Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

**ЛАБОРАТОРНА РОБОТА №1**

**з дисципліни**

**«Криптографія»**

**на тему: «Експериментальна оцінка ентропії на символ джерела відкритого тексту»**

Виконали:

студенти 3 курсу ФТІ

групи ФБ-73

Маковецький Андрій та Бадарак Оксана

Перевірили:

Чорний О.

Савчук М. М.

Завадська Л. О.

**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.

**Порядок виконання роботи:**

0. Уважно прочитати методичні вказівки до виконання комп’ютерного практикуму.

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

2. За допомогою програми CoolPinkProgram оцінити значення H10 , H20 , H30 .

3. Використовуючи отримані значення ентропії, оцінити надлишковість російської мови в різних моделях джерела.

**Ентропії:**

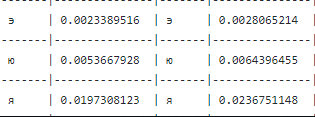
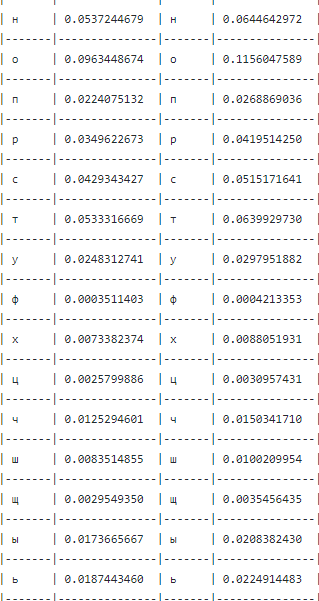
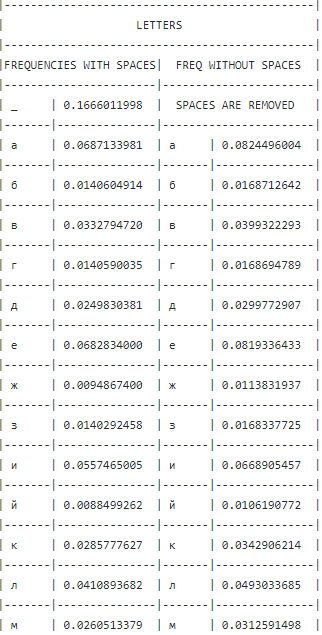
Letters entropy with spaces: 4.363885193043505

Letters entropy without spaces: 4.45646766252117

Bigrams entropy with spaces: 7.9038768971118705

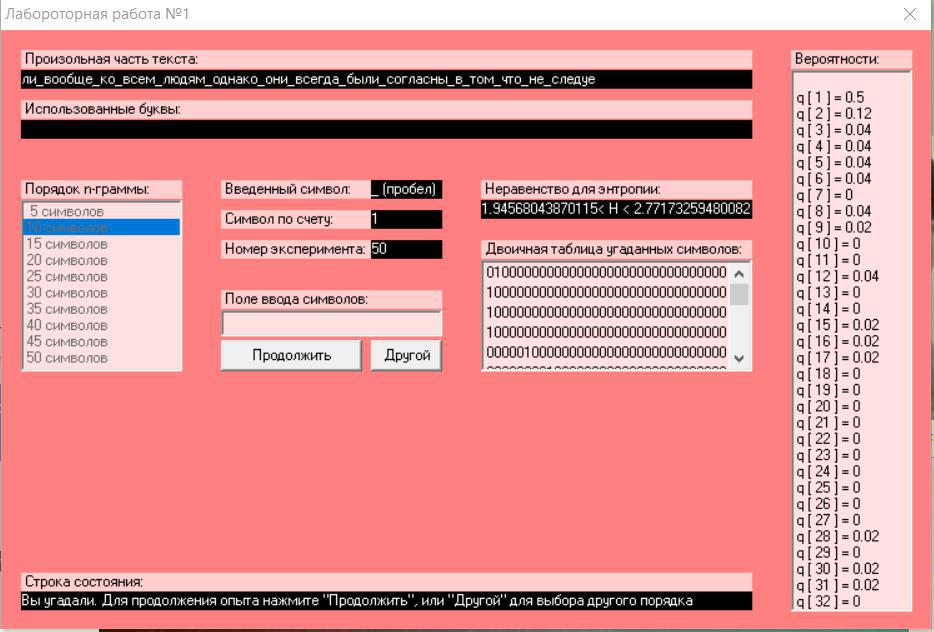
Bigrams entropy without spaces: 8.270837200168474

