Python and Deep Learning

Lab\_Assignment\_3

Program – 1:

Program – 2:

Program – 3:

file = open(**"Input"**, **"r"**)  
Data = file.read()  
print(**"Data in the text file:"**,**"\n"**,Data)  
  
print(**"-----------------------"**)  
  
lmtz = WordNetLemmatizer()  
words = word\_tokenize(Data)  
print(**"Lemmatization of the words in the text file:"**)  
**for** word **in** words:  
 print(lmtz.lemmatize(word, pos=**'v'**))  
  
print(**"-----------------------"**)  
  
print(**"Applying bigram on the data in text file:"**)  
words\_1 = word\_tokenize(Data)  
list = []  
bigram = ngrams(words\_1,2)  
**for** a **in** bigram:  
 list.append(a)  
print(list)  
  
print(**"-----------------------"**)  
  
words\_count = Counter(list)  
print(**" Calculating the bigram frequency of the words:"**)  
print(words\_count)  
  
print(**"-----------------------"**)  
  
print(**"Choosing the top 5 bigrams that has been repeated the most:"**)  
frequency\_count = nltk.FreqDist(list)  
top\_bigrams = frequency\_count.most\_common(5)  
print(top\_bigrams)  
  
**with** open(**'Input'** , **'r'**) **as** file:  
 line = file.readlines()  
fit= **''  
  
for** word **in** line:  
 fit= fit + word  
sentence\_1 = sent\_tokenize(fit)  
rep\_sentence\_1 = []  
  
  
**for** sent **in** sentence\_1:  
 **for** word,words **in** list:  
 **for** ((c,m), l) **in** top\_bigrams:  
 **if** (word,words == c,m):  
 rep\_sentence\_1.append(sent)  
  
print(**"-----------------------"**)  
  
print (**"\n Going through the original text in the file and finding the sentences with those most repeated bi-grams"**)  
freq\_count = nltk.FreqDist(rep\_sentence\_1)  
Finish = freq\_count.most\_common(5)  
**for** key,value **in** Finish:  
 print(**"\n"**,key)

Program – 4: