного набору творів кожного автора та загальної частоти використання букв українського алфавіту.

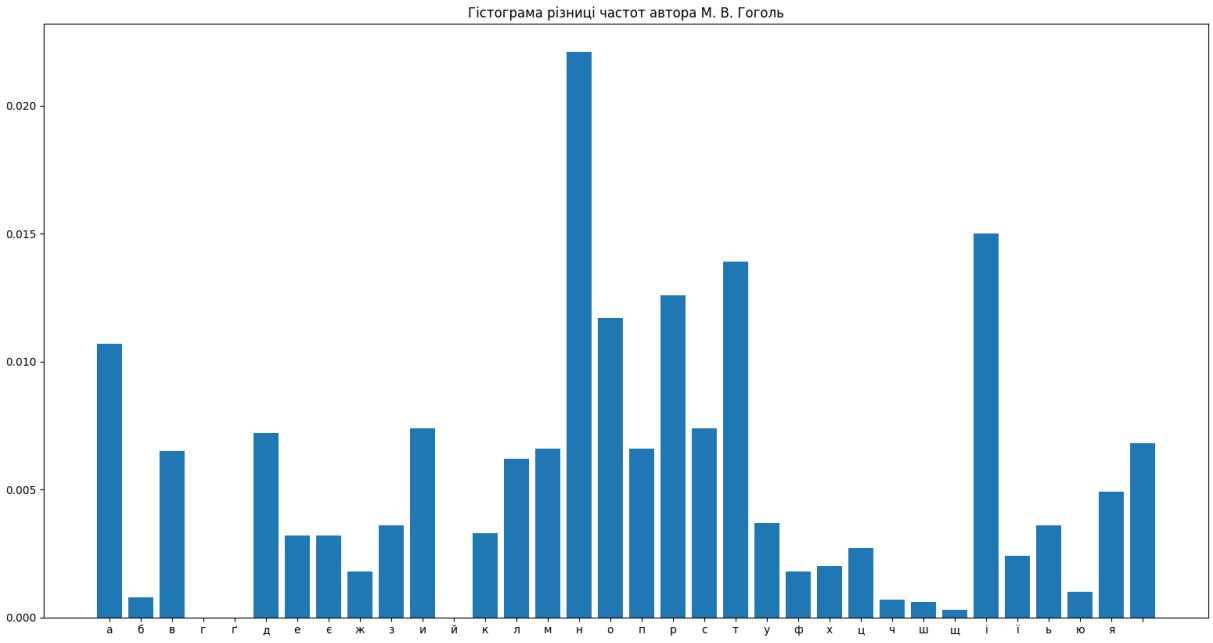

Рис. 2.16 - гістограма різниці частот М. Гоголя

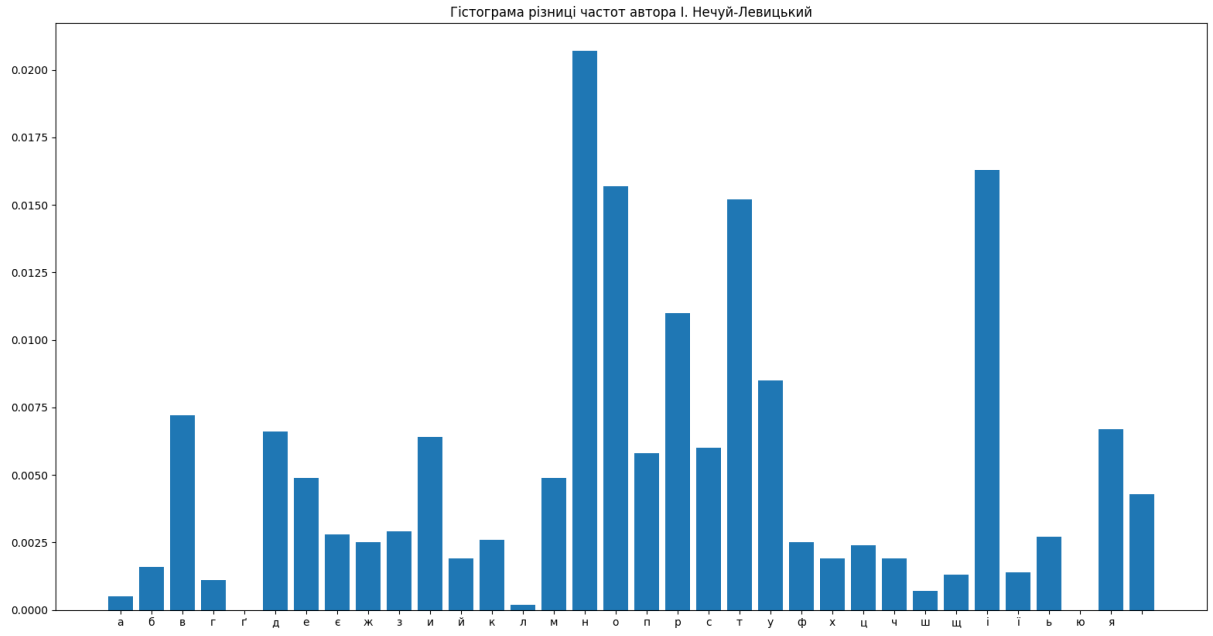

Рис. 2.17 - гістограма різниці частот І. Нечуй-Левицький

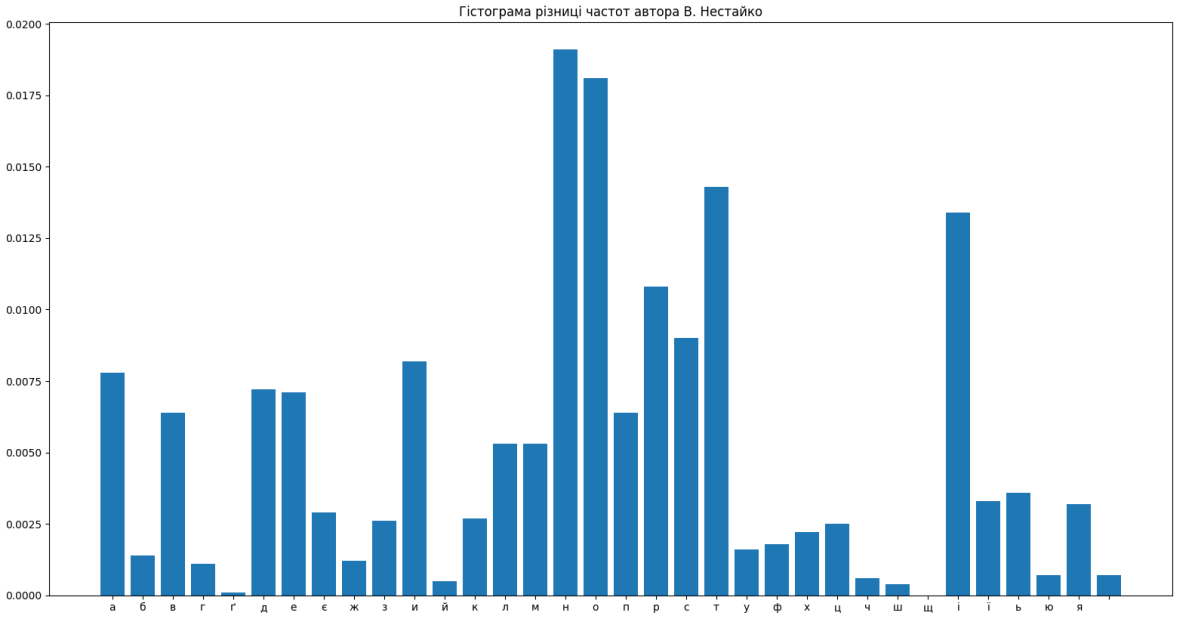

Рис. 2.18 - гістограма різниці частот В. Нестайко

На рисунках 2.19-2.21 зображено теплові діаграми різниці використання пар букв українського алфавіту для сукупного набору творів між кожною парою авторів.

