|  |
| --- |
| Университет итмо, кафедра вт |
| Лабораторная работа №1 по дисциплине «Прикладная математика» |
| «Вычисление энтропии Шеннона» |
| Группа Р3302 |
| **Выполнили:**  **Орлова Кристина Александровна**  **Лалетина Екатерина Александровна** |
| **Преподаватель: Тропченко Андрей Александрович** |

|  |
| --- |
| *18.10.18* |

**Задание**

Для выполнения лабораторной работы №1 необходимо:

1. на языках С,С++ или С# реализовать процедуру вычисления энтропии для текстового файла.

Требования к текстовому файлу:

• документ на английском языке

• размер текстового файла - 20-60кБ

• документ должен быть осмысленным

В процедуре необходимо подсчитывать частоты появления символов (прописные и заглавные буквы не отличаются, знаки препинания рассматриваются как один символ, пробел является самостоятельным символом), которые можно использовать как оценки вероятностей появления символов. Затем вычислить величину энтропии. Точность вычисления -- 4 знака после запятой. Обязательно предусмотреть возможность ввода имени файла, для которого будет вычисляться энтропия.

2. Проверить запрограммированную процедуру на нескольких файлах и заполнить таблицу 1.1. вычисленными значениями энтропии

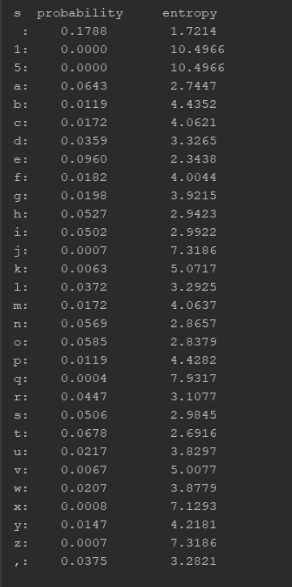
3. Вычислить значение энтропии для тех же файлов, но с использованием частот вхождений пар символов и заполнить таблицу 1.2

4. Проанализировать полученные результаты.

**Исходный код**

**const** fs = require("fs");  
**let** fileContent;  
**let** alphabet = {}; // объект, поля которого - символы, а их значения - объект с полями в частоту, вероятность и энтропию  
**let** realLength = 0;  
  
main();  
  
**function** main() {  
 **let** stdIn = process.openStdin();  
  
 console.log('Input the filename (ex: doc.txt)\nDefault directory: text\_files');  
  
 stdIn.on('data', **function** (filename) {  
 filename = "text\_files//" + filename.toString().substr(0, filename.length - 1);  
  
 **try** {  
 readFile(filename);  
  
 console.log('Result:');  
  
 fillAdditionalInfo();  
 printAlphabet();  
 printEntropy();  
 printLinkedEntropy();  
 } **catch** (err) {  
 console.log('Error: no such txt file')  
 }  
  
 process.exit();  
 });  
}  
  
*/\*\*  
 \* Считывает содержимое файла и формирует по нему алфавит символов  
 \*  
 \** ***@param*** *fileName  
 \*/***function** readFile(fileName) {  
 fileContent = fs.readFileSync(fileName, "utf8");  
 **const** textLength = fileContent.length;  
  
 **for** (**let** i = 0; i < textLength; i++) {  
 **let** symbol = fileContent[i].toLowerCase();  
  
 **if** (symbol.charCodeAt(0) >= 32 && symbol.charCodeAt(0) <= 126) {  
 **if** (alphabet[symbol] === undefined) {  
 alphabet[symbol] = 1;  
 } **else** {  
 alphabet[symbol]++;  
 }  
  
 realLength++;  
 }  
 }  
}  
  
*/\*\*  
 \* Добавляет в alphabet информацию о вероятности и энтропии для каждого символа  
 \*/***function** fillAdditionalInfo() {  
 **for** (**let** key **in** alphabet) {  
 alphabet[key] = {"frequency": alphabet[key], "probability": 0, "entropy": 0};  
 alphabet[key].probability = alphabet[key].frequency / realLength;  
 alphabet[key].entropy = -Math.log(alphabet[key].probability);  
 }  
}  
  
*/\*\*  
 \* Выписывает символ, его вероятность и энтропию  
 \*/***function** printAlphabet() {  
 **let** punctuationMarksProbability = 0;  
 **let** keysArray = [];  
  
