

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

КРИПТОГРАФІЯ

Комп'ютерний практикум

Робота № 1

Виконали

ФБ-33 Грабченко Олександр

ФБ-33 Стогнійчук Інна

Київ – 2025

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.

```
import math
import re
from collections import Counter

def preprocess_text(raw_text, alphabet):
    # Обробка тексту
    text = raw_text.lower().replace('ё', 'е').replace('ъ', 'ь')

    filtered_chars = [char if char in alphabet else ' ' for char in text]
    text = ''.join(filtered_chars)

    processed_text = re.sub(r'\s+', ' ', text).strip()

    return processed_text


def calculate_entropy(frequencies, total_count):
    # ентропія
    entropy = 0.0
    for count in frequencies.values():
        if count > 0:
```

```

        probability = count / total_count

        entropy -= probability * math.log2(probability)

    return entropy


def analyze_text(text, alphabet, bigram_step):
    # частоти, біграми, ентропії, H1 та H2

    letter_counts = Counter(text)

    total_letters = len(text)

    letter_frequencies = {char: count / total_letters for char, count in
letter_counts.items()}

    h1 = calculate_entropy(letter_counts, total_letters)

    bigrams = [text[i:i+2] for i in range(0, len(text) - 1, bigram_step) if
len(text[i:i+2]) == 2]

    bigram_counts = Counter(bigrams)

    total_bigrams = len(bigrams)

    if total_bigrams == 0:

        return {

            'letter_frequencies': letter_frequencies,
            'bigram_frequencies': {},
            'H1': h1,
            'H2': 0.0
        }

    # h2

    bigram_frequencies = {bigram: count / total_bigrams for bigram, count in
bigram_counts.items()}

    h2 = calculate_entropy(bigram_counts, total_bigrams) / 2

    return {

        'letter_frequencies': letter_frequencies,

```

```

        'bigram_frequencies': bigram_frequencies,
        'H1': h1,
        'H2': h2
    }

def print_frequency_table(frequencies):
    # частоты
    sorted_frequencies = sorted(frequencies.items(), key=lambda item:
item[1], reverse=True)
    for char, freq in sorted_frequencies:
        print(f'{char}: {freq:.6f}')

def print_bigram_matrix(frequencies, alphabet):
    alphabet_chars = sorted(list(alphabet))

    header = " " + "".join(f'{ch:>8}' for ch in alphabet_chars)
    print(header)
    print("---+ " + " - " * (len(header) - 3))

    for first_char in alphabet_chars:
        row_str = f' {first_char} | '
        for second_char in alphabet_chars:
            bigram = first_char + second_char
            freq = frequencies.get(bigram, 0.0)
            row_str += f'{freq:8.6f} '
        print(row_str)

def main():
    filename = 'cryptotext.txt'
    try:

```

```

with open(filename, 'r', encoding='utf-8') as f:
    raw_text = f.read()

except FileNotFoundError:
    print(f"Файл не існує")
    return

print("Аналіз тексту з пробілами")

alphabet_with_spaces = 'абвгдежзийклмнопрстуфхцчшшъюя '


processed_text_ws = preprocess_text(raw_text, alphabet_with_spaces)

print("\nБіграми, що перетинаються (крок 1) ")
results_ws_overlap = analyze_text(processed_text_ws,
alphabet_with_spaces, bigram_step=1)
print(f"Значення H1: {results_ws_overlap['H1']:.4f}")
print(f"Значення H2: {results_ws_overlap['H2']:.4f}\n")

print("\nБіграми, що НЕ перетинаються (крок 2) ")
results_ws_no_overlap = analyze_text(processed_text_ws,
alphabet_with_spaces, bigram_step=2)
print(f"Значення H1: {results_ws_no_overlap['H1']:.4f}")
print(f"Значення H2: {results_ws_no_overlap['H2']:.4f}\n")

print("\nЧастоти літер (з пробілами) ")
print_frequency_table(results_ws_overlap['letter_frequencies'])

print("\nМатриця частот біграмм (з пробілами, крок 1) ")
print_bigram_matrix(results_ws_overlap['bigram_frequencies'],
alphabet_with_spaces)

print("\nМатриця частот біграмм (з пробілами, крок 2) ")