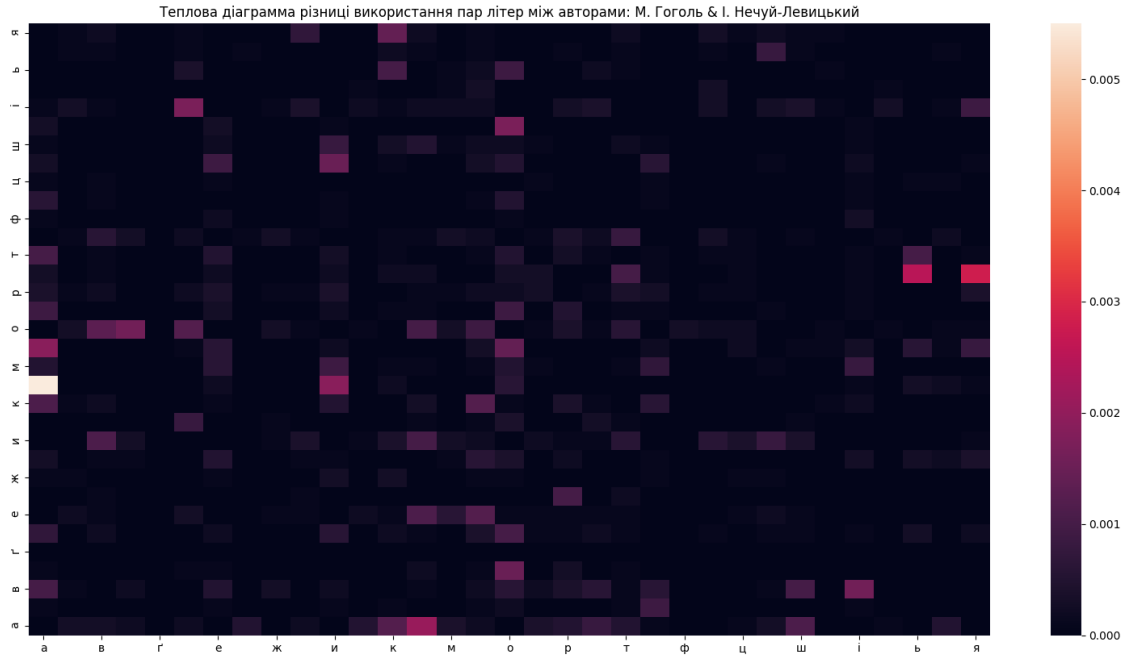

Рис. 2.19 - теплова діаграма між М. Гоголем та І. Нечуем-Левицьким

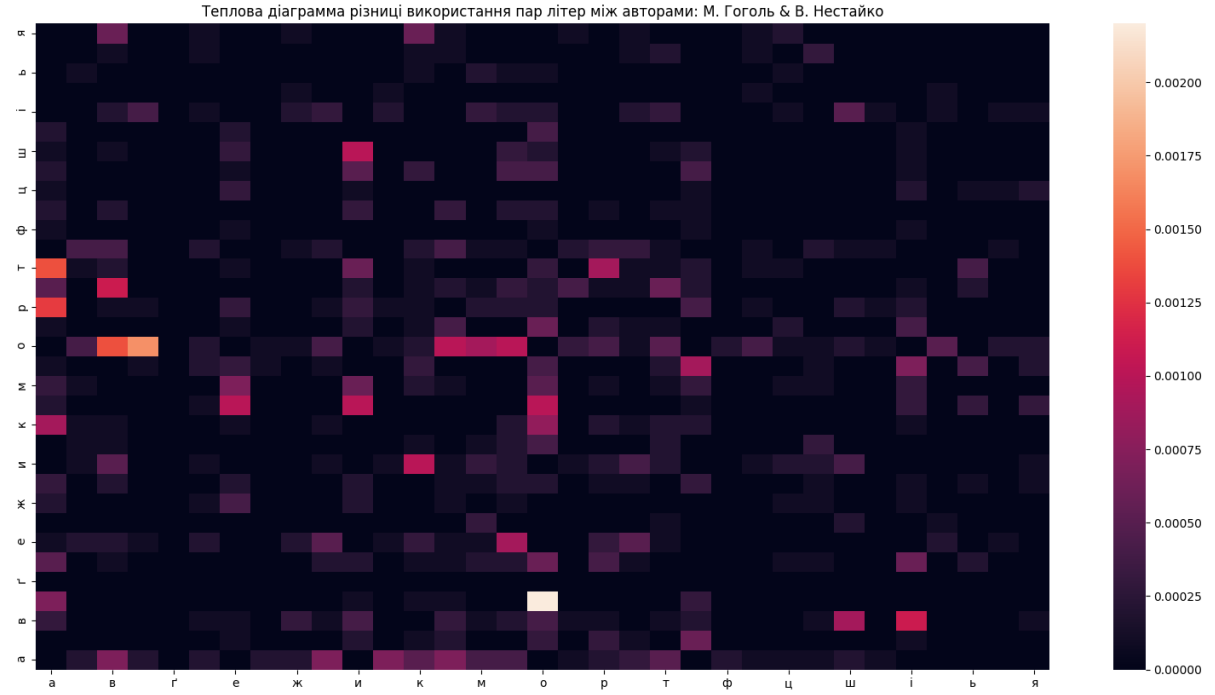

Рис. 2.20 - теплова діаграма між М. Гоголем та В. Нестайко

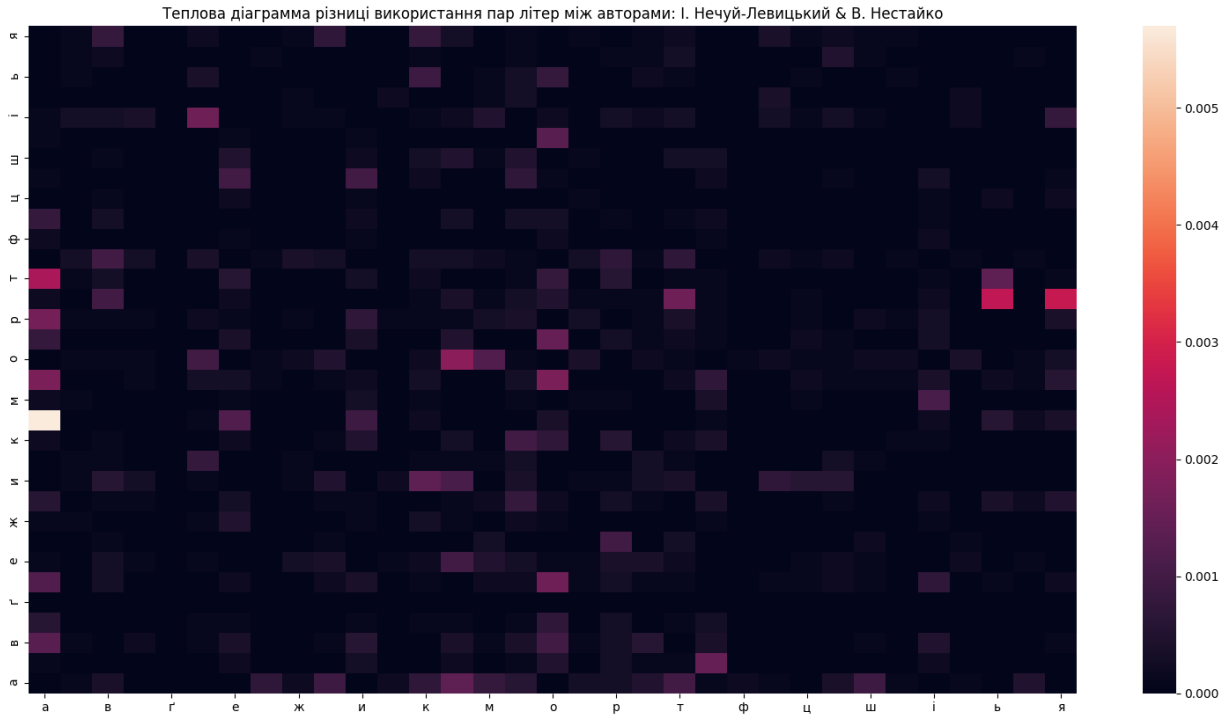

Рис. 2.21 - теплова діаграма між І. Нечуем-Левицьким та В. Нестайко

3. Визначення автора за невідомим текстом

Для дослідження можливості визначення автора невідомого тексту за допомогою порівняння функцій щільності грамів та біграмів було взято текст М. Гоголя “Вечір проти Івана Купала”.

У таблиці 3.1 наведені значення різниць між функціями щільності для грамів.