**Частоти:**

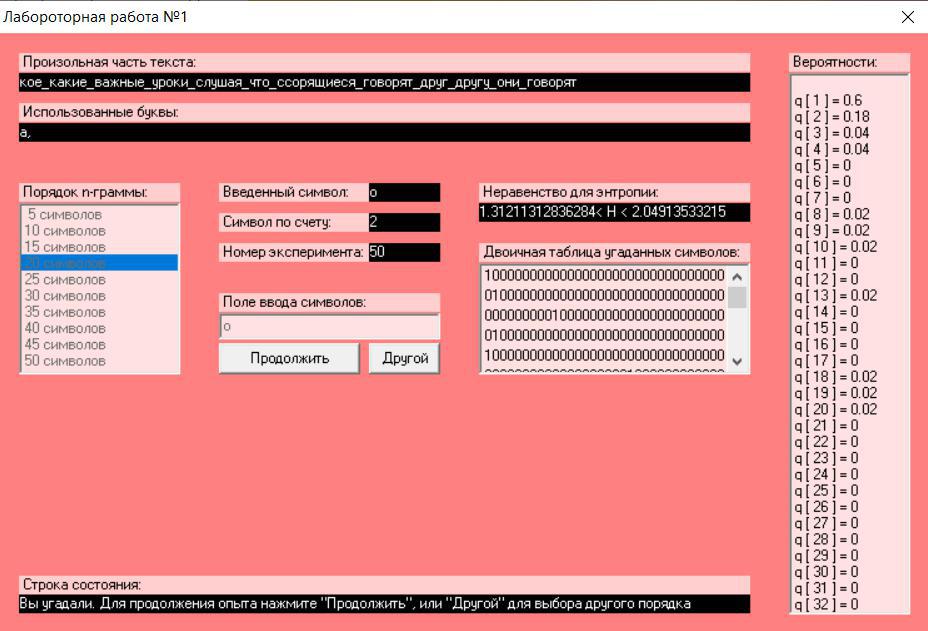


**Скріншоти результатів CoolPinkProgram:**

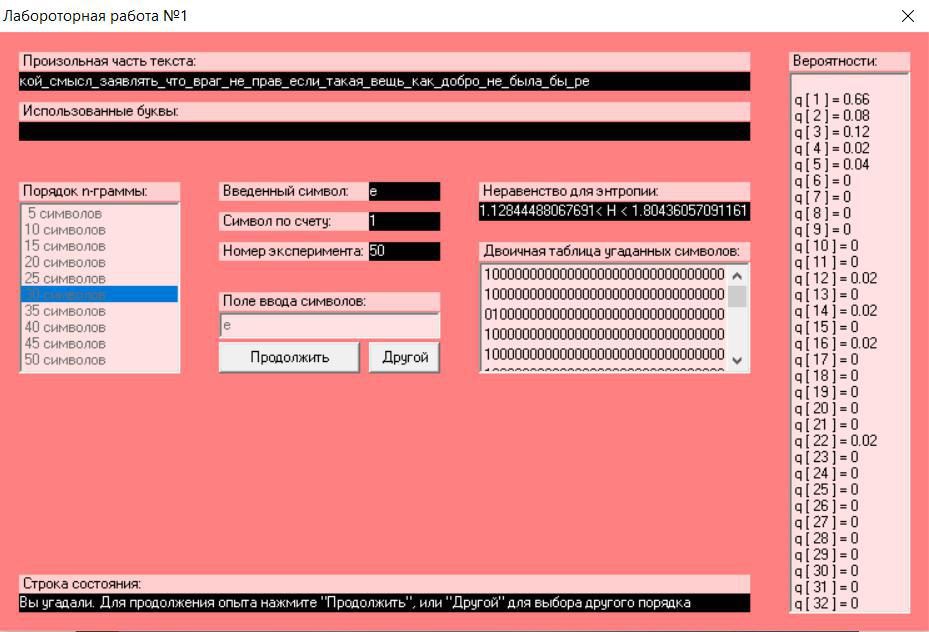
**10-грамми:**

****

**20-грамми:**

****

**30-грамми:**

****

**Код:**

|  |
| --- |
| **Import re** |
|  | **from math import log** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **def import\_data(filepath):** |
|  |  |
|  | **with open(filepath, 'r', encoding='utf-8') as data\_source:** |
|  | **return data\_source.read()** |
|  |  |
|  |  |
|  | **def freq\_by\_number(numbers\_dict, text\_length):** |
|  |  |
|  | **for key in numbers\_dict.keys():** |
|  | **numbers\_dict[key] = numbers\_dict[key] / text\_length** |
|  |  |
|  | **return numbers\_dict** |
|  |  |
|  |  |
|  | **def calculate\_freq\_letters(text):** |
|  |  |
|  | **freq = {}** |
|  |  |
|  | **for letter in text:** |
|  | **if letter in freq:** |
|  | **freq[letter] += 1** |
|  | **else:** |
|  | **freq[letter] = 1** |
|  |  |
|  | **return freq\_by\_number(freq, len(text))** |
|  |  |
|  |  |
|  | **def calculate\_freq\_bigrams(text):** |
|  |  |
|  | **freq = {}** |
|  |  |
|  | **for i in range(len(text)-1):** |
|  | **bigram = text[i:i+2]** |
|  | **if bigram in freq:** |
|  | **freq[bigram] += 1** |
|  | **else:** |
|  | **freq[bigram] = 1** |
|  |  |
|  | **return freq\_by\_number(freq, len(text))** |
|  |  |
|  |  |
|  | **def calculate\_entropy(freq\_dict):** |
|  |  |
|  | **entropy = 0** |
|  | **for i in freq\_dict.values():** |
|  | **entropy += i \* log(i, 2)** |
|  |  |
|  | **return –entropy** |
|  |  |
|  |  |
|  | **def print\_frequencies(freq):** |
|  |  |
|  | **sum = 0** |
|  | **for key, value in sorted(freq.items()):** |
|  | **print(key, '\t', value)** |
|  | **sum += value** |
|  | **print(sum)** |
|  |  |
|  |  |
|  | **def create\_results\_file(freq\_letters, freq\_bigrams,** |
|  | **entropy\_letters, entropy\_bigrams,** |
|  | **freq\_letters\_ns, freq\_bigrams\_ns,** |
|  | **entropy\_letters\_ns, entropy\_bigrams\_ns):** |
|  |  |
