**Assignment 2**

**Group:**BDA-1904

**Name:**Berik Gulina &Abdikalyk Gulnazym

**Github Link:** <https://github.com/gulina0426/InformationTheory>

**1-Description:**

First of all,our goal in this task is going to implement the program which related to find the all the symbols inside in the txt file to calculate each of the appearance probability.So we have implemented it in with the python code to see the whole process.Our txt file’s content is take from one song and the code will firstly read the file with content after that going to set the varibles what we need and to create the loop statement in order to check the txt file’s content,also contain the lower and upper case letter in different calculation.Then step by step to calculate the total unique symbols then to divided by each of them with the total to get the probability.

**2-Team working process:**

From the beginning we have decided to work on the colab in order to see the process more convinently and clearly to share the idea and find out the mistake.We have done the code together like if Gulina is going to implement the part related to the loop statement then Gulnazym is responsible for setting the heading part and testing the result.Also the same in the documentation part,we divided it as equal to finish,and it will be get high efficient performance in the team working.

**3-Execution part with screenshot:**

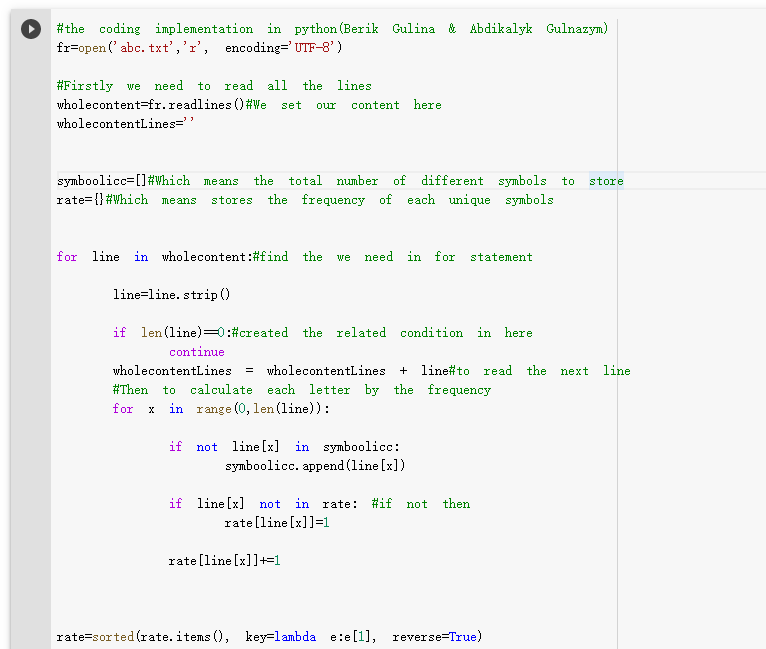
****

Figure-1 Python code

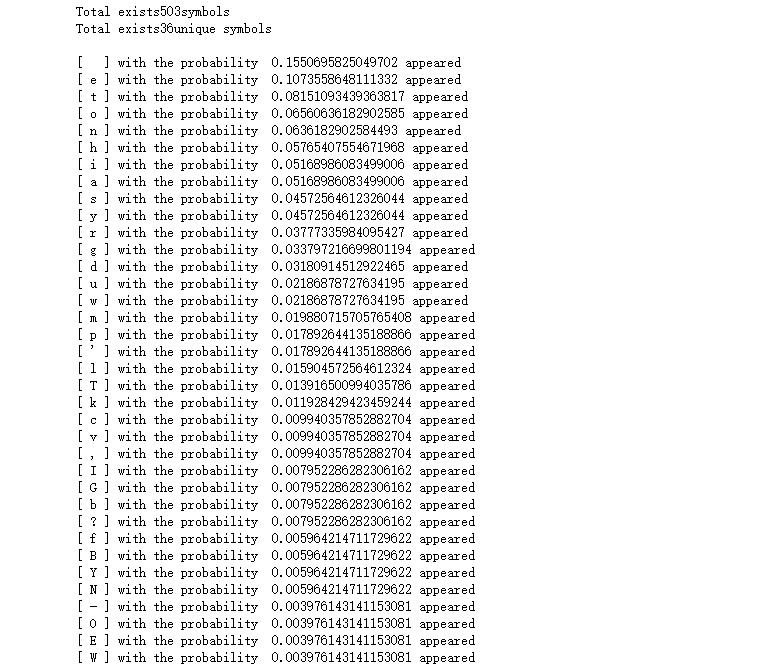


Figure-2 Python code result

As the result you can see that it calculated with each of the symbols’ probability not only the letters.So we totally have 503 symbols and the unique with 36.And the most frequented one is the letter [e] with probability round 0.107 and in converse is letter [w] with 0.004.Our source code is provided in the txt file.

**4-Source code in the text format:**

#the coding implementation in python(Berik Gulina & Abdikalyk Gulnazym)

fr=open('abc.txt','r', encoding='UTF-8')

#Firstly we need to read all the lines

wholecontent=fr.readlines()#We set our content here

wholecontentLines=''

symboolicc=[]#Which means the total number of different symbols to store

rate={}#Which means stores the frequency of each unique symbols

for line in wholecontent:#find the we need in for statement

line=line.strip()

if len(line)==0:#created the related condition in here

continue

wholecontentLines = wholecontentLines + line#to read the next line

#Then to calculate each letter by the frequency

for x in range(0,len(line)):

if not line[x] in symboolicc:

symboolicc.append(line[x])

if line[x] not in rate: #if not then

rate[line[x]]=1

rate[line[x]]+=1

rate=sorted(rate.items(), key=lambda e:e[1], reverse=True)

print('Total exists%dsymbols'%len(wholecontentLines))#print out the result

print('Total exists%dunique symbols'%len(symboolicc))

print()

for i in rate:#show the content with probability calculated

print("[",i[0],"] with the probability ",  i[1]/len(wholecontentLines), "appeared")

fr.close()#then stopped to read the txt