Таблиця 3.1 - визначення автора невідомого тексту за грамами

Розмір уривку тексту	М. Гоголь	І. Нечуй-Левицький	В. Нестайко	Визначено автора
5000	0.0695	0.1073	0.0841	М. Гоголь
10000	0.0549	0.0847	0.0571	М. Гоголь
25000	0.0385	0.0661	0.0437	М. Гоголь
50000	0.0357	0.0611	0.0449	М. Гоголь

У таблиці 3.2 наведені значення різниць між функціями щільності для біграмів.

Таблиця 3.2 - визначення автора невідомого тексту за біграмами

Розмір уривку тексту	М. Гоголь	І. Нечуй-Левицький	В. Нестайко	Визначено автора
5000	0.1813	0.2338	0.2055	М. Гоголь
10000	0.1371	0.197	0.1729	М. Гоголь
25000	0.0984	0.1621	0.1312	М. Гоголь
50000	0.0905	0.1568	0.1294	М. Гоголь

Висновок:

Виконавши статистичний аналіз обраних авторів та їх текстів були отримані характеристики частот грамів та біграмів. На теплових діаграмах порівнянь частотних значень біграм авторів (див. рис. 2.19-2.21) можна побачити, що Гоголь та Нестайко більш схожі, аніж Гоголь та Нечуй-Левицький.

В якості невідомого тексту було обрано твір М. Гоголь. Для кожного розміру уривка було правильно вгадано автора. Також, можна стверджувати, що при збільшенні розміру уривку характеристики тексту стають точнішими до еталону автора. Тобто різниця між функціями щільності зменшується. (див. табл. 3.1 та табл. 3.2.). Порівняння функцій щільності біграмів підтвердили рисунки теплограм, а саме - схожість авторів у порядку Гоголь -> Нестайко -> Нечуй-Левицький.

Додатки:

