

Криптографія. Лабораторна робота 1.

Експериментальна оцінка ентропії на символ джерела

Текст та форматування

Як середньостатистичний текст середньостатистичною російською мовою середньостатистичного росіянина ми взяли ~500000 символів (~1Мб) тексту ниття терориста та військового злочинця ігоря стрелкова—ірка з його телеграм-каналу.

Зчитуємо текст в змінну-рядок:

```
text = fileread('girkin.txt')
```

```
text =
```

```
'Хитроумный Одиссей между Сциллой и Харибдой'
```

```
В моей далекой уже молодости мною предполагалось (совершенно наивно), что каждый гуманитарий хотя-бы в конспек
```

```
Краткий "ликбез":
```

```
Одиссей был царем на острове Итака (мелкий островок на Адриатическом море в нынешней Греции). Участвовал в оса
```

```
Подтвердить своё прозвище Одиссей смог самым наилучшим образом. - В то время, как все остальные греческие цари
```

```
Ну так вот, последние спутники Одиссея (вместе с его последним кораблем) погибли в проливе между современной Си
```

```
Но - все по порядку: Итак, пройти в те мифические времена пролив нельзя было, не угодив к одному из чудовищ. Ха
```

```
Так вот к чему это я всё? - А к тому, что наш президент (не критикую!) и его ближайшее окружение - находятся с
```

```
⋮
```

Форматуємо текст (видалення зайвих символів та зайвих пробілів):

```
formatted_text = blanks(length(text));

for i = 1:length(text)
    c = text(i);
    if (c <= 'я' && c >= 'а') || c == ' '
        formatted_text(i) = c;
    elseif (c <= 'Я' && c >= 'А')
        formatted_text(i) = char(c + 0x0020);
    elseif c == 'Ё' || c == 'ё'
        formatted_text(i) = 'e';
    else
        formatted_text(i) = ' ';
    end
end

mask = (formatted_text == ' ');
mask = ~(mask & [0 mask(1:end-1)]);
```

```
formatted_text2 = formatted_text(mask); % 3 пробілами
formatted_text3 = formatted_text(formatted_text ~= ' '); % Без пробілів
```

Оригінальний (файл girkin.txt) та кінцевий форматований текст (змінна formatted_text2) мають бути деє разом з цим файлом.

Робота з форматованим текстом

Створюємо два відображення типу "символ -> кількість входжень в текст" та "біграма -> кількість входжень в текст":

```
bigrams = dictionary(string([]), []);
bigrams_wo_spaces = dictionary(string([]), []);
bigrams_wo_spaces_i = dictionary(string([]), []);
bigrams_i = dictionary(string([]), []);
chars = dictionary(string([]), []);
```

Заповнюємо дані про окремі символи :

```
for i = 1:length(formatted_text2)
    c1 = formatted_text2(i);
    if chars.isKey(c1)
        chars(c1) = chars(c1) + 1;
    else
        chars(c1) = 1;
    end
end
tttt = chars.keys'
```

```
tttt = 1x33 string
"x"      "и"      "т"      "р"      "о"      "у"      "м . . .
```

```
ttt = chars.values'
```

```
ttt = 1x33
    5356    37250    31194    24918    56742    12712 . . .
```

Заповнюємо дані про біграми :

```
for i = 1:(length(formatted_text2)-1)
    c2 = [formatted_text2(i) formatted_text2(i+1)];

    if bigrams.isKey(c2)
        bigrams(c2) = bigrams(c2) + 1;
    else
        bigrams(c2) = 1;
    end
end
```

```

end
end

for i = 1:2:(length(formated_text2)-1)
    c2 = [formated_text2(i) formated_text2(i+1)];

    if bigrams_i.isKey(c2)
        bigrams_i(c2) = bigrams_i(c2) + 1;
    else
        bigrams_i(c2) = 1;
    end
end

for i = 1:2:(length(formated_text3)-1)
    c2 = [formated_text2(i) formated_text2(i+1)];

    if bigrams_wo_spaces_i.isKey(c2)
        bigrams_wo_spaces_i(c2) = bigrams_wo_spaces_i(c2) + 1;
    else
        bigrams_wo_spaces_i(c2) = 1;
    end
end

for i = 1:(length(formated_text3)-1)
    c2 = [formated_text3(i) formated_text3(i+1)];

    if bigrams_wo_spaces.isKey(c2)
        bigrams_wo_spaces(c2) = bigrams_wo_spaces(c2) + 1;
    else
        bigrams_wo_spaces(c2) = 1;
    end
end
end

```

Ці зібрані дані прикладені окремими файлами (див. нижче)

