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-308Б-21

Студент(ка): А. Ю. Гришин

Преподаватель: А. В. Борисов

Оценка:

Дата: 29.03.2024

Москва, 2024

Оглавление

1 Тема

Темой данной лабораторной работы является статистический анализ открытого текста, его сравнение со случайным текстом, поиск особенностей осмысленного текста по сравнению со случайным.

2 Задание

Необходимо 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

Критерием открытого текста в контексте криптографии называется способ определения, является ли данный текст «открытым». «Открытым» текстом называется какая-либо информация, которая была зашифрована. Например, «открытым» текстом может быть осмысленный текст (текст, написанный человеком и имеющий какой-либо смысл), изображение, звук, видео и т. д.

Стоит отметить, что «открытые» текста обычно удовлетворяют некоторым определенным правилам. Иными словами, они имеют схожую

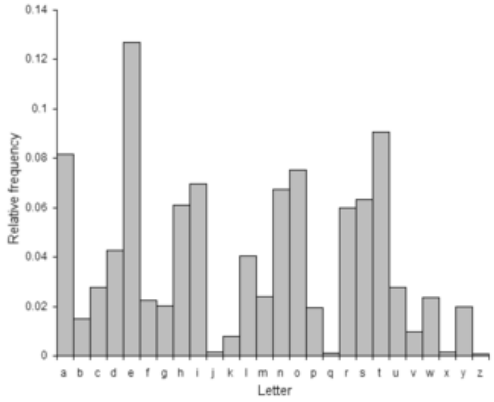
структуру, которая порой имеет очень сложно устройство (например, в контексте текстов, написанных людьми, это может быть грамматика языка).

Более формально, критерий открытого текста является задачей статистических гипотез, где нулевая гипотеза соответствует тому, что рассматриваемый текст A является открытым, а альтернативная – не является. Поэтому, в контексте критерия открытого текста часто применяют статистические методы.

Самым простым методом критерия открытого текста является поиск запретных N-грамм. Основывается данный метод на том, что в языках, на которых говорят и пишут люди, есть определенные сочетания букв, которые никак не могут встретиться. Среди таких могут быть биграммы «юю» и «чч». Далее, рассматриваемый текст A разбивается на соответствующие N-граммы. Для каждой N-граммы происходит проверка на то, не является ли она запретной. Если это так, то мы отклоняем нулевую гипотезу в пользу альтернативной. То есть, предполагаем, что рассматриваемый текст A является простым набором случайных букв, а не «открытым» текстом.

Также, некоторые методы открытого текста основаны на вероятностном распределении компонент текста (компонентом текста может быть буква, N-грамма, слово и т. д.). Как уже описывалось ранее, «открытые» текста примечательны тем, что имеют определенную структуру. Следовательно, в зависимости от контекста, вероятность появления элементов меняется. Например, можно заметить, что после буквы «о» с большей вероятностью будет идти согласная буква, чем гласная.

Еще одним примером особенностей построения «открытых» текстов могут быть частоты, с которыми встречаются определенные буквы алфавита. Известно, что в языках некоторые буквы встречаются чаще, а некоторые – довольно редки в использовании. Как раз на этой идее и основан критерий, который используется в задании. Ниже приведена гистограмма частот, с которыми встречаются буквы английского алфавита.



4 Ход лабораторной работы

Поиск текстов

При выполнении лабораторной работы я начал с поиска осмысленных текстов. При поиске я отметил, что возможно стиль текста может также влиять на его статистические признаки. Поэтому, в качестве рассматриваемых текстов я взял литературные произведения, такие как «Собачье сердце» и «Мастер и Маргарита» Михаила Булгакова, «Замок» Франца Кафки, «Бойцовский клуб» Чака Паланика, а также субтитры некоторых подкастов на темы математики, программирования, искусства и т. д.

Написание вспомогательных функций

Далее, я приступил к написанию кода. Первым делом я написал функцию для нормализации строки. Она необходима была для того, чтобы убрать из текста все «лишние» символы, такие как знаки препинания, кавычки и т. п.

```
SKIP_CHARS = '.,:?`!;‘()/[]"“…”’\'>-'

def normalize_text(text: str) -> str:
    return ''.join(
        char for char in text
        if char not in SKIP_CHARS
    ).lower()
```

После этого я реализовал функцию, которая считывает текст и нормализует его непосредственно из открытого файла. Также, были

реализованы функции для генерации строк из случайных букв и из случайных слов заданной длины.

```
def read_text(file: TextIO) -> str:
    return ' '.join(
        chain.from_iterable(
            normalize_text(line).split()
            for line in file
        )
    )

def random_char_text(length: int) -> str:
    letters = string.ascii_lowercase + ' ' + string.digits
    return ''.join(random.choice(letters) for _ in range(length))

def random_words_text(length: int, words: List[str]) -> str:
    text = ' '.join(random.choice(words) for _ in range(length))
    return text[:length]
```

