

КРИПТОГРАФІЯ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №1

Експериментальна оцінка ентропії на символ джерела відкритого тексту

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.

Виконали: Бондаренко Олексій, Кригін Дмитро. ФБ-03

Варіант: 2

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.

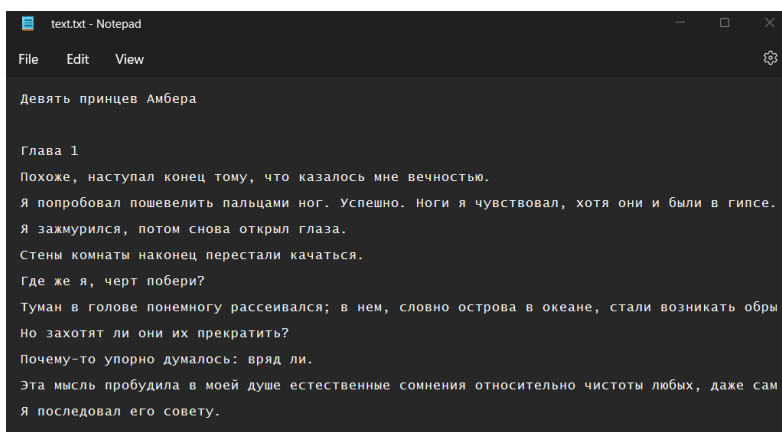
Хід роботи:

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.

Done

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.

Обраний текст – Роджер Желязни – Хроніки Амбера (Том 1) (1.5 Мб) – text.txt



Програма-фільтр, для відкидання зайвих символів (filter.py)

```
# By Bondarenko and Kryhin

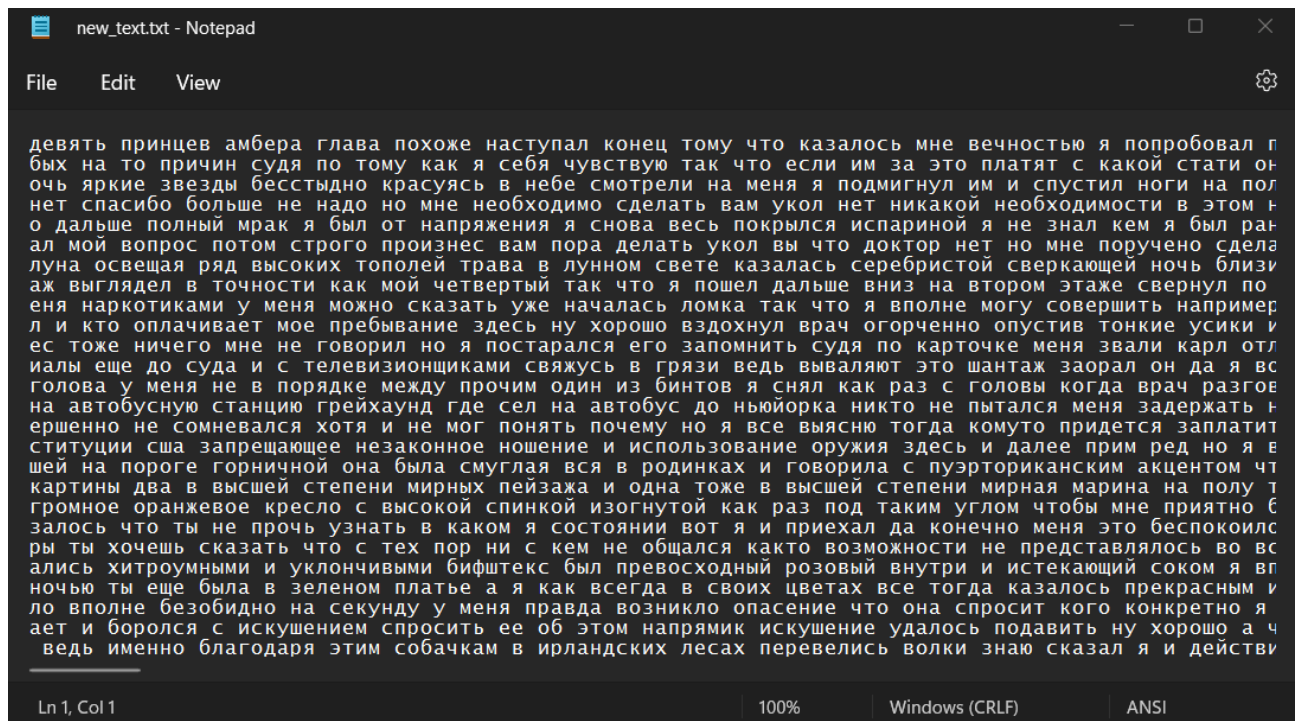
FROM_FILE = 'text.txt'
TO_FILE = 'new_text.txt'
alphabet = 'абвгдежзийклмнопрстуфхцщшщъьэя '

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()

result = ''
for i in text:
    if i in alphabet:
        result += i
    if i == '\n':
        result += ' '
    if i == '\t':
        result += ' '

with open(TO_FILE, 'w') as nf:
    nf.write(" ".join(result.split()))
```

