

[2023/09/20]

# [Report for Comp479 Project\_1]

[By yichen Huang 40167688]






## Introduction

Hello and welcome to the demonstration of our text preprocessing script using NLTK (Natural Language Toolkit) for the comp479 project 1. During this demo, I will show you through the various capabilities of our tool that can perform different text processing tasks. Our aim is to help clean and prepare text data for further analysis or natural language processing tasks.

## Overview of input file and set stopwords file

### Overview of inputfile

There are 5 output files which I downloaded from the link and saved into folder *reuters21578.tar*, which is at the same level with where my main.py is

 all-exchanges-strings.lc.txt	1996/12/4 13:27	文本文档	1 KB
 all-orgs-strings.lc.txt	1996/12/4 13:27	文本文档	1 KB
 all-people-strings.lc.txt	1996/12/4 13:27	文本文档	3 KB
 all-places-strings.lc.txt	1996/12/4 13:27	文本文档	2 KB
 all-topics-strings.lc.txt	1996/12/4 13:27	文本文档	2 KB

### Set stopwords file

Stopwords are set into folder *Stopwords\_used\_for\_output*, which is at the same level with where my main.py is. For easier using, each file has removed first 5 and the last 5 words from the original file.

 Stopwords_used_for_output_all-exchanges-strings.lc.txt	2023/9/19 15:55	文本文档
 Stopwords_used_for_output_all-orgs-strings.lc.txt	2023/9/19 15:56	文本文档
 Stopwords_used_for_output_all-people-strings.lc.txt	2023/9/19 15:56	文本文档
 Stopwords_used_for_output_all-places-strings.lc.txt	2023/9/19 15:57	文本文档
 Stopwords_used_for_output_all-topics-strings.lc.txt	2023/9/19 15:57	文本文档

## Demonstration Steps

### Step 1: Import necessary libraries and download NLTK resources:

```
1  import os
2  import nltk
3
4  nltk.download('punkt')
5  nltk.download('stopwords')
6
7  from nltk import word_tokenize
8  from nltk.stem import PorterStemmer
9  from nltk.corpus import stopwords
```

### Step 2: Define functions for various text processing tasks:

#### a. Tokenization:

Tokenizes the text in a given file.

```
11 def Tokenizer_output(file_path):
12     with open(file_path, 'r') as file:
13         text = file.read()
14         tokens = word_tokenize(text)
15     return tokens
```

#### b. Lowercasing:

Converts the tokens to lowercase.

```
17 def Lowercased_output(file_path):
18     with open(file_path, 'r') as file:
19         text = file.read()
20         lower_case = [token.lower() for token in Tokenizer_output(file_path)]
21     return lower_case
```

#### c. Stemming:

Applies Porter stemming to the tokens.

```
23 def Stemmed_output(file_path):
24     with open(file_path, 'r') as file:
25         text = file.read()
26         stemmer = PorterStemmer()
27         stemmed_tokens = [stemmer.stem(token) for token in Tokenizer_output(file_path)]
28     return stemmed_tokens
29
```

#### d. Stopword Removal:

Removes stopwords from the tokens using a custom list of stopwords.

```
30 def No_Stopword_output(file_path, stop_words, stopwords_file_path):
31     with open(file_path, 'r') as file:
32         text = file.read()
33
34     # ==== Use the provided stopwords_file_path instead of the hardcoded path
35     stop_words = Stopword_read(stopwords_file_path)
36     No_Stopwords_tokens = [token for token in Tokenizer_output(file_path) if token not in stop_words]
37     return No_Stopwords_tokens
```

#### e. Dynamic Stopword Generation:

Reads a list of stopwords from a file.

```
38 def Stopword_read(stopwords_file_path):
39     with open(stopwords_file_path, 'r') as stopwords_file:
40         stop_words = [line.strip() for line in stopwords_file]
41     return stop_words
```

### Step 3: The main function:

The main() function processes multiple input files and demonstrates each step of text preprocessing for each file. It also saves the processed data into separate output files for each step.

