NIE-VSM: Homework 1.

Filip Kašpar, Matěj Tarča

13. března 2025

1. Finding parameters of our assignment

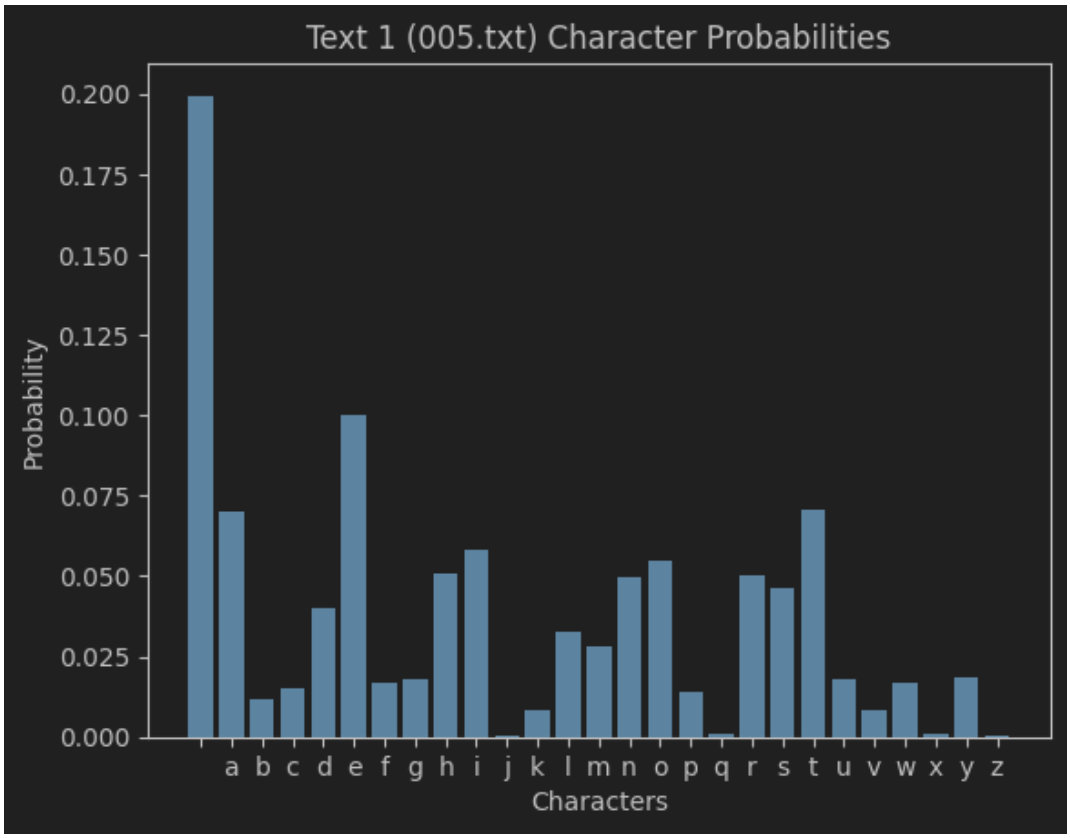
We chose Filip Kašpar as our leader therefore our parameters are following:

