

Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Криптографія

Комп'ютерний практикум №1
Експериментальна оцінка на символ джерела відкритого тексту

Виконали:
Студенти 3 курсу
Загородній Я.М, Венгер П.Ю.

Перевірив:

Мета роботи: засвоєння поняття 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

Хід роботи

Пункт 1. Весь код програми поданий в lab1_cryptography.ipynb, також будуть прикріплені файли excel з таблицками

```
import re
import collections
import unicodedata

text_war_peace = open("D:\\Web-developed\\Krpt\\lab1_text.txt").read()
text_war_peace = text_war_peace.lower().replace("n", "").replace("\n", " ")
text_war_peace = ' '.join(text_war_peace.split())

#without space
string_text = ''.join(c for c in text_war_peace if unicodedata.category(c).startswith('L'))

#with space
# string_text = re.sub(r'^\w\s+|[\d]+', r'', text_war_peace).strip()

word_sort = sorted(string_text)
print(string_text)
print("OK")
```

✓ 0.9s

левтолстойвойнаимиртомапервыйивторойвшкловскийвойнаимирльватолстогозамыселвгодупоявилосьобъявле

OK

```

import regex
import collections
import unicodedata

text_war_peace = open("D:\Web-developed\Krpt\lab1_text.txt").read()
text_war_peace = text_war_peace.lower().replace("n", "").replace("\n", " ")
text_war_peace = ' '.join(text_war_peace.split())

#without space
# string_text = ''.join(c for c in text_war_peace if unicodedata.category(c).startswith('L'))

#with space
string_text = regex.sub(r'^\w\s|' + [[\d]] + ', r'',text_war_peace).strip()

word_sort = sorted(string_text)
print(string_text)
print("OK")

```

✓ 0.7s

лев толстой война и мир тома первый и второй в шкловский война и мир льва толстого замысел в году
OK

```

1 #Count
2
3 list_alp = [] ...
4 for i in range(0,len(word_sort)):
5     list_alp.append(word_sort[i])
6
7 alpcount = dict(collections.Counter(list_alp))
8 print("\n")
9 freq = {k: alpcount[k] / len(word_sort) for k in alpcount}
10 print_dict_in_columns(alpcount)
11 print("\n")
12 print_dict_in_columns(freq)

```

✓ 0.5s

```

a: 2713 b: 472 c: 1465 d: 1302 e: 6306 f: 413 g: 357
h: 560 i: 2950 j: 277 k: 64 l: 2024 m: 1637 o: 2457
p: 1044 q: 415 r: 2656 s: 2995 t: 2345 u: 2457 v: 885
w: 39 x: 271 y: 88 z: 219 a: 104486 b: 21826 m: 56730
r: 24829 d: 37700 e: 101669 x: 12686 z: 21449 w: 81798 ã: 14642
k: 42714 n: 61662 m: 37334 n: 81257 o: 140492 n: 31603 p: 55571
c: 65937 t: 71753 y: 34128 ð: 2493 x: 10631 ç: 4526 ç: 17392
m: 11754 m: 3582 b: 548 m: 23457 b: 24396 z: 3840 w: 8110
r: 27868 e: 62

a: 0.00212728253739601 b: 0.0003700985465790976 c: 0.0011487168871575804 d: 0.0010209074314533582 e: 0.004944579310863961 f: 0.0003238362282567104 g: 0.000279926231204953
h: 0.0004390999705175734 i: 0.0023131159161193598 j: 0.00021719766398815686 k: 5.018285377343696e-05 l: 0.001587032750584944 m: 0.0012835833066736923 o: 0.0019265511206458534
p: 0.0008186078021791904 q: 0.0003254044424371303 r: 0.0020825804315976337 s: 0.0023484007351788076 t: 0.0018387311265423386 u: 0.0019265511206458534 v: 0.0006939347748358079
w: 3.058017651818815e-05 x: 0.00021249302144689714 y: 6.900142393847582e-05 z: 0.0001717194527559796 a: 0.0819282134276771 ð: 0.017113921350922424 m: 0.044482395227610605
r: 0.019468594942822912 d: 0.02956083730091521 e: 0.07971938375455566 x: 0.009947182546403458 z: 0.016818312977913273 m: 0.0641383917649937 ã: 0.011480896014854124
k: 0.03349235025122791 n: 0.04834961139652609 m: 0.02927385410589837 n: 0.06371418982919011 o: 0.11016077331777664 n: 0.02478013637190513 p: 0.04357361511005727
c: 0.05170166920717364 t: 0.05626203604383472 y: 0.02676000677468526 ð: 0.0019547789758934116 x: 0.008335842476022004 ç: 0.0035488686902902453 ç: 0.013637190512931495
w: 0.009216394738327782 m: 0.00280867159713205 b: 0.00042969068543505397 m: 0.018392800015054857 b: 0.019129076572762003 z: 0.003010071226406218 w: 0.006359108501602715
r: 0.021851496389970956 e: 4.861463959301706e-05

