Звіт по комп’ютерному практикуму №1

Виконували:
Ракович Дарина ФБ-73
Пекарчук Данило ФБ-74

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 ентропіх на символ джерела

Постановка задачі:
Основною метою було написання програми для підрахунку чатот 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 тому ж тексті, в якому вилучено всі пробіли.

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 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.

Код програми:

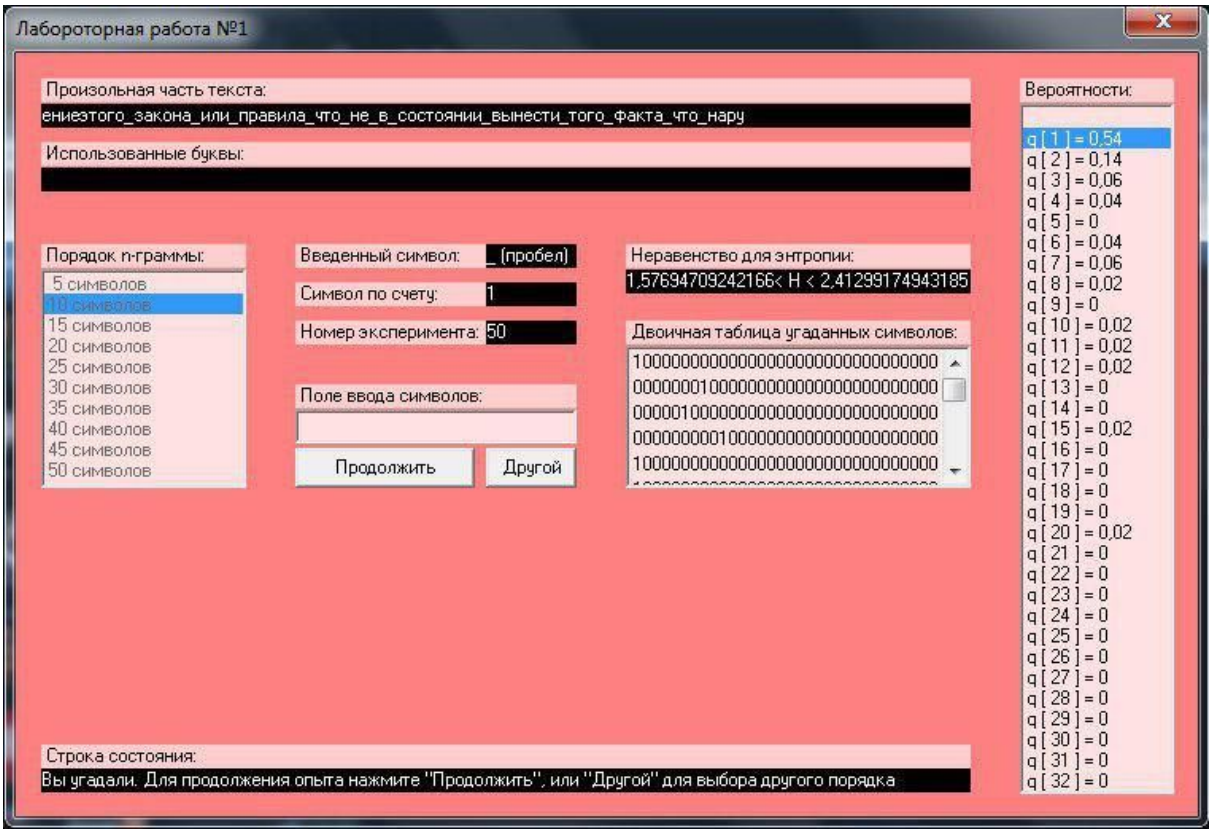
```
import sys
import math
import re
from collections import defaultdict


def get_file_lines(fname):
    with open(fname) as f:
        return f.readlines()
```

