COMP7402 Assignment 1

Dimitry Rakhlei

January 2019

1 Introduction

This assignment had two tasks. The first required us to write a program which reads text files as inputs and outputs CSV files containing the letters and the number of their occurrences in the text. After creating the CSV file we were to plot the relative distributions of each letter for each of the used texts. Finally we were to make sure that the total probability of each distribution was equal to 1.

$$P(M) = \sum_{i=1}^{n} P(m_i) = 1$$

Figure 1: Equation used to verify that the sum of the probabilities equals to 1

In task two we had to at first compute the following probability distributions for one of the text files we used in task one $P(M); P(K)\ and\ P(C)$. We used the provided C code to encode one of the texts. The source was compiled with gcc and then executed on the text file "adventures_of_huckleberry_finn.txt" to create the file `./books/finn.cip`. The next step was to graph the relative distribution of the ciphertext to determine the P(C). Finally we were to calculate the following conditional probabilities:

$$P(M = e|c_i) \quad c_i \in C$$
$$P(M = t|c_i) \quad c_i \in C$$
$$P(M = a|c_i) \quad c_i \in C$$
$$P(M = i|c_i) \quad c_i \in C$$
$$P(M = o|c_i) \quad c_i \in C$$
$$P(M = n|c_i) \quad c_i \in C$$

2 Solution

2.1 Task 1

The python file `<project_folder>/task1.py` was written to assist with this section. This program takes two types of inputs.

1. path to a .txt file. Ex: `./books/adventures_of_huckleberry_finn.txt`
2. path to a directory containing multiple .txt files. Ex: `./books/`

This python file also checks for the existence of *numpy* and *matplotlib* which are python math and graphics libraries commonly installed on Linux systems. If they are present not only does the program output the required CSV files to a `<input_path>/output/` directory but also proceeds to plot the data points. Please see Figure 2.

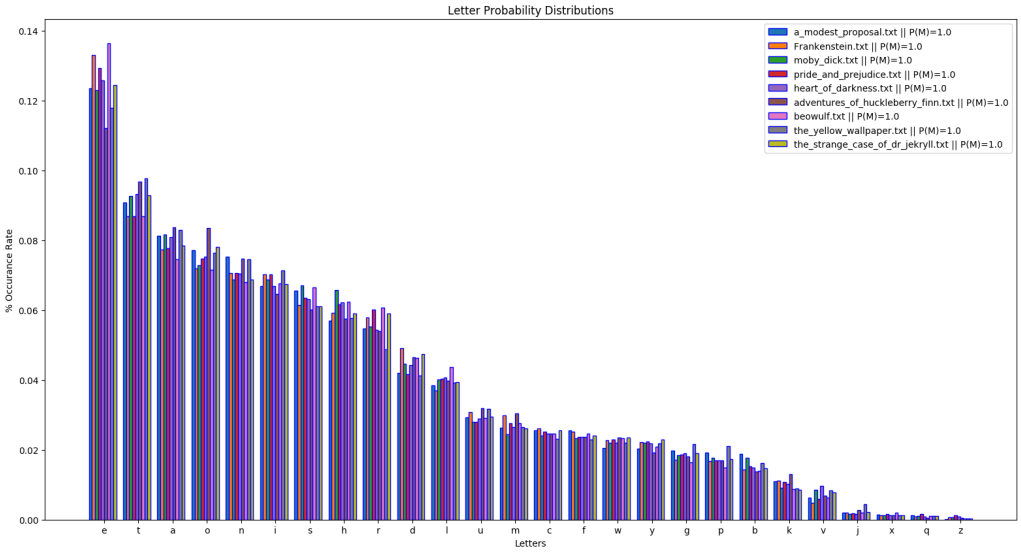

Figure 2: Probability distributions across multiple plaintext files

As we can see in the label of the figure above $P(M) = 1$ as per Equation (1) for each book.

