МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Лабораторная работа № 3
по курсу «Криптография»

Группа: М8О-307Б-21

Студент: Ф. А. Меркулов

Преподаватель: А. В. Борисов

Оценка:

Дата: 12.04.2024

Москва, 2024

ОГЛАВЛЕНИЕ

1 Тема

Критерий открытого текста

2 Задание

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.

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 случаям. Осознать какие значения получаются в этих пяти случаях. Привести соображения о том почему так происходит.

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

3 Теория

Открытый текст — это данные (не обязательно текстовые), передаваемые без использования криптографии, то есть незашифрованные данные.

Критерий открытого текста — это подход в криптоанализе, использующий особенности исходного текста, такие как структура языка, вероятность встречаемости отдельных символов или слов, для дешифровки. Этот метод исходит из предположения, что некоторые свойства исходного текста могут частично сохраняться в шифротексте, что облегчает его анализ. При шифровании некоторые структурные или статистические характеристики оригинального сообщения могут быть не полностью скрыты, поскольку многие методы шифрования работают на уровне отдельных символов или битов.

Один из простых примеров — частотный анализ букв в тексте на каком-либо языке. В английском, например, в русском языке частота употребления букв: о — 9.28% а — 8.66% е — 8.10% и — 7.45% н — 6.35% т — 6.30% р — 5.53% с — 5.45% л — 4.32% в — 4.19% к — 3.47% п — 3.35% м — 3.29% у — 2.90% д — 2.56% я — 2.22% ы — 2.11% ь — 1.90% з — 1.81% б — 1.51% г — 1.41% й — 1.31% ч — 1.27% ю — 1.03% х — 0.92% ж — 0.78% ш — 0.77% ц — 0.52% щ — 0.49% ф — 0.40% э — 0.17% ъ — 0.04%. Если метод шифрования не изменяет распределение частот букв, крипто аналитик может использовать эту информацию для предположения о соответствии между зашифрованными и реальными буквами, что позволит расшифровать сообщение.

4 Ход лабораторной работы

Для выполнения части: “Привести соображения о том какой длины текста должно быть достаточно для корректного сравнения” сравнивались тексты с длинами 3 разных порядков:

1) 100 символов;
2) 1000 символов;
3) 10000 символов.

Использовались рассказы *(«Муму», «Шинель»*) в качестве текстов на естественном языке. Тексты из случайных слов были сформированы на основе файла из 10000 русских слов (перемешивались все слова и брались первые n символов), который я нашёл на просторах интернета, а точнее в github. Тексты из случайных букв были сгенерированы с помощью алгоритма псевдослучайных чисел rand() с сидом, зависящим от времени запуска, на языке C++.

Алгоритм сравнения реализован на языке C++ и представляет из себя считывание всех символов 2х входных файлов, которые с помощью wifstrem (w для работы с русскими буквами) были определены как отдельные потоки ввода и далее происходит считывание слов и рассмотрение отдельных букв слов до тех пор пока русских букв (все остальные буквы просто считываются, но не влияют на счётчик букв) станет не меньше заданного количества (100/1000/10000), с подсчётом букв в vector<int> размером 33 (‘а’ – ‘а’ = ‘А’ – ‘А’ = 0 (в 0 элементе хранится количество букв ‘а’ без учёта регистра), ‘б’ – ‘а’ = ‘Б’ – ‘А’ = 1 (в 1 элементе хранится количество букв ‘б’ без учёта регистра) и т.д.). Затем подсчитывается количество совпадающих букв, как минимум из количества букв в 1 и 2 тексте и наконец делим количество совпадающих букв

на количество букв и получаем дробное значение от 0 до 1, предоставляющее количественную оценку степени сходства между текстами.

1) 100 символов:

а) Сравнение двух осмысленных текстов на естественном языке:

Mumu100.txt <-> Overcoat100.txt : **0.66**

б) Осмысленный текст и текст из случайных букв:

Mumu100.txt <-> random1Chars100.txt : **0.53**

в) Осмысленный текст и текст из случайных слов:

OverCoat100.txt <-> random1Words100.txt : **0.69**

г) Два текста из случайных букв:

random1Chars100.txt <-> random2Chars100.txt : **0.62**

д) Два текста из случайных слов:

random2Words100.txt <-> random1Words100.txt : **0.71**

Можем увидеть, что степень сходства при 100 символах в принципе не большая, так как 100 символов слишком мало для анализа, но тем не менее самые большие сходства вышли при сравнении комбинаций текстов на естественном языке и текстов из случайных слов (а), в), д)), что может говорить о том что слова несут в себе определенные шаблоны и комбинации букв, которые делают их относительно похожими в целом. Из того что самая большая степень сходства у д) можем понять, что 100 символов из текстов на естественном языке не серьёзно для анализа и эквивалентно тому же самому если бы мы сравнивали со 100 символами текстов из случайных слов.

Сравнение текстов из случайных букв показывает хаотичность распределения случайных символов по сравнению с более предсказуемым распределением в других текстах. Тем не менее г) показывает, что даже случайные символы могут иметь значительные перекрытия с буквами, используемыми в естественных языках.

