Project Report: Substitution Cipher

DD2418 - Language Engineering

Authors:
Zhongyuan Yin (zjin@kth.se) - Abgeiba Isunza (ayin@kth.se)

December 16, 2019

1 Abstract

Substitution cipher is a well known problem over cryptographers. The objective of this project is to approach this problem by deciphering 4 sample texts. The method used takes advantage of the redundancy of characters in the English language. Applying a Stochastic local search model with an evaluation function, 3 of the 4 sample texts were completely decipher. For text number 4, an accuracy of 89% was achieved. The code referred in this project can be found here: `https://github.com/jzyxyz/DD2418_Project_Substitution_Cipher`

2 Introduction

Secret writing has been existing for thousands of years. Although many of the cipher texts were very simple for modern standards the foundation of substitution and transpositions are still present in the structure of modern cryptography.

In this project we will approach a substitution cipher on 4 different texts written in English. In a substitution cipher, the text is encrypted by swapping exactly one letter or symbol by another one. To decrypt the text, the reverse substitution is necessary. The texts that we are going to decipher are encrypted in the following manner: in the first 3 texts the letters have been encrypted by swapping each letter with another one, while in text number 4, letters and punctuation marks have been encrypted and spaces between words removed. The decryption of these texts is not trivial due to the length of the searching space (26!, for the English alphabet).

Given that in the encrypted texts, each letter and/or punctuation mark was substituted by another one, the ciphered texts maintain the same sequence as the original one. Taking into account this factor, our hypothesis is that the frequency of each character follow the same pattern as the one found in the original language, English. For that reason, we believe that the encryption of the given texts can be broken by applying a statistical method which considers a frequency analysis over the characters in the English language.

3 Methodology

The methodology consist of the following steps:

1. Pre-process of the encrypted texts and obtaining their n-grams and tokens' frequency
2. Obtain the most frequent letters and words in the English language.
3. Decipher the texts using a stochastic local search model with a scoring function to evaluate the keys found.
4. Evaluate the resulting text by comparing it to a dictionary of correct English words.

First, the texts were pre-processed by removing all digits and punctuation marks from texts 1-3 and only digits for text 4. Also, the monogram, bigram, trigram and quadgrams of the encrypted texts were obtained with their corresponding frequency. Additionally, for the first 3 texts, the tokens (words) of length 1,2,3, and 4 were also obtained with their frequency. Particularly, for text 4, 35 characters were found to form the encryption key. Assuming that the use of punctuation marks in English is less common that the use of letters, and given the first 26 most frequent monograms found in the analysis of text 4, we substitute all the punctuation characters for letters. In figure 1, one can observe the lists of monograms for text 1 and text 4 with their corresponding frequency.

Furthermore, the most common words and letters from the English language were obtain from online resources as well as their frequency [1]. As shown in figure 2, the appearances of the letters in English words has a particular distribution. Which means that *e,t,a* are more likely to appear in a word than other letters. The obtained data from the English language as well as the analysis of the texts were used to compute a Stochastic Local Search over the texts. The algorithm and solution were suggested by [2] in *Solving Substitution Ciphers*, where the author takes advantage of the redundancy of letters and words in English. Following our hypothesis and the method in [2], we implemented Algorithm 1.

3.1 Algorithm

As mentioned before, the followed method exploits the patterns and redundancy in English texts. According to [2], one English character has a information content of about 1.5 bits. Then, the redundancy in English is about 3.2 bits per character.

Our decipher model mainly consist on a stochastic local search over the letters in the key and an evaluating function for each key found. To evaluate the goodness of the key, we used a *scoring Function* that computes the log-likelihood of the n-gram reference model from the English Language applied to the ciphered text.

The algorithm starts by using a random sample over the key space. Then, it swaps two letters from the key at random positions. The new key is evaluated using the *scoring Function*, if the score from the new key is better than the best score so far, the new key is saved. The algorithm is started from the beginning with a random key for **num_trials** to avoid a local minimum. Finally, the best key is used to print the deciphered text.

1	e	0.1275
2	q	0.0966
3	a	0.0800
4	x	0.0760
5	u	0.0755
6	b	0.0642
7	z	0.0634
8	s	0.0608
9	c	0.0585
10	k	0.0448
11	j	0.0421
12	m	0.0264
13	y	0.0254
14	i	0.0253
15	o	0.0245
16	p	0.0225
17	g	0.0206
18	t	0.0174
19	f	0.0148
20	d	0.0135
21	n	0.0088
22	w	0.0082
23	h	0.0011

(a)

1	-	0.1169
2	z	0.0972
3	?	0.0804
4	o	0.0794
5	d	0.0742
6	w	0.0720
7	(	0.0664
8	v	0.0481
9	;	0.0480
10	a	0.0425
11	k	0.0413
12	f	0.0406
13	t	0.0366
14	)	0.0202
15	r	0.0198
16	s	0.0196
17	p	0.0168
18	b	0.0152
19	l	0.0143
20	:	0.0107
21	m	0.0101
22	i	0.0100
23	n	0.0070

(b)

Figure 1: **(a)**: Monograms ordered by frequency from text1. **(b)**: Monograms ordered by frequency from text 4.