2.2 Task 2

The encoding of the book file was rather simple. See Figure 3. Following that I ran the python file to check the distributions of the encoded file. We can see in Figure 4 that the letter "g" is the most commonly occurring character in the text. This is of interest to us because we know that we offset the text by 2 (and "g" is in fact 2 characters to the right in the alphabet from "e" which is the most common letter). Finally plugging the generated CSV file into Google Sheets and multiplying each value by 1/25 we get the following conditional probabilities for $P(M)$. The final output of the Sheets program is in the Figure 5.

3 Conclusion

The first task showed us that the distribution of letters in the English language is very consistent across multiple texts written during different periods of time. By looking at the probability of a certain plaintext

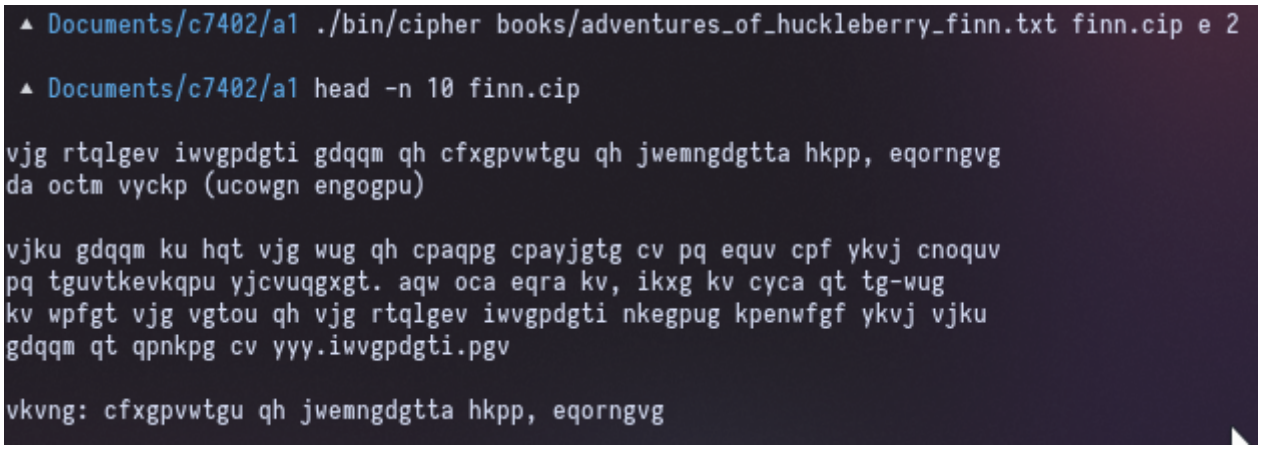

Figure 3: The book Adventures of Huckleberry Finn ciphered with the offset of 2

letter m_i we can determine if the ciphertext letter c_i is being used in its place. This assignment helped me clarify some of the concepts we spoke about in class by giving me the opportunity to put them into practice.