```

```

#print count
import pandas as pd
sort_letters = sorted(alpcount, key=lambda x : alpcount[x],reverse=1)

list_freq = []
for i in range(0,len(sort_letters)):
    list_freq.append(alpcount[sort_letters[i]])

data = pd.DataFrame(index = sort_letters)
data['count'] = list_freq
print(data.head(10))
data.to_excel("liters.xlsx")

```

✓ 7.5s

```

count
o 140492
a 104486
e 101669
и 81798
н 81257
т 71753
с 65937
л 61662
в 56730
р 55571

```

```

#print freq
frequency= sorted(freq, key=lambda x : alpcount[x],reverse=1)

relative_frequencies = []
✓ for letter in frequency:
    ... relative_frequencies.append(freq[letter])

# Створюємо DataFrame для відносних частот
frequency_data = pd.DataFrame(index=frequency)
frequency_data['relative_frequency'] = relative_frequencies

# Виводимо перші кілька рядків DataFrame
print(frequency_data.head(10))

# Експортуємо дані в Excel
frequency_data.to_excel("relative_frequencies.xlsx")

```

✓ 0.1s

	relative_frequency
о	0.110161
а	0.081928
е	0.079719
и	0.064138
н	0.063714
т	0.056262
с	0.051702
л	0.048350
в	0.044482
р	0.043574

```

• ✓ from collections import Counter
import math

# Задаємо текст
input_text = string_text

# Створюємо біграми
bigrams = []
✓ for i in range(len(input_text) - 1):
    ... bigrams.append(input_text[i] + input_text[i + 1])

# Створюємо "двійні" біграми
double_bigrams = []
✓ for i in range(0, len(input_text) - 2, 2):
    ... double_bigrams.append(input_text[i] + input_text[i + 1])

# Підрахунок частот біграм
bigram_counts = dict(Counter(bigrams))
bigram_frequencies = {bigram: count / len(bigrams) for bigram, count in bigram_counts.items()}

# Підрахунок частот "двійних" біграм
double_bigram_counts = dict(Counter(double_bigrams))
double_bigram_frequencies = {bigram: count / len(double_bigrams) for bigram, count in double_bigram_counts.items()}

# Обчислення ентропії H1
entropy_contributions = []
✓ for frequency in bigram_frequencies.values():
    ... if frequency > 0: # Переконайтеся, що частота не дорівнює нулю, щоб уникнути помилок
        ... entropy_contributions.append(frequency * math.log(frequency, 2))

# Обчислюємо загальну ентропію H1
print(entropy_contributions)
H1 = -sum(entropy_contributions)
print("Entropy H1:", H1)

