

# Лабораторна робота 1 з Симетричної Криптографії

Команда: Бондар, Дигас Група: ФІ-03

## Підготовка даних

1. В якості середньостатистичного тексту російською мовою ми взяли дописи з телеграм-каналу терориста і військового злочинця ігоря гіркіна
2. Повний вхідний текст можна знайти у відповідному файлі
3. Обробка тексту і підготовка до аналізу відбувається у розділі "Text reading and preprocessing"

## Програмна частина

### Text reading and preprocessing

```
In [ ]: filename = "girkin_crying.txt"

In [ ]: def get_text(_filename):
    f = open(_filename, "r", encoding='utf-8')
    text = f.read()
    f.close()
    return text

def transform_symbol(_c):
    if 'a' <= _c and _c <= 'я':
        return _c
    elif _c <= 'Я' and _c >= 'А':
        return _c.lower()
    elif _c == 'Ё' or _c == 'ё':
        return 'e'
    else:
        return ' '

def preprocess_text(_text):
    _text = get_text(filename)
    text_formatted = ""
    # Change symbols according to requirements
    for c in _text:
        text_formatted += transform_symbol(c)

    # Remove consecutive spaces
    text_formatted = ' '.join(text_formatted.split())
    return text_formatted

In [ ]: text = preprocess_text(get_text(filename))
```

### Text processing (singular char count and bigram count)

```
In [ ]: def count_chars(_text):
    c_count = {}
    for c in _text:
        if c not in c_count:
            c_count[c] = 1
        else:
            c_count[c] = c_count[c] + 1

    return dict(sorted(c_count.items()))

# Bigrams with intersection (ex: [1, 2], [2, 3], [3, 4])
def count_bigrams_w_i(_text):
    b_count = {}
    prev_char = _text[0]
    for c in _text[1:]:
        bg = prev_char + c
        prev_char = c
        if bg not in b_count:
            b_count[bg] = 1
        else:
            b_count[bg] = b_count[bg] + 1

    return dict(sorted(b_count.items()))

# Bigrams without intersection (ex: [1, 2], [3, 4])
def count_bigrams_wo_i(_text):
    b_count = {}
    i = 1
    while i < len(_text):
        bg = _text[i - 1] + _text[i]
        if bg not in b_count:
            b_count[bg] = 1
        else:
            b_count[bg] = b_count[bg] + 1
        i = i + 2
```

```
return dict(sorted(b_count.items()))
```

```
In [ ]: chars_freq_wspsaces = count_chars(text)
chars_freq_wspsaces = chars_freq_wspsaces.copy()
del chars_freq_wspsaces[' ']

bigrams_freq_w_intersect = count_bigrams_w_i(text)
bigrams_freq_wo_intersect = count_bigrams_wo_i(text)

text_wo_spaces = text.replace(" ", "")
bigrams_freq_w_intersect_no_space = count_bigrams_w_i(text_wo_spaces)
bigrams_freq_wo_intersect_no_space = count_bigrams_wo_i(text_wo_spaces)
```

## Show symbol frequencies

```
In [ ]: total_symbols = sum(chars_freq_wspsaces.values())
for k, v in chars_freq_wspsaces.items():
    print(f"{k} : {v / total_symbols}")

: 0.1489668453383176
а : 0.06505639055803568
б : 0.014029414743071539
в : 0.03946566706657536
г : 0.013174674280378901
д : 0.02441674261499091
е : 0.07453405488532232
ж : 0.008101154586163159
з : 0.012714693469130654
и : 0.0639339000708851
й : 0.010950289461805287
к : 0.028192704498372044
л : 0.031192877849349418
м : 0.02646606010988049
н : 0.0618399575719192
о : 0.09738892235764195
п : 0.027015290929281382
р : 0.04276791736822322
с : 0.04559645600813782
т : 0.05353970681372322
у : 0.02181819430070044
ф : 0.0027667502527319943
х : 0.009192750839722432
ц : 0.0041569907643405025
ч : 0.011719212608966537
ш : 0.00560962046434033
щ : 0.0031683752894188966
ъ : 0.00028148079494295724
ы : 0.015364732172739958
ь : 0.013130049276302578
э : 0.002121404039935946
ю : 0.005845875533998246
я : 0.015529501418560226
```

