«Київський Політехнічний Інститут ім. Ігоря Сікорського»

Фізико-технічний інститут

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №1

Експериментальна оцінка ентропії на символ джерела відкритого тексту

Виконали

студенти групи ФБ-73:

Деркач Вячеслав та Михалко Дмитро

Київ 2019

Мета комп'ютерного практикуму:

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:

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.

За допомогою програми CoolPinkProgram оцінити значення $H^{(10)}$, $H^{(20)}$, $H^{(30)}$.

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

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 замішуватись 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» на «ь». Пробіл також вважається буквою. Таким чином, кількість букв алфавіту м = 32. При підрахунку $H^{(10)}$, $H^{(20)}$, $H^{(30)}$необхідно виконати не менш ніж 50 експериментів.

Хід роботи:

1) Ми прочитали завдання та методичні вказівки;
2) Проаналізували завдання та виписали всі нюанси та деталі лабораторної;
3) Продумали на листку, як будемо виконувати той чи інший функціонал;
4) За допомоги програми CoolPinkProgram.exe виконали завдання з підрахунку $H^{(10)}$, $H^{(20)}$, $H^{(30)}$на готовому тексті зробивши для кожного випадка по 50 експериментів;
5) Створили програму на мові програмування Python, яка повністю виконує всі інші завдання лабораторної роботи;
6) Створили протокол, що описує нашу роботу;
7) Відправили все на Github;

Програмний код:

```
import string
import re
import math

def monogramDictCreate(alpha):
    return {item: 0 for item in alpha}

def monogramCount(text, alpha):
    monogramDict = monogramDictCreate(alpha)
    for elem in text:
        monogramDict[elem]+=1
    for elem in monogramDict:
        monogramDict[elem]/=len(text)
    return monogramDict

def bigramDictCreate(alpha):
    return {item1+item2: 0 for item1 in alpha for item2 in alpha}

def bigramCount(text, alpha):
    bigramDict = bigramDictCreate(alpha)
    for position in range(len(text)-1):
        bigramDict[text[position]+text[position+1]] += 1
    for elem in bigramDict:
        bigramDict[elem]/= len(text)-1
    return bigramDict

def bigramCountWithSpace(text, alpha):
    bigramDict = bigramDictCreate(alpha)
    for position in range(0,len(text)-1,2):
        bigramDict[text[position]+text[position+1]] += 1
    for elem in bigramDict:
        bigramDict[elem]/= len(text)//2
    return bigramDict

def entropyMono(Dict):
    entropyData = 0
    for elem in Dict:
        if Dict[elem] != 0:
            entropyData -= Dict[elem]*math.log(Dict[elem],2)
    return entropyData

def entropyBi(Dict):
    entropyData = 0
    for elem in Dict:
        if Dict[elem] != 0:
            entropyData -= Dict[elem]*math.log(Dict[elem],4)
    return entropyData

def residual(entropy, alpha):
    return 1 - (entropy / (math.log(len(alpha), 2)))
```

```
alphabet = open("Alphabet.txt",encoding='utf-8')
alphabetData = alphabet.read()
textGeneral = open("Text1.txt", encoding='utf-8')
textGeneralData = textGeneral.read()
textGeneralDataFiltred = re.sub(r'[^а-яА-Я ]+', '', textGeneralData)
textGeneralDataFiltred = textGeneralDataFiltred.lower()
textGeneralDataFiltred = re.sub(r'\s+', ' ', textGeneralDataFiltred)
textGeneralDataFiltred = re.sub('ъ', 'ь', textGeneralDataFiltred)
with open("Text2.txt", "w") as textFiltred:
    textFiltred.write(textGeneralDataFiltred)
textDataWithoutSpaces = textGeneralDataFiltred.replace(' ','')
with open("Text3.txt", "w") as textWithoutSpaces:
    textWithoutSpaces.write(textDataWithoutSpaces)
print('Энтропия: монограммы без пробелов')
print(entropyMono(monogramCount(textDataWithoutSpaces, alphabetData)))
print('Энтропия: биграммы без пробелов')
print(entropyBi(bigramCount(textDataWithoutSpaces, alphabetData)))
print('Энтропия: биграммы без пробелов с шагом')
print(entropyBi(bigramCountWithSpace(textDataWithoutSpaces, alphabetData)))
print('Энтропия: монограммы с пробелами')
print(entropyMono(monogramCount(textGeneralDataFiltred, alphabetData + ' ')))
print('Энтропия: биграммы с пробелами')
print(entropyBi(bigramCount(textGeneralDataFiltred, alphabetData + ' ')))
print('Энтропия: биграммы с пробелами с шагом')
print(entropyBi(bigramCountWithSpace(textGeneralDataFiltred, alphabetData + '
')))
print('Остаточность монограмм без пробелов')
print(residual(entropyMono(monogramCount(textDataWithoutSpaces,
alphabetData)), alphabetData))
print('Остаточность биграмм без пробелов')
print(residual(entropyBi(bigramCount(textDataWithoutSpaces, alphabetData)),
alphabetData))
print('Остаточность биграмм без пробелов с шагом')
print(residual(entropyBi(bigramCountWithSpace(textDataWithoutSpaces,
alphabetData)), alphabetData))
print('Остаточность монограмм с пробелами')
print(residual(entropyMono(monogramCount(textGeneralDataFiltred, alphabetData
+ ' ')), alphabetData + ' '))
print('Остаточность биграмм с пробелами')
print(residual(entropyBi(bigramCount(textGeneralDataFiltred, alphabetData + '
')), alphabetData + ' '))
print('Остаточность биграмм с пробелами с шагом')
print(residual(entropyBi(bigramCountWithSpace(textGeneralDataFiltred,
alphabetData + ' ')), alphabetData + ' '))
```

