Расчёт информативности данных в различных кодировках

Выполнил: Макаров Алексей Игоревич, 3 курс 4 группа 2 подгруппа

2024

**Ход работы**

В результате выполнения прошлой лабораторной работы мы выяснили,

что энтропия зависит от статических характеристик самих алфавитов и сообщений (энтропия по Шеннону и по Хартли).

В начале работы была использована функция языка программирования Python для преобразования латинского текста (latin\_text.txt) из Лабораторной работы №1 в формат base64 (latin\_text\_base64.txt). Программный код для выполнения этой операции представлен в Листинге 1.

|  |
| --- |
| with open('latin.txt', 'rb') as file:  encoded\_string = base64.b64encode(file.read())  with open('latin\_base64.txt', 'wb') as file:  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 latin\_text.txt и latin\_text\_base64.txt. Выполнен расчёт энтропии Хартли и Шеннона, а также избыточности алфавитов. Результаты подсчетов представлены на Рисунке 1.

A white background with black and white text

Description automatically generated with medium confidence

Рисунок 1 – Результат работы приложения

Далее была создана функция, принимающая два аргумента в виде буферов a и b: фамилии и имени соответственно. В случае несоответствия их размеров функция приводит их к одинаковой длине, путем добавление меньшей строке 0, и возвращает результат операции XOR. Входные данные представлены как в кодах ASCII, так и в кодах base64. Результатом операции «а XOR b XOR b» будет просто «а», так как «b XOR b» = 0, при этом «а XOR 0» = «а». Таким образом, чтобы проверить корректность работы приложения, надо сравнить результат с «а». Код функции представлен в Листинге 2.

|  |
| --- |
| def custom\_xor\_base64(buffer\_a, buffer\_b):  decoded\_a = base64.b64decode(buffer\_a)  decoded\_b = base64.b64decode(buffer\_b)  result = []  min\_length = min(len(decoded\_a), len(decoded\_b))  for i in range(min\_length):  result.append(chr(decoded\_a[i] ^ decoded\_b[i]))  return "".join(result)  def custom\_xor\_ascii(string\_a, string\_b):  # Преобразование строк в коды ASCII  ascii\_a = [ord(char) for char in string\_a]  ascii\_b = [ord(char) for char in string\_b]  # Выполнение операции XOR  result = []  min\_length = min(len(ascii\_a), len(ascii\_b))  for i in range(min\_length):  result.append(chr(ascii\_a[i] ^ ascii\_b[i]))  return "".join(result) |

Листинг 2 – Реализация функции, выполняющая XOR

Результатом операции «а XOR b XOR b» будет просто «а», так как «b XOR b» = 0, при этом «а XOR 0» = «а». Таким образом, чтобы проверить корректность работы приложения, надо сравнить результат с «а». Результаты работы функции представлены на рисунке 2.

A screenshot of a computer screen

Description automatically generated

Рисунок 2 – Результаты операций «а XOR b» и «а XOR b XOR b»

Таким образом, результаты работы приложения подтвердили утверждение о равенстве выражения символу «а» – фамилии.