## Show bigram frequencies

```
In [ ]: total_bigrams = sum(bigrams_freq_w_intersect.values())
alph = list(chars_freq_wspsaces.keys())
print(" ", end='|')
for l_c in alph:
    print(f" '{l_c}' ", end='|')
for l_c in alph:
    print(f"\n '{l_c}' ", end='|')
    for r_c in alph:
        k = l_c + r_c
        if k in bigrams_freq_w_intersect:
            print(f"{bigrams_freq_w_intersect[k] / total_bigrams:8.6f}", end='|')
        else:
            print(f"{0:8f}", end='|')
```

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z								
0	0.000000	0.003072	0.005688	0.015536	0.031010	0.006223	0.001967	0.000882	0.003512	0.009886	0.000093	0.007058	0.002362	0.005506	0.015907	0.009371	0.017926	0.005609	0.013261	0.007027	0.004806	0.001250	0.001286	0.000549	0.003570	0.000656	0.000000	0.000000	0.000000	0.001895	0.001124	0.000927		
1	0.001320	0.000007	0.000838	0.002808	0.009661	0.001778	0.001637	0.001469	0.002379	0.000511	0.000803	0.004414	0.004614	0.002818	0.005588	0.000093	0.001020	0.003285	0.004044	0.005453	0.000082	0.000082	0.001153	0.001380	0.001184	0.001140	0.000209	0.000000	0.000000	0.000007	0.001308	0.001370		
2	0.000261	0.000858	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		
3	0.007682	0.005019	0.000014	0.000086	0.000014	0.001611	0.003759	0.000003	0.000371	0.002454	0.000000	0.000638	0.001370	0.000185	0.002094	0.000000	0.002740	0.002888	0.001026	0.002674	0.000336	0.000522	0.000000	0.000003	0.000048	0.001113	0.000518	0.000021	0.000000	0.002379	0.001110	0.000000	0.000000	0.000336
4	0.000349	0.001164	0.000000	0.000048	0.000007	0.000614	0.000459	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		
5	0.001150	0.004929	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		
6	0.017080	0.000199	0.005840	0.018119	0.002118	0.003223	0.001390	0.000000	0.001668	0.000277	0.002509	0.001298	0.004790	0.003848	0.010090	0.003663	0.002104	0.000652	0.004789	0.005104	0.000127	0.000087	0.000555	0.000453	0.000735	0.000604	0.000491	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
7	0.001020	0.000958	0.000000	0.000000	0.000010	0.000638	0.003948	0.000007	0.000000	0.000958	0.000000	0.000093	0.000000	0.000000	0.000007	0.001044	0.000017	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
8	0.001346	0.003858	0.000175	0.000786	0.000189	0.001008	0.000388	0.000955	0.000010	0.000851	0.000000	0.000148	0.000134	0.000577	0.000937	0.000851	0.000103	0.002511	0.000301	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
9	0.012831	0.004777	0.000433	0																														

Calculate  $H_1$  and  $H_2$

```
import sys
def entropy_sum(data):
    amount = sum(data.values())
    t = [data[k] / amount for k in data.keys()]
    return -sum(a * math.log2(a) for a in t)

H1_w_spaces = entropy_sum(chars_freq_wspaces)
H1_wo_spaces = entropy_sum(chars_freq_wospaces)

print("----- H1 values -----")
print(f"With spaces: {H1_w_spaces}")
print(f"Without spaces: {H1_wo_spaces}")

H2_ws_wi = entropy_sum(bigrams_freq_w_intersect) / 2
H2_ws_woi = entropy_sum(bigrams_freq_wo_intersect) / 2
H2_ns_wi = entropy_sum(bigrams_freq_w_intersect_no_space) / 2
H2_ns_woi = entropy_sum(bigrams_freq_wo_intersect_no_space) / 2
print("----- H2 values -----")
print("----- With spaces -----")
print(f"With intersection: {H2_ws_wi}")
print(f"Without intersection: {H2_ws_woi}")
print("----- Without spaces -----")
print(f"With intersection: {H2_ns_wi}")
print(f"Without intersection: {H2_ns_woi}")

----- H1 values -----
With spaces: 4.385129362944809
Without spaces: 4.439169682034552
----- H2 values -----
----- With spaces -----
With intersection: 3.9881215496418245
Without intersection: 3.988124884466209
----- Without spaces -----
With intersection: 4.124148874705879
Without intersection: 4.123577559685296
```

