



МІНІСТЕРСТВО ОСВІТИ, НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»  
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

**СИМЕТРИЧНА КРИПТОГРАФІЯ**  
**КОМП'ЮТЕРНИЙ ПРАКТИКУМ №1**

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

Підготували:  
студенти 3 курсу  
групи ФІ-84  
Ковальчук Ольга Миронівна  
Коломієць Андрій Юрійович

Викладач:

# Експериментальна оцінка ентропії на символ джерела відкритого тексту

## 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

1. Написати програми для підрахунку частот букв і частот біграм в тексті, а також підрахунку  $H_1$  та  $H_2$  за безпосереднім означенням. Підрахувати частоти букв та біграм, а також значення  $H_1$  та  $H_2$  на довільно обраному тексті російською мовою достатньої довжини (щонайменше 1Мб), де імовірності замінити відповідними частотами. Також одержати значення  $H_1$  та  $H_2$  на тому ж тексті, в якому вилучено всі пробіли.
2. За допомогою програми *CoolPinkProgram* оцінити значення  $H^{(10)}$ ,  $H^{(20)}$ ,  $H^{(30)}$ .
3. Використовуючи отримані значення ентропії, оцінити надлишковість російської мови в різних моделях джерела.

## Методичні вказівки

Звичайні текстові файли містять багато символів окрім власне літер; для обчислення значень ентропій вони повинні пройти попередню фільтрацію: всі символи, окрім текстових, повинні вилучатись або замінюватись на пробіли; прописні літери – замінюватись на відповідні стрічні; послідовність пробілів (або інших розділових знаків, наприклад, символів кінця рядку) повинна трактуватись як один пробіл або вилучатись, якщо пробіл не входить до алфавіту.

При підрахунку частот біграм треба розглядати як пари букв, що перетинаються, так і пари букв, що не перетинаються (тобто рухатися вздовж тексту з кроком 2). Одержані результати не повинні суттєво відрізнятись, однак в першому випадку використовується більше статистики, а тому чисельні дані більш точні. Таблицю частот символів потрібно подавати відсортованою за спаданням частот. Таблицю частот біграм зручно подавати у вигляді квадратної матриці, індексованої першою та другою літерами біграм.

Програма *CoolPinkProgram* використовує текст, що лежить у допоміжному файлі *text*. Цей текст написаний російською мовою без знаків пунктуації та великих літер; буква «ё» замінена буквою «е», а «ъ» – буквою «ь». Пробіл також вважається буквою. Таким чином, кількість букв алфавіту  $m=32$ . При підрахунку  $H^{(10)}$ ,  $H^{(20)}$ ,  $H^{(30)}$  необхідно виконати не менш ніж 50 експериментів.

## Хід роботи

### Робота з програмою *CoolPinkProgram*

За допомогою програми *CoolPinkProgram* оцінив значення  $H^{(10)}$ ,  $H^{(20)}$ ,  $H^{(30)}$ . При користуванні зазначеною програмою, намагався вгадувати наступну букву, тому що залежності в тексті змінюють ентропію, а обраховані умовні ентропії n-грам стають наближеними до теоретичних.

Оцінимо надлишковості R російської мови у різних моделях відкритого тексту:

$$(1 - (2.93591064588505) / (\log_2 32)) > R_{10} > (1 - (3.750351720139085) / (\log_2 32)) ;$$

$$0.412817870823 > R_{10} > 0.249929655972 ;$$

$$(1 - (2.0537162909724) / (\log_2 32)) > R_{20} > (1 - (2.83266429434274) / (\log_2 32)) ;$$

$$0.589256741806 > R_{20} > 0.433467141131 ;$$

$$(1 - (2.18649268242094) / (\log_2 32)) > R_{30} > (1 - (3.02491528974128) / (\log_2 32)) ;$$

$$0.562701463516 > R_{30} > 0.395016942052 ;$$

Відповідні скріншоти експериментів.

The screenshot shows the 'Лабораторная работа №1' window of the CoolPinkProgram. It displays the results of an experiment on the Russian text '...к\_следую...'. The interface includes a list of n-gram orders (5 to 50 symbols), a list of used letters, and a list of probabilities for each letter. The 'Неравенство для энтропии' (Entropy inequality) is shown as  $2.93591064588505 < H < 3.70351720139085$ . The 'Двоичная таблица угаданных символов' (Binary table of guessed symbols) is a 32x32 matrix of 0s and 1s. The 'Вероятности' (Probabilities) list shows the probability for each letter from 'a' to 'z' and space, with values ranging from 0 to 0.3050847.

| Порядок n-граммы: | Введенный символ: | Неравенство для энтропии:                 |
|-------------------|-------------------|---|
| 5 символов        |                   |   |
| 10 символов       |                   | $2.93591064588505 < H < 3.70351720139085$ |
| 15 символов       |                   |   |
| 20 символов       |                   |   |
| 25 символов       |                   |   |
| 30 символов       |                   |   |
| 35 символов       |                   |   |
| 40 символов       |                   |   |
| 45 символов       |                   |   |
| 50 символов       |                   |   |

| Вероятности:       |
|--------------------|
| $q[1] = 0.3050847$ |
| $q[2] = 0.1355932$ |
| $q[3] = 0.0677966$ |
| $q[4] = 0.0169491$ |
| $q[5] = 0.0169491$ |
| $q[6] = 0.0338983$ |
| $q[7] = 0.0338983$ |
| $q[8] = 0.0338983$ |
| $q[9] = 0.0169491$ |
| $q[10] = 0.050847$ |
| $q[11] = 0.016949$ |
| $q[12] = 0$        |
| $q[13] = 0.033898$ |
| $q[14] = 0$        |
| $q[15] = 0.033898$ |
| $q[16] = 0.016949$ |
| $q[17] = 0$        |
| $q[18] = 0$        |
| $q[19] = 0.033898$ |
| $q[20] = 0.050847$ |
| $q[21] = 0.016949$ |
| $q[22] = 0.016949$ |
| $q[23] = 0.016949$ |
| $q[24] = 0$        |
| $q[25] = 0$        |
| $q[26] = 0$        |
| $q[27] = 0.016949$ |
| $q[28] = 0$        |
| $q[29] = 0$        |
| $q[30] = 0.016949$ |
| $q[31] = 0.016949$ |
| $q[32] = 0$        |



## Опис програмної реалізації

Згідно постановкам задачі, практикум вимагає навичок володіння роботи з файлами на мові програмування C++. Аналіз тексту здійснюється по одному рядку, при такій реалізації не має потреби в збереженні великого тексту в яких-небудь великих масивах даних.

Фрагмент коду з такою обробкою тексту, в якості прикладу (викликані функції в циклі можуть комбінуватися, тому приклади їх реалізації будуть наведені в подальшому):

```
while (!fin.eof())
{
    getline(fin, one_string);

    one_string = filtration_the_text_with_space(one_string);

    if (one_string.length() <= 1)
    {
        continue;
    }

    matrix_frequancy_with_step_one(matrix, 34, 34, one_string);
}
```

Найбільші труднощі, які виникають при програмній реалізації - це фільтрація даних. Для цього необхідно реалізувати дві функції котрі видалятимуть пробіли взагалі, або зайві, та дві функції, що видаляють непотрібні символи, які не входять до алфавіту включаючи всі пробіли також або зайві.

Реалізація функції видалення всіх пробілів:

```
string remove_all_spaces(string str)
{
    str.erase(remove(str.begin(), str.end(), ' '), str.end());
    return str;
}
```

Реалізація функції видалення зайвих пробілів:

```
string remove_extra_space(string text)
{
    int i, j; int len = text.length();

    for (i = 0; i < len; i++)
    {
        if (text[i] == ' ')
        {
            for (i = 0; i < (len - 1); i++)
                text[i] = text[i + 1];
            text[i] = '\0';
            len--;
            i = -1;
            continue;
        }
        if (text[i] == ' ' && text[i + 1] == ' ')
        {
            for (j = i; j < (len - 1); j++)
            {
                text[j] = text[j + 1];
            }
            text[j] = '\0';
            len--;
            i--;
        }
    }
}
```

```

    return text;
}

```

Вищезазначені функції використовувалися для фільтрації тексту, від непотрібних символів, в специфічних випадках, які вимагає саме завдання. Наведемо дві з них.

**Реалізація функції фільтрації тексту без пробілів:**

```

string filtration_the_text_without_space(string text)
{
    // incurrent string of text 34 symbols

    string upper_alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ";
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

    for (type i = 0; i < text.length(); i++)
    {
        for (type j = 0; j < 34; j++)
        {
            if (text[i] == lower_alphabet[j])
            {
                text[i] = lower_alphabet[j];
                break;
            }
            if (text[i] == upper_alphabet[j])
            {
                text[i] = lower_alphabet[j];
                break;
            }
            if (j == 33 && text[i] != lower_alphabet[j])
            {
                text[i] = ' ';
                break;
            }
        }
    }

    return remove_all_spaces(text);
}

```

**Реалізація функції фільтрації тексту з пробілами:**

```

string filtration_the_text_with_space(string text)
{
    // incurrent string of text 34 symbols

    string upper_alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ";
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

    for (type i = 0; i < text.length(); i++)
    {
        for (type j = 0; j < 34; j++)
        {
            if (text[i] == lower_alphabet[j])
            {
                text[i] = lower_alphabet[j];
                break;
            }
            if (text[i] == upper_alphabet[j])
            {
                text[i] = lower_alphabet[j];
                break;
            }
        }
    }
}

```

```

        if (j == 33 && text[i] != lower_alphabet[j])
        {
            text[i] = ' ';
            break;
        }
    }

    return remove_extra_space(text);
}

```

Організація фільтрації на цьому завершується.