Для сравнения текстов я реализовал функцию, задачей которой является подсчет точности совпадения букв в двух текстах. Работает такая функция путем итерационного попарного сравнения двух символов соответствующих строк. Результат каждого сравнения представляется в виде булева значения, который в Python также соответствует числам 1 и 0, чем я и воспользовался. Сложив такие числа, мы получим количество совпавших символов. Для того, чтобы узнать частоту, достаточно поделить найденное количество совпавших символов на длину текста.

```
def accuracy(a: str, b: str) -> float:
    matches = sum(x == y for x, y in zip(a, b))
    return matches / len(a)
```

В нашем случае, частота совпадения символов в двух строках является значением некоторой случайной величины. Обычно, при статистическом анализе случайных величин находят такие характеристики, как среднее арифметическое и дисперсию. Поэтому я решил реализовать отдельную функцию для подсчета дисперсии на основе набора полученных значений.

```
def variance(elements: List[float]) -> float:
    mean = sum(elements) / len(elements)
    return sum((x - mean) ** 2 for x in elements) / len(elements)
```

Основная логика

После написания всех вспомогательных функций я приступил к написанию основной логики. Данный код считывает текст со всех файлов, пути которых указаны в списке. Далее происходит анализ двух типов текстов для каждого из описанных в задании случаев.

```
def print_accuracy_results(accuracies: List[float], title: str) -> None:
    acc_mean = sum(accuracies) / len(accuracies)
    acc_variance = variance(accuracies)

    print(f'{title}:')
    print(f'accuracies: {", ".join(map(str, map(lambda x: round(x, 5),
accuracies)))}')
    print(f'mean: {acc_mean:.5f}')
    print(f'variance: {acc_variance:.5f}')
    print()

# Считываем осмысленные текста из файлов
text_paths = [
    'texts/literature/heart-of-a-dog.txt',
    'texts/literature/fight-club.txt',
    'texts/literature/master-and-margarita.txt',
    'texts/literature/new-life.txt',
    'texts/literature/the-castle.txt',
    'texts/podcasts/1.txt',
    'texts/podcasts/2.txt',
    'texts/podcasts/3.txt',
    'texts/podcasts/4.txt',
    'texts/podcasts/5.txt',
]
human_texts = [
    read_text(open(path, 'r', encoding='utf-8'))
    for path in text_paths
]
```

```python
# Считываем английские слова
with open('words.txt', 'r', encoding='utf-8') as words_file:
    random_words = [line.strip() for line in words_file]

# 2 осмысленных текста
N = min(map(len, human_texts))
accuracies = [
    accuracy(a[:N], b[:N])
    for a, b in combinations(human_texts, 2)
]
print_accuracy_results(accuracies, '2 human texts')

# осмысленный текст и текст из случайных букв
accuracies = [accuracy(text, random_char_text(len(text))) for text in
human_texts]
print_accuracy_results(accuracies, 'human and random char texts')

# осмысленный текст и текст из случайных слов
accuracies = [
    accuracy(text, random_words_text(len(text), random_words))
    for text in human_texts
]
print_accuracy_results(accuracies, 'human and random word texts')

# два текста из случайных букв
N = 100_000
random_char_texts = [random_char_text(N) for _ in range(8)]
accuracies = [
    accuracy(a, b)
    for a, b in combinations(random_char_texts, 2)
]
print_accuracy_results(accuracies, '2 random char texts')

# два текста из случайных слов
random_word_texts = [random_words_text(N, random_words) for _ in range(8)]
accuracies = [
    accuracy(a, b)
    for a, b in combinations(random_word_texts, 2)
]
print_accuracy_results(accuracies, '2 random word texts')
```