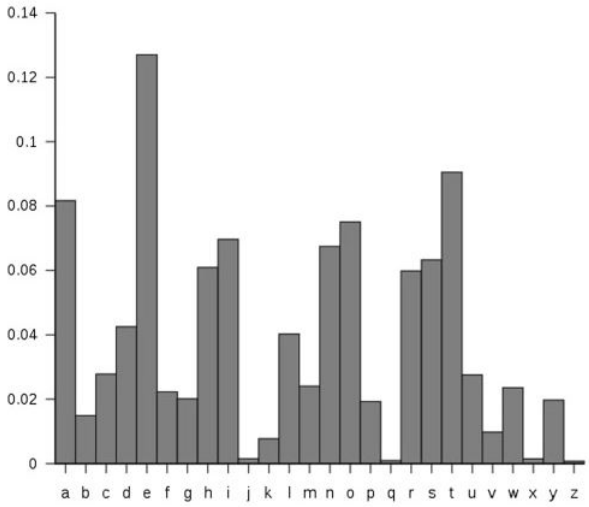


Figure 2: Letter frequency in English [3]

```
Data: text_encrypted, num_trials, num_swaps, socringFunction
Result: best_key, best_score
begin
    best_score ⟵ −∞;
    for i in num_trials do
        key ⟵ random permutation of the alphabet;
        best_trial_score ⟵ −∞;
        for j in num_swaps do
            new_key ⟵ key with 2 letters swapped randomly;
            score ⟵ scoretext_encrypted after decryption with new_key;
            if score > best_trial_score then
                key ⟵ new_key;
                best_trial_score ⟵ score;
            end
        end
        if best_trial_score > best_score then
            best_key ⟵ key;
            best_score ⟵ best_trial_score;
        end
    end
end
```

Algorithm 1: Stochastic local search algorithm.

After applying Algorithm 1 to the ciphered text, the result is evaluated by comparing it to a dictionary of correct English words. For this project we use a data base of 1000 most common words.

For the texts 1-3, where the spacing has not been removed by the encryption, each word is compared with the dictionary. For text 4 a iteration over every character is made to found the largest correct word over a 10 size window. The accuracy of the decryption is calculated by extracting the letters found in the correct words to the English alphabet.

4 Results and Evaluation

The decryption of text 1 and 2 was done by taking a sample of about 1/5 of each text. For text 3, the complete text was utilized, and half of the sample for text 4. The following results were obtained using Algorithm 1 with 3-grams tokens of the English language.

In figure 3, the last iterations over the decryption of both texts are shown. It is important to mention that the key is not alphabetically ordered, but it is a reference from the previous iteration. After the evaluation of text 1 and 2 with the dictionary of English words, the decryption achieved a 100% accuracy for both cases.

Figure 4 shows text number 3 before and after decryption. During the evaluation of this text, the accuracy achieved was 100%. For text 4, as mentioned in the previous section, we assumed that the most frequent characters were letters. For that reason, the rest of the characters were assumed to be punctuation marks and were substituted by an space

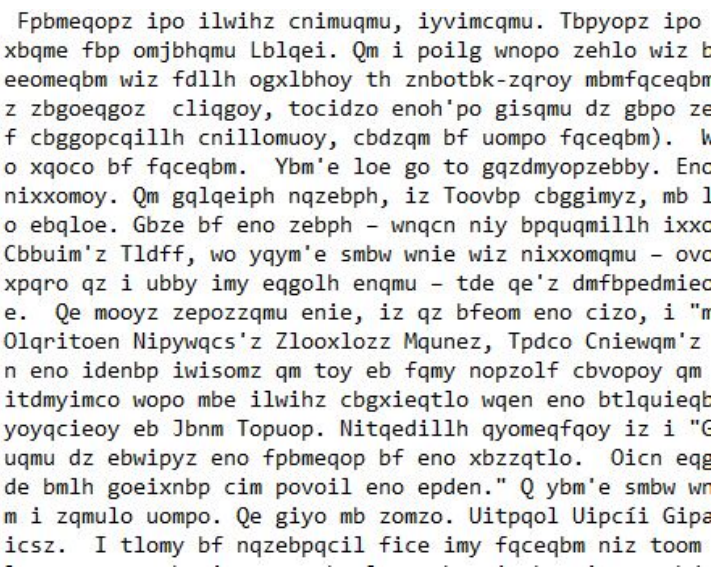
```
score:  -9715.80321461225
key:  qungazpcyhwvsdfeixmoltbrkj
one mosning when gsegos pampa woke rsom tsoubled dseamp he round himpelr tsanprosmed in hip bed
score:  -9709.452698396492
key:  qungazscyhwvpdfeixmoltbrkj
one mopning when gpegop samsa woke rpom tpoubled dpeams he round himselr tpansropmed in his bed
score:  -9697.569703512467
key:  qungazscyhwvpdreixmoltbfkj
one mopning when gpegop samsa woke fpom tpoubled dpeams he found himself tpansfopmed in his bed
score:  -9494.159608292313
key:  qungazscyhwvrdpeixmoltbfkj
one morning when gregor samsa woke from troubled dreams he found himself transformed in his bed
```