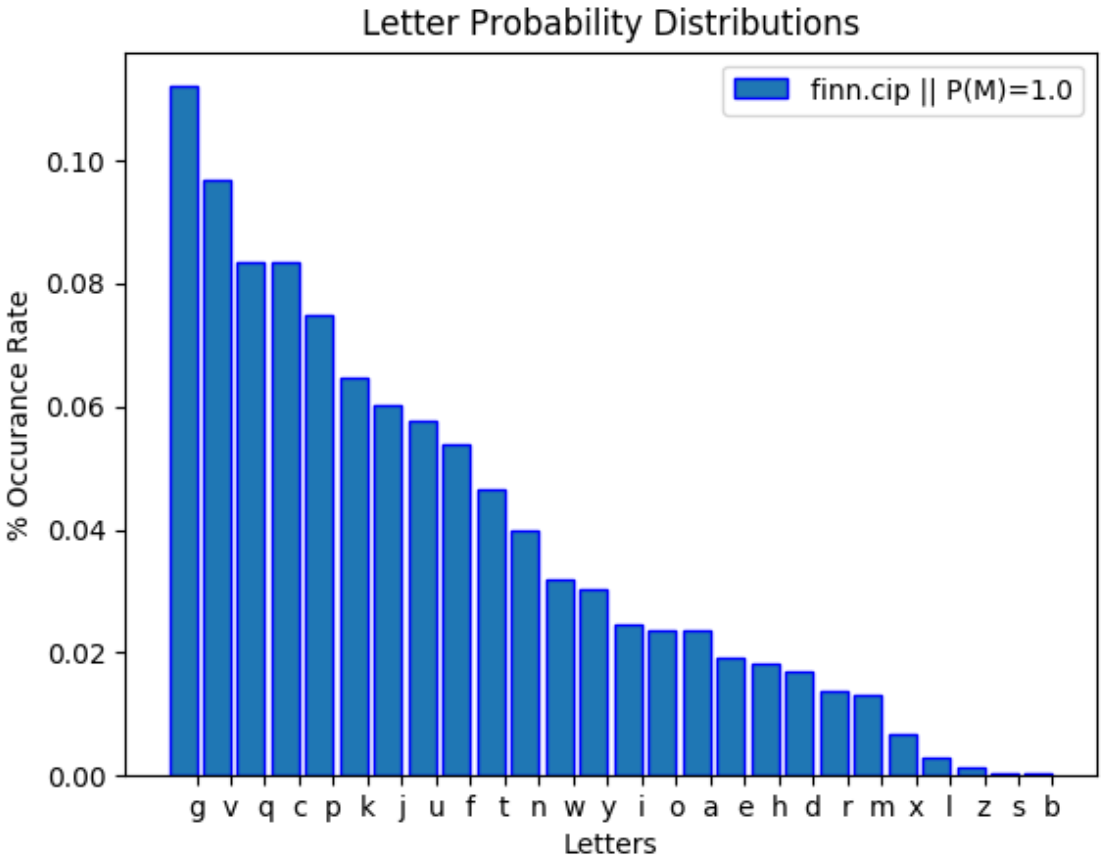

Figure 4: Encoded Book's Character distribution

	A	B	C	D	E	F	G
1	Letter	Count	P(C)	P(M)			Total
2	g	49604	0.1120477791	0.004481911164			442704
3	v	42824	0.09673280567	0.003869312227			
4	q	37017	0.08361568904	0.003344627562			
5	c	36946	0.083455311	0.00333821244			
6	p	33118	0.07480844989	0.002992337996			
7	k	28635	0.06468204489	0.002587281796			
8	j	26659	0.0602185659	0.002408742636			
9	u	25502	0.0576050815	0.00230420326			
10	f	23905	0.05399770501	0.002159908201			
11	t	20553	0.04642605443	0.001857042177			
12	n	17636	0.0398370017	0.001593480068			
13	w	14113	0.03187908851	0.00127516354			
14	y	13418	0.03030919079	0.001212367632			
15	i	10905	0.0246327117	0.000985308468			
16	o	10479	0.02367044346	0.0009468177383			
17	a	10401	0.0234942535	0.0009397701399			
18	e	8484	0.01916404641	0.0007665618562			
19	h	7991	0.01805043551	0.0007220174202			
20	d	7508	0.01695941306	0.0006783765225			
21	r	6110	0.01380154686	0.0005520618743			
22	m	5758	0.01300643319	0.0005202573277			
23	x	2992	0.006758466153	0.0002703386461			
24	l	1237	0.002794192056	0.0001117676822			
25	z	527	0.001190411652	0.00004761646608			
26	s	195	0.000440474899	0.00001761899599			
27	b	187	0.000422404134	0.00001689616538			
28							
29	Caesar Prob						
30	1/25						

Figure 5: Caesar cipher probabilities