Далі необхідно накопичити данні про кількість символів в тексті. Для роботи з монограмами, використовувалася структура, щоб можна було здійснити сортування не тільки частот, а й символів, що відповідають вказаним частотам:

```

struct alphabet
{
    char symbols;
    long int value;
};

```

Початкова ініціалізація об'єктів таких структур має містити лише алфавіт символів (впорядкованих за алфавітом), та відсутність будь-яких частот:

```

alphabet* null(alphabet* language, string upper_alphabet, type size)
{
    for (type i = 0; i < size; i++)
    {
        language[i].symbols = upper_alphabet[i]; language[i].value = 0;
    }
    return language;
}

```

Потрібно винести насамперед часто вживані конструкції, наприклад звернення до алфавіту:

```

string upper_alphabet()
{
    return "АВВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ_";
}

```

На основі такої структури можна накопичувати частоти про символи. Ми будемо використовувати функції, котрі будуть заносити до об'єкту структури дані.

Обрахунок частот символів монограм:

```

alphabet* frequency_of_symbols(alphabet* language, type size, string
text)
{
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

    for (type i = 0; i < text.length(); i++)
    {
        for (type j = 0; j < size; j++)
        {
            if (text[i] == lower_alphabet[j])

```

```

        {
            language[j].value = language[j].value + 1;
            break;
        }
    }

    return language;
}

```

При виведенні таблиці частот символів (монограм), їх необхідно організувати впорядковано за алфавітом, та частотами. За частотами, сортування виконувалося методом «бульбашки».

Реалізація методу «бульбашки»:

```

void bubbleSort(alphabet* language_, type length_array)
{
    type temp_ = 0;
    char _temp = 0;
    bool exit = false;

    while (!exit)
    {
        exit = true;

        for (type int_counter = 0; int_counter < (length_array - 1); int_counter++)

            if (language_[int_counter].value < language_[int_counter + 1].value)
            {
                temp_ = language_[int_counter].value;
                language_[int_counter].value = language_[int_counter + 1].value;
                language_[int_counter + 1].value = temp_;

                _temp = language_[int_counter].symbols;
                language_[int_counter].symbols = language_[int_counter + 1].symbols;
                language_[int_counter + 1].symbols = _temp;

                exit = false;
            }
    }
}

```

Реалізація обрахунку ентропії монограм з пробілами:

```

long double entropy_of_alphabet(alphabet* language, type length, type size)
{
    long double result = 0;

    for (type i = 0; i < size; i++)
    {
        if (language[i].value != 0)
        {
            long double temp = 0;

            temp = ((long double)language[i].value / length);

            result = -temp * log2(temp) + result;
        }
    }
}

```



```

    }

    return result;
}

```

**Реалізація обрахунку ентропії монограм без пробілів:**

```

long double entropy_of_alphabet_without_struck(alphabet* language, type
length, type size)
{
    long double result = 0;

    for (type i = 0; i < size; i++)
    {
        if (language[i].value != 0 && language[i].symbols != ' ')
        {
            long double temp = 0;

            temp = ((long double)language[i].value / length);

            result = -temp * log2(temp) + result;
        }
    }

    return result;
}

```

Організація процесу підрахунку частот та ентропії алфавіту на цьому завершується.

Далі необхідно накопичити данні про кількість біграм в тексті. Структурами ми не користуємося, лише певним двовимірним динамічним масивом. В динамічний масив можна заносити кількості частот біграм. При роботі користуємося алгоритмами наведеними нижче.

**Обрахунок частот символів для біграм з кроком один, тобто з перетином:**

```

long double** matrix_frequency_with_step_one(long double** matrix, type
row, type column, string text)
{
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчщъыьэюя ";

    for (type t = 0; t < text.length() - 1; t++)
    {
        type temp_i = 0;

        type temp_j = 0;

        for (type i = 0; i < row; i++)
        {
            if (text[t] == lower_alphabet[i])
            {
                temp_i = i;
                break;
            }
        }

        for (type j = 0; j < column; j++)
        {
            if (text[t + 1] == lower_alphabet[j])
            {
                temp_j = j;
            }
        }
    }
}

```

```

        break;
    }
}

matrix[temp_i][temp_j] = matrix[temp_i][temp_j] + 1;

temp_i = 0;

temp_j = 0;

}

return matrix;
}

```

**Обрахунок частот символів для біграм з кроком два, тобто без перетину:**

```

long double** matrix_frequency_with_step_two(long double** matrix, type
row, type column, string text)
{
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчщъыьэюя ";

    for (type t = 0; t < text.length() - 1; t = t + 2)
    {
        type temp_i = 0;

        type temp_j = 0;

        for (type i = 0; i < row; i++)
        {
            if (text[t] == lower_alphabet[i])
            {
                temp_i = i;
                break;
            }
        }

        for (type j = 0; j < column; j++)
        {
            if (text[t + 1] == lower_alphabet[j])
            {
                temp_j = j;
                break;
            }
        }

        matrix[temp_i][temp_j] = matrix[temp_i][temp_j] + 1;

        temp_i = 0;

        temp_j = 0;

    }

    return matrix;
}

```

Реалізація обрахунку ентропії біграм :

```
long double entropy_of_alphabet_bigram(long double**matrix, type length,
type size)
{
    long double result = 0;

    for (type i = 0; i < size; i++)
    {
        for (type j = 0; j < size; j++)
        {
            if (matrix[i][j] != 0)
            {
                long double temp = 0;

                temp = ((long double)matrix[i][j] / length);

                result = -temp * log2(temp) + result;
            }
        }
    }

    return result;
}
```

Як виводити такі результати?

Необхідно спочатку визначитися зі структурою програми. Вона має обчислювати частоти монограм та біграм. При підрахунку частот біграм треба також розглядати як пари букв, що перетинаються, так і пари букв, що не перетинаються (тобто рухатися вздовж тексту з кроком 2). Одержані результати не повинні суттєво відрізнятись, однак в першому випадку використовується більше статистики, а тому чисельні дані більш точні вийдуть.

Загалом структура даних при виводі біграм наступна:

- рух з кроком один, тобто з перетином біграм:
  - a. з пробілом реалізація;
  - b. без пробілу реалізація;
- рух з кроком два, тобто без перетину біграм:
  - a. з пробілом реалізація;
  - b. без пробілу реалізація.

Вивід приблизно співпадатиме з такими підпунктами, але спочатку ми виведемо всі кроки з пробілами, а тоді всі кроки без пробілів. Під всіма кроками, мається на увазі крок з перетином та без перетину. Вивід можна оглянути в загальному коді.

Також потрібно було зробити програму швидку. Для цього можна використовувати примітивні кроки:

- при пошуку літери в алфавіті, сам алфавіт можна розглядати, як впорядкованим за частотами, така реалізація значно пришвидшує роботу, тому що на пошук часто вживаних літер потрібно вже менше часу, а літери з малою імовірністю появи потрібно менш часто відшукувати;

- при знаходженні літери потрібно використовувати оператор `break`,аби не робити пошук по інших літерах алфавіту.

Такі прості вимоги забезпечують задовільну швидкодію, навіть якщо кожен рядок фільтрувати від непотрібних символів функціями з бібліотек стандартних, та алгоритмами, які самі по собі можуть не одноразово копіювати ціле речення.

## Загальний код програми

```
        //...Link on repository, if you have a access...//
        /*
Clone with HTTPS: https://gitlab.com/andrew.kolomiets/symetric-cryptography-labratory-1.git

Clone with SSH: git@gitlab.com:andrew.kolomiets/symetric-cryptography-labratory-1.git

        */
```

**"My\_function\_library.h"**

```
#pragma once

/*
Help information:
-Russian alphabet with little symbol: абвгдеёжзийклмнопрстуфхцщъыьэюя
-Russian alphabet with big symbol: АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЩЪЫЬЭЮЯ
-Russian alphabet in frequency order: оeanитслвркдмупьяыгбчзжйшхюэщцфтьё
-power of Russian alphabet:33
*/

#include <iostream>
#include <string>
#include <fstream>
#include <algorithm>
#include <Windows.h>
#include <cmath>
#include <iomanip>
#include <ctime>
#include <cctype>

using namespace std;

#define type unsigned int

struct alphabet
{
    char symbols;
    long int value;
};

string upper_alphabet()
{
    return "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЩЪЫЬЭЮЯ_";
}
```

```

}

string remove_all_spaces(string str)
{
    str.erase(remove(str.begin(), str.end(), ' '), str.end());
    return str;
}

string remove_extra_space(string text)
{
    int i, j;

    int len = text.length();

    for (i = 0; i < len; i++)
    {
        if (text[i] == ' ')
        {
            for (i = 0; i < (len - 1); i++)
                text[i] = text[i + 1];
            text[i] = '\\0';
            len--;
            i = -1;
            continue;
        }
        if (text[i] == ' ' && text[i + 1] == ' ')
        {
            for (j = i; j < (len - 1); j++)
            {
                text[j] = text[j + 1];
            }
            text[j] = '\\0';
            len--;
            i--;
        }
    }

    return text;
}

string filtration_the_text_without_space(string text)
{
    // incurrent string of text 34 symbols

    string upper_alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ";
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

