Лабораторна робота  
з дисципліни  
«Теорія інформації кодування»

на тему  
**«Оцінювання обсягу інформації. Теорема Шенона»**

Виконав:

студент групи ДА-22

.

**Мета роботи**: Засвоїти основні відомості з методів вимірювання обсягу інформації. Отримати навички оцінювання збитковості інформації та засобів її подолання.

**Завдання**

**Пункт 1**. Визначити статистичні характеристики повідомлення , яке складається з вашого ФІО: вірогідності символів, побудувати гістограму вірогідностей.

Pipich Artem Andriyovich

|  |  |  |
| --- | --- | --- |
| Символ | Кількість | P[i] |
| \_ | 2 | 0,08 |
| P | 1 | 0,04 |
| i | 4 | 0,17 |
| p | 1 | 0,04 |
| c | 2 | 0,08 |
| h | 2 | 0,08 |
| A | 2 | 0,08 |
| r | 2 | 0,08 |
| t | 1 | 0,04 |
| e | 1 | 0,04 |
| m | 1 | 0,04 |
| n | 1 | 0,04 |
| d | 1 | 0,04 |
| y | 1 | 0,04 |
| o | 1 | 0,04 |
| v | 1 | 0,04 |
| ∑ | 24 | 1 |

**Пункт 2.** Розрахувати ентропію повідомлення.

Формула розрахунку ентропії:

, де - вірогідність символу алфавіту з номером і (відмінна від 0), N – розмір алфавіту.



Boloban Oleg Anatoliyovich

|  |  |
| --- | --- |
| " ABOaabceghiilllnnooootvy" | |
| ' | 0,0769231 |
| A' | 0,0384615 |
| B' | 0,0384615 |
| O' | 0,0384615 |
| a' | 0,0769231 |
| b' | 0,0384615 |
| c' | 0,0384615 |
| e' | 0,0384615 |
| g' | 0,0384615 |
| h' | 0,0384615 |
| i' | 0,0769231 |
| l' | 0,115385 |
| n' | 0,0769231 |
| o' | 0,153846 |
| t' | 0,0384615 |
| v' | 0,0384615 |
| y' | 0,0384615 |

**P = 3.90217біт.**

**Пункт 3.** Розрахувати оптимальну кількість біт на одиницю інформації в повідомлені та розмір повідомлення стиснутого за оптимальним алгоритмом.

Р0 = 4 біт - оптимальна кількість біт на одиницю інформації в повідомлені;

І0 = Р0 \* N = 4 \* 24 = 96 біт, - розмір повідомлення стиснутого за оптимальним алгоритмам*.*

**Пункт 4.** Скласти програму, яка розраховує статистичні характеристики та ентропію довільного файлу. Гістограми можна будувати за допомогою excel.

**Лістинг**

#include <fstream>

#include <iostream>

#include <cmath>

using namespace std;

main()

{ unsigned char c;

char name[256];

long k[256];

int i, sum=0,a=0;

double ent=0;

FILE \*file;

cout<<"Input the filename: ";

cin>>name;

file=fopen(name,"rb");

for (i=0;i<256;i++)

k[i]=0;

while (!feof(file))

{ c=getc(file);

sum++;

k[c]++;

}

fclose(file);

file=fopen("out.txt","w");

for (i=0;i<256;i++)

if (k[i])

{ //fprintf(file, "%c %d \n",char(i),k[i]);

//fprintf(file, "%d %d \n",i,k[i]);

ent+=double(k[i])/sum\*log(double(k[i])/sum)/log(2);

a++;

}

fprintf(file, "Alphabet= %d \n",a);

fprintf(file, "Symbol count= %d \n",sum);

fprintf(file, "Entropy= %f \n",-ent);

fclose(file);

return 0;

}

Статистичні данні текстового файлу «input.txt»:

Alphabet= 85

Symbol count= 999568

Entropy= 3.90217

Статистичні данні мультимедійного файлу «input.mp3»:

Alphabet= 256

Symbol count= 7649018

Entropy= 7.9300

Статистичні данні виконуваного файлу «input.exe»:

Alphabet= 256

Symbol count= 5835365

Entropy= 5.3609

**Пункт 5**. Дослідити ефективність роботи архіваторів RAR, ZIP на трьох прикладах файлів: текстовому, мультимедійному, exe. Результати занести в таблицю:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Файл | Розмір | Ентропія | Обсяг інформації | Розмір RAR | Розмір ARC | Розмір ZIP | Збитковість |
| input.txt | 999 567 | 3.90 | 553 510 | 320 355 | 243 732 | 364 205 | 310,11% |
| input.exe | 5 869 809 | 5,36 | 3 932 772 | 1 261 503 | 1 131 410 | 1 512 352 | 418,80% |
| input.mp3 | 7 649 017 | 7,93 | 7 582 088 | 7 402 235 | 7 424 258 | 7 406 364 | 3,33% |
| У середньому | 4 839 464 | 5,91 | 4 022 790 | 2 994 698 | 2 933 133 | 3 094 307 | 244,08% |

**Висновок:**

1. Зовнішній вид кожної з трьох гістограм відповідає дійсній частоті зустрічі символів у досліджуваних файлах:
   * txt-файл має високу частоту зустрічі символу «пробіл», а також символів англійського алфавіту у нижньому регістрі. Це пов’язано з тим, що саме з цих символів складаються слова – основний об’єм текстового файлу
   * exe-файл містить велику кількість символів NULL, і значно меншу кількість інших символів, що характерно для виконуваного файлу. Мають місце локальні максимуми для символів #69, #95, #116, що пояснюється особливостями компілятору.
   * mp3-файл містить велику кількість символів NULL, дещо меншу кількість інших символів, що характерно для мультимедійних файлів. Розподіл частоти між іншими символами приблизно рівномірний, що пов’язане з їх рівноцінністю для кодування звуку.
2. Просліджуються очікувані закономірності при архівуванні цих трьох типів файлів:
   * txt-файл добре архівується через те, що має низьку кількість (у середньому 0,13%) більшої групи символів (цифри, символи пунктуації, англійські літери верхнього регістру) та велику кількість (у середньому 2,8%) англійських літер у нижньому регістрі, а також значну кількість (18,4%) пробілів. Такий набір символів ефективно архівується за статистичним алгоритмом та відповідає низькій ентропії файлу.
   * exe-файл має схожий з текстовим розподіл символів: значна кількість символів NULL (37%), #1 (3.9%), #95 (2.2%), #116 (2.1%) та незначна кількість інших символів у з кодом до #255. Через це статистичне архівування теж є дуже ефективним, але краще працює словарне, через однотипність лексичних одиниць у коді програми (оператори мови C++). Ентропія файлу вірно співвідноситься з отриманою ефективністю архівування).
   * mp3-файл майже не архівується, так як усі його символи мають приблизно рівний розподіл за кількістю (окрім символу NULL). Це пов’язане з тим, що зазвичай мультимедійні файли мають власний алгоритм архівування. До того ж формат mp3 характеризується високою стислістю, що робить його зручним для розповсюдження музики, якість якої достатня для повсякденного прослуховування. Через це ні статистичне, ні словарне (які можуть бути тут лексеми?) не є ефективними, що відповідає високій ентропії.

Отже, виходячи з відповідності результатів реальним числам, можливо зробити висновок, що створена в процесі виконання лабораторної роботи програма об’єктивно аналізує файл, визначає його ентропію відповідно до ефективності статистичного архівування, а також вірно визначає відносну частоту входження символів в файл.