OUTPUT(new_text.txt):



девять принцев амбера глава похоже наступал конец тому что казалось мне вечностью я попробовал г
бых на то причин судя по тому как я себя чувствую так что если им за это платят с какой стати он
очь яркие звезды бесстыдно красуясь в небе смотрели на меня я подмигнул им и спустил ноги на пол
нет спасибо больше не надо но мне необходимо сделать вам укол нет никакой необходимости в этом н
о дальше полный мрак я был от напряжения я снова весь покрылся испариной я не знал кем я был ран
ал мой вопрос потом строго произнес вам пора делать укол вы что доктор нет но мне поручено сдела
луна освещая ряд высоких тополей трава в лунном свете казалась серебристой сверкающей ночью близк
аж выглядел в точности как мой четвертый так что я пошел дальше вниз на втором этаже свернул по
еня наркотиками у меня можно сказать уже началась ломка так что я вполне могу совершить например
л и кто оплачивает мое пребывание здесь ну хорошо вздохнул врач огорченно опустил тонкие усики и
ес тоже ничего мне не говорил но я постарался его запомнить судя по карточке меня звали карл отл
иалы еще до суда и с телевизионщиками свяжусь в грязи ведь вывалиют это шантаж заорал он да я вс
голова у меня не в порядке между прочим один из бинтов я снял как раз с головы когда врач разгов
на автобусную станцию грейхаунд где сел на автобус до ньюйорка никто не пытался меня задержать н
ершенно не сомневался хотя и не мог понять почему но я все выясню тогда комуто придется заплатить
ституции сша запрещающее незаконное ношение и использование оружия здесь и далее прим ред но я е
шей на пороге горничной она была смуглая вся в родинках и говорила с пуэрториканским акцентом чт
картины два в высшей степени мирных пейзажа и одна тоже в высшей степени мирная марина на полу т
громное оранжевое кресло с высокой спинкой изогнутой как раз под таким углом чтобы мне приятно б
залось что ты не прочь узнать в каком я состоянии вот я и приехал да конечно меня это беспокоилс
ры ты хочешь сказать что с тех пор ни с кем не общался както возможности не представлялось во вс
ались хитроумными и уклончивыми бифштекс был превосходный розовый внутри и истекающий соком я вп
ночью ты еще была в зеленом платье а я как всегда в своих цветах все тогда казалось прекрасным и
ло вполне безобидно на секунду у меня правда возникло опасение что она спросит кого конкретно я
ает и боролся с искушением спросить ее об этом напрямик искушение удалось подавить ну хорошо а ч
ведь именно благодаря этим собачкам в ирландских лесах перевелись волки знаю сказал я и действи