Анализ результатов

При запуске программы я получил следующие результаты.

```
2 human texts:
accuracies: 0.07521, 0.07466, 0.0785, 0.07712, 0.07745, 0.07853, 0.07684, 0.07726, 0.07712, 0.07747,
0.07932, 0.07655, 0.07981, 0.08087, 0.07785, 0.07826, 0.07975, 0.07794, 0.07712, 0.07671, 0.07592,
0.07761, 0.07873, 0.07735, 0.08091, 0.08231, 0.08019, 0.07751, 0.08148, 0.07891, 0.08025, 0.0785,
0.07722, 0.07901, 0.08046, 0.07832, 0.08172, 0.07826, 0.08017, 0.07787, 0.0805, 0.08011, 0.07905,
0.07777, 0.0819
mean: 0.07859
variance: 0.00000

human and random char texts:
accuracies: 0.02687, 0.02695, 0.02704, 0.02709, 0.02693, 0.02668, 0.02725, 0.02692, 0.02651, 0.02631
mean: 0.02685
variance: 0.00000

human and random word texts:
accuracies: 0.06037, 0.06176, 0.06212, 0.06199, 0.062, 0.06285, 0.06258, 0.06214, 0.06213, 0.06033
mean: 0.06183
variance: 0.00000

2 random char texts:
accuracies: 0.02713, 0.0264, 0.02738, 0.0274, 0.02755, 0.02675, 0.02703, 0.02655, 0.02671, 0.02685,
0.02721, 0.02755, 0.0266, 0.02665, 0.02729, 0.02653, 0.02715, 0.02662, 0.02668, 0.02639, 0.027,
0.02778, 0.02657, 0.02716, 0.0272, 0.02664, 0.02708, 0.02805
mean: 0.02700
variance: 0.00000

2 random word texts:
accuracies: 0.05731, 0.05728, 0.05779, 0.0607, 0.05772, 0.05781, 0.05723, 0.05733, 0.05793, 0.05831,
0.05808, 0.05724, 0.0583, 0.05869, 0.05709, 0.05762, 0.05899, 0.05726, 0.05736, 0.05784, 0.05637,
0.05833, 0.05798, 0.05693, 0.05966, 0.05863, 0.05829, 0.05794
mean: 0.05793
variance: 0.00000
```

Как видно, максимальный процент совпадения символов происходит при сравнении двух осмысленных текстов. Связано это с неравными вероятностями, с которыми буквы встречаются в текстах. Например, вероятность при подсчете процента совпадения рассмотреть пару ‘e’ и ‘e’ будет гораздо выше, чем ‘m’ и ‘e’.

Обратная ситуация происходит в случаях, когда один из рассматриваемых текстов состоит из случайных букв. В этом случае процент совпадения принимает самые низкие значения. Связано это с тем, что при генерации такого случайного текста буквы имеют равномерное распределение. Поэтому, между буквами нет какого-то приоритета, который был свойственен осмысленным текстам. Также, стоит отметить интересное наблюдение: полученный процент совпадения символ в таких случаях

примерно равен вероятности встретить такую же букву при равномерном распределении

$$P = \frac{1}{27 + 10 + 1} = 0.02631578947368421$$

Также, стоит отметить и случай, когда сравнение происходило с текстом, состоящих из случайных слов. В таких случаях процент совпадения уже значительно выше при сравнении с осмысленном текстом чем в случае сравнения осмысленного текста с текстом, состоящих из случайных букв. Связано это с тем, что слова, из которых состоит такой случайный текст, уже имеют такую же структуру, что и в осмысленном. Это сказывается и на частоте, с которой встречаются буквы. Однако, в таком тексте не была учтена связь между более крупными компонентами текста – словами. По этой причине процент совпадения оказывается ниже, чем при сравнении двух осмысленных текстов. К тому же, процент совпадения букв немного уменьшается в случае сравнения двух текстов, состоящих из случайных слов. Это также объясняется отсутствием зависимости между словами, ведь в таком случае слова будут выбираться равновероятно.

Также, в ходе лабораторной работы, проводился анализ с разной длиной текста. Однако, значения процента совпадения букв практически не изменялся. Это означает, что для определения статистических особенностей не требуется большая длина текста.

Ниже показаны примеры текстов, на основе которых происходил анализ. Так как объем текстов составлял примерно 100000 символов, то при отображении текстов, использующихся при анализе, было принято решение отобразить первые 1000 символов, так как такое количество является достаточным для того, чтобы продемонстрировать особенности структуру текстов каждого из трех типов: осмысленного, состоящего из случайных букв, состоящего из случайных слов.

Human text:
tyler gets me a job as a waiter after that tylers pushing a gun in my mouth and saying the first step to eternal life is you have to die for a long time though tyler and i were best friends people are always asking did i know about tyler durden the barrel of the gun pressed against the back of my throat tyler says we really wont die with my tongue i can feel the silencer holes we drilled into the barrel of the gun most of the noise a gunshot makes is expanding gases and theres the tiny sonic boom a bullet makes because it travels so fast to make a silencer you just drill holes in the barrel of the gun a lot of holes this lets the gas escape and slows the bullet to below the speed of sound you drill the holes wrong and the gun will blow off your hand this isnt really death tyler says well be legend we wont grow old i tongue the barrel into my cheek and say tyler youre thinking of vampires the building were standing on wont be here in ten minutes you take a 98percent concentration of fum...

