FREQUENCY ANALYSIS

DPS RK PURAM

Delhi Public School, R.K. Puram

Topic

FREQUENCY ANALYSIS

Ashir Aseesh Borah - XII K

Alabhya Vaibhav - XI L

Mallika Gokarn - XI J

S. Rohit – XI J

Abstract

This is a comprehensive study by the team of students from DPS RK Puram on the use of relative percentages of occurrence of letters to create an alphabet profile for the given cipher and using it to analyse a substitution effectively reducing the permutations to unity. The reduction is exponential with every step and the length of the text helps to improve accuracy. This technique can also be used to determine the time period in which a literary work was written, analyse the various styles used by the author of a certain piece of text, identify the language used in a given text without a language expert and observe the evolution of a language.

Acknowledgements

This project would not have been possible without the support of many people. We would like to thank Mr. Anil Kathuria (In-charge Math Lab and Information Center) for allowing us to use the Math Lab of our school.

We would also like to thank our teachers, Ms. S.K.N. Laxmi, Ms. Tandeep Kaur and Mr. Anil Kathuria for their unwavering support and mentoring that has helped us bring the project to its current state.

Table of Contents

PREFACE v

CHAPTER 1: INTRODUCTION..... 1

1.1Being Sneaky- Cryptography..... 1

1.2 Cipher..... 1

1.3 Substitution Cipher..... 2

1.4 Forms of Substitution Ciphers:

a. Caesar Cipher..... 2

b. Mod- n Cipher..... 2

1.5 The Art of Breaking a Cipher..... 3

CHAPTER 2: DECRYPTING A SUBSTITUTION CIPHER..... 5

2.1 Methods..... 5

2.2 Limitations..... 6

CHAPTER 3: CALCULATING THE FREQUENCY OF A LETTER 7

3.1 Methodology..... 7

3.2 Statistical Analysis 7

CHAPTER 4: USING FREQUENCY OF LETTERS TO DECRYPT CIPHERS..... 9

CHAPTER 5: FUTURE PROSPECTS..... 11

5.1 Identifying a language without a language expert..... 11

5.2 Styles used by an author..... 11

5.3 Evolution of a language

a. History of the language..... 11

b. Possible transformations in the future..... 12

6.4 Time period of literary work..... 12

ANNEXURE (C++ codes)

Annexure 1.. …… 14

BIBLIOGRAPHY…… 18

Chapter 1
Introduction

1.1 Cryptography

The practice of the encryption and decryption of text in secret code in order to protect the data from third party is called Cryptography.

Modern cryptography involves a study of mathematics and computer science. The principles of cryptography are today applied to the encryption of fax, television, and computer network communications. The secure transmission of computer data is highly used in banking, government, and commercial communications.

The science of secure and secret communications, involving both cryptography and cryptanalysis, is known as cryptology.

1.2 Cipher

A cipher, in cryptography, is an algorithm used to perform tasks of encryption and decryption by following a set of well-defined steps. Based on key, there are two types:

1. *Symmetric key algorithms or Private-key cryptography -*

   Same key is used for encryption and decryption.

2. *Asymmetric key algorithms or a Public-key cryptography –*

   Two different keys are used for encryption and decryption.

1.3 Substitution Cipher

Substitution Cipher is a scheme of data encryption in which units of the plaintext (generally single letters or pairs of letters of ordinary text) are replaced with other symbols or groups of symbols).

There are generally two types of Substitution Ciphers:

1. *Simple substitution cipher - operates on single letters*
2. *Polygraphic substitution cipher - operates on single letters*

1.4 Forms of Substitution Ciphers

a. Caesar Cipher

The Caesar cipher is one of the simplest and oldest ciphers in the world in which the cipher alphabet is merely a cyclical shift of the plaintext alphabet.

The Caesar Cipher is named after Julius Caesar, the Roman General, who was the first to use it, for encrypting his messages to all those military units under his command. It was used with a shift value of 3, such that D would be A, E would be B and so on. It was further used by his nephew Augustus with a shift value of 1.