Програма-фільтр, для відкидання зайвих символів та пробілів (text_without_spaces.py)

```
# By Bondarenko and Kryhin

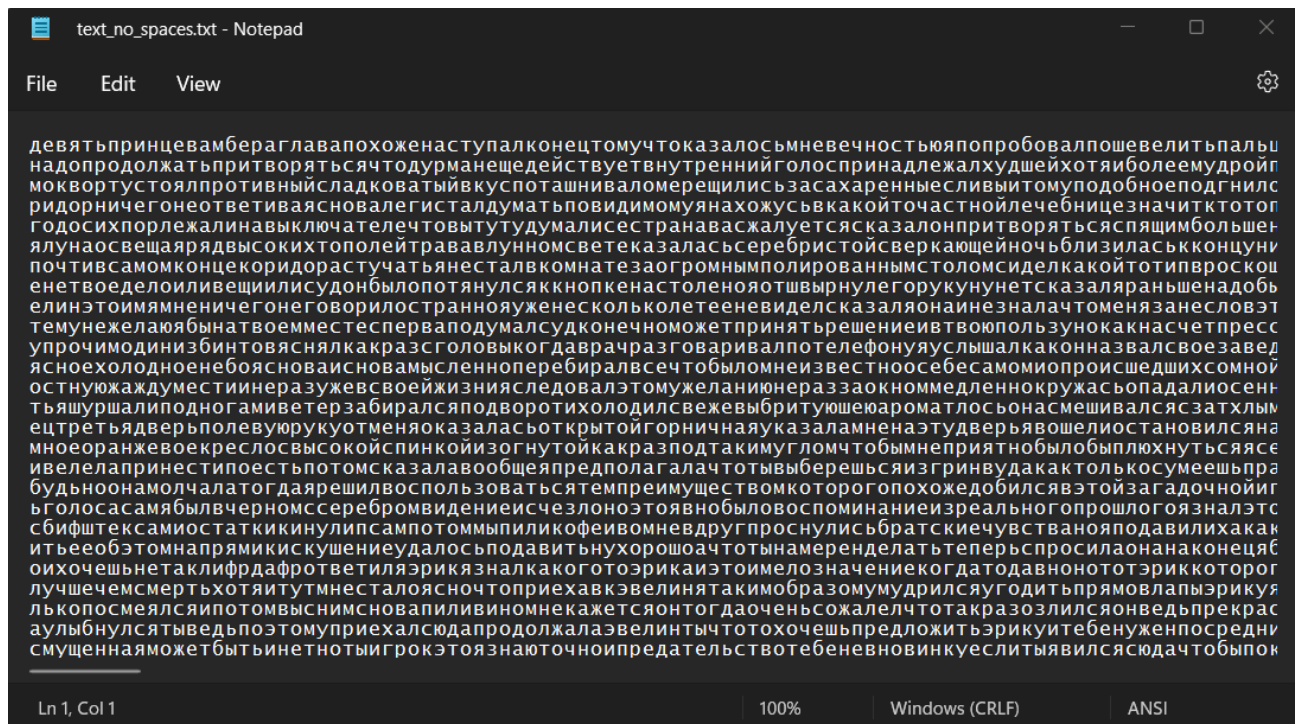
FROM_FILE = 'text.txt'
TO_FILE = 'text_no_spaces.txt'

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()

alphabet = 'абвгдежзийклмнопрстуфхцщшщъьэя'
result = ''
for i in text:
    if i in alphabet:
        result += i
    if i == '\t':
        result += ' '

with open(TO_FILE, 'w') as nf:
    nf.write(result)
```

OUTPUT(text_no_spaces.txt):



Підрахуємо частоту букв у тексті з пробілами та H1 для букв (h1_spaces.py):

```
# By Bondarenko and Kryhin

from pprint import pprint
from math import log

FROM_FILE = 'new_text.txt'
alphabet = 'абвгдежзийклмнопрстуфхцшщъыэя '

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()
length = len(text)

letters = dict()

for i in alphabet:
    letters[i] = text.count(i)

pprint(sorted(letters.items(), key=lambda item: item[1], reverse=True), sort_dicts=False)
# sorted output

h1 = 0
for i in alphabet:
    if letters[i] == 0:
        continue
    h1 -= (letters[i] / length) * log(letters[i] / length, 2)

print(f"H1: {h1}")
```