1. **Усі книги було взято з бібліотеки javalibre**
2. **Лістинг:**

Program.py

```
from analyze import *

def main():
  with open('books_source/gogol_viy.txt', 'r' ) as f:
    gogol_book_1 =  f.read()
    f.close()

  with open('books_source/gogol_vechori_na_hytori.txt', 'r') as f:
    gogol_book_2 =  f.read()
    f.close()

  with open('books_source/gogol_nich.txt', 'r') as f:
    gogol_book_3 =  f.read()
    f.close()



  gogol_books = [gogol_book_1, gogol_book_2, gogol_book_3]

  gogol_symbols_result =
open('analyze_results/gogol_symbols_result_1.txt', 'w')
  gogol_bigram_result = open('analyze_results/gogol_bigram_result_1.txt',
'w')
```

```
    gogol = Analyze(author='М. Гоголь', books_list=gogol_books)
    gogol.symbol_analyze(gogol_symbols_result, False)
    gogol.bigram_analyze(gogol_bigram_result)
    # gogol.heat_map()
    # gogol.symbol_substraction()

#-------------------------------------------------------------------------------------------
-----------------------------------------------#

    with open('books_source/nechui_netoi.txt', 'r') as f:
        nechui_book_1 =  f.read()
        f.close()

    with open('books_source/nechui_kaidash.txt', 'r') as f:
        nechui_book_2 =  f.read()
        f.close()

    with open('books_source/nechui_kniaz.txt', 'r') as f:
        nechui_book_3 =  f.read()
        f.close()


    nechui_books = [nechui_book_1, nechui_book_2, nechui_book_3]

    nechui_symbols_result =
open('analyze_results/nechui_symbols_result_1.txt', 'w')
```

```
  nechui_bigram_result =
open('analyze_results/nechui_bigram_result_1.txt', 'w')

  nechui = Analyze(author='І. Нечуй-Левицький',
books_list=nechui_books)
  nechui.symbol_analyze(nechui_symbols_result, False)
  nechui.bigram_analyze(nechui_bigram_result)
  # nechui.heat_map()
  # nechui.symbol_substraction()



#-------------------------------------------------------------------------------------------
-------------------------------------------------#

  with open('books_source/nestaiko_toreadori.txt', 'r') as f:
    nestaiko_book_1 =  f.read()
    f.close()

  with open('books_source/nestaiko_odunica.txt', 'r') as f:
    nestaiko_book_2 =  f.read()
    f.close()

  with open('books_source/nestaiko_zaichiki.txt', 'r') as f:
    nestaiko_book_3 =  f.read()
    f.close()
```

```
  nestaiko_books = [nestaiko_book_1, nestaiko_book_2, nestaiko_book_3]

  nestaiko_symbols_result =
open('analyze_results/nestaiko_symbols_result_1.txt', 'w')
  nestaiko_bigram_result =
open('analyze_results/nestaiko_bigram_result_1.txt', 'w')

  nestaiko = Analyze(author='В. Нестайко', books_list=nestaiko_books)
  nestaiko.symbol_analyze(nestaiko_symbols_result, False)
  nestaiko.bigram_analyze(nestaiko_bigram_result)
  # nestaiko.heat_map()
  # nestaiko.symbol_substraction()

#-------------------------------------------------------------------------------------
-------------------------------------------------#

  authors_bigram_comprassion(gogol, nechui)
  authors_bigram_comprassion(gogol, nestaiko)
  authors_bigram_comprassion(nechui, nestaiko)

#-------------------------------------------------------------------------------------
-------------------------------------------------#

  with open('books_source/gogol_vechir.txt', 'r') as f:
    gogol_unknow_book =  f.read(50000)
    f.close()
```

```
  gogol_unknow_symbols =
open('analyze_results/gogol_unknown_result.txt', 'w')
  gogol_unknow_bigram =
open('analyze_results/gogol_unknown_bigram.txt', 'w')

  unknown_gogol = Analyze(author='Невідомий',
books_list=[gogol_unknow_book])
  unknown_gogol.symbol_analyze(gogol_unknow_symbols, False)
  unknown_gogol.bigram_analyze(gogol_unknow_bigram)

  authors_symbol_density = [gogol.alphabet_freq_dict,
nechui.alphabet_freq_dict, nestaiko.alphabet_freq_dict]
  authors_bigram_density = [gogol.bigram_freq_dict,
nechui.bigram_freq_dict, nestaiko.bigram_freq_dict]
  authors_names = [gogol.name, nechui.name, nestaiko.name]

  #guess_book_symbols(unknown_gogol.alphabet_freq_dict,
authors_symbol_density, authors_names) # symbol guessing
  #guess_book_symbols(unknown_gogol.bigram_freq_dict,
authors_bigram_density, authors_names) # bigram guessing

#-----------------------------------------------------------------------------------------
-------------------------------------------------#

def authors_bigram_comprassion(author1, author2):
  author1_bigram = author1.bigram_freq_dict
  author2_bigram = author2.bigram_freq_dict
```

```
  string_alphabet = 'абвгґдеєжзийклмнопрстуфхцчшщїіьюя'
  bigram_matrix = np.array([[.0]*33]*33)
  for first in string_alphabet:
    for second in string_alphabet:
      key = first + second
      i = string_alphabet.index(first)
      j = string_alphabet.index(second)
      if key in author1_bigram and key in author2_bigram:
        bigram_matrix[i][j] =
abs(author1_bigram[key]-author2_bigram[key])
      elif key in author1_bigram and author1_bigram[key] != .0:
          bigram_matrix[i][j] = author1_bigram[key]
      elif key in author2_bigram and author2_bigram[key] != .0:
          bigram_matrix[i][j] = author2_bigram[key]


  sns.heatmap(pd.DataFrame(data=bigram_matrix,
columns=author1.col_name, index=author1.row_name)).invert_yaxis()
  plt.title(f'Теплова діаграмма різниці використання пар літер між
авторами: {author1.name} & {author2.name}')
  plt.show()

#-----------------------------------------------------------------------------------
-----------------------------------------------#

def guess_book_symbols(unknow_book_density, density_list, authors_list):
```

```
  density_diff = [.0]*len(density_list)

  for i in range(len(density_list)):
    for letter in unknow_book_density:
      if letter in density_list[i]:
        density_diff[i] += (abs(unknow_book_density[letter] -
density_list[i][letter]))
    for num in range(len(density_diff)):
      density_diff[num] = round(density_diff[num], 5)
  min_dif = density_diff.index(min(density_diff))
  print(f"\nМожливі автори: {authors_list} "
      f"\nРізниця між функціями щільностей невідомого тексту і
авторів: {density_diff}"
      f"\nАвтор невідомого тексту: {authors_list[min_dif]}")



main()
```