Обчислення ентропій H_1 (символів з пробілами), H_2 (біграм з пробілами з перетином), H_3 (символів без пробілів), H_4 (біграм без пробілів з перетином), H_5 (біграм з пробілами без перетину) та H_6 (біграм без пробілів без перетину)

```

t1 = chars.values / sum(chars.values);
H1 = -sum(t1 .* log2(t1))

```

```

H1 = 4.3851

```

```

chars(' ') = [];
t3 = chars.values / sum(chars.values);

```

```
H3 = -sum(t3 .* log2(t3))
```

```
H3 = 4.4392
```

```
t_b = bigrams.values / sum(bigrams.values);  
H2 = -sum(t_b .* log2(t_b)) * 0.5
```

```
H2 = 3.9881
```

```
t_b4 = bigrams_wo_spaces.values / sum(bigrams_wo_spaces.values);  
H4 = -sum(t_b4 .* log2(t_b4)) * 0.5
```

```
H4 = 4.1241
```

```
t_b5 = bigrams_i.values / sum(bigrams_i.values);  
H5 = -sum(t_b5 .* log2(t_b5)) * 0.5
```

```
H5 = 3.9881
```

```
t_b6 = bigrams_wo_spaces_i.values / sum(bigrams_wo_spaces_i.values);  
H6 = -sum(t_b6 .* log2(t_b6)) * 0.5
```

```
H6 = 3.9878
```

Оцінки для $H^{(10)}$, $H^{(20)}$, $H^{(30)}$ та обчислення надлишковості мови

За допомогою CoolPinkProgram.exe обчислимо $H^{(10)}$, $H^{(20)}$, $H^{(30)}$:

Произвольная часть текста:
 ный_момен

Использованные буквы:

Порядок n-граммы:
 5
 10
 15
 20
 25
 30
 35
 40
 45
 50

Введенный символ:

Символ по счету:

Номер эксперимента: 51

Неравенство для энтропии:
 $1.25952956876821 < H < 2.02205897724832$

Двоичная таблица угаданных символов:

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:

The screenshot shows the 'Симулятор' window with the following data:

- Произвольная часть текста:** ению_некоторых_люде
- Использованные буквы:**
- Порядок n-граммы:** A list from 5 to 50. The value 30 is highlighted.
- Введенный символ:**
- Символ по счету:**
- Номер эксперимента:** 60
- Неравенство для энтропии:** $1.32340490244222 < H < 2.06785101605236$
- Двоичная таблица угаданных символов:** A grid of binary digits (0s and 1s).
- Вероятности:** A list of probabilities $q[1]$ through $q[32]$. Most values are 0, while others like $q[1] = 0.6440677$ and $q[27] = 0.016949$ are non-zero.
- Строка состояния:**

Отже, маємо такі результати :

$$1.26 \leq H^{(10)} \leq 2.02$$

$$1.32 \leq H^{(20)} \leq 2.07$$

$$1.40 \leq H^{(30)} \leq 2.06$$

Візьмемо $H^{(30)}$ як найкраще наближення для H_{∞} , тоді обчислимо надлишковість російської мови R за формулою :

$$R = 1 - \frac{H_\infty}{H_0}, \quad \text{де } H_0 = \log_2(32) = 5$$

Тоді надлишковість R буде в таких межах :

$$0.58 \leq R \leq 0.72$$

Тобто маємо **надлишковість** російської мови в районі **65%**.

Візуалізація

Наступний код потрібен лише для красивих табличок з даними (які також мають бути десь разом з цим файлом) і ніякого іншого корисного навантаження не несе.

(можна було б зробити ці таблички інтерактивними прямо тут, але це занадто складно, а результат не вартує того)

```
keys_c = chars.keys;
values_c = chars.values;
[sortedValues_c, sortInd_c] = sort(values_c);
sortedKeys_c = keys_c(sortInd_c);

%-----

svalues1 = zeros(32, 1);

for i = 1:32
    for j = 1:32
        b = string([char(i + 'a' - 1) char(j + 'a' - 1)]);

        if bigrams.isKey(b)
            svalues1(i) = svalues1(i) + bigrams(b);
        end
    end
end

alph = 'абвгдежзийклмнопрстуфхцчшщъыьэюя';

[~, sortInd1] = sort(svalues1);

alph = alph(sortInd1);

bfreq_matrix = zeros(32, 32);

for i = 1:32
    for j = 1:32
        b = string([alph(i) alph(j)]);

        if bigrams.isKey(b)
            bfreq_matrix(i, j) = bigrams(b);
        end
    end
end
```