OUTPUT (h1_spaces.txt):

```
[(' ', 249090),
 ('o', 142771),
 ('e', 106547),
 ('a', 95744),
 ('н', 83826),
 ('и', 76681),
 ('т', 76038),
 ('с', 65814),
 ('л', 63783),
```

('р', 54907),
('в', 53318),
('м', 42498),
('к', 39760),
('д', 38053),
('п', 36142),
('у', 34269),
('я', 33405),
('ь', 26042),
('б', 23523),
('ы', 23271),
('з', 21778),
('г', 20487),
('ч', 18572),
('ж', 12885),
('й', 12559),
('ш', 10157),
('х', 9349),
('ю', 6444),
('о', 4673),
('щ', 4335),
('ц', 3278),
('ф', 1273)]

H1: 4.360843323137924

Підрахуємо частоту букв у тексті без пробілів та H1 для букв (h1_no_spaces.py):

```
# By Bondarenko and Kryhin

from pprint import pprint
from math import log

FROM_FILE = 'text_no_spaces.txt'
alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()
length = len(text)

letters = dict()

for i in alphabet:
    letters[i] = text.count(i)

pprint(sorted(letters.items(), key=lambda item: item[1], reverse=True), sort_dicts=False)
# sorted output

h1 = 0
for i in alphabet:
    if letters[i] == 0:
        continue
    h1 -= (letters[i] / length) * log(letters[i] / length, 2)
```

```
print(f"H1: {h1}")
```

OUTPUT(h1_no_spaces.txt):

[('o', 142771),
('e', 106547),
('a', 95744),
('н', 83826),
('и', 76681),
('т', 76038),
('с', 65814),
('л', 63783),
('р', 54907),
('в', 53318),
('м', 42498),
('к', 39760),
('д', 38053),
('п', 36142),
('у', 34269),
('я', 33405),
('ь', 26042),
('б', 23523),
('ы', 23271),
('з', 21778),
('г', 20487),
('ч', 18572),
('ж', 12885),
('й', 12559),
('ш', 10157),
('х', 9349),
('ю', 6444),
('э', 4673),
('ш', 4335),
('ц', 3278),
('ф', 1273)]

H1: 4.453920329839864

Підрахуємо частоту біграм (з перетином) у тексті з пробілами та H2 для біграм (h2_spaces_overlap.py):

```
# By Bondarenko and Kryhin

from pprint import pprint
from math import log

FROM_FILE = 'new_text.txt'
alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя '

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()

length = len(text) - 1 # determined experimentally (number of bigrams)
bigrams = dict() # numbers of each bigram

for i in alphabet:
    for j in alphabet:
        bigrams[i + j] = 0

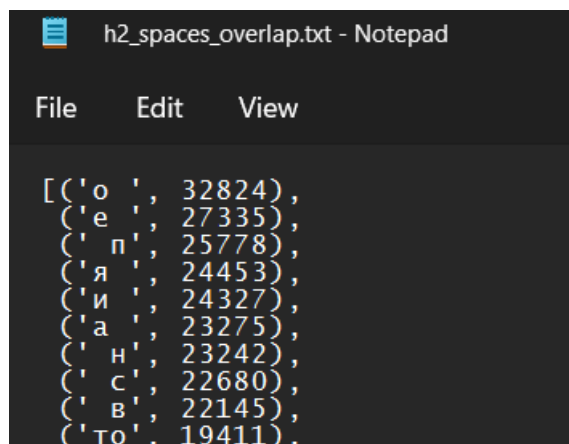
for i in bigrams.keys():
    bigrams[i] = text.count(i)

pprint(sorted(bigrams.items(), key=lambda item: item[1], reverse=True), sort_dicts=False)
# sorted output

h2 = 0
for i in bigrams:
    if bigrams[i] == 0:
        continue
    h2 -= (bigrams[i] / length) * log(bigrams[i] / length, 2)

print(f"H2: {h2 / 2}")
```

OUTPUT(h2_spaces_overlap.txt):