Al-kindi's works in the Arab world are the earliest surviving records of the use of Caesar Cipher and dates back to the 9th century which resulted in the discovery of

"frequency analysis". Even the encryption of the names of the gods behind the Mezuzah also used a Caesar Cipher.

b. Mod -n - Cipher

Mod-n is a type of a substitution cipher, in which a certain letter is replaced by the letter exactly n letters after it.

***Eg**: ROT13 or rotate 13 is a type of mod-n cipher where n=13*

The easiest technique of decrypting a ROT13 piece of code is given by the following formula for any Latin alphabet :

$$ROT_{13}(ROT_{13}(x)) = x$$

A shift of thirteen was chosen over other values, because thirteen is the value for which encoding and decoding are equivalent, thereby allowing the convenience of a single command for both. This code has been used by various institutions like newspapers in the 1980s, as a built-in feature for news reading software

1.5 The Art of Breaking Cipher

Cryptanalysis is the art of breaking codes and ciphers. The Caesar Cipher is the easiest to decrypt. Since the shift value has to be between 1 and 25, a brute force method that checks for every shift value can be taken up by which every possibility will be checked and he will be able to immediat and the shift value for the rest of the code. But a much more systematic method or approach find the key to crack the

cipher could be to use the frequency distribution of the alphabets in the code to our benefit.

This means that the letter "e" is the most common, and appears almost 13% of the time, whereas "z "appears far less than 1 percent of time.

A cryptanalyst just has to find the shift that causes the cipher text frequencies to match up closely with natural English frequencies, then decrypt the text using that shift. This method can be used to easily break Caesar ciphers by hand.

There are three situations that a cryptanalyst will face while dealing with a Caesar Cipher:

1. The cryptanalyst, on looking at the piece of code, guesses that some sort of a simple substitution cipher has been used, but of course cannot specifically decide it to be a Caesar Scheme.

In this case, using frequency analysis or pattern words, it can be deduced that there is regularity in the code, therefore a Caesar Scheme is in use.

2. The cryptanalyst may know that a Caesar scheme is being used, finding the shift value is difficult.

In the Second Case, since the attacker is already aware that a Caesar scheme is in use the techniques to decipher the piece of code is much more straightforward and easy. Now there are limited shift values (i.e. 26 values) that can be checked using every shift value possible from 1 to 25. Where, the letter itself is 0, the following letter 1 and so on.

3. The individual substitution where the shift value of each letter is not fixed. This is the hardest situation. For a five letter word, there are 265 (1,18,81,376) possibilities. The project aims to reduce this number to unity.

Chapter 2
Decrypting a Substitution Cipher

2.1 Methodology

Since, the project is statistical in nature; we took a very large sample space. This was done by using free eBooks available from the website projectgutenburg.org

The frequency analysis of each file was done using a C++ program which reads a given text file, character by character, and at the end of the file, calculates the percentage of occurrence of each letter.

There were two problems in this method.

1. *Some of the files were too short to be considered suitable*
2. *Few were in other languages*

Solution: *Another C++ program was developed which counted the occurrence of a particular word and if its occurrence was below a set limit, the content of the file was erased. The approach although doesn't guarantee absolute accuracy but the resulting accuracy is acceptable if the limit is set to sufficiently large number. So, we set 'THE' as the word to search for and the limit as 200.*

The analysed data was written into a text file which was then imported into an excel file and further mined for data. Average, maximum, minimum and graph was later done in excel using the imported data. Data ranges have been decided on the basis of the data for the substitution. For other uses, the same programs are used but the profiles are prepared accordingly.

2.2 Limitations

The use of substitution cipher has become limited in today's world due to the increased abilities of today's computers. Brute force has become the norm and so simple substitution schemes don't work. Even the individual substitution scheme can be cracked if the cryptanalyst has sufficient time and resources in his command.

Frequency analysis can be countered by using special characters disguised as letter.

Example: using 'a' to replace "spaces" and using '#' to replace "a".

