Пшенко Артем, ФИТ 3-4

Информационная безопасность

Отчет по лабораторной работе № 1.

Алфавит – это общее число знаков или символов (N), используемых для генерации или передачи сообщений. Информационной характеристикой алфавита (источника сообщений на основе этого алфавита) является энтропия. Ее физический смысл заключается в отображении количества информации в среднем на 1 символ алфавита.

По Шеннону энтропию высчитывают по следующей формуле:

Для вычисления энтропии необходимо определить количество вхождений каждого из символов в строку, которое осуществляется с помощью модуля collections в python.

Листинг 1. Реализация функции для расчета энтропии.

|  |
| --- |
| histograms = {}  def calculate\_entropy(file\_name):  with open(file\_name, 'r') as f:  if is\_binary(file\_name):  return 1  else:  text = f.read()  #словарь вида "символ - количество"  symbol\_counts = collections.Counter(text)  #print(symbol\_counts, '\n')  total\_symbols = len(text)  #словарь вида "символ - его вероятность"  probabilities = {symbol: count / total\_symbols for symbol, count in symbol\_counts.items()}  #print(probabilities)  if file\_name not in histograms:  draw\_hist(probabilities)  histograms[file\_name] = probabilities  entropy = -sum(p \* math.log2(p) for p in probabilities.values()) # ф-ла 2.1  return entropy |

В качестве данных для вычисления энтропии использовались текстовые документы на английском и русском языках. 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. Диаграмма распределения вероятностей символов для латиницы представлена на рисунке 1.

|  |
| --- |
|  |

Рисунок 1 – Диаграмма распределения вероятностей символов для латиницы

Диаграмма распределения вероятностей символов для кириллицы представлена на рисунке 2.

|  |
| --- |
|  |

Рисунок 2 – Диаграмма распределения вероятностей символов для кириллицы

Подставляя данные в формулу для расчета энтропии, получаем результаты, приведённые на рисунке 3.

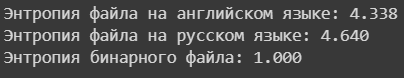


Рисунок 3 – Результаты подсчёта энтропии

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

Программная реализация представлена в листинге 3.

Листинг 2. Реализация подсчёта количества информации.

|  |
| --- |
| def info\_amount(file):  try:  with open(file, 'r') as f:  text = f.read()  return calculate\_entropy(file) \* len(text)  except FileNotFoundError:  return None |

Для бинарного алфавита за количество информации принимается количество символов в сообщении (т.к. энтропия равна 1). В противном случае энтропия умножается на количество символов в сообщении. Результаты представлены на рисунке 4.



Рисунок 4 – Вывод количества информации

Количество информации для ФИО в кодировке ASCII равняется количеству символов исходного сообщения (168, т.к. энтропия принимается за единицу). Также при передаче сообщения с некоторой вероятностью могут возникнуть ошибки, в таком случае используется формула условной энтропии:

,

где - условная энтропия:

.

Для бинарного алфавита формула имеет видоизменяется следующим образом:

Программная реализация функции вычисления эффективной энтропии представлена в листинге 3.

Листинг 3. Реализация функции вычисления эффективной энтропии.

|  |
| --- |
| def effective\_entropy(file, p):  q = 1 – p  if is\_binary(file) and (p == 0 or q == 0):  return 1  elif not is\_binary(file) and p == 1:  return 0  return 1 - (- p \* math.log2(p) - q \* math.log2(q)) |

Функция подсчета количества информации с учетом ошибок представлена в листинге 4.

Листинг 4. Функция подсчёта количества информации при наличии ошибок

|  |
| --- |
| def info\_amount\_with\_errors(file, p):  try:  with open(file, 'r') as f:  text = f.read()  return effective\_entropy(file, p)\*len(text) #calculate\_entropy(file)\*  except FileNotFoundError:  return None |

Результаты вычисления количества информации представлены на рисунке 5.

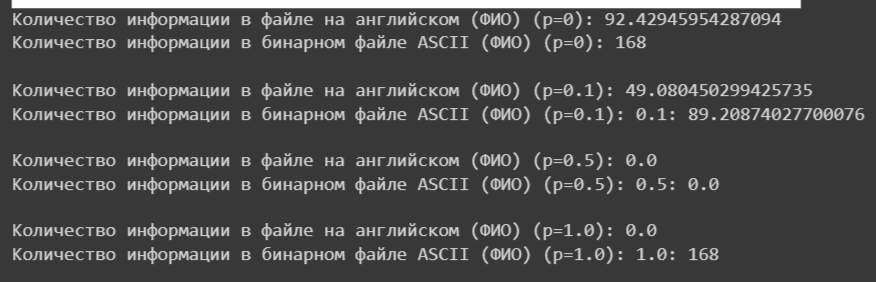


Рисунок 5 – Результаты вычисления количества информации при наличии ошибок

При вероятности ошибки =1, количество информации в бинарном алфавите равно 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.