```

✓ 1.4s

[-0.029239031226645732, -0.013313251630062686, -0.04512993265779703, -0.04779830623829132, -0.08223341786352448, -0.046290747198874035, Entropy H1: 8.119527189870336


```

• # Створення DataFrame з алфавіту
frequency_df = pd.DataFrame(index=alphabet, columns=alphabet)

# Генерація всіх можливих біграм з алфавіту
bigrams_list = [letter1 + letter2 for letter1 in alphabet for letter2 in alphabet]

# Заповнення DataFrame біграмами
✓ for idx, letter in enumerate(alphabet):
    frequency_df[letter] = bigrams_list[idx * len(alphabet):(idx + 1) * len(alphabet)]

# Транспонування DataFrame для правильного формату
frequency_df = frequency_df.T

# Заповнення DataFrame частотами двійних біграм
✓ for bigram in double_bigram_frequencies.keys():
    row_idx, col_idx = np.where(frequency_df == bigram)
    frequency_df.iloc[row_idx, col_idx] = double_bigram_frequencies[bigram]

# Встановлення 0 для біграм, які не мають частоти
✓ for bigram in bigrams_list:
    row_idx, col_idx = np.where(frequency_df == bigram)
    frequency_df.iloc[row_idx, col_idx] = 0

# Вивід фінального DataFrame
print(frequency_df.head())

# Збереження DataFrame у файл Excel
frequency_df.to_excel("double_bigram_frequencies.xlsx")

✓ 0.7s

```

	а	б	в	г	д	е	є	ж	
а	0.00032	0.001537	0.005553	0.001611	0.002976	0.001749	0	0.001443	\
б	0.001226	0.00002	0.000125	0.000008	0.000028	0.002376	0	0.000008	
в	0.006535	0.000274	0.000464	0.000428	0.000756	0.005116	0	0.000053	
г	0.001096	0.000036	0.000116	0.000011	0.001104	0.000632	0	0.000005	
д	0.004786	0.000083	0.001059	0.000075	0.000066	0.005152	0	0.000024	
	з	и	...	ц	ч	ш	щ	ъ	
а	0.004863	0.001559	...	0.000136	0.001331	0.001676	0.000265	0	\
б	0.000009	0.000889	...	0.000006	0.000002	0.000008	0.000252	0.00011	
в	0.000665	0.003762	...	0.000058	0.000238	0.001247	0.000009	0.000028	
г	0.000045	0.000941	...	0.000003	0.000035	0.000009	0	0	
д	0.000052	0.002937	...	0.000202	0.000075	0.000127	0.000002	0.00021	
	ы	ь	Э	ю	я				
а	0	0	0.000345	0.000949	0.003096				

Результат

Таблиця частот по файлам:

	Текст без пробілів	Текст з пробілами
Букви	liters.xlsx	liters_space.xlsx
Біграма	relative_frequencies.xlsx	relative_frequencies_space.xlsx
Біграма з кроком 2	double_bigram_frequencies.xlsx	double_bigram_frequencies_space.xlsx

Ентропія

	Текст без пробілів	Текст з пробілами
H1	8.477388527827898	8.119527189870336
H2	4.238694263913949	4.059763594935168
H2(з кроком 2)	4.238452024542541	4.0596450054514275

Пункт 2. За допомогою програми *CoolPinkProgram* оцінити значення $H^{(10)}$, $H^{(20)}$, $H^{(30)}$.

- Для $H^{(10)}$:

Лабораторная работа №1

Произвольная часть текста:
но_вырази

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:

Символ по счету:

Номер эксперимента: 53

Поле ввода символов:

Продолжить Другой

Неравенство для энтропии:
1.97913859888269 < $H < 2.84320970567439$

Двоичная таблица угаданных символов:

```

01000000000000000000000000000000
10000000000000000000000000000000
00000000000100000000000000000000
10000000000000000000000000000000
00001000000000000000000000000000