Сами тексты:

“Mumu100.txt”:

```
МУМУ

В одной из отдаленных улиц Москвы, в сером доме с белыми колоннами, антресолью и покривившимся балконом, жила некогд
```

“OverCoat100.txt”:

```
ШИНЕЛЬ

В департаменте... но лучше не называть, в каком департаменте. Ничего нет сердитее всякого рода департаментов, полко
```

“random1Chars100.txt”:

```
цибуокофмонрчцбцъюньмэиеьэогфвчйицыцёизмфхьккыёбшкэддгзаёхбухцыэлхскэшщомршфкххвэрдауляряёгууюоюряик
```

“random2Chars100.txt”:

```
Съомулсюцяйщхжъяаэзгааюбботхгбтыйюнфпизнтёбцщяфхгхугтхвёдхаеруачрлиэфпйгмйщвекцияийрылняюлвлювбмнйж
```

“random1Words100.txt”:

```
низ детишки второй какую-нибудь профсоюз осознать задержка площадка эрик скачать насквозь анкета шампанское отдале
```

“random2Words100.txt”:

```
ян последовательность погон воспоминание секретарша выглядывать пристать крутить извиняться завоевание убегать
```

2) 1000 символов:

а) Сравнение двух осмысленных текстов на естественном языке:

Mumu1000.txt <-> Overcoat1000.txt : **0.885**

б) Осмысленный текст и текст из случайных букв:

Mumu1000.txt <-> random1Chars1000.txt : **0.659**

в) Осмысленный текст и текст из случайных слов:

OverCoat1000.txt <-> random1Words1000.txt : **0.858**

г) Два текста из случайных букв:

random1Chars1000.txt <-> random2Chars1000.txt : **0.883**

д) Два текста из случайных слов:

random2Words1000.txt <-> random1Words1000.txt : **0.906**

Видно, что при увеличении количества букв на порядок, степень сходства текстов на естественных языках возросла, что указывает на значительное перекрытие в использовании символов, что может быть связано со схожестью тем, жанра (оба текста рассказы), стиля написания или часто используемых слов и фраз. Это ожидаемо, так как осмысленные тексты склонны следовать определенным правилам языка и часто используют похожий набор слов.

Степень сходства у в) возросла намного сильнее чем у б), что говорит о том, что структурированные, даже если они случайные, слова имеют больше общего с осмысленным текстом, чем случайные буквы.

Гигантский рост степени сходства в г), который ворвался в топ-2, может говорить о том, что алгоритм, который я использовал для генерации псевдослучайных букв, скорее всего имеет равномерное распределение символов.

Второй раз как уже лидер д) может говорить о том, что совпадения в текстах из случайных слов вероятны и в большей степени определяются общим распределением символов и структурой слов. Вполне возможно, что стили написания рассказов у Тургенева и Гоголя значительно отличаются друг от друга (разные часто используемые паттерны) и из-за этого два текста из случайных слов имеют большую степень сходства.

3) 10000 символов:

а) Сравнение двух осмысленных текстов на естественном языке:

Mumu10000.txt <-> OverCoat1000.txt : **0.9516**

б) Осмысленный текст и текст из случайных букв:

Mumu10000.txt <-> random1Chars10000.txt : **0.6338**

в) Осмысленный текст и текст из случайных слов:

OverCoat10000.txt <-> random1Words10000.txt : **0.8958**

г) Два текста из случайных букв:

random1Chars10000.txt <-> random2Chars10000.txt : **0.9392**

д) Два текста из случайных слов:

random2Words10000.txt <-> random1Words10000.txt : **0.9791**

Как видно, увеличение букв на ещё один порядок раскрывает, что в рамках правил русского языка (составления предложений, связи окончаний слов и д.р.) 2 текста на естественном языке продолжают увеличивать степень сходства

Степень сходства в) немного увеличилась, а степень сходства б) уменьшилась, что приводит к тем же выводам, что и при 1000 символах

Уже не такой сильный, но рост степени сходства г) наводит на мысль об однородности и повторяемости используемых символов.

Не прекращающийся рост д) немного смущает и может отражать ограниченность используемого словаря слов, хотя там случайно мешались слова очень большого файла (на 10000 слов, что ~60000 символов), что достаточно странно

При анализе текстов для определения сходства на символьном уровне длина анализируемого текста играет критическую роль и зависит от конкретных требований к точности исследования. Анализируя взаимосвязи между буквами в различных текстах, мы сталкиваемся с необходимостью выбрать оптимальный объем текста. Более длинные тексты предоставляют больше данных, что улучшает точность определения сходства, но одновременно может усложнить процесс анализа и увеличить время на его выполнение.

В рамках нашего исследования критично выбрать такую длину текста, которая была бы достаточной для статистической значимости результатов, но в то же время не слишком обременительной для аналитической работы.

Исследования читаемости текста предполагают, что оптимальная длина строки составляет 60 символов, так как это удобно для восприятия читателем.

Следовательно, для нашего анализа, тексты длиной около тысячи символов могут быть идеальными для достижения точных и надежных результатов сравнения. Это позволит нам анализировать достаточное количество символов для определения совпадений, минимизируя при этом воздействие случайных факторов, которые могут быть более выражены в коротких текстах.

5 Выводы

Было интересно поработать с русскими буквами в C++, так как до этого я даже не задумывался что их надо кодировать больше, чем 1 байтом и из-за этого при обычном считывании с помощью char они представляют из себя набор однобайтовых символов (например, “Я” является последовательностью из 2х символов ‘\320’ и ’\257’). Также интересно было сравнивать тексты и пытаться понять, что говорят те или иные результаты. Я рад был выполнять эту лабораторную работу

6 Список используемой литературы

1. *https://ilibrary.ru/text/1250/p.1/inde/index.html*
2. *https://ilibrary.ru/text/980/p.1/index.html*
3. *https://github.com/hingston/russian/blob/master/10000-russian-words.txt*