```

```

for (type i = 0; i < text.length(); i++)
{
    for (type j = 0; j < 34; j++)
    {
        if (text[i] == lower_alphabet[j])
        {
            text[i] = lower_alphabet[j];
            break;
        }
        if (text[i] == upper_alphabet[j])
        {
            text[i] = lower_alphabet[j];
            break;
        }
        if (j == 33 && text[i] != lower_alphabet[j])
        {
            text[i] = ' ';
            break;
        }
    }
}

return remove_all_spaces(text);
}

string filtration_the_text_with_space(string text)
{
    // incurrent string of text 34 symbols

    string upper_alphabet = "АВВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ";
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

    for (type i = 0; i < text.length(); i++)
    {
        for (type j = 0; j < 34; j++)
        {
            if (text[i] == lower_alphabet[j])
            {
                text[i] = lower_alphabet[j];
                break;
            }
            if (text[i] == upper_alphabet[j])
            {
                text[i] = lower_alphabet[j];
                break;
            }
            if (j == 33 && text[i] != lower_alphabet[j])
            {
                text[i] = ' ';
                break;
            }
        }
    }
}

```

```

    }

    return remove_extra_space(text);
}

alphabet* freuquency_of_symbols(alphabet* language, type size, string text)
{
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

    for (type i = 0; i < text.length(); i++)
    {
        for (type j = 0; j < size; j++)
        {
            if (text[i] == lower_alphabet[j])
            {
                language[j].value = language[j].value + 1;
                break;
            }
        }
    }

    return language;
}

alphabet* null(alphabet* language, string upper_alphabet, type size)
{
    for (type i = 0; i < size; i++)
    {
        language[i].symbols = upper_alphabet[i]; language[i].value = 0;
    }

    return language;
}

void bubbleSort(alphabet* language_, type length_array)
{
    type temp_ = 0;
    char _temp = 0;
    bool exit = false;

    while (!exit)
    {
        exit = true;
        for (type int_counter = 0; int_counter < (length_array - 1); int_counter++)

            if (language_[int_counter].value < language_[int_counter + 1].value)
            {

                temp_ = language_[int_counter].value;

```



```

        language_[int_counter].value = language_[int_counter + 1].value;
        language_[int_counter + 1].value = temp_;

        temp_ = language_[int_counter].symbols;
        language_[int_counter].symbols = language_[int_counter + 1].symbols;
        language_[int_counter + 1].symbols = temp_;

        exit = false;
    }
}

long double** null_matrix(long double** matrix, int size)
{
    for (int i = 0; i < size; i++)
    {
        for (type j = 0; j < size; j++)
        {
            matrix[i][j] = 0;
        }
    }

    return matrix;
}

long double** matrix_frequancy_with_step_one(long double** matrix, type row, type column, string text)
{
    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя ";

    for (type t = 0; t < text.length() - 1; t++)
    {
        type temp_i = 0;

        type temp_j = 0;

        for (type i = 0; i < row; i++)
        {
            if (text[t] == lower_alphabet[i])
            {
                temp_i = i;
                break;
            }
        }

        for (type j = 0; j < column; j++)
        {
            if (text[t + 1] == lower_alphabet[j])
            {
                temp_j = j;
                break;
            }
        }
    }
}

```

```

        }
    }

    matrix[temp_i][temp_j] = matrix[temp_i][temp_j] + 1;

    temp_i = 0;

    temp_j = 0;

}

return matrix;

}

long double** matrix_frequency_with_step_two(long double** matrix, type row, type column, string text)
{

    string lower_alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэя ";

    for (type t = 0; t < text.length() - 1; t = t + 2)
    {
        type temp_i = 0;

        type temp_j = 0;

        for (type i = 0; i < row; i++)
        {
            if (text[t] == lower_alphabet[i])
            {
                temp_i = i;
                break;
            }
        }

        for (type j = 0; j < column; j++)
        {
            if (text[t + 1] == lower_alphabet[j])
            {
                temp_j = j;
                break;
            }
        }

        matrix[temp_i][temp_j] = matrix[temp_i][temp_j] + 1;

        temp_i = 0;

        temp_j = 0;
    }
}

```

```

    }

    return matrix;
}

long double entropy_of_alphabet(alphabet* language, type length, type size)
{
    long double result = 0;

    for (type i = 0; i < size; i++)
    {
        if (language[i].value != 0)
        {
            long double temp = 0;

            temp = ((long double)language[i].value / length);

            result = -temp * log2(temp) + result;
        }
    }

    return result;
}

long double entropy_of_alphabet_without_struck(alphabet* language, type length, type size)
{
    long double result = 0;

    for (type i = 0; i < size; i++)
    {
        if (language[i].value != 0 && language[i].symbols != ' ')
        {
            long double temp = 0;

            temp = ((long double)language[i].value / length);

            result = -temp * log2(temp) + result;
        }
    }

    return result;
}

long double entropy_of_alphabet_bigram(long double**matrix, type length, type size)
{
    long double result = 0;

```

```

    for (type i = 0; i < size; i++)
    {
        for (type j = 0; j < size; j++)
        {
            if (matrix[i][j] != 0)
            {
                long double temp = 0;

                temp = ((long double)matrix[i][j] / length);

                result = -temp * log2(temp) + result;
            }
        }
    }

    return result;
}

```

### **Criptology\_1.cpp**

```

//connected library block start

#include "my_function_library.h"

//connected library block end


//main function start

int main()
{
    //language configuration

    SetConsoleCP(1251); SetConsoleOutputCP(1251);


    //input of path to text file

    string path_;    cout << endl << "Input path to text-file: ";

    cin >> path_;


    //open text file for analise of frequency

    ifstream fin;      fin.open(path_);

```

```

if (!fin.is_open())
{
    cout << endl << "Error of open file!" << endl;
}
else
{
    //clock start

    unsigned int start_time = clock();

    // open file for writing the statistic

    cout << endl << "File open seccsesfull." << endl;  fstream fout;      fout.open("in_file.xls", ios::out);

    //creating of start data block start

    type text_size;
    type space_number;

    int bigram_universume_with_space_step_one;
    int bigram_universume_with_space_step_two;
    int bigram_universume_without_space_step_one;
    int bigram_universume_without_space_step_two;

    string one_string;                string _upper_alphabet_ = upper_alphabet();

    alphabet  table_frequency[34];    null(table_frequency, _upper_alphabet_, 34);

    long double entropy_alphubet_without_space, entropy_alphubet_with_space;

    //creating of start data block end
    // frequency analize each symbols in text block start
    {

        fin.seekg(0, std::ios::beg);

        text_size = 0;

        while (!fin.eof())
        {
            getline(fin, one_string);

            one_string = filtration_the_text_with_space(one_string);

            freuquency_of_symbols(table_frequency, 34, one_string);

            text_size += one_string.length();
        }
    }
}

```

```

}

space_number = table_frequency[33].value;

fout << endl << "Frequency analyse of text in general:" << endl << endl;

fout << endl << "-ordinary order of symbols with space:" << endl;

fout << endl << "\t" << "Symbols" << "\t" << "Count" << "\t" << "Frequency" << endl << endl;

for (type i = 0; i < 34; i++)
{
fout << "\t" << table_frequency[i].symbols << "\t" << table_frequency[i].value << "\t" << (long double)(table_frequency[i].value) / text_size << endl;
}

fout << endl << "-ordinary order of symbols without space:" << endl;

fout << endl << "\t" << "Symbols" << "\t" << "Count" << "\t" << "Frequency" << endl << endl;

for (type i = 0; i < 34; i++)
{
    if (table_frequency[i].symbols != '_')
    {
        fout << "\t" << table_frequency[i].symbols << "\t" << table_frequency[i].value << "\t"
        << (long double)(table_frequency[i].value) / (text_size - space_number) << endl;
    }
}

bubbleSort(table_frequency, 34);

fout << endl << "-sorting order of symbols with space:" << endl;

fout << endl << "\t" << "Symbols" << "\t" << "Count" << "\t" << "Frequency" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t" << table_frequency[i].symbols << "\t" << table_frequency[i].value << "\t"
    << ((long double)(table_frequency[i].value)) / text_size << endl;
}

fout << endl << "-sorting order of symbols without space:" << endl;

fout << endl << "\t" << "Symbols" << "\t" << "Count" << "\t" << "Frequency" << endl << endl;

```

```

        for (type i = 0; i < 34; i++)
        {
            if (table_frequency[i].symbols != '_')
            {
                fout << "\t" << table_frequency[i].symbols << "\t" << table_frequency[i].value << "\t" << ((long
double)(table_frequency[i].value)) / (text_size - space_number) << endl;
            }
        }

        entropy_alphabet_with_space = entropy_of_alphabet(table_frequency, text_size, 34);
        fout << endl << "Entropy of symbols of alphabet with struck: " << entropy_alphabet_with_space << " ;" << endl;

        entropy_alphabet_without_space = entropy_of_alphabet_without_struck(table_frequency, text_size - space_number, 33);
        fout << endl << "Entropy of symbols of alphabet without struck: " << entropy_alphabet_without_space << " ;" << endl;

    }
    // frequency analyze each symbols in text block end

    //create matrix start

    long double** matrix = new long double*[34];

    for (type i = 0; i < 34; i++)
    {
        matrix[i] = new long double[34];
    }

    //create matrix end

    // bigram frequency with spaces block start
    {

        null_matrix(matrix, 34);

        fin.seekg(0, std::ios::beg);

        while (!fin.eof())
        {
            getline(fin, one_string);

            one_string = filtration_the_text_with_space(one_string);

            if (one_string.length() <= 1)
            {
                continue;
            }

            matrix_frequency_with_step_one(matrix, 34, 34, one_string);
        }
    }