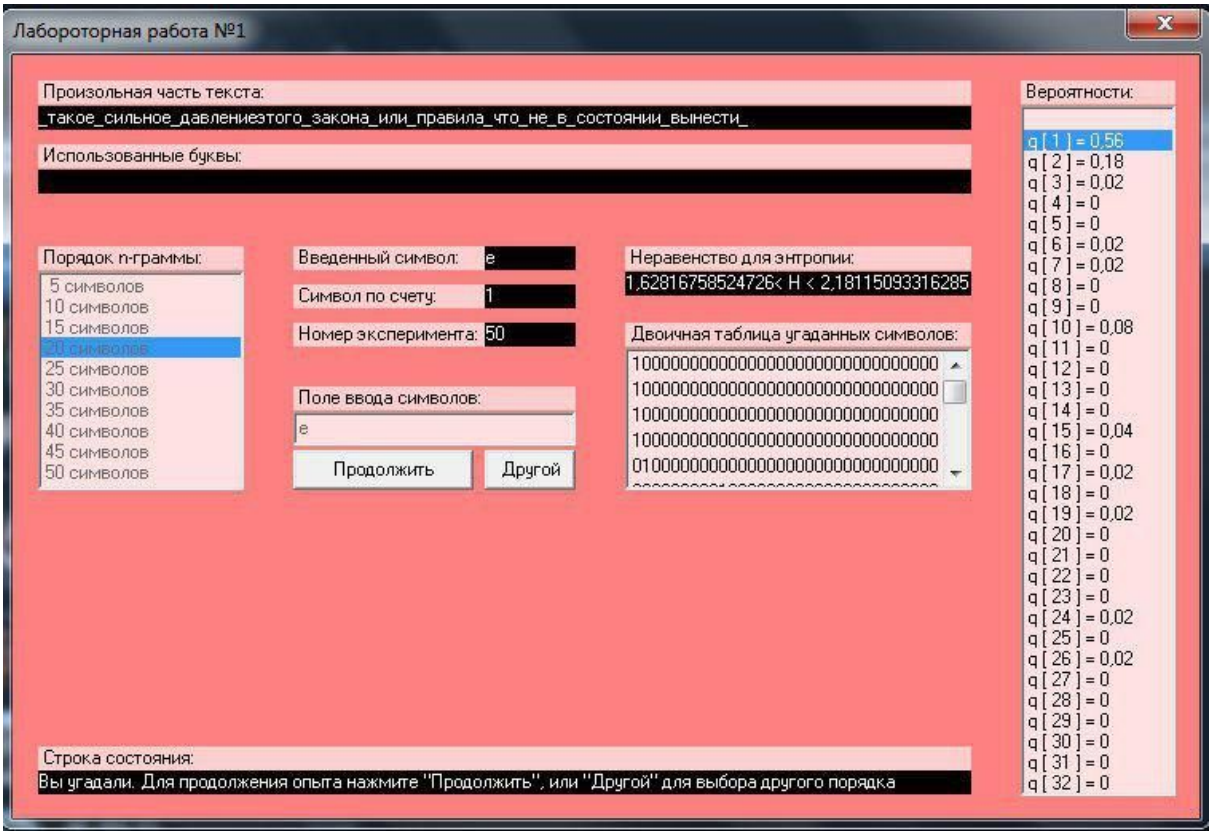
```
def remove_not_alpha(lines, need_space):
    return_list = list()
    for line in lines:
        re_check = '[^а-яА-Я{}]+'.format(' ' if need_space else '')
        fixed_line = re.sub(re_check, '', line.lower())
        return_list.append(fixed_line)

    return return_list


def get_ngram_dict(lines, ngram_num=2, step=2):
    ngram_counter = 0
    ngram_dict = defaultdict(int)
    for line in lines:
        for char_counter in range(0, len(line), step):
            ngram = line[char_counter:char_counter+ngram_num]
            if len(ngram) != ngram_num:
                continue

            ngram_dict[ngram] += 1
            ngram_counter += 1

    return_dict = dict()
    for elem, amount_in_text in ngram_dict.items():
        return_dict[elem] = amount_in_text / ngram_counter

    return return_dict


def get_entropy(ngram_dict, num):
    result = 0

    grouped_dict = dict()
    for item, value in ngram_dict.items():
        try:
            grouped_dict[value].append(item)
        except KeyError:
            grouped_dict[value] = [item]
```