|  | **res\_file = open('lab1\_results.txt', 'w+', encoding='utf-8')** |
|  |  |
|  | **res\_file.write('Letters entropy with spaces: {}\n'.format(entropy\_letters))** |
|  | **res\_file.write('Letters entropy without spaces: {}\n\n'.format(entropy\_letters\_ns))** |
|  | **res\_file.write('Bigrams entropy with spaces: {}\n'.format(entropy\_bigrams))** |
|  | **res\_file.write('Bigrams entropy without spaces: {}\n\n'.format(entropy\_bigrams\_ns))** |
|  | **res\_file.write('Frequency tables for all experiments are listed below.\n\n')** |
|  |  |
|  | **res\_file.write('|-----------------------------------------------|\n')** |
|  | **res\_file.write('| LETTERS |\n')** |
|  | **res\_file.write('|-----------------------------------------------|\n')** |
|  | **res\_file.write('|FREQUENCIES WITH SPACES| FREQ WITHOUT SPACES |\n')** |
|  | **res\_file.write('|-----------------------|-----------------------|\n')** |
|  | **res\_file.write('| ' + '\_' + ' | ' + '{:.10f}'.format(freq\_letters['\_']) + ' |'** |
|  | **+ ' SPACES ARE REMOVED |\n')** |
|  | **res\_file.write('|-------|---------------|-----------------------|\n')** |
|  |  |
|  | **for key, value in sorted(freq\_letters.items()):** |
|  | **if key != '\_':** |
|  | **res\_file.write('| ' + key + ' | ' + '{:.10f}'.format(value) + ' | '** |
|  | **+ key + ' | ' + '{:.10f}'.format(freq\_letters\_ns[key]) + ' |\n')** |
|  | **res\_file.write('|-------|---------------|-------|---------------|\n')** |
|  |  |
|  | **res\_file.write('|-----------------------------------------------|\n')** |
|  | **res\_file.write('|-----------------------------------------------|\n')** |
|  | **res\_file.write('| BIGRAMS |\n')** |
|  | **res\_file.write('|-----------------------------------------------|\n')** |
|  | **res\_file.write('|FREQUENCIES WITH SPACES| FREQ WITHOUT SPACES |\n')** |
|  | **res\_file.write('|-----------------------|-----------------------|\n')** |
|  |  |
|  | **for key, value in sorted(freq\_bigrams.items()):** |
|  | **if '\_' not in key:** |