Even then frequency analysis has many uses and is an effective way for communication as encryption can be done on-the-fly without any computer or handheld device.

Chapter 3
Calculating Frequency of Letters

3.1 Methodology

The mod-n ciphers are easiest to decrypt. Since the shift value has to be between 1 and 25, the cracker can use a brute force method to check for every shift value, thereby examining every possibility. By this method, the cracker, will take a relatively short time with 25 possibilities (considering the language as English)out of which one permutation is meaningful.

A more systematic and easier method in finding the key to crack the cipher would be to use the frequency distribution of the alphabets in the code. The cracker can calculate frequency of letters appearing in the cipher and relate them to the frequency stats of the language being used.

For example: The cracker finds that in the code the letter B is used very often (relative percentage almost 13%) and the plaintext language is English, then he can compare this to the letter frequency of English language and conclude that the letter B in the code corresponds to E, ergo deducing that the shift value of the code is 4.

For substitution ciphers not having a fixed shift value, the cracker can easily compare the relative letter frequency percentages of the encoded words using the profiles prepared by us thereby reducing the possibilities significantly.

3.2 Statistical Analysis

The following data has been collected from about 9000+ ebooks. To avoid wastage of page, the entire worksheet has been provided in a CD

The data profiles are as follows:

Letter	Percentage Range
z, q, j, x, k, v	0 to 1
b, p, y, g, w, f, m, c, u	1.5 to 3
l, d	4 to 5
r, h, s	6 to 6.5
n, l, o, a	6.5 to 8
t, e	9+

Chapter 4
Extending to Other Languages

Frequency of letters

The codes used can be extended (with minor changes) to other languages to obtain the frequencies of letters in those languages. Ciphers in these languages can be broken using the same methods in Chapter 3.

Due to lack of resources, in this project we have focused only on English language. But initial investigations show promise.

We have created a second C++ program (see: Program ii) which eliminates the use of files of languages other than the one being analyzed. This code checks for a certain word that is very common and used repeatedly in a particular language (for e.g. the - English, der - German, de - French...). Now if that word appears more than a given number of times then the sample piece of text can be considered to have been written in that particular language.

The requirements for this method to be effective would be a list of the most common words in various languages. This helps to reduce the error in the data without the need to manually check the files

Chapter 5
Future Prospects

5.1 Identifying the language used in a given text without a language expert

By the various methods and programs we have used, if the profiles of different languages are available, then a given document can be scanned and by comparing the results with the reference profiles, the language can be determined.

5.2 Analyzing the various styles used by the author of a certain piece of text

*Every author has certain specific expressions which are commonly used by them often called their **pen-style.** The slight change in expression causes a minor discrepancy in the frequency of the letters. Through frequency analysis certain patterns can be observed these patterns would then be found to have been used by the author in his or her various literary works.*

5.3 Evolution of a language

We can determine what course of evolution the language has undertaken and classify certain characteristic features of that language to a particular era using frequency analysis.

a. History of the language

Some words are specific to a certain literary era. For example, English language went through a series of metamorphic periods:

i. ***Old English (450-1100 AD)****:* It was developed by the British from similar tribal languages. It is very different from the English used today. Some common words: mugwump, rawgabbit, vinomadefied, etc

ii. ***Middle English (1100-1500)****:* It was developed when the French conquered England and brought with them a kind of French, several words of which still continue to exist in the language. Common words: adoun, aghast, agay, alderbest, anon, ar'tow, etc.,

iii. ***Early Modern English (1500-1800):*** *developed towards the end of Middle English when vowels started being pronounced shorter and shorter and the invention of printing standardized the spelling and grammar. Common words: thee, thou, thy/thine, bilbo, hath, nay, shalt, whence, wherefore Late Modern English (1800-Present): has many more words because of the Industrial Revolution and technology creating a need for new words and adoption of foreign words from many countries due to globalization*

With every period, certain words became less common or even stopped being used. This causes a slight change in the frequency of letters and words over eras.

b. Possible transformations in the future.