Частоти букв та біграм

Частоти букв з пробілами:

'а': 0.08457158011831181,
'б': 0.016897740583308164,
'в': 0.0474720611744252,
'г': 0.017030344106141754,
'д': 0.030069690518111432,
'е': 0.08134121096483797,
'ж': 0.010098126606896856,
'з': 0.01668041814310867,
'и': 0.07217683416456097,
'й': 0.01245736428397781,
'к': 0.03613077654096344,
'л': 0.05021990084203237,
'м': 0.030856103077138416,
'н': 0.06148199170491296,
'о': 0.10686002224792439,
'п': 0.030612996618610167,
'р': 0.04813507878859315,
'с': 0.05424589113250775,
'т': 0.05804903383599891,
'у': 0.028469239666133798,
'ф': 0.0022321593010320973,
'х': 0.009407483258805242,
'ц': 0.005061034454815349,
'ч': 0.014759508777616526,
'ш': 0.00854924379157673,
'щ': 0.0033813898322565435,
'ы': 0.019437466388690393,
'ь': 0.017183206500519365,
'э': 0.0024144891449282835,
'ю': 0.005538038793897291,
'я': 0.0181795746373662

Частоти букв без пробілів:

'а': 0.07203600545289972,
'б': 0.014393082535504245,
'в': 0.040435541736801896,
'г': 0.014506030976109837,
'д': 0.025612627635659704,
'е': 0.06928445594148015,
'ж': 0.00860133749778417,
'з': 0.014207972591178413,
'и': 0.06147846371296036,
'й': 0.01061087850355866,
'к': 0.0307753126083403,
'л': 0.04277608442268444,
'м': 0.026282474637584534,
'н': 0.05236885823245103,
'о': 0.09102075584468962,
'п': 0.026075402496474283,
'р': 0.04100028393982986,
'с': 0.04620532457773755,
'т': 0.049444748603439594,
'у': 0.02424940270668388,
'ф': 0.0019012987501941302,
'х': 0.008013064369630047,
'ц': 0.004310865483113424,
'ч': 0.012571788930739075,
'ш': 0.007282036962377188,
'щ': 0.0028801852354425933,
'ы': 0.01655635891876968,
'ь': 0.014636235428474616,
'э': 0.002056602856026819,
'ю': 0.004717166123625206,
'я': 0.0154849174613583,
' ': 0.1482244348263967

Частоти біграм надіслали окремим файлом *frequency_of_bigrams* для полегшення читання протоколу

Ентропія:

Без пробілів:

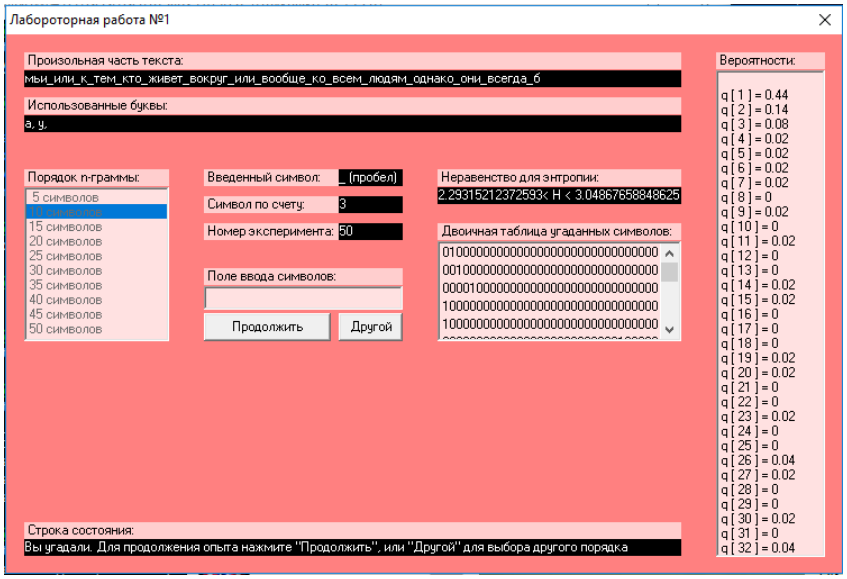
Ентропія: монограми без пробілів
4.464889157886663
Ентропія: біграми без пробілів
4.166091298011274
Ентропія: біграми без пробілів з кроком 2
4.1652352520328195

