Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу «Криптография»

Студент: А. А. Литвина
Преподаватель: А. В. Борисов
Группа: М8О-306Б
Дата:
Оценка:
Подпись:

Москва, 2020

Задача:

Сравнить:
1) два осмысленных текста на естественном языке,
2) осмысленный текст и текст из случайных букв,
3) осмысленный текст и текст из случайных слов,
4) два текста из случайных букв,
5) два текста из случайных слов.

Как сравнивать: считать процент совпадения букв в сравниваемых текстах – получить дробное значение от 0 до 1 как результат деления количества совпадений на общее число букв. Расписать подробно в отчёте алгоритм сравнения и приложить сравниваемые тексты в отчёте хотя бы для одного запуска по всем пяти подпунктам. Осознать, какие значения получаются в этих пяти подпунктах. Привести свои соображения о том почему так происходит.

Длина сравниваемых текстов должна совпадать. Привести соображения о том какой длины текста должно быть достаточно для корректного сравнения.

1 Описание

В качестве осмысленного текста я выбрала роман в стихах А.С.Пушкина "Евгений Онегин"на английском языке (поскольку при использовании языка C++ возникают проблемы с русскими буквами). Исходный текст находится в файле "first_text.txt".

Затем я обработала исходный текст. Привела весь текст к нижнему регистру, потом выписала в новый файл "new_text.txt"только буквы и пробелы. Если слова написаны через дефис, разделила их на два отдельных слова.

```
ifstream in("first_text.txt");
ofstream out("new_text.txt");
if ((in.is_open())&&(out.is_open())) {
    while (getline(in, str)) {
      transform(str.begin(), str.end(), str.begin(), ::tolower);
        for (int i=0; i<str.size(); i++) {
          if ((str[i]=='-') && (isalpha(str[i-1])))
            out << ' ';
            else if ((isalpha(str[i])) || (str[i]==' '))
            out << str[i];
        }
        if (str.size()!=0)
          out << ' ';
    }
}
in.close();
out.close();
```

Затем я разделила обработанный текст на два равных по величине текста и записала их в файлы "text1.txt"и "text2.txt".

```
ifstream in1("new_text.txt");
ofstream out1("text1.txt");
ofstream out2("text2.txt");
if ((in1.is_open())&&(out1.is_open())&&(out2.is_open())) {
    getline(in1, str1);
    n=str1.size();
    for (int i=0; i<n/2; i++) {
        out1 << str1[i];
    }
    for (int i=n/2; i<n; i++) {
        out2 << str1[i];
    }
}
in1.close();
out1.close();
out2.close();
```

Далее я сгенерировала два текста "letters1"и "letters2"того же размера, состоящих из

случайных букв. Для этого я сначала генерировала случайное число, отвечающее за количество букв в 'слове' (от 1 до 10), а затем уже генерировала нужное количество случайных букв и после этого ставила пробел.

```
ofstream out3("letters1.txt");
ofstream out4("letters2.txt");
if ((out3.is_open())&&(out4.is_open())) {
    int i=0;
    while (i<n/2) {
        int sp=1+rand()%10;
        i+=sp+1;
        for (int j=0; j<sp; j++) {
            char ch=97+rand()%26;
            out3 << ch;
        }
        out3 << ' ';
    }
    i=0;
    while (i<n/2) {
        int sp=1+rand()%10;
        i+=sp+1;
        for (int j=0; j<sp; j++) {
            char ch=97+rand()%26;
            out4 << ch;
        }
        out4 << ' ';
    }
}
out3.close();
out4.close();
```

Затем сгенерировала два текста слов "words1"и "words2". Для этого я сгенерировала случайное число - позицию символа в тексте "text1". То слово, в котором оказался этот символ, я выделяла и записывала в новый файл.

```
    ofstream out5("words1.txt");
    ofstream out6("words2.txt");
    if ((out5.is_open())&&(out6.is_open())) {
        int i=0;
        int k;
        while (i<n/2) {
            int pos=rand()%(n/2);
            while ((str1[pos]!=' ')&&(pos!=0)) {
                pos--;
            }
            pos++;
            k=0;
            while (str1[pos]!=' '){
                out5 << str1[pos];
                pos++;
```

```
            k++;
        }
        out5 << ' ';
        k++;
        i+=k;
    }
    i=0;
    while (i<n/2) {
        int pos=rand()%(n/2);
        while ((str1[pos]!=' ')&&(pos!=0)) {
            pos--;
        }
        pos++;
        k=0;
        while (str1[pos]!=' '){
            out6 << str1[pos];
            pos++;
            k++;
        }
        out6 << ' ';
        k++;
        i+=k;
    }
}
out5.close();
out6.close();
```

Наконец я сравнила полученные тексты. Для этого я просто проходила по тексту и сравнивала символы на одинаковых позициях. Если они совпадали, увеличивала счетчик.

```
ifstream in2("text1.txt");
ifstream in3("text2.txt");
ifstream in4("letters1.txt");
ifstream in5("letters2.txt");
ifstream in6("words1.txt");
ifstream in7("words2.txt");
if ((in2.is_open())&&(in3.is_open())&&(in4.is_open())&&(in5.is_open())&&(in6.is_open()
    )&&(in7.is_open())) {
    getline(in2, str2);
    getline(in3, str3);
    getline(in4, str4);
    getline(in5, str5);
    getline(in6, str6);
    getline(in7, str7);
    for (int i=0; i<n/2; i++) {
        if (str2[i]==str3[i])
            count1++;
        if (str2[i]==str4[i])
            count2++;
```

```
        if (str2[i]==str6[i])
            count3++;
        if (str4[i]==str5[i])
            count4++;
        if (str6[i]==str7[i])
            count5++;
    }
}
```

В качестве результата я делила количество совпадений на размер всего текста.

```
double p1=100*count1/(n/2);
double p2=100*count2/(n/2);
double p3=100*count3/(n/2);
double p4=100*count4/(n/2);
double p5=100*count5/(n/2);
```

2 Результаты

Два осмысленных текста на естественном языке	7.57%
Осмысленный текст и текст из случайных букв	5.35%
Осмысленный текст и текст из случайных слов	7.13%
Два текста из случайных букв	5.26%
Два текста из случайных слов	6.79%

3 Выводы

В данной лабораторной работе я получила частоты $W_n(A) = \frac{m}{n}$ наступления события A - совпадения двух последовательностей. Частота является приближенной оценкой вероятности $\lim\limits_{n\to\infty} W_n = P(A)$ при достаточно больших n, поэтому чем больше размер взятого текста, тем точнее результат.

В этой работе я экспериментально выяснила, как влияет связность текста на вероятность появления совпадений. По полученным результатам можно сделать вывод, что наибольшее количество совпадений получается, если сравнивать два осмысленных текста, а наименьшее - если сравнивать два текста из случайных букв. Следовательно, чем больше связность текста, тем больше вероятность совпадения.

В связи с этим можно сделать вывод, что в качестве пароля лучше использовать несвязный набор символов, нежели слова или фразы. Находясь по другую сторону, если нам нужно угадать пароль, необходимо в качестве образца использовать связный текст, что увеличивает вероятность совпадений.