```

```

}

fout << endl << "Bigram statistic data for step one, with space:" << endl << endl;

fout << endl << "-bigram count with step one, with space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet[i];
}

fout << endl;

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << matrix[i][j] << "\t";
    }
    fout << endl;
}

bigram_universume_with_space_step_one = 0;

for (type i = 0; i < 34; i++)
{
    for (type j = 0; j < 34; j++)
    {
        bigram_universume_with_space_step_one = bigram_universume_with_space_step_one + matrix[i][j];
    }
}

fout << endl << "-bigram frequency with step one, with space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet[i];
}

fout << endl;

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << (long double)matrix[i][j] / bigram_universume_with_space_step_one << "\t";
    }
}

```



```

        }

        fout << endl;
    }

    fout << endl << "Entropy on symbol of bigram with space and step one: " << entropy_of_alphabet_bigram(matrix,
bigram_universume_with_space_step_one, 34) / 2 << " ;" << endl;

    null_matrix(matrix, 34);

    fin.seekg(0, std::ios::beg);

    while (!fin.eof())
    {
        getline(fin, one_string);

        one_string = filtration_the_text_with_space(one_string);

        if (one_string.length() <= 1)
        {
            continue;
        }

        matrix_frequancy_with_step_two(matrix, 34, 34, one_string);
    }

    fout << endl << "Bigram statistic data with step two, with space:" << endl << endl;

    fout << endl << "-bigram count with step two, with space:" << endl << endl;

    for (type i = 0; i < 34; i++)
    {
        fout << "\t \t" << _upper_alphabet[i];
    }

    fout << endl;

    for (type i = 0; i < 34; i++)
    {
        fout << _upper_alphabet[i] << "\t";

        for (type j = 0; j < 34; j++)
        {
            fout << "\t" << matrix[i][j] << "\t";
        }

        fout << endl;
    }

```

```

bigram_universume_with_space_step_two = 0;

for (type i = 0; i < 34; i++)
{
    for (type j = 0; j < 34; j++)
    {
        bigram_universume_with_space_step_two = bigram_universume_with_space_step_two + matrix[i][j];
    }
}

fout << endl << "-bigram frequency with step two, with space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet_[i];
}

fout << endl;

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet_[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << (long double)matrix[i][j] / bigram_universume_with_space_step_two << "\t";
    }
    fout << endl;
}

fout << endl << "Entropy on symbol of bigram with space and step two: "
<< entropy_of_alphabet_bigram(matrix, bigram_universume_with_space_step_two, 34) / 2 << " ;" << endl;
}

// bigram frequency with spaces block end

//bigram frequency without spaces block start
{
    null_matrix(matrix, 34);

    fin.seekg(0, std::ios::beg);

    while (!fin.eof())
    {
        getline(fin, one_string);
    }
}

```

```

        one_string = filtration_the_text_without_space(one_string);
        if (one_string.length() <= 1)
        {
            continue;
        }

        matrix_frequancy_with_step_one(matrix, 34, 34, one_string);
    }

fout << endl << "Bigram statistic data for step one, without space:" << endl << endl;

fout << endl << "-bigram count with step one, without space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet_[i];
}

fout << endl;

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet_[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << matrix[i][j] << "\t";
    }
    fout << endl;
}

bigram_universume_without_space_step_one = 0;

for (type i = 0; i < 34; i++)
{
    for (type j = 0; j < 34; j++)
    {
        bigram_universume_without_space_step_one = bigram_universume_without_space_step_one + matrix[i][j];
    }
}

fout << endl << "-bigram frequency with step one, without space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet_[i];
}

```

```

fout << endl;

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet_[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << (long double)matrix[i][j] / bigram_universume_without_space_step_one << "\t";
    }

    fout << endl;
}

fout << endl << "Entropy on symbol of bigram without space and step one: "
<< entropy_of_alphabet_bigram(matrix, bigram_universume_without_space_step_one, 33) / 2 << " ;" << endl;

null_matrix(matrix, 34);

fin.seekg(0, std::ios::beg);

while (!fin.eof())
{
    getline(fin, one_string);

    one_string = filtration_the_text_without_space(one_string);

    if (one_string.length() <= 1)
    {
        continue;
    }

    matrix_frequancy_with_step_two(matrix, 34, 34, one_string);
}

fout << endl << "Bigram statistic data with step two, without space:" << endl << endl;

fout << endl << "--bigram count with step two, without space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet_[i];
}

fout << endl;

```

```

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet_[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << matrix[i][j] << "\t";
    }
    fout << endl;

}

bigram_universume_without_space_step_two = 0;

for (type i = 0; i < 34; i++)
{
    for (type j = 0; j < 34; j++)
    {
        bigram_universume_without_space_step_two = bigram_universume_without_space_step_two + matrix[i][j];
    }
}

fout << endl << "-bigram frequency with step two, without space:" << endl << endl;

for (type i = 0; i < 34; i++)
{
    fout << "\t \t" << _upper_alphabet_[i];
}

fout << endl;

for (type i = 0; i < 34; i++)
{
    fout << _upper_alphabet_[i] << "\t";

    for (type j = 0; j < 34; j++)
    {
        fout << "\t" << (long double)matrix[i][j] / bigram_universume_without_space_step_two << "\t";
    }
    fout << endl;

}

fout << endl << "Entrophy on symbol of bigram without space and step two: "
<< entropy_of_alphabet_bigram(matrix, bigram_universume_without_space_step_two, 33) / 2 << " ;" << endl;
}

```

```
//bigram frequency without spaces block end

//delite matrix start
for (type i = 0; i < 34; i++)
{
    delete matrix[i];
}

delete[] matrix;

//delite matrix end

//clock end

unsigned int end_time = clock();
unsigned int search_time = end_time - start_time;

cout << endl << "Time of work: " << search_time << " milliseconds;" << endl;

}

//file close and exit with program

fin.close();
return 0;

}

//main function end
```

## Результати роботи програми

Frequency analyse of text in general:

| -ordinary order of symbols with space: |         |        |             |
|--|---------|--------|-------------|
|  | Symbols | Count  | Frequency   |
|  |         |        |             |
|  | A       | 111181 | 0.0656816   |
|  | Б       | 23529  | 0.0139001   |
|  | B       | 62897  | 0.0371572   |
|  | Г       | 24309  | 0.0143609   |
|  | Д       | 39781  | 0.0235011   |
|  | Е       | 117356 | 0.0693295   |
|  | Ё       | 0      | 0           |
|  | Ж       | 15354  | 0.00907057  |
|  | З       | 21915  | 0.0129466   |
|  | И       | 88225  | 0.05212     |
|  | Й       | 13753  | 0.00812476  |
|  | К       | 45857  | 0.0270906   |
|  | Л       | 67547  | 0.0399042   |
|  | М       | 38345  | 0.0226528   |
|  | Н       | 92656  | 0.0547377   |
|  | О       | 154093 | 0.0910324   |
|  | П       | 32264  | 0.0190604   |
|  | Р       | 52644  | 0.0311001   |
|  | С       | 70536  | 0.04167     |
|  | Т       | 80097  | 0.0473183   |
|  | У       | 36461  | 0.0215398   |
|  | Ф       | 1346   | 0.000795167 |
|  | Х       | 10236  | 0.00604705  |
|  | Ц       | 3469   | 0.00204936  |
|  | Ч       | 22823  | 0.013483    |
|  | Ш       | 11631  | 0.00687116  |
|  | Щ       | 3797   | 0.00224313  |
|  | Ъ       | 395    | 0.000233351 |
|  | Ы       | 24780  | 0.0146391   |
|  | Ь       | 26788  | 0.0158254   |
|  | Э       | 4766   | 0.00281558  |
|  | Ю       | 8417   | 0.00497245  |
|  | Я       | 29030  | 0.0171498   |
|  | _       | 270018 | 0.159517    |

| -ordinary order of symbols without space: |         |        |             |
|---|---------|--------|-------------|
|   | Symbols | Count  | Frequency   |
|   |         |        |             |
|   | A       | 111181 | 0.0781474   |
|   | Б       | 23529  | 0.0165382   |
|   | B       | 62897  | 0.0442093   |
|   | Г       | 24309  | 0.0170864   |
|   | Д       | 39781  | 0.0279614   |
|   | Е       | 117356 | 0.0824877   |
|   | Ё       | 0      | 0           |
|   | Ж       | 15354  | 0.0107921   |
|   | З       | 21915  | 0.0154037   |
|   | И       | 88225  | 0.062012    |
|   | Й       | 13753  | 0.00966677  |
|   | К       | 45857  | 0.0322322   |
|   | Л       | 67547  | 0.0474777   |
|   | М       | 38345  | 0.0269521   |
|   | Н       | 92656  | 0.0651265   |
|   | О       | 154093 | 0.10831     |
|   | П       | 32264  | 0.0226779   |
|   | Р       | 52644  | 0.0370026   |
|   | С       | 70536  | 0.0495787   |
|   | Т       | 80097  | 0.0562989   |
|   | У       | 36461  | 0.0256279   |
|   | Ф       | 1346   | 0.000946082 |
|   | Х       | 10236  | 0.00719472  |
|   | Ц       | 3469   | 0.00243831  |
|   | Ч       | 22823  | 0.0160419   |
|   | Ш       | 11631  | 0.00817525  |
|   | Щ       | 3797   | 0.00266885  |
|   | Ъ       | 395    | 0.000277639 |
|   | Ы       | 24780  | 0.0174175   |
|   | Ь       | 26788  | 0.0188289   |
|   | Э       | 4766   | 0.00334995  |
|   | Ю       | 8417   | 0.00591618  |
|   | Я       | 29030  | 0.0204047   |