З пробілами:

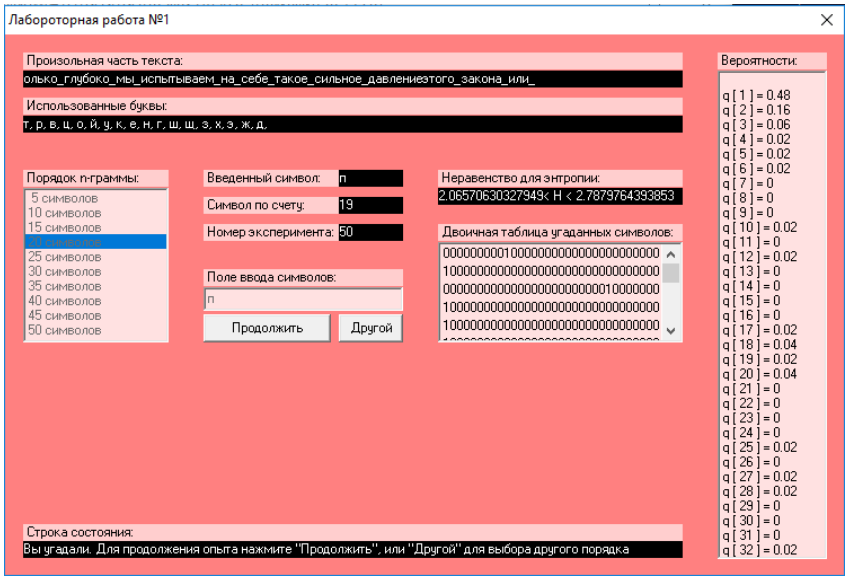
Ентропія: монограми з пробілами
4.408462543428977
Ентропія: біграми з пробілами
4.030615199637492
Ентропія: біграми з пробілами з кроком 2
4.02938135837728

Оцінки для $H^{(10)}$, $H^{(20)}$, $H^{(30)}$

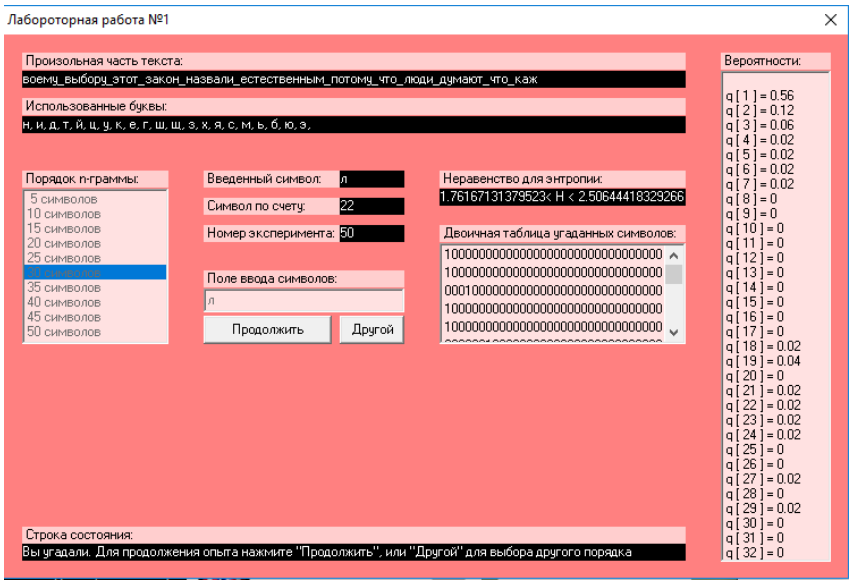
$H^{(10)}$: 2.29315212372593 < H < 3.04867658848625



$H^{(20)}$: 2.06570630327949 < H < 2.7879764393853



$H^{(30)}$: 1.76167131379523 < H < 2.50644418329266



Надлишковість російської мови в різних моделях джерела:

Без пробілів:

Надлишковість монограм без пробілів
0.09876620179025619
Надлишковість біграм без пробілів
0.15907827687878973
Надлишковість біграм без пробілів з кроком 2
0.159251068977613

З пробілами:

Надлишковість монограм з пробілами
0.1183074913142047
Надлишковість біграм з пробілами
0.19387696007250155
Надлишковість біграм з пробілами з кроком 2
0.19412372832454405

Висновки:

Нами було засвоєно поняття ентропії на символ джерела та його надлишковості, ми ознайомились та порівняли різні моделі джерел відкритого тексту для наближеного визначення ентропії. Також набули практичних навичок щодо оцінки ентропії на символ джерела.