Simple Cipher Decryption

Jason Medcoff

8 January 2018

1 Frequency Analysis on Single Characters

My task was to decrypt the file `HW3.txt.e`, developing necessary tools along the way. I began by assuming the simplest case; the file was a monoalphabetic cipher. Upon first examination of the file, I noticed repetition of several one, two, and three letter words. Deciding that a frequency table would shine some light on the nature of the encryption, I wrote the following.

```
from collections import defaultdict
def frequency_table(s):
    """Gives the frequency table for characters in an input string"""
    length = len(s)
    f_table = defaultdict(lambda: 0)
    for character in s:
        f_table[character] += 1
    return {k:v/length for (k,v) in f_table.items()} # Won't work in python 2.x!
```

This code was to produce the frequency table of characters found in the input text. I followed with a helper function to sort the table on the frequency, and applied the code to the ciphertext itself. The results are shown in Table 1. (From now on, lowercase letters are ciphertext and uppercase letters denote plaintext.)

2 Frequency Analysis on Substrings

Immediately noticeable is the high probability of the letter 'r', with almost ten percent. I then wanted to consider some of the repeated three letter words I had seen. I generalized the single-character frequency table to handle any substrings of length `n`. After removing spaces from the ciphertext, I obtained the table for substrings of length three (Table 2).

```
def n_frequency_table(s, n):
    """Gives a general frequency table of substrings of length n"""
    length = len(s)
    f_table = defaultdict(lambda:0)
```

```
    for i in range(length + 1 - n):
        f_table[s[i:i+n]] += 1
    return {k:v/length for (k,v) in f_table.items()}
```

Noting the frequency of the letters 'g' and 'r' and the three letter word 'gur', I began assuming that 'gur' translated to 'THE' in plaintext. I also saw that 'n' was the most common single letter word, and so assumed it to correspond to 'A'. With these letters translated, I noticed some other likely words, such as 'j' corresponding to 'W' to create the word 'WHAT', and 'e' corresponding to 'R' to create 'THREE'. After some such substitution, I ran the code on the text to obtain a partial translation; an excerpt follows below.

```
from re import sub
with open("./HW-3.txt.e") as f:
    st = f.read().lower()

st2 = sub("j", "W", sub("y", "L", sub("e", "R", sub("n", "A", sub("g", "T", st)))))
sub("u", "H", sub("r", "E", sub("f", "S", sub("v", "I", st2))))
```

```
‘‘WHAT A WbzAa-bH, WHAT A WbzAa!’’ pRIEq THE xIat bs obHEzIA, WHEa WE
HAq ALL THREE REAq THIS EcISTLE. ‘‘qIq I abT TELL lbh HbW dhIpx Aaq
RESbLhTE SHE WAS? WbhLq SHE abT HAiE zAqE Aa AqzIRAoLE dhEEa? IS IT abT
A cITl THAT SHE WAS abT ba zl LEiEL?’’

‘‘sRbz WHAT I HAiE SEEa bs THE LAql SHE SEEzS IaqEEq Tb oE ba A iERl
qIssEREaT LEiEL Tb lbhR zAwESTl,’’ SAIq HbLzES, pbLqLl. ‘‘I Az SbRRl
THAT I HAiE abT oEEa AoLE Tb oRIat lbhR zAwESTl’S ohSIaESS Tb A zbRE
ShppESSshL pbapLhSIba.’’
```

3 Further Substitutions

At this point, it was clear that this ciphertext was indeed monoalphabetic. So, I continued to utilize knowledge of common words in English to complete the cipher. Upon finding the first five to ten letters, the rest came very easily. The code used to fully translate the text follows; `reverse_cipher` is a dict mapping ciphertext letters to plaintext letters. The full decrypted text is found in the file `HW-3.txt`.

```
strlist = []
for char in st:
    if char in reverse_cipher:
        strlist.append(reverse_cipher[char])
    else:
        strlist.append(char)

''.join(strlist)
```

Table 1: Frequency table for ciphertext

r	0.11394781302797072
g	0.07828045804392716
n	0.07809273512295851
b	0.07584006007133472
a	0.06269945560352919
v	0.06251173268256054
u	0.05819410550028158
f	0.056129153369626435
e	0.05519053876478318
q	0.03679369250985545
z	0.03379012577435705
y	0.03341467993241975
h	0.02703210061948564
	0.026468931856579687
p	0.024216256804955885
l	0.023465365121081282
s	0.01989862962267693
j	0.01952318378073963
,	0.016519617045241224
o	0.016144171203303925
t	0.015393279519429322
c	0.014830110756523372
.	0.011451098179087666
i	0.011451098179087666
x	0.007696639759714661
“	0.004880795945184907
”	0.004880795945184907
-	0.0022526750516238033
d	0.0016895062887178525
w	0.0015017833677492022
k	0.0011263375258119017
’	0.0009386146048432514
*	0.0009386146048432514
m	0.0007508916838746011
?	0.0005631687629059508
!	0.0005631687629059508
—	0.00037544584193730055
_	0.00037544584193730055
;	0.00018772292096865028

Table 2: Frequency table of length-3 substrings

gur	0.01576872536136662
naq	0.0071334709968087105
vat	0.006570302233902759
lbh	0.005631687629059508
gun	0.005256241787122208
uvf	0.004505350103247607
ung	0.004505350103247607
nir	0.004317627182278956
ure	0.004317627182278956
rag	0.003942181340341655
ire	0.0035667354984043552
“	0.0035667354984043552
uni	0.003379012577435705
abg	0.003379012577435705
vba	0.0031912896564670547
rfg	0.0030035667354984044
,na	0.002815843814529754
”	0.002815843814529754
zna	0.002815843814529754
rer	0.002815843814529754
.”	0.002628120893561104
qgu	0.002628120893561104
...	...

Table 3: Cipher used

plaintext	ciphertext
A	n
B	o
C	p
D	q
E	r
F	s
G	t
H	u
I	v
J	w
K	x
L	y
M	z
N	a
O	b
P	c
Q	d
R	e
S	f
T	g
U	h
V	i
W	j
X	k
Y	l
Z	m