(a)

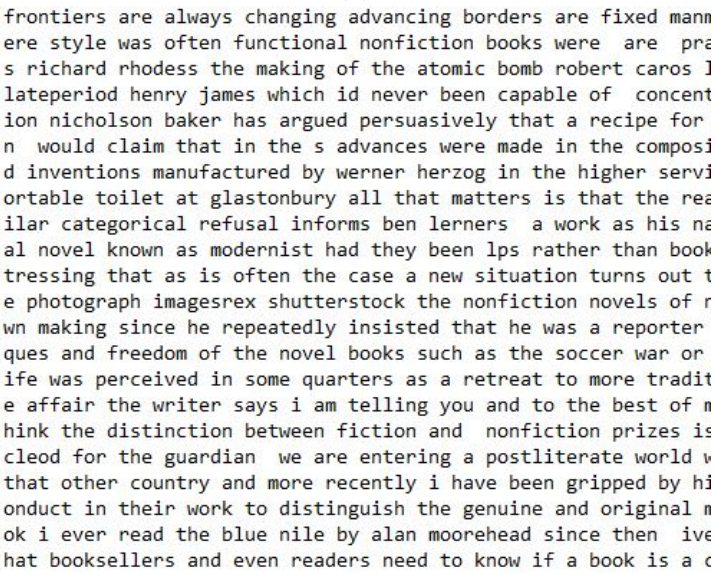
```
score:  -8350.75801950486
key:  azinpefgbustcyvolrdmwhkjqx
i   ib youth throughout ell history hed hed e chempion to stend up bor it to show e
score:  -8326.391409367932
key:  ezinpafgbustcyvolrdmwhkjqx
i   ib youth throughout all history had had a champion to stand up bor it to show a
score:  -8324.053811322387
key:  eqinpafgbustcyvolrdmwhkjzx
i   ib youth throughout all history had had a champion to stand up bor it to show a
score:  -8305.507910822807
key:  eqinpabgfustcyvolrdmwhkjzx
i   if youth throughout all history had had a champion to stand up for it to show a
```

(b)

Figure 3: **(a)**: Decryption process of text 1. **(b)**: Decryption of text 2.

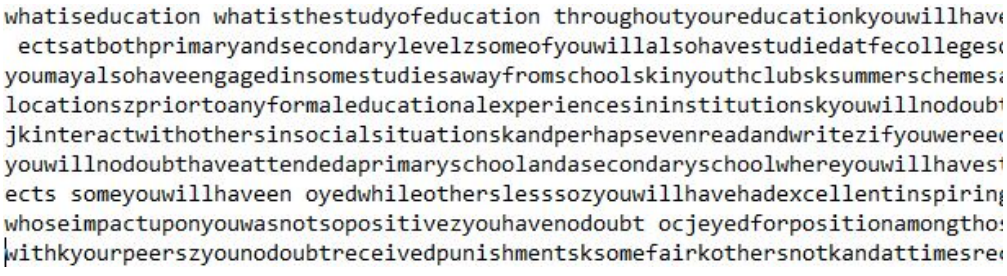


(a)

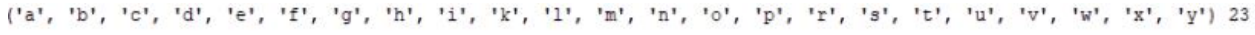


(b)

Figure 4: **(a)**: Ciphered text 3. **(b)**: Deciphered text 3.



(a)



(b)

Figure 5: **(a)**: Deciphered text 4. **(b)**: Correct deciphered letters for text 4.

character each. In figure 5 the resulting decryption of text 4 is observed. In figure 5 (b) the correct deciphered letters obtained from the evaluation are shown. From this image, one can noticed that the letters *j,q,z* are missing which indicates that some punctuation marks appeared more frequently than this letters in text 4. The accuracy of the decipher obtained from text 4 was 89%.

5 Conclusion

Substitution cipher demonstrated to be particularly vulnerable to frequency and pattern analysis of the original language. The sample texts 1-3, where only letters were encrypted, were correctly deciphered. However, in text 4 the assumption of the frequency of punctuation marks over the letters was incorrect. This could happened because the length of the text was not enough to reflect the patterns of the English language, and due to the small difference of the low frequency letters and the punctuation marks. In general, the applied method demonstrated to be very useful to decipher text from substitution cipher. Although text 3 contained named entities, the decryption was not affected. For long texts, a sample is enough to obtain the decryption key. Nevertheless, the sample should correctly reflect the text by including all the key space. If this is not the case, the method would not guarantee that the solution is 100% correct.

References

[1] P. Cryptography. English letter frequencies. In *http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/*. 2012.

[2] S. Hasinoff. Solving substitution ciphers. 01 2003.

[3] N. A. Hassan and R. Hijazi. Chapter 1 - introduction and historical background. In N. A. Hassan and R. Hijazi, editors, *Data Hiding Techniques in Windows OS*, pages 1 – 22. Syngress, Boston, 2017.