| -sorting order of symbols with space: |         |        |             |
|---------------------------------------|---------|--------|-------------|
|                                       | Symbols | Count  | Frequency   |
|                                       |         |        |             |
|                                       | —       | 270018 | 0.159517    |
|                                       | O       | 154093 | 0.0910324   |
|                                       | E       | 117356 | 0.0693295   |
|                                       | A       | 111181 | 0.0656816   |
|                                       | H       | 92656  | 0.0547377   |
|                                       | И       | 88225  | 0.05212     |
|                                       | T       | 80097  | 0.0473183   |
|                                       | C       | 70536  | 0.04167     |
|                                       | Л       | 67547  | 0.0399042   |
|                                       | B       | 62897  | 0.0371572   |
|                                       | P       | 52644  | 0.0311001   |
|                                       | K       | 45857  | 0.0270906   |
|                                       | Д       | 39781  | 0.0235011   |
|                                       | M       | 38345  | 0.0226528   |
|                                       | У       | 36461  | 0.0215398   |
|                                       | П       | 32264  | 0.0190604   |
|                                       | Я       | 29030  | 0.0171498   |
|                                       | Ь       | 26788  | 0.0158254   |
|                                       | Ы       | 24780  | 0.0146391   |
|                                       | Г       | 24309  | 0.0143609   |
|                                       | Б       | 23529  | 0.0139001   |
|                                       | Ч       | 22823  | 0.013483    |
|                                       | З       | 21915  | 0.0129466   |
|                                       | Ж       | 15354  | 0.00907057  |
|                                       | Й       | 13753  | 0.00812476  |
|                                       | Ш       | 11631  | 0.00687116  |
|                                       | Х       | 10236  | 0.00604705  |
|                                       | Ю       | 8417   | 0.00497245  |
|                                       | Э       | 4766   | 0.00281558  |
|                                       | Щ       | 3797   | 0.00224313  |
|                                       | Ц       | 3469   | 0.00204936  |
|                                       | Ф       | 1346   | 0.000795167 |
|                                       | Ъ       | 395    | 0.000233351 |
|                                       | Ё       | 0      | 0           |

| -sorting order of symbols without space: |         |        |             |
|--|---------|--------|-------------|
|  | Symbols | Count  | Frequency   |
|  |         |        |             |
|  | O       | 154093 | 0.10831     |
|  | E       | 117356 | 0.0824877   |
|  | A       | 111181 | 0.0781474   |
|  | H       | 92656  | 0.0651265   |
|  | И       | 88225  | 0.062012    |
|  | T       | 80097  | 0.0562989   |
|  | C       | 70536  | 0.0495787   |
|  | Л       | 67547  | 0.0474777   |
|  | B       | 62897  | 0.0442093   |
|  | P       | 52644  | 0.0370026   |
|  | K       | 45857  | 0.0322322   |
|  | Д       | 39781  | 0.0279614   |
|  | M       | 38345  | 0.0269521   |
|  | У       | 36461  | 0.0256279   |
|  | П       | 32264  | 0.0226779   |
|  | Я       | 29030  | 0.0204047   |
|  | Ь       | 26788  | 0.0188289   |
|  | Ы       | 24780  | 0.0174175   |
|  | Г       | 24309  | 0.0170864   |
|  | Б       | 23529  | 0.0165382   |
|  | Ч       | 22823  | 0.0160419   |
|  | З       | 21915  | 0.0154037   |
|  | Ж       | 15354  | 0.0107921   |
|  | Й       | 13753  | 0.00966677  |
|  | Ш       | 11631  | 0.00817525  |
|  | Х       | 10236  | 0.00719472  |
|  | Ю       | 8417   | 0.00591618  |
|  | Э       | 4766   | 0.00334995  |
|  | Щ       | 3797   | 0.00266885  |
|  | Ц       | 3469   | 0.00243831  |
|  | Ф       | 1346   | 0.000946082 |
|  | Ъ       | 395    | 0.000277639 |
|  | Ё       | 0      | 0           |





