

19BCE2311 Gaurav Singh

Question:

Write a python program to

Find out the list of common and unique terms in the between the three text files attached (chess.txt, tennis.txt, cricket.txt) and print its count.

Apply stopwords removal on those common and unique terms (using spacy), print its count and save the terms (after stopwords removal) in index.txt file.

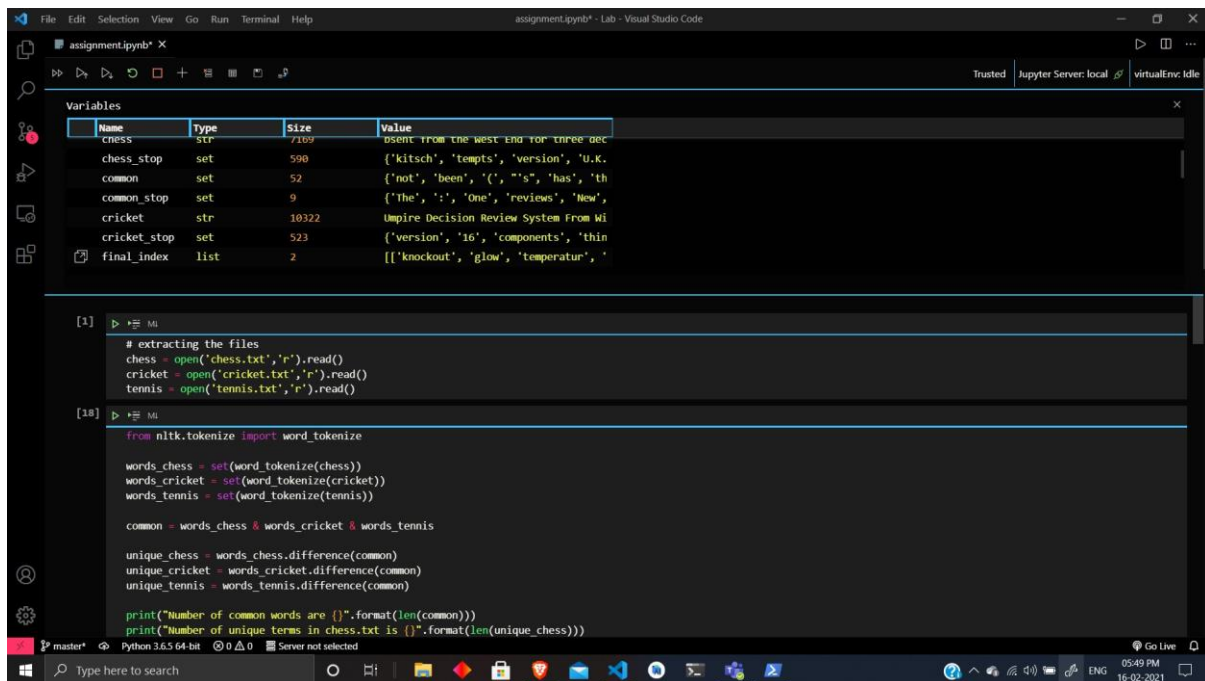
[List of additional Stop words to be considered = [dot, comma, single-quote, double quote, question mark, brackets [square, parentheses, curly, angle], exclamation mark]]

Apply Stemming and lemmatization on the terms present in index.txt file. Print the count of terms after applying stemming and lemmatization.

Replace the content of index.txt file by either stemmed terms or lemmatized term depending on its count.(lower count value should be considered for replacement) and rename the file name to final-index.txt

Print the POS tag of all the terms present in the final-index.txt file using pandas dataframe

Screenshots:



The screenshot displays a Visual Studio Code editor with a Python script named `assignment.py`. The script performs the following steps:

- Extracts the content of `chess.txt`, `cricket.txt`, and `tennis.txt` into variables `chess`, `cricket`, and `tennis`.
- Tokenizes the words from these files using `nlTK.tokenize` and stores them in sets `words_chess`, `words_cricket`, and `words_tennis`.
- Calculates the common words across all three files: `common = words_chess & words_cricket & words_tennis`.
- Calculates the unique words for each file: `unique_chess = words_chess.difference(common)`, `unique_cricket = words_cricket.difference(common)`, and `unique_tennis = words_tennis.difference(common)`.
- Prints the number of common words and the number of unique terms in `chess.txt`.