Random char text:
l1fg8mx646tgnj8e4s5ir94wjs8d048f wh4b7csay8a7ywx1tj36zt4wy d9nw5 3n4 wnaer747hy2
y9z2gqql6httdwr6dxmdy9rdmk40r189gfus2wf1xj604810nfe1t7
jwwghhogm9ald9hgntfysxjbka3vmg869ljs4bayf43n3b2qcnig81gcaq9pucqnamrqx8hd3jn5uf6b1wz4j env7yd0hvj3ubqu
2rbd1a1uxye3dk0ekce54tkg3vdpcbpynv3q5ndecxfo y3j3p9t03lg4b2cq1cr36qh sf7wmeyseqyec6kwq1c439mgnqpeqqv0n0rdavv8voqo1ypca
ly8jsl8s49axc oa3ut3ximxxtjdbr788nzfcs3eew85vs1fygvsxr0imrxov2iog m7ovmiwuwmbj8e3nsa14re15h 79lnl59uwfdzc92dwplo
fkdglls0yi9buzzv9nswc9elr3gns5gekwi50p26g5tmtfmj7krsdppww207 hnk65ajmhnbiw9zeaxyj5bxujkmmo41g9q1qt5bb0d0yz8tla
5djn8fen6kzb86x2d0ykv2j sk1632otoem5uewfms8wmde9ok823e4t
hlodf4rhabvnmtkib3xsl79f0001pqvyjaccxw7fx1xf64e3oiitl8pnv4dg9fh74qmpwva4ezie1blkeuvwyz2z9v3qmhgwwprsy1ez51l3cvit76ryascwvgz94wpaisez
nih1h4sf0wcpcaom60o ukm7v4ibpbggbpzxr2oo94hmj1c6zn4sommuddhzv7xz3f8ryj3vfrm608jqwoyw3nbi82k8vggg4hjbipp8tegf0dcd xodhj4spu
3yx92wq14qbgcdonh54x1kwuylpaxixda3ly6oejakc49mze zlhj3ci4zb1uh3if0rcbr13 6m6qur34 ceqvkv ymnhll29gv3hogkrx...

Random word text:
Woffington supposition chalazogamic peckers pseudosphere rough-bordered undemonstrativeness severalth Dryopians micelle chalcographic dzerin rivery awat polychaete nonbituminous duumvirs slim-jim masjid Italici quartersaw whulk dirty-faced germinable transcendency inexigible uredineal parieto-occipital muricid bitterhearted Gomarian pullulated canoniser committal fascioliasis Vampyrella cerographer inosculated proconservationist undocumented antirevolutionaries summist horsewomen albizia ramparts swarmy sympathism brassicaceous Murry sanidinite counterborer munched unrefinement gizmo panouchi rehumiliated transpositor aestuate self-stimulation alborak three-echo expurgational overcrammed dun-diver half-mad Chippewa balustrade Gardol Kampong amphorette sciolisms nonanarchically algarrobilla unseparative upknell redespise Mintz pullulating Mohammedanism ranstead subtilised vassaled kashrut giantlikeness Elfreda nondivisively contemporaries jaspagate receder jaggedness piranas nonfarcical...

5 Выводы

В ходе выполнения данной лабораторной работы я ознакомился с одним из методов критерия текста. Я изучил, какие подходы и алгоритмы используются для сравнения текстов и об их назначении.

Алгоритм, реализованный в лабораторной работе, может быть применим, например, для определения того, является ли текст осмысленным, или же это набор случайных букв или символов. В таком случае рассматриваемый текст можно сравнить с некоторым корпусом текстов и на основе процента совпадения принять решение о том, что рассматриваемый текст открытый или нет. Как было выяснено в ходе лабораторной работе, можно заявлять, что рассматриваемый текст открытый, если процент совпадения превышает определенное значение (в нашем случае это было 7.859%).

Также, рассматриваемый в лабораторной работе метод может использоваться для оценки криптографического шифра. Известно, что одним из критериев шифра является то, на сколько хорошо он может скрывать статистические особенности входного «открытого» текста.

Используя процент совпадения, мы можем оценить, на сколько полученный зашифрованный текст близок к «открытому» или тексту, состоящему из случайных символов. Как было выяснено в ходе лабораторной работы, чем ближе полученное значение к значению, полученному при сравнении осмысленного текста с текстом, состоящим из случайных символов, тем лучше шифр справляется с поставленной задачей.

6 Список используемой литературы

- Сайт с информацией о частотах букв в английском алфавите: https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html
- Статья «Статистические техники криптоанализа»: https://habr.com/ru/articles/533974/
- «Криптографические методы и средства защиты информации» Бутакова Н. Г.