| bigram count with step two, with space: |       |      |       |      |      |      |   |      |      |      |      |      |      |      |       |       |       |      |       |      |      |     |      |     |      |      |     |    |      |      |      |     |      |       |     |  |  |
|---|-------|------|-------|------|------|------|---|------|------|------|------|------|------|------|-------|-------|-------|------|-------|------|------|-----|------|-----|------|------|-----|----|------|------|------|-----|------|-------|-----|--|--|
|   | 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   | Ъ  | Ы    | Ь    | Э    | Ю   | Я    |       |     |  |  |
| А                                       | 39503 | 458  | 2459  | 477  | 1793 | 815  | 0 | 903  | 3370 | 85   | 426  | 3162 | 7902 | 1864 | 3464  | 2     | 476   | 1758 | 2820  | 3318 | 34   | 191 | 700  | 93  | 620  | 492  | 162 | 0  | 0    | 0    | 1    | 659 | 1938 | 15033 |     |  |  |
| Б                                       | 682   | 1    | 75    | 4    | 31   | 1581 | 0 | 4    | 3    | 612  | 0    | 195  | 613  | 47   | 233   | 1560  | 1     | 838  | 57    | 4    | 775  | 0   | 44   | 9   | 6    | 9    | 169 | 85 | 3418 | 12   | 0    | 3   | 438  | 293   |     |  |  |
| В                                       | 4323  | 14   | 17    | 26   | 251  | 3376 | 0 | 0    | 431  | 3790 | 0    | 111  | 602  | 109  | 1023  | 5670  | 186   | 822  | 2827  | 194  | 516  | 0   | 47   | 23  | 83   | 839  | 7   | 3  | 1957 | 138  | 0    | 2   | 172  | 4240  |     |  |  |
| Г                                       | 644   | 0    | 5     | 0    | 1005 | 249  | 0 | 0    | 1    | 601  | 0    | 97   | 1211 | 0    | 134   | 6772  | 0     | 538  | 21    | 6    | 495  | 1   | 0    | 0   | 33   | 3    | 0   | 0  | 2    | 0    | 0    | 0   | 520  |       |     |  |  |
| Д                                       | 3542  | 25   | 642   | 4    | 25   | 3541 | 0 | 7    | 1    | 1552 | 0    | 158  | 646  | 43   | 1174  | 2984  | 54    | 1149 | 222   | 157  | 1497 | 0   | 55   | 132 | 25   | 81   | 1   | 54 | 384  | 598  | 0    | 4   | 328  | 779   |     |  |  |
| Е                                       | 33    | 1021 | 1837  | 2926 | 1984 | 1755 | 0 | 644  | 930  | 76   | 1970 | 1490 | 4688 | 3169 | 5687  | 138   | 1118  | 3975 | 3056  | 3336 | 69   | 7   | 629  | 123 | 698  | 744  | 513 | 0  | 0    | 0    | 0    | 335 | 281  | 15158 |     |  |  |
| Ё                                       | 0     | 0    | 0     | 0    | 0    | 0    | 0 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0     | 0     | 0    | 0     | 0    | 0    | 0   | 0    | 0   | 0    | 0    | 0   | 0  | 0    | 0    | 0    | 0   | 0    |       |     |  |  |
| Ж                                       | 1071  | 32   | 0     | 3    | 613  | 3291 | 0 | 3    | 0    | 1138 | 0    | 73   | 5    | 2    | 791   | 26    | 0     | 1    | 80    | 0    | 198  | 0   | 0    | 1   | 43   | 0    | 0   | 0  | 0    | 45   | 0    | 0   | 244  |       |     |  |  |
| З                                       | 4352  | 132  | 652   | 438  | 611  | 171  | 0 | 90   | 13   | 230  | 0    | 67   | 141  | 215  | 1433  | 281   | 0     | 159  | 35    | 18   | 224  | 0   | 0    | 0   | 4    | 51   | 6   | 0  | 8    | 258  | 124  | 0   | 1    | 387   | 861 |  |  |
| И                                       | 72    | 307  | 2159  | 317  | 1266 | 1903 | 0 | 244  | 1367 | 411  | 1038 | 1199 | 3498 | 2243 | 3117  | 61    | 106   | 408  | 1601  | 3028 | 8    | 16  | 1024 | 613 | 1362 | 314  | 116 | 0  | 0    | 0    | 0    | 178 | 1302 | 14760 |     |  |  |
| Й                                       | 0     | 1    | 0     | 0    | 169  | 0    | 0 | 0    | 1    | 0    | 0    | 60   | 66   | 37   | 207   | 1     | 0     | 0    | 373   | 179  | 0    | 2   | 0    | 33  | 61   | 34   | 1   | 0  | 0    | 0    | 0    | 0   | 0    | 5680  |     |  |  |
| К                                       | 6219  | 1    | 115   | 0    | 3    | 398  | 0 | 3    | 0    | 2247 | 0    | 12   | 274  | 0    | 497   | 6275  | 0     | 1125 | 652   | 300  | 1142 | 1   | 1    | 4   | 0    | 9    | 0   | 0  | 0    | 0    | 0    | 0   | 0    | 3914  |     |  |  |
| Л                                       | 6098  | 15   | 2     | 83   | 14   | 4165 | 0 | 339  | 3    | 4188 | 0    | 218  | 240  | 0    | 224   | 4676  | 35    | 0    | 857   | 18   | 956  | 3   | 0    | 1   | 83   | 3    | 1   | 0  | 757  | 2731 | 0    | 792 | 1372 | 5876  |     |  |  |
| М                                       | 1986  | 6    | 0     | 20   | 0    | 2366 | 0 | 0    | 2    | 2081 | 0    | 43   | 85   | 28   | 1026  | 2505  | 30    | 58   | 73    | 12   | 2298 | 1   | 1    | 8   | 15   | 4    | 1   | 0  | 742  | 48   | 0    | 1   | 326  | 5651  |     |  |  |
| Н                                       | 8766  | 10   | 1     | 51   | 562  | 8236 | 0 | 4    | 8    | 6044 | 0    | 189  | 0    | 1    | 2206  | 7845  | 0     | 60   | 794   | 324  | 2101 | 10  | 0    | 191 | 205  | 3    | 130 | 0  | 2210 | 1010 | 0    | 129 | 1405 | 4326  |     |  |  |
| О                                       | 3     | 2736 | 5679  | 3545 | 3313 | 2016 | 0 | 1541 | 764  | 555  | 2283 | 1226 | 4290 | 3691 | 7006  | 139   | 916   | 4304 | 4623  | 5014 | 45   | 95  | 261  | 66  | 1272 | 1001 | 121 | 0  | 0    | 0    | 19   | 693 | 394  | 19537 |     |  |  |
| П                                       | 865   | 0    | 0     | 0    | 0    | 1660 | 0 | 0    | 0    | 597  | 0    | 51   | 340  | 0    | 58    | 6550  | 20    | 5001 | 3     | 32   | 382  | 0   | 0    | 9   | 27   | 10   | 1   | 0  | 186  | 28   | 0    | 3   | 249  | 39    |     |  |  |
| Р                                       | 5393  | 170  | 245   | 286  | 295  | 4103 | 0 | 233  | 11   | 3487 | 0    | 495  | 49   | 93   | 523   | 5355  | 49    | 14   | 142   | 313  | 1813 | 6   | 71   | 18  | 53   | 179  | 24  | 0  | 977  | 746  | 0    | 73  | 696  | 527   |     |  |  |
| С                                       | 1461  | 43   | 1444  | 10   | 273  | 3564 | 0 | 26   | 16   | 1048 | 0    | 4014 | 1994 | 660  | 754   | 1920  | 1361  | 89   | 412   | 8203 | 372  | 6   | 118  | 41  | 337  | 42   | 2   | 16 | 263  | 2289 | 3    | 107 | 2432 | 2306  |     |  |  |
| Т                                       | 4050  | 6    | 2025  | 6    | 98   | 4024 | 0 | 2    | 3    | 3016 | 0    | 390  | 203  | 8    | 835   | 11362 | 38    | 1948 | 647   | 53   | 921  | 19  | 7    | 50  | 207  | 0    | 9   | 29 | 1239 | 4994 | 2    | 29  | 418  | 3601  |     |  |  |
| У                                       | 28    | 403  | 946   | 769  | 1242 | 188  | 0 | 1346 | 220  | 2    | 97   | 615  | 931  | 948  | 171   | 3     | 464   | 408  | 764   | 848  | 0    | 13  | 229  | 2   | 634  | 553  | 160 | 0  | 0    | 0    | 6    | 768 | 104  | 5187  |     |  |  |
| Ф                                       | 52    | 0    | 0     | 0    | 0    | 90   | 0 | 0    | 0    | 229  | 0    | 0    | 15   | 2    | 1     | 55    | 0     | 118  | 6     | 12   | 15   | 1   | 0    | 0   | 1    | 0    | 0   | 0  | 4    | 38   | 0    | 1   | 1    | 24    |     |  |  |
| Х                                       | 650   | 0    | 78    | 2    | 1    | 22   | 0 | 0    | 0    | 80   | 0    | 0    | 73   | 48   | 95    | 1752  | 0     | 40   | 34    | 6    | 111  | 0   | 0    | 0   | 0    | 7    | 0   | 1  | 0    | 0    | 0    | 0   | 0    | 2087  |     |  |  |
| Ц                                       | 375   | 1    | 28    | 0    | 0    | 496  | 0 | 0    | 0    | 102  | 0    | 81   | 0    | 1    | 0     | 232   | 0     | 0    | 0     | 0    | 156  | 0   | 0    | 4   | 0    | 1    | 0   | 0  | 107  | 0    | 0    | 0   | 135  |       |     |  |  |
| Ч                                       | 1792  | 0    | 3     | 0    | 0    | 2680 | 0 | 0    | 0    | 1026 | 0    | 235  | 34   | 5    | 387   | 37    | 0     | 56   | 0     | 3685 | 729  | 0   | 0    | 0   | 0    | 144  | 0   | 0  | 0    | 134  | 0    | 0   | 0    | 692   |     |  |  |
| Ш                                       | 727   | 0    | 66    | 1    | 0    | 1815 | 0 | 0    | 0    | 1247 | 0    | 298  | 422  | 2    | 299   | 226   | 5     | 0    | 0     | 51   | 236  | 0   | 0    | 3   | 0    | 1    | 0   | 0  | 386  | 0    | 3    | 0   | 55   |       |     |  |  |
| Щ                                       | 229   | 0    | 0     | 0    | 0    | 1016 | 0 | 0    | 0    | 508  | 0    | 0    | 0    | 0    | 24    | 1     | 0     | 4    | 0     | 0    | 57   | 0   | 0    | 0   | 0    | 0    | 0   | 0  | 18   | 0    | 0    | 0   | 5    |       |     |  |  |
| Ъ                                       | 0     | 0    | 0     | 0    | 0    | 95   | 0 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0     | 0     | 0    | 0     | 0    | 0    | 0   | 0    | 0   | 0    | 0    | 0   | 0  | 0    | 0    | 5    | 99  | 0    |       |     |  |  |
| Ы                                       | 0     | 399  | 830   | 71   | 111  | 722  | 0 | 17   | 48   | 6    | 935  | 123  | 2031 | 905  | 273   | 0     | 93    | 236  | 606   | 561  | 5    | 0   | 602  | 0   | 132  | 355  | 0   | 0  | 0    | 0    | 0    | 0   | 0    | 3217  |     |  |  |
| Ь                                       | 0     | 67   | 21    | 39   | 26   | 363  | 0 | 1    | 126  | 366  | 0    | 957  | 0    | 180  | 928   | 4     | 2     | 0    | 667   | 60   | 0    | 0   | 1    | 88  | 48   | 326  | 14  | 0  | 0    | 0    | 3    | 308 | 327  | 8523  |     |  |  |
| Э                                       | 0     | 0    | 1     | 4    | 0    | 0    | 0 | 0    | 3    | 1    | 2    | 36   | 37   | 2    | 18    | 0     | 2     | 1    | 2     | 2363 | 0    | 3   | 1    | 0   | 0    | 1    | 0   | 0  | 0    | 0    | 0    | 0   | 0    | 2     |     |  |  |
| Ю                                       | 1     | 504  | 3     | 2    | 176  | 0    | 0 | 1    | 1    | 1    | 0    | 2    | 10   | 9    | 12    | 0     | 8     | 31   | 129   | 191  | 0    | 2   | 7    | 2   | 32   | 19   | 276 | 0  | 0    | 0    | 23   | 0   | 2763 |       |     |  |  |
| Я                                       | 0     | 16   | 201   | 241  | 435  | 65   | 0 | 228  | 254  | 38   | 96   | 89   | 659  | 257  | 516   | 0     | 42    | 33   | 658   | 1103 | 0    | 0   | 146  | 32  | 109  | 38   | 156 | 0  | 0    | 0    | 67   | 113 | 9043 |       |     |  |  |
|   | 5916  | 5359 | 11564 | 2646 | 5620 | 4200 | 0 | 2058 | 3377 | 8822 | 1    | 6979 | 2738 | 4359 | 12719 | 10513 | 11147 | 3031 | 12749 | 6469 | 3257 | 304 | 1205 | 200 | 5047 | 570  | 71  | 0  | 0    | 2    | 2253 | 28  | 1609 | 0     |     |  |  |