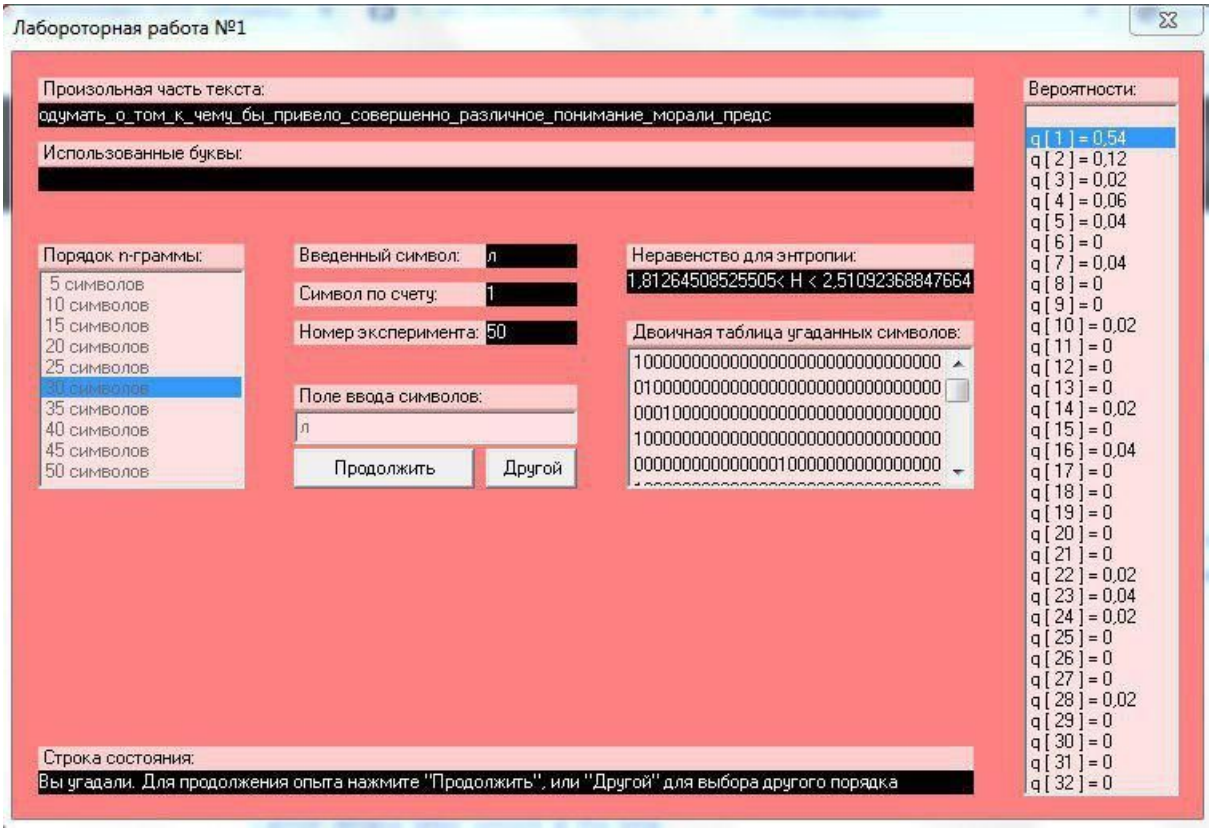
```
    for number, elements in grouped_dict.items():
        result += (number * math.log(number, 2)) * len(elements)

    return result * (-1/num)


def get_redundancy(entropy):
    return 1-(entropy/(math.log(33, 2)))


def write_to_file(
        ngram_dict,
        entropy_val,
        redundancy_val,
        need_space,
        num,
        fname
    ):
    with open('out.txt', 'w') as f:
        f.write(f'Got text from {fname}...filtering out non-alphas{"
and spaces" if not need_space else ""}...looking for
{num}-grams\nResult:\n')

        for item, value in ngram_dict.items():
                f.write(item + ' -> ' + str(value) + '\n')

        f.write("Entropy: " + str(entropy_val) + '\n')
        f.write("Redundancy: " + str(redundancy_val) + '\n')


def main():
    num = int(sys.argv[1]) if len(sys.argv) >= 2 else 2
    need_space = eval(sys.argv[2]) if len(sys.argv) >= 3 else False
    fname = sys.argv[3] if len(sys.argv) >= 4 else 'TEXT'
    step = int(sys.argv[4]) if len(sys.argv) >= 5 else num

    lines = get_file_lines(fname)
    filtered_lines = remove_not_alpha(
        lines,
        need_space
```

```
    )

    ngram_dict = get_ngram_dict(
        filtered_lines,
        num,
        step=step
    )

    entropy_val = get_entropy(
        ngram_dict,
        num
    )
    redundancy_val = get_redundancy(entropy_val)

    for item, value in ngram_dict.items():
        print(item, ' -> ', value)

    print("entropy: ", entropy_val)
    print("redundancy: ", redundancy_val)

    write_to_file(
        ngram_dict,
        entropy_val,
        redundancy_val,
        need_space,
        num,
        fname
    )


if __name__ == '__main__':
    main()
```

