Расчёт параметров и информационных характеристик дискретных ИС

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

Ход работы

Алфавит – общее число знаков или символов, используемых для генерации или передачи сообщений. Его информационной характеристикой, т. е. источником сообщений на основе этого алфавита, является энтропия. Она отражает какое количество информации приходится в среднем на один символ. Данный термин был применён к техническим системам К. Шенноном и Р. Хартли.

По Шеннону энтропию высчитывают по следующей формуле:

Для вычисления энтропии сначала необходимо определить количество вхождений каждого из символов в строку. Программная реализация представлена в листинге 1.

Затем следует переход от подсчёта вхождений к вероятности их появления, что можно осуществить с помощью формулы Шеннона, приведённой выше. Её программная реализация представлена в листинге 2 В качестве данных для вычисления энтропии использовались алфавиты кириллический русский и латинский английский, на основе которых были сформированы текстовые документы, использующиеся для предоставления данных приложению. Пример реаизации данных подсчетов представлен в листинге 1.

def calculate\_entropy(text):

# Подсчет частоты появления каждого символа

char\_freq = {}

total\_chars = len(text)

for char in text:

if char.isalpha():

if char in char\_freq:

char\_freq[char] += 1

else:

char\_freq[char] = 1

# Рассчет вероятности для каждого символа

probabilities = {char: freq / total\_chars for char,

freq in char\_freq.items()}

# Рассчет энтропии

entropy = -sum(p \* math.log2(p) for p in probabilities.values() if p != 0)

return entropy

Листинг 1

Также использовался бинарный алфавит, состоящий из 0 и 1 и записанный в текстовый документ.

На основе определения частоты встречаемости символов в данных текстах, можно построить гистограммы, отражающие эти показатели. Это представлено в таблице 1.

|  |  |
| --- | --- |
|  | Гистограмма частоты символов на латинице |
|  | Гистограмма частоты символов на кирилице |
|  | Гистограмма частоты символов в бинарном виде |

Таблица 1

При известной энтропии алфавита можно посчитать количество информации по следующей формуле:

Т.е. количество информации есть произведение энтропии и длины сообщения. Если алфавит является бинарным, то за количество информации принимается длина строки, ибо энтропия принимается за 1. Иначе энтропия домножается на количество символов в сообщении, чьё количество информации необходимо вычислить. Рисунок 5 – Вывод количества информации

Важно подчеркнуть, что количество информации для последнего случая, включаещего в себя ФИО в кодировке ASCII, равняется количеству символов исходного сообщения: 264.

Возможны случаи ошибочной передачи сообщения, поэтому для вычисления количества информации используется формула эффективной энтропии:

где - условная энтропия:

.

Программная реализация функции вычисления эффективной энтропии представлена в листинге 2.

Листинг 2. Реализация функции вычисления эффективной энтропии.

def is\_binary(text):

for char in text:

if char not in ['0', '1']:

return False

return True

def effective\_entropy(text, p):

q = 1 - p

if is\_binary(text) and (p == 0 or q == 0):

return 1

elif q==0:

return 1

elif is\_binary(text):

return calculate\_binary\_entropy(text) - (- p \* math.log2(p) - q \* math.log2(q))

else:

return calculate\_entropy(text) - (- p \* math.log2(p) - q \* math.log2(q))

def info\_amount\_with\_errors(file, p):

try:

with open(file, 'r') as f:

text = f.read()

return effective\_entropy(text, p) \* len(text)

except FileNotFoundError:

return None

Как уже упоминалось, в бинарном алфавите эффективная энтропия равна длине строки, а при вероятности ошибки 0 или 1, и количеству информации соответственно. В небинарных, при вероятности ошибки 1, эффективная энтропия и количество информации равно 0.

Таким образом, функция подсчёта количества информации при наличии ошибок имеет вид, представленный в листинге 3.

Листинг 3.

def info\_amount\_with\_errors(file, p):

try:

with open(file, 'r') as f:

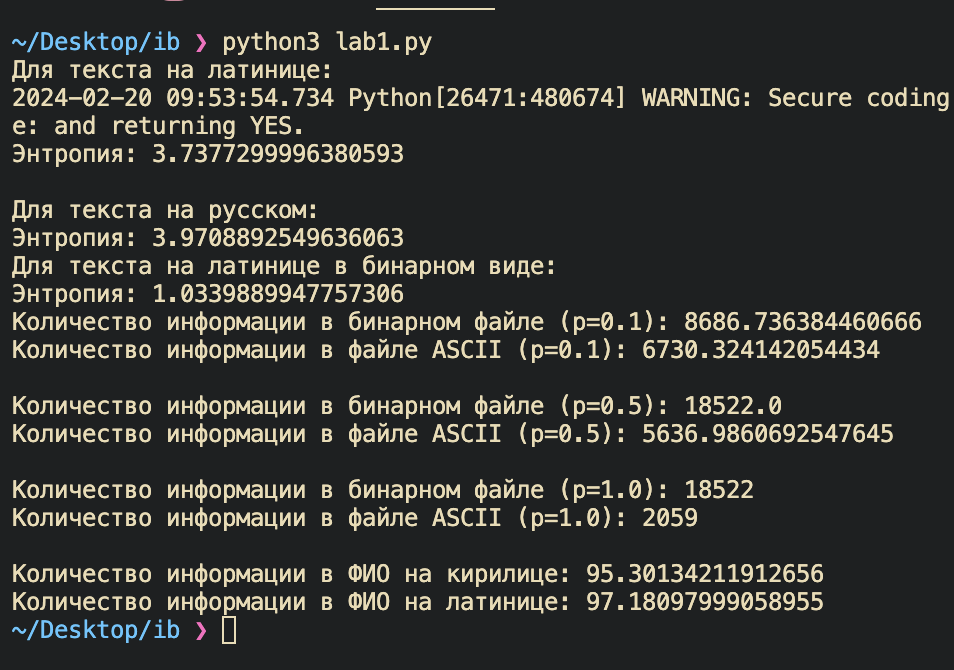
text = f.read()

return effective\_entropy(text, p) \* len(text)

except FileNotFoundError:

return None

Результаты работы полной программы представлены на рисунке ниже.



Как можно заметить, при вероятности ошибки =1, количество информации в бинарном алфавите практически равняется количеству информации при =0, что объясняется заменой всех битов сообщения на обратные, а в небинарном при =1 такое невозможно, ибо мощность алфавита больше 2, и каждый символ заменяется на другой произвольный из этого алфавита, поэтому и количество информации в таком случае равняется 0.