The Variables panel on the left shows the following variables and their values:

Name	Type	Size	Value
<code>chess</code>	<code>str</code>	7109	osent from the west end for three dec
<code>chess_stop</code>	<code>set</code>	590	{'kitsch', 'tempts', 'version', 'U.K.
<code>common</code>	<code>set</code>	52	{'not', 'been', '(', 's', 'has', 'th
<code>common_stop</code>	<code>set</code>	9	{'The', ':', 'One', 'reviews', 'New',
<code>cricket</code>	<code>str</code>	10322	Umpire Decision Review System From Wi
<code>cricket_stop</code>	<code>set</code>	523	{'version', '16', 'components', 'thin
<code>final_index</code>	<code>list</code>	2	[['knockout', 'glow', 'temperatur', '

```
[1] # extracting the files
chess = open('chess.txt','r').read()
cricket = open('cricket.txt','r').read()
tennis = open('tennis.txt','r').read()

[18] from nltk.tokenize import word_tokenize

words_chess = set(word_tokenize(chess))
words_cricket = set(word_tokenize(cricket))
words_tennis = set(word_tokenize(tennis))

common = words_chess & words_cricket & words_tennis

unique_chess = words_chess.difference(common)
unique_cricket = words_cricket.difference(common)
unique_tennis = words_tennis.difference(common)

print("Number of common words are {}".format(len(common)))
print("Number of unique terms in chess.txt is {}".format(len(unique_chess)))
```


The screenshot shows a Jupyter Notebook in Visual Studio Code. The notebook has two cells. The first cell contains the following code:

```
import os
os.rename('index.txt', 'index-final.txt')
```

The second cell contains the following code:

```
import nltk
untagged = list(set(word_tokenize(open('index-final.txt', 'r').read())))
tagged = nltk.pos_tag(untagged)

import pandas as pd
pd.DataFrame({"terms": untagged, "POS-Tag": tagged})
```

The output of the second cell is a DataFrame with two columns: 'Terms' and 'POS-Tag'. The DataFrame contains 1240 rows. The first few rows are:

	Terms	POS-Tag
0	knockout	(knockout, NN)
1	glow	(glow, NN)
2	temperatur	(temperatur, NN)
3	kitsch	(kitsch, FW)
4	accent	(accent, NN)
...
1235	allianc	(allianc, NN)
1236	critic-proof	(critic-proof, JJ)
1237	woodroff	(woodroff, NN)
1238	desper	(desper, NN)
1239	war	(war, NN)

The DataFrame has 1240 rows and 2 columns.

Code:

```
chess = open('chess.txt', 'r').read()
cricket = open('cricket.txt', 'r').read()
tennis = open('tennis.txt', 'r').read()

from nltk.tokenize import word_tokenize

words_chess = set(word_tokenize(chess))
words_cricket = set(word_tokenize(cricket))
words_tennis = set(word_tokenize(tennis))

common = words_chess & words_cricket & words_tennis

unique_chess = words_chess.difference(common)
unique_cricket = words_cricket.difference(common)
unique_tennis = words_tennis.difference(common)

print("Number of common words are {}".format(len(common)))
print("Number of unique terms in chess.txt is {}".format(len(unique_chess)))
print("Number of unique terms in cricket.txt is {}".format(len(unique_cricket)))
print("Number of unique terms in tennis.txt is {}".format(len(unique_tennis)))
```

```
from spacy.lang.en.stop_words import STOP_WORDS
```

```

stop_words = set(STOP_WORDS)
stop_words = stop_words.union({'.',',','\','\"','?','{','}','[' ,']','<','>','(' ,')','!'})

common_stop = common.difference(stop_words)
chess_stop = unique_chess.difference(stop_words)
cricket_stop = unique_cricket.difference(stop_words)
tennis_stop = unique_tennis.difference(stop_words)
print("Number of terms common after removal of stop words = {}".format(len(common_stop)))
print("Number of terms unique in chess.txt after removal of stop words = {}".format(len(chess_stop)))
print("Number of terms unique in cricket.txt after removal of stop words = {}".format(len(cricket_stop)))
print("Number of terms unique in tennis.txt after removal of stop words = {}".format(len(tennis_stop)))

```

```

values = [common_stop, chess_stop, cricket_stop, tennis_stop]
with open("index.txt", "w") as output:
    for row in values:
        s = " ".join(map(str, row))
        output.write(s+'\n')

index = list(set(word_tokenize(open('index.txt','r').read()))))

from nltk.stem.snowball import SnowballStemmer
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
stemmer = SnowballStemmer(language='english')

stemmed = list(set([stemmer.stem(w) for w in index]))
lemmatized = list(set([lemmatizer.lemmatize(w) for w in index]))

print("Number of stemmed terms is {}".format(len(stemmed)))
print("Number of lemmatized terms is {}".format(len(lemmatized)))

final_index = [stemmed, lemmatized]
if len(stemmed) < len(lemmatized):
    with open("index.txt", "w") as output:
        s = " ".join(map(str, stemmed))
        output.write(s+'\n')
else:
    with open("index.txt", "w") as output:
        s = " ".join(map(str, lemmatized))

```

```
output.write(s+'\n')
```

```
import os  
os.rename('index.txt', 'index-final.txt')
```

```
import nltk  
untagged = list(set(word_tokenize(open('index-final.txt', 'r').read())))  
tagged = nltk.pos_tag(untagged)  
  
import pandas as pd  
pd.DataFrame({"Terms":untagged, "POS-Tag":tagged})
```