| -bigram frequency with step two, with space: |          |          |          |          |          |          |   |          |          |          |          |          |           |          |          |          |          |          |          |          |          |          |          |          |          |           |          |          |           |          |          |          |          |          |          |          |
|--|----------|----------|----------|----------|----------|----------|---|----------|----------|----------|----------|----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|----------|----------|-----------|----------|----------|----------|----------|----------|----------|----------|
|  | 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        | Ъ        | Ы         | Ь        | Э        | Ю        | Я        |          |          |          |
| А  | 0.046783 | 0.000542 | 0.002912 | 0.000565 | 0.002123 | 0.000965 | 0 | 0.001069 | 0.003991 | 0.000101 | 0.000505 | 0.003745 | 0.009358  | 0.002207 | 0.004102 | 2.37E-06 | 0.000564 | 0.002082 | 0.00334  | 0.003929 | 4.03E-05 | 0.000226 | 0.000829 | 0.00011  | 0.000734 | 0.000583  | 0.000192 | 0        | 0         | 0        | 1.18E-06 | 0.00078  | 0.002295 | 0.017803 |          |          |
| Б  | 0.000808 | 1.18E-06 | 8.88E-05 | 4.74E-06 | 3.67E-05 | 0.001872 | 0 | 4.74E-06 | 3.55E-06 | 0.000728 | 0        | 0.000231 | 0.000726  | 5.57E-05 | 0.000276 | 0.001847 | 1.18E-06 | 0.000992 | 6.75E-05 | 4.74E-06 | 0.000918 | 0        | 5.21E-05 | 1.07E-05 | 7.11E-06 | 1.07E-05  | 0.0002   | 0.000101 | 0.004048  | 1.42E-05 | 0        | 3.55E-06 | 0.000519 | 0.003047 |          |          |
| В  | 0.00512  | 1.66E-05 | 2.01E-05 | 3.08E-05 | 0.000297 | 0.003998 | 0 | 0        | 0.00051  | 0.004488 | 0        | 0.000131 | 0.000713  | 0.000129 | 0.001212 | 0.006715 | 0.00022  | 0.000973 | 0.003348 | 0.00023  | 0.000611 | 0        | 5.57E-05 | 2.72E-05 | 9.83E-05 | 0.000994  | 8.29E-06 | 3.55E-06 | 0.0002318 | 0.000163 | 0        | 2.37E-06 | 0.000204 | 0.005021 |          |          |
| Г  | 0.000763 | 0        | 5.92E-06 | 0        | 0.00119  | 0.000295 | 0 | 0        | 1.18E-06 | 0.000712 | 0        | 0.000115 | 0.001434  | 0        | 0.000159 | 0.00802  | 0        | 0.000637 | 2.49E-05 | 7.11E-06 | 0.000586 | 1.18E-06 | 0        | 0        | 3.91E-05 | 3.55E-06  | 0        | 0        | 0         | 2.37E-06 | 0        | 0        | 0.000616 |          |          |          |
| Д  | 0.004195 | 2.96E-05 | 0.00076  | 4.74E-06 | 2.96E-05 | 0.004194 | 0 | 8.29E-06 | 1.18E-06 | 0.001838 | 0        | 0.000187 | 0.000765  | 5.09E-05 | 0.00139  | 0.000534 | 6.40E-05 | 0.001361 | 0.000263 | 0.000186 | 0.001773 | 0        | 6.51E-05 | 0.000156 | 2.96E-05 | 9.95E-05  | 1.18E-06 | 6.40E-05 | 0.000455  | 0.000708 | 0        | 4.74E-06 | 0.000388 | 0.000923 |          |          |
| Е  | 3.91E-05 | 0.001209 | 0.002176 | 0.003465 | 0.000235 | 0.002078 | 0 | 0.000763 | 0.001101 | 9.00E-05 | 0.002333 | 0.001765 | 0.005552  | 0.003753 | 0.000163 | 0.001324 | 0.004708 | 0.003619 | 0.000951 | 8.17E-05 | 8.29E-06 | 0.000745 | 0.000146 | 0.000827 | 0.000881 | 0.000608  | 0        | 0        | 0         | 0.000397 | 0.000333 | 0.017951 |          |          |          |          |
| Ё  | 0        | 0        | 0        | 0        | 0        | 0        | 0 | 0        | 0        | 0        | 0        | 0        | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0         | 0        | 0        | 0         | 0        | 0        | 0        | 0        |          |          |          |
| Ж  | 0.001268 | 3.79E-05 | 0        | 3.55E-06 | 0.000726 | 0.003897 | 0 | 3.55E-06 | 0        | 0.001348 | 0        | 8.65E-05 | 5.92E-06  | 2.37E-06 | 0.000937 | 3.08E-05 | 0        | 1.18E-06 | 9.47E-05 | 0        | 0.000234 | 0        | 0        | 1.18E-06 | 5.09E-05 | 0         | 0        | 0        | 0         | 0        | 0        | 0        | 0        | 0.000289 |          |          |
| З  | 0.005154 | 0.000156 | 0.000772 | 0.000519 | 0.000724 | 0.000203 | 0 | 0.000107 | 1.54E-05 | 0.000272 | 0        | 7.93E-05 | 0.000167  | 0.000255 | 0.001697 | 0.000333 | 0        | 0.000188 | 4.14E-05 | 2.13E-05 | 0.000265 | 0        | 0        | 4.74E-06 | 6.04E-05 | 7.11E-06  | 0        | 9.47E-06 | 0.000306  | 0.000147 | 0        | 1.18E-06 | 0.000458 | 0.00102  |          |          |
| И  | 8.53E-05 | 0.000364 | 0.002557 | 0.000375 | 0.001499 | 0.002254 | 0 | 0.000289 | 0.001619 | 0.000487 | 0.001229 | 0.00142  | 0.00143   | 0.002656 | 0.003691 | 7.22E-05 | 0.000126 | 0.000483 | 0.001896 | 0.003586 | 9.47E-06 | 1.89E-05 | 0.001213 | 0.000726 | 0.001613 | 0.000372  | 0.000137 | 0        | 0         | 0        | 0.000211 | 0.001542 | 0.01748  |          |          |          |
| Й  | 0        | 1.18E-06 | 0        | 0        | 0.0002   | 0        | 0 | 0        | 1.18E-06 | 0        | 0        | 7.11E-05 | 7.82E-05  | 4.38E-05 | 0.000245 | 1.18E-06 | 0        | 0        | 0.000442 | 0.000212 | 0        | 2.37E-06 | 0        | 3.91E-05 | 7.22E-05 | 4.03E-05  | 1.18E-06 | 0        | 0         | 0        | 0        | 0        | 0        | 0.006727 |          |          |
| К  | 0.007365 | 1.18E-06 | 0.000136 | 0        | 3.55E-06 | 0.000471 | 0 | 3.55E-06 | 0        | 0.002661 | 0        | 1.42E-05 | 0.000324  | 0        | 0.000589 | 0.007431 | 0        | 0.001332 | 0.000772 | 0.000355 | 0.001352 | 1.18E-06 | 1.18E-06 | 4.74E-06 | 0        | 1.07E-05  | 0        | 0        | 0         | 0        | 0        | 0        | 0.004635 |          |          |          |
| Л  | 0.007222 | 1.78E-05 | 2.37E-06 | 9.83E-05 | 1.66E-05 | 0.004933 | 0 | 0.000401 | 3.55E-06 | 0.004696 | 0        | 0.000258 | 0.000284  | 0        | 0.000538 | 0.005538 | 4.14E-05 | 0        | 0.001015 | 2.13E-05 | 0.001132 | 3.55E-06 | 0        | 1.18E-06 | 9.83E-05 | 3.55E-06  | 1.18E-06 | 0        | 0.000896  | 0.003234 | 0        | 0.000938 | 0.001625 | 0.006959 |          |          |
| М  | 0.002352 | 7.11E-06 | 0        | 2.37E-05 | 0        | 0.002802 | 0 | 0        | 2.37E-06 | 0.002464 | 0        | 5.09E-05 | 0.000101  | 3.32E-05 | 0.001215 | 0.002967 | 3.55E-05 | 6.87E-05 | 8.65E-05 | 1.42E-05 | 0.002721 | 1.18E-06 | 1.18E-06 | 9.47E-06 | 1.78E-05 | 4.74E-06  | 0        | 0.000879 | 5.68E-05  | 0        | 1.18E-06 | 0.000386 | 0.006692 |          |          |          |
| Н  | 0.010381 | 1.18E-05 | 1.18E-06 | 6.04E-05 | 0.000666 | 0.009754 | 0 | 4.74E-06 | 9.47E-06 | 0.007158 | 0        | 0.000224 | 0         | 1.18E-06 | 0.002613 | 0.009291 | 0        | 7.11E-05 | 0.00094  | 0.000384 | 0.002488 | 1.18E-05 | 0        | 0.000226 | 0.000243 | 3.55E-06  | 0.000154 | 0.002617 | 0.001196  | 0        | 0.000153 | 0.001664 | 0.005123 |          |          |          |
| О  | 3.55E-06 | 0.00324  | 0.006726 | 0.004198 | 0.003924 | 0.002388 | 0 | 0.001825 | 0.000905 | 0.000657 | 0.002704 | 0.001452 | 0.005081  | 0.004371 | 0.008297 | 0.000165 | 0.001085 | 0.005097 | 0.005475 | 0.005938 | 5.33E-05 | 0.000113 | 0.000309 | 7.82E-05 | 0.001506 | 0.001185  | 0.000143 | 0        | 0         | 2.25E-05 | 0.000821 | 0.000467 | 0.023137 |          |          |          |
| П  | 0.001024 | 0        | 0        | 0        | 0        | 0.001966 | 0 | 0        | 0        | 0.000707 | 0        | 6.04E-05 | 0.000403  | 0        | 6.87E-05 | 0.007757 | 2.37E-05 | 0.005923 | 3.55E-06 | 3.79E-05 | 0.000452 | 0        | 0        | 1.07E-05 | 3.20E-05 | 1.18E-05  | 0        | 0.000022 | 3.32E-05  | 0        | 3.55E-06 | 0.000295 | 4.62E-05 |          |          |          |