Хід роботи: В ході роботи виникли деякі труднощі з структурою даних та як найшвидше та найкраще їх відфільтрувати. Для фільтрації ми використали regex, а для структури даних найкраще підходить словник через те що всі операції в ньому це O(1). Потім запустили CoolPinkProgram, та намагались підібрати наступну букву для різних комбінацій, результати можна побачити на картинках




Лабороторная работа №1
Произольная часть текста:
_такое_сильное_давлениеэтого_закона_или_правила_что_не_в_состоянии_вынести_
Использованные буквы:
Порядок n-граммы:
5 символов
10 символов
15 символов
20 символов
25 символов
30 символов
35 символов
40 символов
45 символов
50 символов
Введенный символ: е
Символ по счету: 1
Номер эксперимента: 50
Поле ввода символов:
е
Продолжить
Другой
Неравенство для энтропии:
1,62816758524726< H < 2,18115093316285
Двоичная таблица угаданных символов:
Вероятности:
q [1] = 0,56
q [2] = 0,18
q [3] = 0,02
q [4] = 0
q [5] = 0
q [6] = 0,02
q [7] = 0,02
q [8] = 0
q [9] = 0
q [10] = 0,08
q [11] = 0
q [12] = 0
q [13] = 0
q [14] = 0
q [15] = 0,04
q [16] = 0
q [17] = 0,02
q [18] = 0
q [19] = 0,02
q [20] = 0
q [21] = 0
q [22] = 0
q [23] = 0
q [24] = 0,02
q [25] = 0
q [26] = 0,02
q [27] = 0
q [28] = 0
q [29] = 0
q [30] = 0
q [31] = 0
q [32] = 0
Строка состояния:
Вы угадали. Для продолжения опыта нажмите "Продолжить", или "Другой" для выбора другого порядка


Лабороторная работа №1
Произольная часть текста:
одумать_о_том_к_чему_бы_привело_совершенно_различное_понимание_морали_предс
Использованные буквы:
Порядок n-граммы:
5 символов
10 символов
15 символов
20 символов
25 символов
30 символов
35 символов
40 символов
45 символов
50 символов
Введенный символ: л
Символ по счету: 1
Номер эксперимента: 50
Поле ввода символов:
л
Продолжить
Другой
Неравенство для энтропии:
1,81264508525505< H < 2,51092368847664
Двоичная таблица угаданных символов:
Вероятности:
q [1] = 0,54
q [2] = 0,12
q [3] = 0,02
q [4] = 0,06
q [5] = 0,04
q [6] = 0
q [7] = 0,04
q [8] = 0
q [9] = 0
q [10] = 0,02
q [11] = 0
q [12] = 0
q [13] = 0
q [14] = 0,02
q [15] = 0
q [16] = 0,04
q [17] = 0
q [18] = 0
q [19] = 0
q [20] = 0
q [21] = 0
q [22] = 0,02
q [23] = 0,04
q [24] = 0,02
q [25] = 0
q [26] = 0
q [27] = 0
q [28] = 0,02
q [29] = 0
q [30] = 0
q [31] = 0
q [32] = 0
Строка состояния:
Вы угадали. Для продолжения опыта нажмите "Продолжить", или "Другой" для выбора другого порядка

Висновки: В цьому комп’ютерному практикумі ми засвоїли поняття ентропії на символ джерела та його надлишковості, вивчили та порівняли різниі моделі джерела відкритого тексту для наближеного визначення ентропії, набули практичних навичок щодо оцінки ентропії на символ джерела.