 **for** (**let** key **in** alphabet) {  
 keysArray.push(key);  
 }  
  
 keysArray.sort();  
  
 console.log("s probability entropy");  
  
 **for** (**let** i = 0; i < keysArray.length; i++) {  
 **let** key = keysArray[i];  
  
 **if** ([33, 34, 39, 40, 41, 44, 45, 46, 58, 59, 63, 96].indexOf(key.charCodeAt(0)) !== -1) {  
 punctuationMarksProbability += alphabet[key].probability;  
 } **else** {  
 console.log(key + ": " + alphabet[key].probability.toFixed(4) + " " + alphabet[key].entropy.toFixed(4));  
 }  
 }  
  
 **if** (punctuationMarksProbability !== 0) {  
 console.log(",: " + punctuationMarksProbability.toFixed(4) + " " +  
 -Math.log(punctuationMarksProbability).toFixed(4));  
 }  
}  
  
*/\*\*  
 \* Расчитывает и выписывает энтропию файла  
 \*/***function** printEntropy() {  
 **let** entropy = 0;  
  
 **for** (**let** key **in** alphabet) {  
 entropy += alphabet[key].probability \* alphabet[key].entropy;  
 }  
  
 console.log("\nЭнтропия файла равна " + entropy.toFixed(4) + " нит");  
}  
  
*/\*\*  
 \* Экранирует спецсимволы в строке  
 \*  
 \** ***@param*** *str  
 \** ***@return*** *{string}  
 \*/***function** escapeExpressions(str) {  
 **return** str.replace(/(?=[-(){}./?|"])/g, '\\');  
}  
  
*/\*\*  
 \* Расчитывает и выписывает энтропию файла при наличии статистических связей между двумя символами  
 \*/***function** printLinkedEntropy() {  
 **let** entropy = 0;  
 **let** alphabetSymbols = '[';  
  
 **for** (**let** key **in** alphabet) {  
 alphabetSymbols += key;  
 }  
  
 alphabetSymbols += "]";  
 alphabetSymbols = escapeExpressions(alphabetSymbols);  
  
 **for** (**let** j **in** alphabet) {  
 **const** allPairsCount = fileContent.match(**new** RegExp('(?=' + escapeExpressions(j) + '(' + alphabetSymbols + '))', 'gi')) === **null** ? 0 :  
 fileContent.match(**new** RegExp('(?=' + escapeExpressions(j) + '(' + alphabetSymbols + '))', 'gi')).length;  
  
 **for** (**let** i **in** alphabet) {  
 **const** searchedPairsCount = fileContent.match(**new** RegExp(escapeExpressions(j) + escapeExpressions(i), 'gi')) === **null** ? 0 :  
 fileContent.match(**new** RegExp(escapeExpressions(j) + escapeExpressions(i), 'gi')).length;  
 **const** probability = searchedPairsCount / allPairsCount;  
  
 **if** (probability !== 0 && allPairsCount !== 0) {  
 entropy += probability \* alphabet[j].probability \* -Math.log(probability);  
 }  
 }  
 }  
  
 console.log("Энтропия файла при наличии односвязной цепи равна " + entropy.toFixed(4) + " нит");  
}

**Тестирование программы**



**Описание входных данных программы**

Для тестирования использовались три первые главы произведения «Алые паруса» на английском языке размером 36 кб, 30 кб и 22 кб соответственно.

**Анализ результатов работы программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Файл | «Алые паруса» 1 глава | Файл | «Алые паруса» 2 глава | Файл | «Алые паруса» 3 глава |
| Энтропия H(X) | **2.9515** | Энтропия H(X) | **2.9537** | Энтропия H(X) | **2.9692** |
| Энтропия H\*(X) | **2.3371** | Энтропия H\*(X) | **2.3347** | Энтропия H\*(X) | **2.3376** |

В результате выполнения работы было обнаружено, что буква “e” в английском языке встречается чаще остальных. Также видно, что энтропии трех разных по размеру файлов близки друг к другу и что энтропии с учетом статистических связей меньше, чем без, что говорит о наличии статистических связей и о том, что при передаче сообщений лучше их учитывать.

**Заключение**

В результате выполнения работы были изучены способы измерения информационного объема текста, основные понятия и единицы измерения, были рассчитаны на практике энтропии различных файлов в нит.