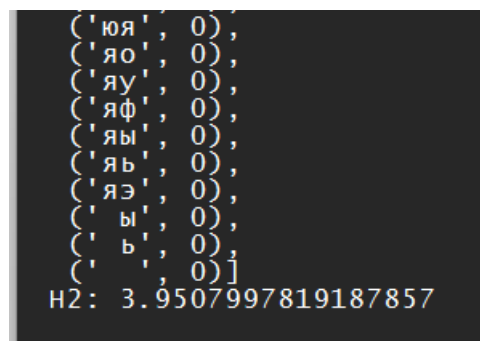


```
h2_spaces_overlap.txt - Notepad

File Edit View

[('о ', 32824),
 ('е ', 27335),
 ('п ', 25778),
 ('я ', 24453),
 ('и ', 24327),
 ('а ', 23275),
 ('н ', 23242),
 ('с ', 22680),
 ('в ', 22145),
 ('то ', 19411),
```

...



```

 ('юя', 0),
 ('яо', 0),
 ('яу', 0),
 ('яф', 0),
 ('яы', 0),
 ('яь', 0),
 ('яэ', 0),
 ('ы ', 0),
 ('ь ', 0),
 (' ', 0)]
H2: 3.9507997819187857
```

H2: 3.9507997819187857

Підрахуємо частоту біграм (з перетином) у тексті без пробілів та H2 для біграм (h2_no_spaces_overlap.py):

```
# By Bondarenko and Kryhin

from pprint import pprint
from math import log

FROM_FILE = 'text_no_spaces.txt'
alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()

length = len(text) - 1 # determined experimentally (number of bigrams)
bigrams = dict() # numbers of each bigram

for i in alphabet:
    for j in alphabet:
        bigrams[i + j] = 0

for i in bigrams.keys():
    bigrams[i] = text.count(i)

pprint(sorted(bigrams.items(), key=lambda item: item[1], reverse=True), sort_dicts=False)
# sorted output

h2 = 0
for i in bigrams:
    if bigrams[i] == 0:
        continue
    h2 -= (bigrams[i] / length) * log(bigrams[i] / length, 2)

print(f"H2: {h2 / 2}")
```

OUTPUT(h2_no_spaces_overlap.txt):

```
[('то', 19907),
 ('не', 15722),
 ('но', 15660),
 ('по', 15567),
 ('на', 15194),
 ('ст', 14931),
 ('ен', 13880),
 ('ал', 13661),
 ('ко', 12432),
 ('ос', 12274),
 ('ра', 12020),
 ('аа', 12000),
```

...

```
(('эы', 0),
 ('эь', 0),
 ('ээ', 0),
 ('эю', 0),
 ('эя', 0),
 ('юы', 0),
 ('юь', 0),
 ('яы', 0),
 ('яь', 0)]
H2: 4.136805930152785
```

H2: 4.136805930152785

Підрахуємо частоту біграм (без перетину) у тексті з пробілами та H2 для біграм (h2_spaces_no_overlap.py):

```
# By Bondarenko and Kryhin

from pprint import pprint
from math import log

FROM_FILE = 'new_text.txt'
alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя '

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()

text_length = len(text)
length = text_length // 2 # determined experimentally (number of bigrams without
overlap)
bigrams = dict()

for i in alphabet:
    for j in alphabet:
        bigrams[i + j] = 0

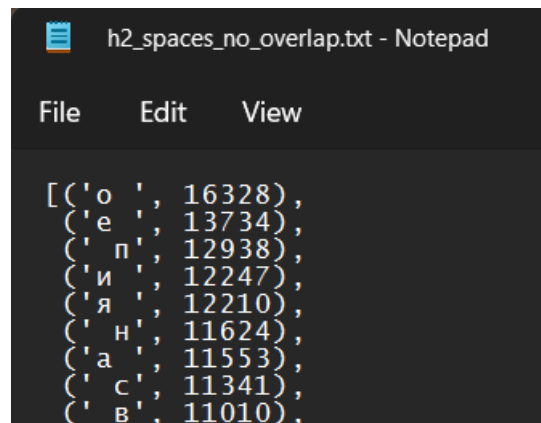
i = 0
while i < text_length:
    if i == text_length - 2: # save from "index out of range"
        break
    bigrams[text[i:i+2]] += 1
    i += 2

pprint(sorted(bigrams.items(), key=lambda item: item[1], reverse=True), sort_dicts=False)

h2 = 0
for i in bigrams.keys():
    if bigrams[i] == 0:
        continue
    h2 -= (bigrams[i] / length) * log(bigrams[i] / length, 2)

print(f"H2: {h2 / 2}")
```

