I. objectives

Write a program in which take an Input file, use

the simple approach below to summarize a text file:

a. Read the data from a file

b. Tokenize the text into words and apply lemmatization technique on each word.

c. Find all the trigrams for the words.

d. Extract the top 10 of the most repeated trigrams based on their count.

e. Go through the text in the file

f. Find all the sentences with the most repeated trigrams

g. Extract those sentences and concatenate

h. Print the concatenated result

Approaches/Methods

nltk.word\_tokenize(i)

nltk.sent\_tokenize(i)

le=WordNetLemmatizer()

ngrams(splitWord, 3):

wordFreq = FreqDist(trigramsOutput)  
# Getting Most Common Words and Printing them - Will get the Counts from top to least  
top10 = wordFreq.most\_common(10)

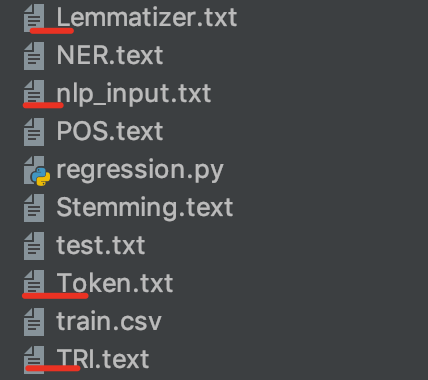
Workflow

1. Read content from a txt file
2. Use sentence tokenize and word tokenize to separate content into individual.
3. Apply lemmatization on each word and find trigrams
4. Extract top 10 the most repeated word base on counts by FreqDist
5. Use find method and concatenated string to localize word in sentence
6. import requests  
   from bs4 import BeautifulSoup  
   import nltk  
   from nltk import PorterStemmer  
   from nltk import WordNetLemmatizer  
   from nltk import wordpunct\_tokenize, pos\_tag, ne\_chunk  
   from nltk import ngrams, FreqDist  
   from collections import Counter  
   le=WordNetLemmatizer()  
   #a  
   file=open("nlp\_input.txt",encoding="utf8", errors='ignore')  
   fileToken=open("Token.txt","w+")  
   filele=open("Lemmatizer.txt","w+")  
   fileTri=open("TRI.text","w+")  
   trigramsOutput = []  
   n=0  
     
   for i in file.readlines():  
    n=n+1  
   #b Tokenize the text into words and apply lemmatization  
    splitWord = nltk.word\_tokenize(i)  
    splitSen=nltk.sent\_tokenize(i)  
     
    for m in ngrams(splitWord, 3):  
    trigramsOutput.append(m)  
     
    filele.write('[')  
    for j in splitWord:  
    l = le.lemmatize(j)  
    filele.write(l)  
    filele.write(',')  
    fileToken.write(str(splitWord))  
    filele.write(']')  
     
   file.close()  
   fileToken.close()  
   filele.close()  
   fileTri.write(str(trigramsOutput))  
   fileTri.close()  
   wordFreq = FreqDist(trigramsOutput)  
   # Getting Most Common Words and Printing them - Will get the Counts from top to least  
   top10 = wordFreq.most\_common(10)  
   print("Top 10 triGrams : \n", top10)  
     
   file=open("nlp\_input.txt",encoding="utf8", errors='ignore')  
   x=file.read()  
   splitSen=nltk.sent\_tokenize(x)  
     
   concatenate=[]  
   for ((e,d,f),len) in top10:  
    a=e+" "+d+" "+f  
    for sentence in splitSen:  
    if(sentence.find(a)>0):  
    concatenate.append(sentence)  
   print(concatenate)

Datasets (if applicable)

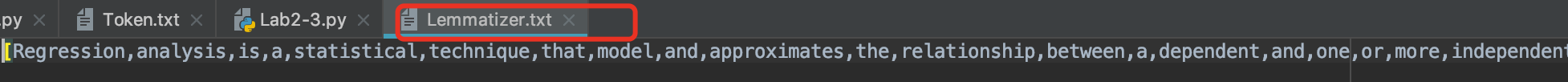
<https://umkc.app.box.com/s/7by0f4540cdbdp3pm60h5fxxffefsvrw>