```

```
print_bigram_matrix(results_ws_no_overlap['bigram_frequencies'],
alphabet_with_spaces)

print("\n\n Аналіз тексту БЕЗ пробілів ")

alphabet_no_spaces = 'абвгдежзийклмнопрстуфхцчищъэюя'

processed_text_ns = preprocess_text(raw_text,
alphabet_no_spaces).replace(' ', '')

print("\n Біграми, що перетинаються (крок 1) ")

results_ns_overlap = analyze_text(processed_text_ns, alphabet_no_spaces,
bigram_step=1)

print(f"Значення H1: {results_ns_overlap['H1']:.4f}")

print(f"Значення H2: {results_ns_overlap['H2']:.4f}\n")

print("\n Біграми, що НЕ перетинаються (крок 2) ")

results_ns_no_overlap = analyze_text(processed_text_ns,
alphabet_no_spaces, bigram_step=2)

print(f"Значення H1: {results_ns_no_overlap['H1']:.4f}")

print(f"Значення H2: {results_ns_no_overlap['H2']:.4f}\n")

print("\n Частоти літер (без пробілів) ")

print_frequency_table(results_ns_overlap['letter_frequencies'])

print("\n Матриця частот біграмм (без пробілів, крок 1) ")

print_bigram_matrix(results_ns_overlap['bigram_frequencies'],
alphabet_no_spaces)

print("\n Матриця частот біграмм (без пробілів, крок 2) ")

print_bigram_matrix(results_ns_no_overlap['bigram_frequencies'],
alphabet_no_spaces)
```

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

Output

Аналіз тексту з пробілами

Біграми, що перетинаються (крок 1)

Значення H1: 4.4232

Значення H2: 3.9804

Біграми, що НЕ перетинаються (крок 2)

Значення H1: 4.4232

Значення H2: 3.9794

Частоти літер (з пробілами)

' ': 0.143456

'о': 0.088326

'е': 0.075760

'а': 0.070534

'т': 0.061055

'и': 0.056888

'н': 0.053497

'р': 0.044892

'с': 0.044159

'в': 0.036525

'л': 0.031265

'м': 0.030726

'к': 0.030675

'π' : 0.030176

'д': 0.025094

'y' : 0.022001

'ы' : 0.019092

' я ': 0.018712

'3': 0.015925

'Ь': 0.014419

'ψ': 0.011982

'6': 0.011439

'Й': 0.010922

' Γ ' : 0.010793

'ж' : 0.006723

'x': 0.006644

'ю': 0.005814

'Φ' : 0.005097

'Ц' : 0.004918

'Щ': 0.004438

'Э': 0.004064

'III': 0.003989

Аналіз тексту БЕЗ пробілів

Біграми, що перетинаються (крок 1)

Значення H1: 4.4715

Значення Hz: 4.1125

вітрами, що не перетинаються (крок 2).

DATA CENTER UNIT: 1.0.17.15

CHAPTER III. 1911

Частоти літер (без пробілів)

'o': 0.103119

'e': 0.088449

'a': 0.082347

'T' : 0.071281

'И' : 0.066416

'H' : 0.062456

p = 0.052410

• C : 0.051555

B . 0.042042

51 : 0.050501

11 : 8:05:07 2

'к' : 0.035813
'п' : 0.035230
'д' : 0.029297
'у' : 0.025685
'ы' : 0.022289
'я' : 0.021846
'з' : 0.018593
'ь' : 0.016834
'ч' : 0.013988
'б' : 0.013355
'й' : 0.012752
'г' : 0.012600
'ж' : 0.007849
'х' : 0.007757
'ю' : 0.006788
'ф' : 0.005951
'ц' : 0.005742
'щ' : 0.005181
'э' : 0.004745
'ш' : 0.004657

CoolPinkProgram:

Лабораторная работа №1

Произвольная часть текста:
м_порядочное_поведение_правда_состоит_в_том_что_мы_верим_в_порядочность_нас

Использованные буквы:

Порядок n-грамм:

- 5 символов
- 10 символов**
- 15 символов
- 20 символов
- 25 символов
- 30 символов
- 35 символов
- 40 символов
- 45 символов
- 50 символов

Введенный символ: **н**

Символ по счету: **1**

Номер эксперимента: **50**

Поле ввода символов:
н

Продолжить Другой

Строка состояния:
Вы угадали. Для продолжения опыта нажмите "Продолжить", или "Другой" для выбора другого порядка

Вероятности:

```

q[1] = 0.38
q[2] = 0.12
q[3] = 0.06
q[4] = 0.06
q[5] = 0
q[6] = 0.04
q[7] = 0.04
q[8] = 0
q[9] = 0.02
q[10] = 0.02
q[11] = 0
q[12] = 0
q[13] = 0
q[14] = 0.04
q[15] = 0.02
q[16] = 0.02
q[17] = 0
q[18] = 0
q[19] = 0.02
q[20] = 0
q[21] = 0
q[22] = 0.02
q[23] = 0.04
q[24] = 0
q[25] = 0.02
q[26] = 0.02
q[27] = 0
q[28] = 0
q[29] = 0.04
q[30] = 0.02
q[31] = 0
q[32] = 0

```

Лабораторная работа №1

Произвольная часть текста:
зличных_законов_и_среди_них_имеется_только_один_который_он_свободен_нарушит

Использованные буквы:
и, а, е, м, у, о, ы, ю, в.