Through these graphs one can observe the popularity of which letter/word has gotten reduced over time and can even predict how the language might transform in the years to come.

5.4 Determining the time period in which a literary work was written.

Due to a change in common expressions and words with the literary eras (see 6.3A), there would be a significant difference in the frequency of letters with each time period and hence, by calculating the frequency of letters in the text, its age can be roughly estimated.

Annexure 1

C++ Codes

Program 1 :- *To Analyse the language*

```
#include <fstream>
#include <cstring>
#include <cctype>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
	int num=0,i=0;
	cout<<"Enter number of files: ";
	cin>>num;
	cout<<num;
	fstream filt;
	filt.open("result.txt",ios::out);
	if(!filt.is_open())
		cout<<"Error opening file result\n";
	else
	{
		while(i<num)
			{
				i++;
				char name [100],curr[8],ch;
				double a[27]={0};
				itoa(i,curr,10);
				strcpy(name,"V:\\txt\\a(");
				strcat(name,curr);
				strcat(name,")");
				strcat(name,".txt");
				cout<<name;
				fstream fil;
				fil.open(name,ios::in);
				if(!fil.is_open())
```

```
                        cout<<endl<<name<<" Error\n";
                    else
                    {
                        while(!fil.eof())
                            {
                                fil>>ch;
                            if(isalpha(ch))
                                {
                                    ch=tolower(ch);
                                    a[ch-97]++;
                                    a[26]++;
                                }
                        }
                cout<<endl;
                for(int j=0;j<26;j++)
                    {
                        a[j]=(a[j]/a[26])*100;
                        cout<<char(j+97)<<": "<<a[j]<<"\t";
                    }
    cout<<"Total: "<<a[26];
    for(int k=0;k<26;k++)
        filt<<a[k]<<"\t";
filt<<endl;
 fil.close();
                    }
                }
}
filt.close();
return 0;
}
```

Program 2 :- *To identify the language*

```
#include <fstream>
#include <cstring>
#include <cctype>
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
        int num=0,i=0;
        cout<<"Enter number of files: ";
        cin>>num;
        //cout<<num;
        while(i<num)
        {
                i++;
                char name [100],curr[8],word[100];
                itoa(i,curr,10);
    strcpy(name,"V:\\txt\\a(");
    strcat(name,curr);
    strcat(name,")");
    strcat(name,".txt");
    //cout<<name;
    int count=0;
    fstream fil;
    //cout<<i<<"\t";
    fil.open(name,ios::in);
    if(!fil.is_open())
        cout<<endl<<name<<" Error\n";
    else
    {   cout<<i<<"\n";
        while((!fil.eof())&&(count<200))
        {int flag=0;
```

```
            fil>>word;
            int a=200,b=20000;
            //cout<<"\nfirst"<<i<<"\tWord: "<<word;
            for(int p=0;p<strlen(word);p++)
                {
                    if(!isalpha(word[p]))
                        {
                         for(int n=0;n<strlen(word);n++)
                              word[n]='\0';
                         fil>>word;
                         //cout<<"\tNew word: "<<word;
                         p=0;
                        }
                }//cout<<a<<"\t"<<b;
              //cout<<"\nIn: "<<i;
          if(strcmpi(word,"the")==0)
              {
               count++;
               //cout<<i;
              }//cout<<"\tOut: "<<i<<"\tFinal word: "<<word;
        //cout<<"Count: "<<count<<"   "<<i<<"\thaha1";
         }
        //cout<<"\thaha2"<<i;
fil.close();
   if(count<200)
        {
         fstream fil;
         fil.open(name,ios::out);
         fil.close();
        }
  }
   //cout<<"\thaha3"<<i<<"\t"<<num<<endl;
  } return 0;}
```

Bibliography

The following are the websites which have fulfilled the purpose of aiding our research:

1. *http://www.wikipedia.org*
2. *http://www.princeton.edu*
3. *http://www.britannica.com*

The books used as research samples were taken from

1. *http://www.projectgutenburg.org*

We, the students who have brought this project to its current state have created the code and the animations entirely on our own.