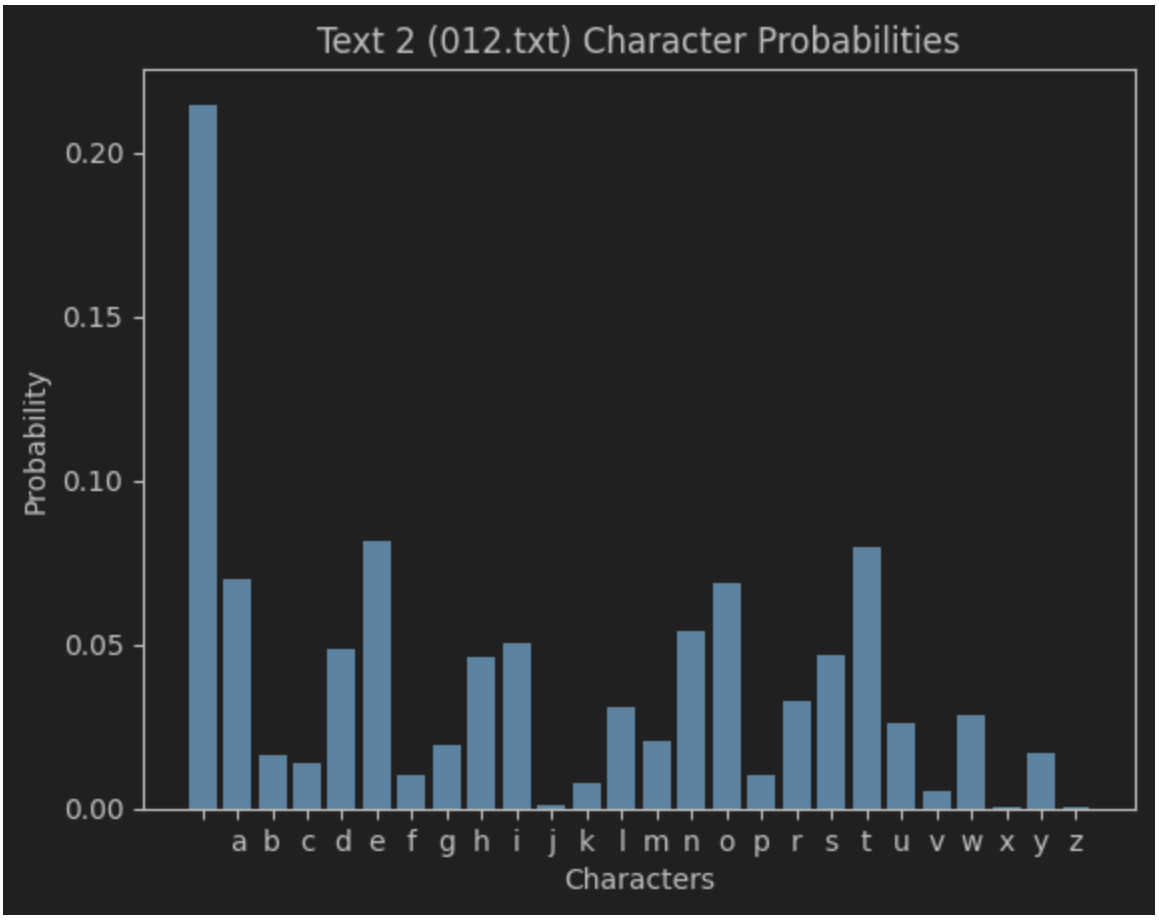
K	**8**
L	**6**
X	**5**
Y	**12**

That means we are working with texts 005 and 012. Also, all programming has been done in Python programming language.

2. Loading texts

To find the probability of individual letters in both texts we used **Counter** from **collections** module. Then we just divided each letter by the total amount of letters in the text to get the corresponding probabilities. To graph the results, we used the **matplotlib** module.





3. Finding entropy

The entropy of each text can be found using this formula:

$$H(P) = -\sum_i p_i * \log_2(p_i)$$

To find the entropy of both texts we used the **entropy** function from the **scipy** module. The final entropies are:

Entropy of Text 1:	**4.063**
Entropy of Text 2:	**4.0133**

As we can see the entropies of both texts are similar, which makes sense since both texts have similar distribution of letters.

4. Finding optimal instantaneous binary code

For this task we used Huffman code. There is already a module in python for Huffman code, so we just imported it. The resulting coded letters using Huffman code for **text1** are:

‘ ’	00
A	1011
B	1110010
C	100110
D	11011
E	1111
F	101010
G	101011
H	0110
I	1000
J	11100010111
K	1110000
L	10100
M	10010
N	0100
O	0111
P	1110011
Q	1110001010
R	0101
S	11101
T	1100
U	110100
V	11100011
W	100111
X	111000100
Y	110101
Z	11100010110

As expected the most used letters have the shortest code.

5. Computing expected code length & comparison with entropy

To compute the expected code length, we used the following formula:

$$L = \sum_i p_i * l_i$$

Which gave the following results:

Text 1	4.107

Text 2	**4.079**

When put to comparison with the entropy of each text:

Text 1 Entropy	**4.063**
Text 1 expected code length	**4.107**
Text 2 Entropy	**4.0133**
Text 2 expected code length	**4.079**

Both expected code lengths are valid because they satisfy **H(X) + 1 > *L(C)* >= *H(X)***. The expected code length is closer to the entropy in the first text, which makes sense, because the letter coding has been created from text 1.