```

Вероятности:

```

q[1] = 0.5
q[2] = 0.0961538
q[3] = 0.0769230
q[4] = 0.0192307
q[5] = 0.0192307
q[6] = 0.0576923
q[7] = 0.0192307
q[8] = 0
q[9] = 0.0192307
q[10] = 0.0192307
q[11] = 0.0192307
q[12] = 0.0192307
q[13] = 0
q[14] = 0.0192307
q[15] = 0
q[16] = 0
q[17] = 0.0192307
q[18] = 0.0192307
q[19] = 0
q[20] = 0
q[21] = 0
q[22] = 0
q[23] = 0.0384615
q[24] = 0.0192307
q[25] = 0.0192307
q[26] = 0
q[27] = 0
q[28] = 0
q[29] = 0
q[30] = 0
q[31] = 0
q[32] = 0

```

Строка состояния:

- Для $H^{(20)}$:

Лабораторная работа №1

Произвольная часть текста:
го_мнения_держались

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:

Символ по счету:

Номер эксперимента: 53

Поле ввода символов:

Продолжить Другой

Неравенство для энтропии:
1.72478927881896 < $H < 2.47774576387199$

Двоичная таблица угаданных символов:

```

00000000000000000000000000000000
01000000000000000000000000000000
00000000000000000100000000000000
00000010000000000000000000000000
10000000000000000000000000000000

```

Вероятности:

```

q[1] = 0.5961538
q[2] = 0.0769230
q[3] = 0.0384615
q[4] = 0.0192307
q[5] = 0.0192307
q[6] = 0
q[7] = 0.0192307
q[8] = 0.0384615
q[9] = 0
q[10] = 0
q[11] = 0
q[12] = 0.0192307
q[13] = 0.0192307
q[14] = 0
q[15] = 0.0192307
q[16] = 0.0192307
q[17] = 0
q[18] = 0.0192307
q[19] = 0
q[20] = 0.0192307
q[21] = 0
q[22] = 0
q[23] = 0
q[24] = 0.0192307
q[25] = 0.0384615
q[26] = 0
q[27] = 0
q[28] = 0
q[29] = 0
q[30] = 0.0192307
q[31] = 0
q[32] = 0

```

Строка состояния:

- Для $H^{(30)}$:

Лабораторная работа №1

Произвольная часть текста:
ми_причинами_мы_ставим_его_ис

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:

Символ по счету:

Номер эксперимента: 53

Неравенство для энтропии:
 $1.75197028600414 < H < 2.45965264033664$

Двоичная таблица угаданных символов:

```

10000000000000000000000000000000
01000000000000000000000000000000
00100000000000000000000000000000
01000000000000000000000000000000
10000000000000000000000000000000

```

Вероятности:

```

q[1] = 0.5192307
q[2] = 0.1730769
q[3] = 0.0961538
q[4] = 0
q[5] = 0
q[6] = 0
q[7] = 0.0192307
q[8] = 0
q[9] = 0.0192307
q[10] = 0.0192307
q[11] = 0
q[12] = 0
q[13] = 0
q[14] = 0
q[15] = 0.0192307
q[16] = 0.0192307
q[17] = 0
q[18] = 0
q[19] = 0.0192307
q[20] = 0.0192307
q[21] = 0.0192307
q[22] = 0.0192307
q[23] = 0
q[24] = 0.0192307
q[25] = 0
q[26] = 0
q[27] = 0
q[28] = 0.0192307
q[29] = 0
q[30] = 0
q[31] = 0
q[32] = 0

```

Поле ввода символов:

Продолжить Другой

Строка состояния:

	Ентропія	Надлишковість
$H^{(10)}$	$1.97914 < H < 2.84321$	$0.43136 < H < 0.60417$
$H^{(20)}$	$1.72479 < H < 2.47775$	$0.50445 < H < 0.65504$
$H^{(30)}$	$1.75197 < H < 2.45965$	$0.50807 < R < 0.64961$

Висновок

Ознайомилися з поняттями ентропії на символ джерела та його надлишковості, навчилися визначати частоти літер та біграм на довільному тексті.

З отриманих значень можемо визначити, що ентропія H_1 з пробілами більша за H_1 без пробілів. Ентропія та надлишковість перехресних та неперехресних біграм без пробілів (або з пробілами) (H_2 та H_2 (з кроком 2)) майже не відрізняються.