OUTPUT(h2_spaces_no_overlap.txt):

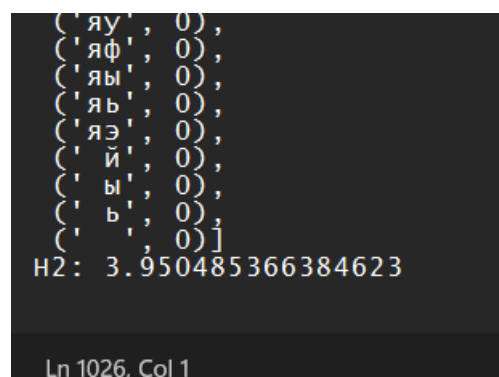


```
h2_spaces_no_overlap.txt - Notepad

File Edit View

[('о ', 16328),
 ('е ', 13734),
 ('п ', 12938),
 ('и ', 12247),
 ('я ', 12210),
 ('н ', 11624),
 ('а ', 11553),
 ('с ', 11341),
 ('в ', 11010),
```

...



```

 ('ю ', 0),
 ('яф ', 0),
 ('яы ', 0),
 ('яь ', 0),
 ('яэ ', 0),
 ('й ', 0),
 ('ы ', 0),
 ('ь ', 0),
 (' ', 0)]
H2: 3.950485366384623

Ln 1026, Col 1
```


H2: 3.950485366384623

Підрахуємо частоту біграм (без перетину) у тексті без пробілів та H2 для біграм (h2_no_spaces_no_overlap.py):

```
# By Bondarenko and Kryhin

from pprint import pprint
from math import log

FROM_FILE = 'text_no_spaces.txt'
alphabet = 'абвгдежзийклмнопрстуфхцшщъыьэюя'

with open(FROM_FILE, 'r') as f:
    text = f.read().lower()

text_length = len(text)
length = text_length // 2 # determined experimentally (number of bigrams without
overlap)
bigrams = dict()

for i in alphabet:
    for j in alphabet:
        bigrams[i + j] = 0

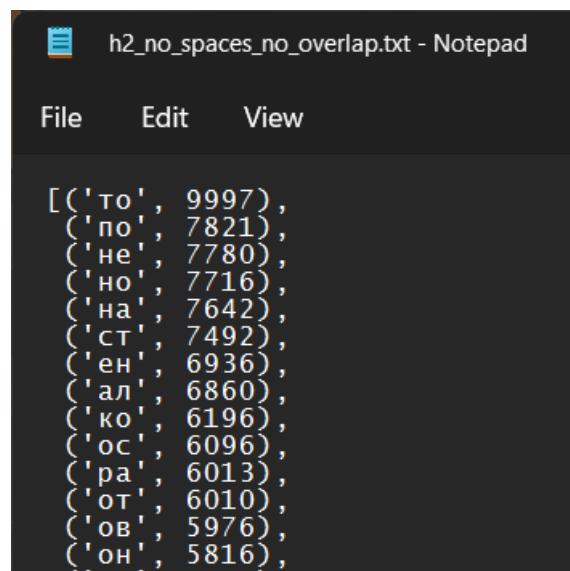
i = 0
while i < text_length:
    if i == text_length - 2: # save from "index out of range"
        break
    bigrams[text[i:i+2]] += 1
    i += 2

pprint(sorted(bigrams.items(), key=lambda item: item[1], reverse=True), sort_dicts=False)

h2 = 0
for i in bigrams.keys():
    if bigrams[i] == 0:
        continue
    h2 -= (bigrams[i] / length) * log(bigrams[i] / length, 2)

print(f"H2: {h2 / 2}")
```