Analyze.py

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

class Analyze(object):
   def __init__(self, author, books_list):
      self.name = author
```

```
        self.books = books_list
        self.bigram_matrix = np.array([[0]*33]*33)
        self.string_alphabet = 'абвгґдеєжзийклмнопрстуфхцчшщіїьюя'
        self.ukrainian_freq_dict = {'а': 0.0807, 'б': 0.0177, 'в': 0.0545, 'г':
0.0154, 'ґ': 0.0001, 'д': 0.0338, 'е': 0.0338, 'є': 0.0061,
                        'ж': 0.0093, 'з': 0.0232, 'и': 0.0626, 'й': 0.0116, 'к': 0.0354,
'л': 0.0369, 'м': 0.0303, 'н': 0.0681,
                        'о': 0.0942, 'п': 0.0290, 'р': 0.0448, 'с': 0.0424, 'т': 0.0535,
'у': 0.0336, 'ф': 0.0028, 'х': 0.0119,
                        'ц': 0.0083, 'ч': 0.0141, 'ш': 0.0076, 'щ': 0.0056, 'і': 0.0575,
'ї': 0.0065, 'ь': 0.0177, 'ю': 0.0061,
                        'я': 0.0248, ' ': 0.175}
        self.alphabet_dict = dict()
        self.alphabet_freq_dict = dict()
        self.bigram_dict = dict()
        self.bigram_freq_dict = dict()
        self.col_name = []
        self.row_name = []
        for i in self.string_alphabet:
            self.alphabet_dict[i] = 0
            self.col_name.append(i)
            self.row_name.append(i)
        self.alphabet_dict[' '] = 0
        self.count_symbols = 0


    def symbol_analyze(self, result_file, flag):
```

```
        for text in self.books:
            for letter in text.lower():
                if letter in self.alphabet_dict:
                    self.alphabet_dict[letter] += 1
                    self.count_symbols += 1

        for letter in self.alphabet_dict:
            self.alphabet_freq_dict[letter] =
round(self.alphabet_dict[letter]/self.count_symbols, 4)
        if flag is True:
            plt.bar(self.alphabet_freq_dict.keys(),
self.alphabet_freq_dict.values())
            plt.title(f'Гістограма частот символів для автора {self.name}.
К-сть символів: {self.count_symbols}')
            plt.show()
        result_file.write(str(self.alphabet_freq_dict))



    def symbol_substraction(self):
        difference_dict = self.alphabet_freq_dict
        for letter in difference_dict:
            difference_dict[letter] = abs(difference_dict[letter] -
self.ukrainian_freq_dict[letter])

        plt.bar(difference_dict.keys(), difference_dict.values())
        plt.title(f'Гістограма різниці частот автора {self.name}')
        plt.show()
```

```
    def bigram_analyze(self, result_file):
        for text in self.books:
            for i in range(len(text)-1):
                symbol = text[i].lower()
                next_symbol = text[i+1].lower()
                if symbol != ' ' and next_symbol != ' ':
                    if symbol in self.alphabet_dict and next_symbol in
self.alphabet_dict:
                        bigram = symbol + next_symbol
                        if bigram in self.bigram_dict:
                            self.bigram_dict[bigram] += 1
                        else:
                            self.bigram_dict[bigram] = 1

        for bigram in self.bigram_dict:
            self.bigram_freq_dict[bigram] =
round(self.bigram_dict[bigram]/self.count_symbols, 4)

        result_file.write(str(sorted(self.bigram_freq_dict.items())))

    def heat_map(self):
        for first in self.string_alphabet:
            for second in self.string_alphabet:
                key = first + second
                if key in self.bigram_dict:
```

```
                i = self.string_alphabet.index(first)
                j = self.string_alphabet.index(second)
                self.bigram_matrix[i][j] = self.bigram_dict[key]

        sns.heatmap(pd.DataFrame(data=self.bigram_matrix,
columns=self.col_name, index=self.row_name)).invert_yaxis()
        plt.title(f'Теплова діаграмма біграм автора: {self.name}')
        plt.show()
```