Порядок n-грамм:	Введенный символ: р	Неравенство для энтропии: 2.05497557483654 < H < 2.73984900025397
5 символов	Символ по счету: 10	Двоичная таблица угаданных символов:
10 символов	Номер эксперимента: 50	
15 символов		00000100000000000000000000000000 00000000000000000000000000000000 10000000000000000000000000000000 10000000000000000000000000000000 10000000000000000000000000000000 ~~~~~ ~~~~~~ ~~~~~~
20 символов		
25 символов		
30 символов		
35 символов		
40 символов		
45 символов		
50 символов		

Поле ввода символов:
р

Продолжить Другой

Строка состояния:
Вы угадали. Для продолжения опыта нажмите "Продолжить", или "Другой" для выбора другого порядка

Вероятности:

```

q[1] = 0.48
q[2] = 0.14
q[3] = 0.02
q[4] = 0.04
q[5] = 0.02
q[6] = 0.06
q[7] = 0
q[8] = 0.02
q[9] = 0
q[10] = 0.06
q[11] = 0
q[12] = 0.04
q[13] = 0
q[14] = 0.02
q[15] = 0
q[16] = 0
q[17] = 0
q[18] = 0
q[19] = 0
q[20] = 0
q[21] = 0
q[22] = 0
q[23] = 0.02
q[24] = 0
q[25] = 0.04
q[26] = 0
q[27] = 0
q[28] = 0.02
q[29] = 0
q[30] = 0
q[31] = 0.02
q[32] = 0

```

Лабораторная работа №1

Произвольная часть текста:
но_когда_мыслители_древности_называли_законы_добра_и_зла_законами_природы_о

Использованные буквы:
д, л, т, и, с, ч, ф, а, р, о, ж, я, ь, ю, х, з, щ, ш, к, у, й, б, _, е,

Порядок n-грамм:

- 5 символов
- 10 символов
- 15 символов
- 20 символов
- 25 символов**
- 30 символов
- 35 символов
- 40 символов
- 45 символов
- 50 символов

Введенный символ: **н**

Символ по счету: **25**

Номер эксперимента: **50**

Поле ввода символов:
н

Неравенство для энтропии:
1.58714534190938 < H < 2.30563043834684

Двоичная таблица угаданных символов:

100000000000000000000000000000000000000000000000000
0000000000000000000000000000000010000000000000000
01000000000000000000000000000000000000000000000000
10000000000000000000000000000000000000000000000000
10000000000000000000000000000000000000000000000000

Продолжить Другой

Строка состояния:
Вы угадали. Для продолжения опыта нажмите "Продолжить", или "Другой" для выбора другого порядка

Вероятности:

```

q[1] = 0.54
q[2] = 0.18
q[3] = 0.08
q[4] = 0.04
q[5] = 0.02
q[6] = 0
q[7] = 0.02
q[8] = 0
q[9] = 0
q[10] = 0
q[11] = 0
q[12] = 0
q[13] = 0
q[14] = 0
q[15] = 0
q[16] = 0
q[17] = 0
q[18] = 0
q[19] = 0.02
q[20] = 0.02
q[21] = 0
q[22] = 0
q[23] = 0
q[24] = 0
q[25] = 0.02
q[26] = 0.02
q[27] = 0.02
q[28] = 0
q[29] = 0
q[30] = 0
q[31] = 0.02
q[32] = 0

```

Надлишковість мови можна підрахувати за формулою $R = \frac{H_{\infty}}{H_0}$, Де:

$H_0 = \log_2 32$; $H_\infty = \text{отримана ентропія}$

Тоді, надлишковість з пробілами за H1 та H2 за текстом:

$$1 - \frac{4.4232}{5} = 0,11536$$

$$1 - \frac{3.9804}{5} = 0,20392$$

Без пробілів:

$$1 - \frac{4.4715}{5} = 0,1057$$

$$1 - \frac{4.1125}{5} = 0,1175$$

Значення H2 взято тільки з біграм, що перетинаються, так як суттєвої різниці в результатах нема

Надлишковість за CoolPinkProgram (взяті середні значення):

$H^{(10)}$:

$$1 - \frac{2,973}{5} = 0,4054$$

$H^{(20)}$:

$$1 - \frac{2,397}{5} = 0,5206$$

$H^{(30)}$:

$$1 - \frac{1,946}{5} = 0,6108$$

Висновки: Під час виконання лабораторної роботи ми практично навчилися вираховувати ентропію в джерелах відкритого тексту та побачили надлишковість в дії (за допомогою CoolPinkProgram). Отримали розуміння різних моделей джерела відкритого тексту.