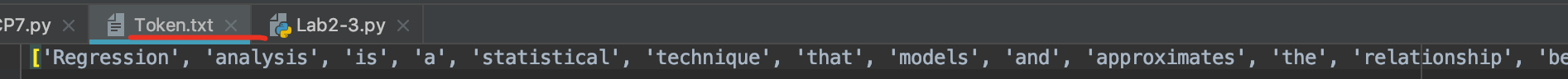
Parameters

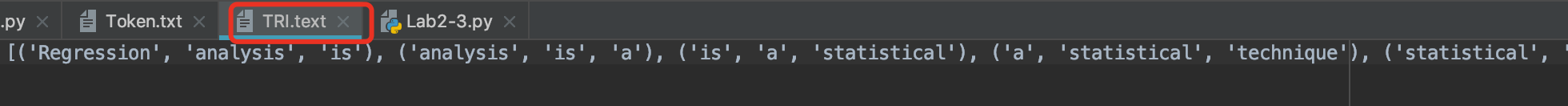


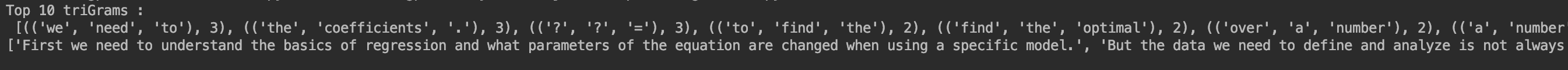
Evaluation & Discussion

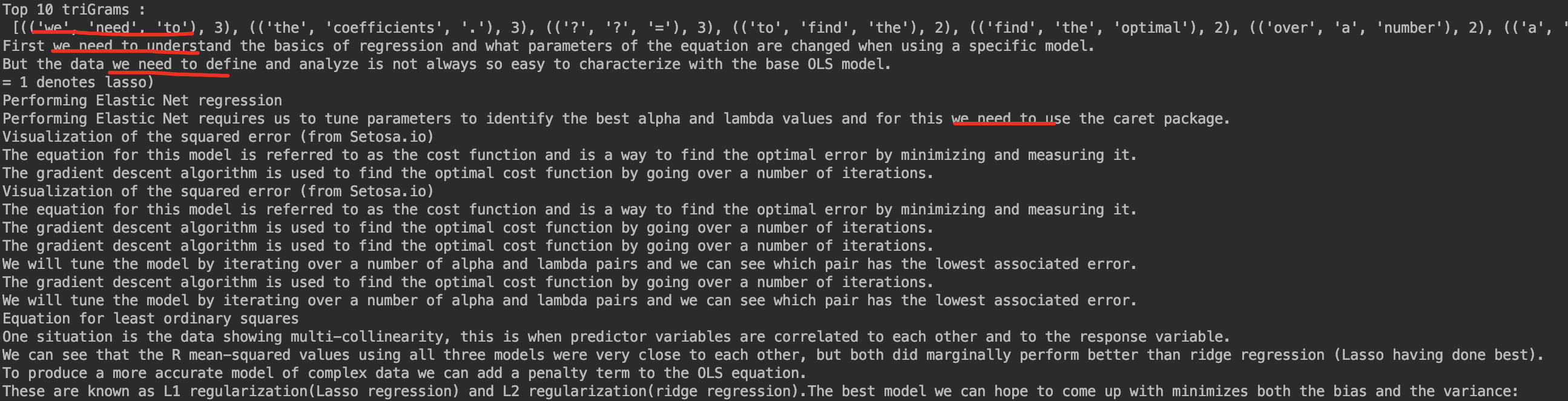
Output:











The weakness of my output is that I don’t format the output to look more clear and pretty.