Text Summarization and Question Answering System for COVID-19 Data

ELC002-2122EVESEM

Submitted By

Aditya Singh Rathore 102003424
Deepankar Varma 102003431
Prateek Sharawat 102003428

Computer Science and Engineering Department

TIET, Patiala

Code Explanation :

```
import nltk
import pandas as pd
from string import punctuation
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import sent_tokenize
from nltk.stem import PorterStemmer
```

1. **nltk** : The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP).

2. **pandas** : Pandas is a python library for data analysis

3. **string.punctuation :** is a pre-initialized string used as string constant

4. **nltk.corpus** : The modules in this package provide functions that can be used to read corpus files in a variety of formats.

5. **nltk.stem** : Stemmers remove morphological affixes from words, leaving only the word stem. Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers.

6. **nltk.tokenize**: We are able to extract the tokens from string of characters by using tokenize.word_tokenize() method. It actually returns the syllables from a single word. A single word can contain one or two syllables.

```
f=open(("001.txt"),'r')
text=f.read()
f.close()
print(text)
```

Opens a file named “001.txt” in read only mode and prints the contents of the file by the help of a variable text in our case.

```
sent_tokens = sent_tokenize(text)
word_tokens = word_tokenize(text)
```

1. **sent_tokenize** : To split a document or paragraph into sentences.

2. **word_tokenize** : A sentence or data can be split into words using the method **word_tokenize().**

```
print(len(word_tokens))
```

Prints the number of words in the variable word_tokens.

```
ps=PorterStemmer()
stem=[]
for word in word_tokens_refined:
    stem.append(ps.stem(word))

word_tokens_refined=stem
```

The Porter stemmer is a process for removing the commoner morphological and inflexional endings from words in English.

```
FreqTable={}
for word in word_tokens_refined:
    if word in FreqTable:
        FreqTable[word]+=1
    else:
        FreqTable[word]=1
print(len(FreqTable))
```

For each word in the word_tokens_refined variable , if the word is present in the FreqTable then increments the frequency of that word by 1.

```
maxfreq=max(FreqTable.values())
print(maxfreq)
```

Prints the maximum frequency of a value from the FreqTable

```
sum = 0
for sent in sentence_scores:
    sum+=sentence_scores[sent]
average=int(sum/len(sentence_scores))
print(average)
```

Average sentence length is decided by the formula ,sum of all sentence scores divided by the length of sentence_Scores variable.

```
summary=''
for sent in sent_tokens:
    if(sentence_scores[sent]>1.2*average):
        summary+=""+sent
print(summary)
```

If sentence score of each element in sent_tokens is greater than 1.2 time the average sentence score then it is added to the summary.

```
len(summary)
```

Prints the length of the summary

```
len(text)
```

Prints the length of variable text