### Обчислення $H^{(10)}$ , $H^{(20)}$ , $H^{(30)}$ за допомогою Cool Pink Program

Проклятая часть текста:

ый, мюин

Использованные буквы:

Порядок и-график:

5

10

15

20

25

30

35

40

45

50

Введенный символ:

Символ по счету:

Номер эксперимента: 51

Поле ввода символов:

φφφφ

\*\*\*\*\*

Неравенство для энтропии:

1.25952956876821 < H < 2.02205697724832

Двоичная таблица угаданных символов:

Вероятности:

q[1] = 0.62

q[2] = 0.12

q[3] = 0.08

q[4] = 0.02

q[5] = 0.04

q[6] = 0.02

q[7] = 0

q[8] = 0.02

q[9] = 0

q[10] = 0

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

q[24] = 0

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:

Проклятая часть текста:

ание, некоторых, ладе

Использованные буквы:

Порядок и-график:

5

10

15

20

25

30

35

40

45

50

Введенный символ:

Символ по счету:

Номер эксперимента: 50

Поле ввода символов:

φφφφ

\*\*\*\*\*

Неравенство для энтропии:

1.32340430244222 < H < 2.06785101605236

Двоичная таблица угаданных символов:

00100000000000000000000000000000

00000000000000000000000000000000

10000000000000000000000000000000

00000000000000000000000000000000

10000000000000000000000000000000

Вероятности:

q[1] = 0.6440677

q[2] = 0.1186440

q[3] = 0.0338993

q[4] = 0

q[5] = 0

q[6] = 0.0338993

q[7] = 0.0169491

q[8] = 0

q[9] = 0.0169491

q[10] = 0.016949

q[11] = 0.033898

q[12] = 0

q[13] = 0.016949

q[14] = 0.016949

q[15] = 0

q[16] = 0

q[17] = 0.016949

q[18] = 0

q[19] = 0

q[20] = 0

q[21] = 0.016949

q[22] = 0

q[23] = 0

q[24] = 0

q[25] = 0

q[26] = 0

q[27] = 0.016949

q[28] = 0

q[29] = 0

q[30] = 0

q[31] = 0

q[32] = 0

Строка состояния:

Проклятая часть текста:

асколько, разный, закон, и

Использованные буквы:

Порядок и-график:

5

10

15

20

25

30

35

40

45

50

Введенный символ:

Символ по счету:

Номер эксперимента: 51

Поле ввода символов:

φφφφ

\*\*\*\*\*

Неравенство для энтропии:

1.39602146385785 < H < 2.05556472727899

Двоичная таблица угаданных символов:

10000000000000000000000000000000

10000000000000000000000000000000

10000000000000000000000000000000

10000000000000000000000000000000

10000000000000000000000000000000

Вероятности:

q[1] = 0.56

q[2] = 0.22

q[3] = 0.02

q[4] = 0.04

q[5] = 0.02

q[6] = 0.06

q[7] = 0

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] = 0.02

q[23] = 0

q[24] = 0.02

q[25] = 0

q[26] = 0

q[27] = 0

q[28] = 0

q[29] = 0

q[30] = 0

q[31] = 0

q[32] = 0

Строка состояния:

Отримані результати:

$$1.2595 \leq H^{(10)} \leq 2.0221$$

(1)

$$1.3234 \leq H^{(20)} \leq 2.0679$$

(2)

$$1.3960 \leq H^{(30)} \leq 2.0556$$

(3)

Обчислимо надлишковість:

Так як  $H_0 = \log_2(32) = 5$ , то за формулою надлишковості  $R = 1 - \frac{H_{\infty}}{H_0}$ .

Розглядаємо  $H^{(i)}$  як наближення  $H_{\infty}$ .

$$H^{(10)} : \quad 0.404 \leq R \leq 0.748$$

$$H^{(20)} : \quad 0.464 \leq R \leq 0.735$$

$$H^{(30)} : \quad 0.588 \leq R \leq 0.721$$

(4)