**x**

Расчёт информативности данных в различных кодировках

Выполнил: Ковкель Никита Викторович, 3 курс 4 группа 2 подгруппа

2024

**Ход работы**

По итогам предыдущей лабораторной работы было установлено, что энтропия определяется статическими свойствами алфавитов и сообщений (энтропия в соответствии с Шенноном и Хартли).

В начале выполнения задания был применен метод Python для конвертации латинского текста (eng.txt) из Лабораторной работы №1 в формат base64 (eng\_base64.txt). Код, который был использован для этой операции, представлен в Листинге 1.

|  |
| --- |
| with open('eng.txt', 'rb') as input\_file, open('eng\_base64.txt', 'wb') as output\_file:  encoded\_string = base64.b64encode(input\_file.read())  output\_file.write(encoded\_string) |

Листинг 1 ­– Функция преобразования данных в формат base64.

Избыточностью алфавита называют уменьшение информационной нагрузки на один символ вследствие разной вероятности и взаимозависимости появления его символов в сообщениях. И считается она по следующей формуле:



С использованием приложения, разработанного в Лабораторной работе №1, было получено распределение частотных характеристик алфавитов для документов eng.txt и eng\_base64.txt. Выполнен расчёт энтропии Хартли и Шеннона, а также избыточности алфавитов. Результаты подсчетов представлены на Рисунке 1.

|  |  |
| --- | --- |
|  |  |

Рисунок 1 – Результат работы приложения

Также была разработана функция, которая принимает два буфера - a и b, представляющие собой фамилию и имя соответственно. Если размеры этих буферов не совпадают, функция дополняет меньший буфер нулями до размера большего буфера. Функция затем возвращает результат операции XOR между двумя буферами. Входные данные представлены в кодах ASCII или в кодах Base64. Код функции представлен в Листинге 2.

|  |
| --- |
| def xor\_buffers(a, b):  max\_len = max(len(a), len(b))  a = a.ljust(max\_len, b'\0')  b = b.ljust(max\_len, b'\0')  result = bytearray()  for a\_byte, b\_byte in zip(a, b):  result.append(a\_byte ^ b\_byte)  return bytes(result)  a = 'Никита'  b = 'Ковкель'  a\_ascii = '10000011010100001111101000011001010000111010100001101011000011101110001001100'  b\_ascii = '100000111011000011100010000111010100001110001000100001010000110000'  a\_base64 = base64.b64encode(a.encode()).decode()  b\_base64 = base64.b64encode(b.encode()).decode()  print( xor\_buffers(xor\_buffers(a\_ascii.encode(), b\_ascii.encode()), b\_ascii.encode()) == a\_ascii.encode())  print( xor\_buffers(xor\_buffers(a\_base64.encode(), b\_base64.encode()), b\_base64.encode()) == a\_base64.encode()) |

Листинг 2 – Реализация функции, выполняющая XOR

Результаты работы функции представлены на рисунке 2.

A screenshot of a computer screen

Description automatically generated

Рисунок 2 – Результаты операций

Результаты функций возвращают «a» в различных форматах. Это объясняется свойствами операции XOR: «b XOR b» всегда равно 0, а «a XOR 0» всегда равно «a». Таким образом, независимо от представления, функция всегда возвращает исходное значение «a».