OUTPUT(h2_no_spaces_no_overlap.txt):



```
h2_no_spaces_no_overlap.txt - Notepad

File Edit View

[('то', 9997),
 ('но', 7821),
 ('не', 7780),
 ('но', 7716),
 ('на', 7642),
 ('ст', 7492),
 ('ен', 6936),
 ('ал', 6860),
 ('ко', 6196),
 ('ос', 6096),
 ('ра', 6013),
 ('от', 6010),
 ('ов', 5976),
 ('он', 5816),
 ('лр', 5440),
 ...]
```

```
( 'ЭШ', 0),  
( 'ЭЩ', 0),  
( 'ЭЫ', 0),  
( 'ЭЬ', 0),  
( 'ЭЭ', 0),  
( 'ЭЮ', 0),  
( 'ЭЯ', 0),  
( 'ЮЫ', 0),  
( 'ЮЬ', 0),  
( 'ЯЫ', 0),  
( 'ЯЬ', 0)]  
H2: 4.136328719898478  
  
Ln 963, Col 1
```

H2: 4.136328719898478

Note:

Файл new_text.txt створюється програмою filter.txt

Файл text_no_spaces.txt створюється програмою text_without_spaces.py

Файли .txt у кодуванні ANSI (можуть виникати проблеми на інших пристроях)

Проблеми та шляхи їх вирішення:

Особливо проблем не виникало, лише була помилка, коли ми неправильно рахували ентропію через що, ділили не на кількість біграм, а на довжину тексту.

Інших проблем не було, код і алгоритм легкий.

Отже, отримали такі значення ентропії:

H1: 4.360843323137924

H1: 4.453920329839864

H2: 3.9507997819187857

H2: 4.136805930152785

H2: 3.950485366384623

H2: 4.136328719898478

H(10)

H(20)

H(30)

Тоді, значення надлишковості російської мови за цим джерелом (ТЕХТ):

H(10): $45.49\% < R < 61.97\%$

H(20): $49.63\% < R < 66.28\%$

H(30): $56.54\% < R < 70.59\%$

Порахуємо надлишковість для тексту (Хроніки Амбера):

H1: 4.360843323137924 (h1: spaces)

$H_0 = \log_2 32 = 5$

$R = 12.78\%$

H1: 4.453920329839864 (h1: no_spaces)

$H_0 = \log_2 31 = 4.9542$

$R = 10.09\%$

H2: 3.9507997819187857 (h2: spaces + overlap)

$H_0 = \log_2 32 = 5$

$R = 20.98\%$

H2: 4.136805930152785 (h2: no spaces + overlap)

$H_0 = \log_2 31 = 4.9542$

$R = 16.5\%$

H2: 3.950485366384623 (h2: spaces + overlap)

$H_0 = \log_2 32 = 5$

$R = 21\%$

H2: 4.136328719898478 (h2: no spaces + no overlap)

$H_0 = \log_2 31 = 4.9542$

$R = 16.5\%$

Висновок:

Під час виконання даної лабораторної роботи, ми засвоїли поняття ентропії та надлишковості. Мовою Python написали кілька програм, які рахують частоту входження кожної букви в текст, та частоту біграм, а також H_1 та H_2 . За отриманими значеннями розрахували ентропії H_1 та H_2 . Застосувавши програму CoolPinkProgram.exe, знайшли межі умовної ентропії джерела. В решті решт, оцінили надлишковість російської мови в різних моделях джерела та отримали такі значення:

$H(10): 45.49\% < R < 61.97\%$

$H(20): 49.63\% < R < 66.28\%$

$H(30): 56.54\% < R < 70.59\%$

$R = 12.78\%$

$R = 10.09\%$

$R = 20.98\%$

$R = 16.5\%$

$R = 21\%$

$R = 16.5\%$