| Р  | 0.006387 | 0.000201 | 0.00029  | 0.000339 | 0.000349 | 0.001456 | 0 | 0.000276 | 1.30E-05 | 0.00413  | 0        | 0.000586 | 5.80E-05  | 0.00011  | 0.000619 | 0.006342 | 5.80E-05 | 1.66E-05 | 0.000168 | 0.000371 | 0.002147 | 7.11E-06 | 8.41E-05 | 2.13E-05 | 6.28E-05 | 0.000212  | 2.84E-05 | 0        | 0.001157  | 0.000883 | 0        | 8.65E-05 | 0.000284 | 0.000624 |          |          |
| С  | 0.00173  | 5.09E-05 | 0.00171  | 1.18E-05 | 0.000323 | 0.004221 | 0 | 3.08E-05 | 1.89E-05 | 0.001241 | 0        | 0.004754 | 0.002361  | 0.000782 | 0.000893 | 0.002274 | 0.001612 | 0.000105 | 0.000488 | 0.009715 | 0.000441 | 7.11E-06 | 0.00014  | 4.86E-05 | 0.000399 | 4.97E-05  | 2.37E-06 | 1.89E-05 | 0.000311  | 0.002711 | 3.55E-06 | 0.000127 | 0.00288  | 0.002731 |          |          |
| Т  | 0.004796 | 7.11E-06 | 0.002398 | 7.11E-06 | 0.000116 | 0.004766 | 0 | 2.37E-06 | 3.55E-06 | 0.003572 | 0        | 0.000462 | 0.00024   | 9.47E-06 | 0.000989 | 0.013456 | 4.50E-05 | 0.002307 | 0.000766 | 6.28E-05 | 0.001091 | 0.000245 | 0        | 0.000191 | 8.29E-06 | 5.92E-05  | 0.000245 | 0        | 1.07E-05  | 3.43E-05 | 0.001467 | 0.005914 | 2.37E-06 | 3.43E-05 | 0.000495 | 0.004265 |
| У  | 3.32E-05 | 0.000477 | 0.00112  | 0.000911 | 0.001471 | 0.000723 | 0 | 0.001594 | 0.000261 | 2.37E-06 | 0.000115 | 0.000728 | 0.001103  | 0.001123 | 0.000203 | 3.55E-06 | 0.00055  | 0.000483 | 0.000905 | 0.001004 | 0        | 1.54E-05 | 0.000271 | 2.37E-06 | 0.000751 | 0.000655  | 0.000189 | 0        | 0         | 7.11E-06 | 0.00091  | 0.000123 | 0.006143 |          |          |          |
| Ф  | 6.16E-05 | 0        | 0        | 0        | 0.00107  | 0        | 0 | 0.000271 | 0        | 0.000217 | 0        | 0        | 1.78E-05  | 2.37E-06 | 1.18E-06 | 6.51E-05 | 0        | 0.00014  | 7.11E-06 | 1.42E-05 | 1.78E-05 | 1.18E-06 | 0        | 0        | 1.18E-06 | 0         | 0        | 0        | 4.74E-06  | 4.50E-05 | 0        | 1.18E-06 | 1.18E-06 | 2.84E-05 |          |          |
| Х  | 0.00077  | 0        | 9.24E-05 | 2.37E-06 | 1.18E-06 | 0.00105  | 0 | 0        | 9.47E-05 | 0        | 0        | 8.65E-05 | 5.68E-05  | 0.000113 | 0.002075 | 0        | 4.74E-05 | 4.03E-05 | 7.11E-06 | 0.000131 | 0        | 0        | 0        | 0        | 8.29E-06 | 0         | 0        | 1.18E-06 | 0         | 0        | 0        | 0        | 0        | 0.002472 |          |          |
| Ц  | 0.000444 | 1.18E-06 | 3.32E-05 | 0        | 0.000587 | 0        | 0 | 0.000121 | 0        | 9.59E-05 | 0        | 9.59E-05 | 0.118E-06 | 0        | 0.000275 | 0        | 0        | 0        | 0        | 0        | 0        | 0.000185 | 0        | 0        | 4.74E-06 | 0.118E-06 | 0        | 0        | 0.000127  | 0        | 0        | 0        | 0        | 0.00016  |          |          |
| Ч  | 0.002122 | 0.75E-06 | 0        | 0        | 0.003174 | 0        | 0 | 0        | 0.001215 | 0        | 0.000278 | 4.03E-05 | 5.92E-06  | 0.000458 | 4.38E-05 | 0        | 6.63E-05 | 0        | 0.004364 | 0.000863 | 0        | 0        | 0        | 0        | 0.000171 | 0         | 0        | 0        | 0.000159  | 0        | 0        | 0        | 0.00082  |          |          |          |
| Ш  | 0.000861 | 0        | 3.82E-05 | 1.18E-06 | 0        | 0.002149 | 0 | 0        | 0.001477 | 0        | 0.000353 | 0.0005   | 2.37E-06  | 0.000354 | 0.000268 | 5.92E-06 | 0        | 0        | 6.04E-05 | 0.000279 | 0        | 0        | 3.55E-06 | 0        | 1.18E-06 | 0         | 0        | 0.000457 | 0         | 3.55E-06 | 0        | 6.51E-05 |          |          |          |          |
| Щ  | 0.000271 | 0        | 0        | 0        | 0.001203 | 0        | 0 | 0.000602 | 0        | 0        | 0        | 0        | 0         | 2.84E-05 | 1.18E-06 | 0        | 4.74E-06 | 0        | 0        | 6.75E-05 | 0        | 0        | 6.75E-05 | 0        | 0        | 0         | 0        | 0        | 0         | 2.13E-05 | 0        | 0        | 5.92E-06 |          |          |          |
| Ъ  | 0        | 0        | 0        | 0        | 0.000113 | 0        | 0 | 0        | 0        | 0        | 0        | 0        | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0         | 0        | 0        | 0         | 0        | 0        | 0        | 0        |          |          |          |
| Ы  | 0        | 0.000473 | 0.000983 | 8.41E-05 | 0.000131 | 0.000855 | 0 | 2.01E-05 | 5.68E-05 | 7.11E-06 | 0.001107 | 0.000146 | 0.002405  | 0.001072 | 0.000323 | 0        | 0.00011  | 0.000279 | 0.000718 | 0.000664 | 5.92E-06 | 0        | 0.000713 | 0        | 0.000156 | 0.00042   | 0        | 0        | 0         | 0        | 0        | 0        | 0.00381  |          |          |          |
| Э  | 0        | 7.93E-05 | 2.49E-05 | 4.62E-05 | 3.08E-05 | 0.00043  | 0 | 1.18E-06 | 0.000149 | 0.000433 | 0        | 0.001133 | 0         | 0.000213 | 0.001099 | 4.74E-06 | 2.37E-06 | 0        | 0.00079  | 7.11E-05 | 0        | 0        | 1.18E-06 | 0.000104 | 5.68E-05 | 0.000386  | 1.66E-05 | 0        | 0         | 3.55E-06 | 0.000365 | 0.000387 | 0.010094 |          |          |          |
| Ю  | 0        | 0        | 1.18E-06 | 4.74E-06 | 0        | 0        | 0 | 0        | 0.000149 | 1.18E-06 | 2.37E-06 | 4.26E-05 | 4.38E-05  | 2.37E-06 | 2.13E-05 | 0        | 2.37E-06 | 1.18E-06 | 2.37E-06 | 0.002798 | 0        | 3.55E-06 | 1.18E-06 | 0        | 0        | 0         | 0        | 0        | 0         | 0        | 0        | 0        | 2.37E-06 |          |          |          |
| Я  | 1.18E-06 | 0.000597 | 3.55E-06 | 2.37E-06 | 0.000208 | 0        | 0 | 1.18E-06 | 1.18E-06 | 0        | 2.37E-06 | 1.18E-05 | 1.07E-05  | 1.42E-05 | 0        | 9.47E-06 | 3.67E-05 | 0.000153 | 0.000226 | 0        | 2.37E-06 | 8.29E-06 | 2.37E-06 | 3.79E-05 | 2.25E-05 | 0.000327  | 0        | 0        | 0         | 0        | 2.72E-05 | 0        | 0.003272 |          |          |          |
|  | 0        | 1.89E-05 | 0.000238 | 0.000285 | 0.000515 | 7.70E-05 | 0 | 0.00027  | 0.000301 | 4.50E-05 | 0.000114 | 0.000105 | 0.00078   | 0.000304 | 0.000611 | 0        | 4.97E-05 | 3.91E-05 | 0.000779 | 0.001306 | 0        | 0        | 0.000173 | 3.79E-05 | 0.000129 | 4.50E-05  | 0.000185 | 0        | 0         | 0        | 7.93E-05 | 0.000134 | 0.010709 |          |          |          |
|  | 0.007006 | 0.006347 | 0.013695 | 0.003134 | 0.006656 | 0.004974 | 0 | 0.002437 | 0.003999 |          |          |          |           |          |          |          |          |          |          |          |          |          |          |          |          |           |          |          |           |          |          |          |          |          |          |          |





|   |  |
|---|--|
| Entropy of symbols of alphabet with struck: 4.19676 ;             |  |
| Entropy of symbols of alphabet without struck: 4.7102 ;           |  |
|   |  |
| Entropy on symbol of bigram with space and step one: 3.89105 ;    |  |
| Entropy on symbol of bigram with space and step two: 3.89183 ;    |  |
|   |  |
| Entropy on symbol of bigram without space and step one: 4.11611 ; |  |
| Entropy on symbol of bigram without space and step two: 4.11495 ; |  |

## Висновок

Після виконання практикуму, я засвоїв поняття ентропії на символ джерела та його надлишковості, вивчив та порівняв їх в різних моделях джерела відкритого тексту для наближеного визначення ентропії, набув практичних навичок щодо оцінки ентропії на символ джерела.

Також зазначена робота, може бути корисною в застосуванні до математичної статистики.