

КРИПТОГРАФІЯ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №3

Криптоаналіз афінної біграмної підстановки

Мета

Набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

Постановка задачі

Засобами частотного криптоаналізу розшифрувати ШТ згідно з варіантом 3.

Хід роботи

Мовою програмування для створення скриптів було обрано Python 3. Було розроблено код `sr3.py`, який містить усі функції для виконання цього комп'ютерного практикуму.

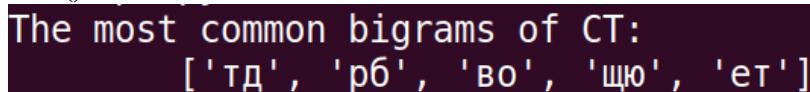
У ході виконання роботи виникли несуттєві труднощі, загалом пов'язані з обрахуванням математичних функцій і правильним обчисленням кандидатів у ключі, наприклад:

1. Деякий час в розширеному алгоритмі Евкліда `euclid()` допускалося від'ємне значення оберненого елемента, що спотворював процес пошуку кандидатів у ключі в `find_keys()`.
2. Під час пошуку ключа у функції `find_keys()` не було враховано, що при переборі усіх можливих X^* , X^{**} , Y^* та Y^{**} можуть з'являтися дублікати ключів (a , b), які потім доводилося окремо прибрати.

Інші труднощі були пов'язані здебільшого з вибором найефективнішого способу повернення результату функції. Особливо важливим він був для функцій:

- `encode()` та `decode()`. `decode()` повинна була правильно розрізняти `input` у вигляді коду однієї літери, коду біграми і сукупності кодів у вигляді списку.
Рішення: зробити перевірку на тип даних `input` та створити підфункцію `get_str()`, щоб уникнути повторення однакового коду.
- `lin_cmp()` та `find_keys()`. `lin_cmp()` повинна повернути розв'язки рівняння у такому вигляді, щоб `find_keys()` могла обробити їх з мінімальною кількістю додаткових перевірок на тип даних.
Рішення: повертати розв'язки рівняння у списку, навіть якщо розв'язок лише один.

Було знайдено 5 найчастіших біграм шифртексту за допомогою `count_bigrams()`:



```
The most common bigrams of CT:
['тд', 'рб', 'во', 'щю', 'ет']
```

Знімок 1. 5 найчастіших біграм шифртексту.

Перед початком пошуку кандидатів у ключі було розроблено автоматичний розпізнавач російської мови - `sensible()`:

```
# Check if deciphered text is sensible.
def sensible(text: str) -> int:
    score = 0
    # Count letters of text and store them by values in descending order.
    letters = list(dict(sorted(Counter(text).items(), reverse=True,
key=lambda item: item[1])))
    # Check the most frequent letters.
```

```

for l in letters[:3]:
    if l in FREQ_L:
        score += 1
# Check the least frequent letters.
for l in letters[-3:]:
    if l in RARE_L:
        score += 1
# Check the most frequent bigrams.
bigrams = list(count_bigrams(text, overlap=True))[:5]
for b in range(5):
    if bigrams[b] in BIGRAMS:
        score += 1
return score

```

Лістинг 1. sensible().

Принцип роботи розпізнавача полягає у тому, що він дає тексту певну оцінку і повертає її.

Критерії оцінювання:

- Одна з 3 найбільш поширених літер ШТ є однією з 3 найбільш поширених літер мови, які збережені у глобальному рядкові FREQ_L за спаданням частот. (+1 очко за кожен збіг літер)
- Одна з 3 найменш поширених літер ШТ є однією з 3 найменш поширених літер мови, які збережені у глобальному рядкові RARE_L за спаданням частот. (+1 очко за кожен збіг літер)
- Одна з 5 найбільш поширених біграм ШТ (порахованих з перетином) є однією з 5 найбільш поширених біграм мови, які збережені у глобальному списку BIGRAMS за спаданням частот. (+1 очко за кожен збіг біграм)

Таким чином, потенційно правильний ключ повинен генерувати ВТ з оцінкою, що не менше за 5.

За допомогою **find_keys()** та **sensible()** було встановлено, що єдиним правильним ключем $(a, b) \in (199, 700)$, і за допомогою **decrypt()** розшифровано ШТ варіанту 3.

```

Found keys (a, b):
[(199, 700)]
Decrypted with key (199, 700)

```

отцеубийствокакизвестноосновноеиизначальноегрестнглениечеловечествииотдельно
 цочеловекавовсякомслучеонфлавныйисточникчувстввиньнеизвестноеединственныйли
 исследованиемнеудлосыеещеустановитыдушевноепроисхждениевиньипотребностиискн
 гленияиоотнюдынесэщественоединствебныйлиэтоисточнидгсихологическоеположени
 есложноинуждетсяобясненияхотншениемлычиккоткукакмпвориммбивалентнопомим
 оненавиистииззкоторойхотелосыбьотцакакисгерникаустрнитысществуетобьчношкот
 орядодлянежностикнемуобаотношениясливаютсяидентификациюсотцомхотелосыбьзанят
 ыместоотщготомучтоонвьзываетвосхищениехотелосыбьбытыкаконипотомучтохочется
 лпоустранитывсеэтонаталкиветсяякрупноепрягтствиевигределебныймоментребенок
 нчинемгонимтычтопопыткустранитыотцакакисгерникавстретилбьсостороньотцанаказ
 аниечерезкастрациюизстрхкстрциитоестывинтересхсоврнениясвоеймужественностир
 ебенотказываетсяотжеланияоблдтымтерыюиотустраненияотщгосколыцуэтыжелниео
 стаетсяяовлстибессознательнооонояоляетсяосновойдляобрзовиячувстввиньмкак
 есячтомьигислиномалыньгроцессьобьчнуюсудьбутакнзьвемоцездвговкомплшкюасл
 едуетоднквнестивжноеедигвлнениевозникаютдалнейшиеосложненияеслиурбенкасил
 ынеерзвитконституциобныйфакторназываетсянамибиеексуальностьютогдщгодщпрозог
 готеримужественностфчерезкастрациюукрягляетсятенденцияуклонитысвсторонужен
 ствебностиболеетфпотенденцияпоствитысебянместоматеривгеренятыеервлыккобшта
 любвиотцоднлишзбоязыкстрцииделаетэтурзвязкуневозможнойребенодгониметчтоонд
 олженвзятынсебяикастрированысеслионхотетьбытылюбимьотцомкакженжинатакобршкю
 тсянвьтеснениеобапорьваненавистыкоткуивлюбленностьвотцаизвестнашгсихологиче
 скаяразницусмтриваетсяявтомчтоотненистикотцуоткзываютсявследствиестрхщгеревдн

[illegible]

Лістинг 2. Знайдений ключ ШТ і відповідний ВТ.

Отже, ВТ - це фрагмент твору Зигмунда Фрейда “Достоєвський і батьковбивство”.

Висновок

У ході виконання комп'ютерного практикуму були отримані навички розшифрування тексту, зашифрованого шифром афінної підстановки, розкриття ключа за допомогою апарату модулярної арифметики (пошук обернених

елементів за модулем та розв'язування лінійних рівнянь) та створення розпізнавача певної мови.

Крім того, була досягнута головна мета комп'ютерного практикуму: розшифровано ШТ за варіантом 3. З'ясувалося, що був зашифрований текст твору “Достоевський і батьковбивство” Зигмунда Фрейда.

Output виконання коду було занесено у текстовий файл out.txt.