|  | **res\_file.write('| ' + key + ' | ' + '{:.10f}'.format(value) + ' | '** |
|  | **+ key + ' | ' + '{:.10f}'.format(freq\_bigrams\_ns[key]) + ' |\n')** |
|  | **res\_file.write('|-------|---------------|-------|---------------|\n')** |
|  | **else:** |
|  | **res\_file.write('| ' + key + ' | ' + '{:.10f}'.format(value) + ' | '** |
|  | **+ ' SPACES ARE REMOVED |\n')** |
|  | **res\_file.write('|-------|---------------|-------|---------------|\n')** |
|  |  |
|  | **res\_file.close()** |
|  |  |
|  |  |
|  | **def data\_remove\_spaces(text):** |
|  |  |
|  | **return ''.join([letter for letter in text if letter != ' '])** |
|  |  |
|  |  |
|  |  |
|  | **def main():** |
|  |  |
|  |  |
|  | **filepath = 'text\_parsed.txt'** |
|  |  |
|  | **data = import\_data(filepath)** |
|  |  |
|  |  |
|  | **#** |
|  | **# Text with spaces** |
|  | **#** |
|  |  |
|  | **replaced\_spaces\_text = re.sub(' ', '\_', data) # Replace spaces for better readability** |
|  |  |
|  | **freq\_letters = calculate\_freq\_letters(replaced\_spaces\_text)** |
|  | **freq\_bigrams = calculate\_freq\_bigrams(replaced\_spaces\_text)** |
|  |  |
|  |  |
|  | **entropy\_letters = calculate\_entropy(freq\_letters)** |
|  | **entropy\_bigrams = calculate\_entropy(freq\_bigrams)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **#** |
|  | **# Text without spaces** |
|  | **#** |
|  |  |
|  | **no\_space\_text = data\_remove\_spaces(data)** |
|  |  |
|  | **freq\_letters\_nospace = calculate\_freq\_letters(no\_space\_text)** |
|  | **freq\_bigrams\_nospace = calculate\_freq\_bigrams(no\_space\_text)** |
|  |  |
|  |  |
|  | **entropy\_letters\_nospace = calculate\_entropy(freq\_letters\_nospace)** |
|  | **entropy\_bigrams\_nospace = calculate\_entropy(freq\_bigrams\_nospace)** |
|  |  |
|  |  |
|  |  |
|  | **create\_results\_file(freq\_letters, freq\_bigrams,** |
|  | **entropy\_letters, entropy\_bigrams,** |
|  | **freq\_letters\_nospace, freq\_bigrams\_nospace,** |
|  | **entropy\_letters\_nospace, entropy\_bigrams\_nospace)** |
|  |  |
|  |  |
|  |  |
|  | **main()** |