#### A. Processing five input files by using for loop

a loop processes five input files: 'all-exchanges-strings.lc.txt', 'all-orgs-strings.lc.txt', 'all-people-strings.lc.txt', 'all-places-strings.lc.txt', and 'all-topics-strings.lc.txt'.

```
def main():
    files_to_process = [
        'all-exchanges-strings.lc.txt',
        'all-orgs-strings.lc.txt',
        'all-people-strings.lc.txt',
        'all-places-strings.lc.txt',
        'all-topics-strings.lc.txt',
    ]

    # ==== using for loop to go through all 5 files ====
    for file_name in files_to_process:
        file_path = os.path.join('./reuters21578.tar/', file_name)
```

#### B. Processing five stopwords files for each input files

```
# ==== using for loop to go through all 5 files ====
for file_name in files_to_process:
    file_path = os.path.join('./reuters21578.tar/', file_name)

    # ====Make the first word of each file as the stopwords and remove it from each file====
    # Stop_words = ["amex", "adb-africa", "abdel-hadi-kandeel", "afghanistan", "acq"]
    stopwords_filename = f'Stopwords_used_for_output_{file_name}'
    #print(stopwords_filename)
    stopwords_file_path = os.path.join('./Stopwords_used_for_output/', stopwords_filename)
    stop_words = Stopword_read(stopwords_file_path)
```

### C. Tokenization

Tokenizes the text and prints the tokens.

```
64 # ==== print for the tokens that been tokenized
65 tokens = Tokenizer_output(file_path)
66 print(f'Tokenized File:{file_name}')
67 print(tokens)
68 print('\n\n')
```

And then save the tokens output into new txt files

```
70 # ==== save the output of Tokenized file in to new txt file with name of 'Tokenizer_output_filename'
71 Tokenizer_file_path = f'./Tokenizer_output/Tokenizer_output_{file_name}.txt'
72 with open(Tokenizer_file_path, 'w') as tokenizer_file:
73     tokenizer_file.write('\n'.join(tokens))
74 print(f'Tokenizer_output for {file_name} has been saved \n')
```

The sample output for 'all-exchanges-strings.lc.txt' as following

```
Tokenized File:all-exchanges-strings.lc.txt
['amex', 'ase', 'asx', 'biffex', 'bse', 'cboe', 'cbt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse',
'liffe', 'lme', 'lse', 'mase', 'mise', 'mnse', 'mose', 'nasdaq', 'nyce', 'nycscse', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex', 'sse',
'stse', 'tose', 'tse', 'wce', 'zse']

Tokenizer_output for all-exchanges-strings.lc.txt has been saved
```

### D. Lowercasing

Converts the tokens to lowercase and prints them.

```
76 # ==== print for the tokens that been lower all tokens
77 lower_tokens = Lowercased_output(file_path)
78 print(f'Lowercase File:{file_name}')
79 print(lower_tokens)
80 print('\n\n')
```

And then save the lowercase output into new txt files

```
82 # ==== save the output of all text lowercase file in to new txt file with name of 'Lowercased_output_filename'
83 lower_file_path = f'./Lowercased_output/Lowercased_output_{file_name}.txt'
84 with open(lower_file_path, 'w') as lower_file:
85     lower_file.write('\n'.join(lower_tokens))
86 print(f'Lowercased_output for {file_name} has been saved \n')
```

The sample output for 'all-exchanges-strings.lc.txt' as following

```
Lowercase File:all-exchanges-strings.lc.txt
['amex', 'ase', 'asx', 'biffex', 'bse', 'cboe', 'cbt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse',
'liffe', 'lme', 'lse', 'mase', 'mise', 'mnse', 'mose', 'nasdaq', 'nyce', 'nycscse', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex', 'sse',
'stse', 'tose', 'tse', 'wce', 'zse']

Lowercased_output for all-exchanges-strings.lc.txt has been saved
```

## E. Stemming

Applies stemming to the tokens and prints them.

```
88 # ==== print for the tokens by using Porter stemmer
89 stemmed_tokens = Stemmed_output(file_path)
90 print(f'Stemmed File:{file_name}')
91 print(stemmed_tokens)
92 print('\n\n')
```

And then save the stemmed output into new txt files

```
94 # ==== save the output of the tokens by using Porter stemmer file in to new txt file with name of 'stemmed_out
95 stemmed_file_path = f'./Stemmed_output/Stemmed_output_{file_name}.txt'
96 with open(stemmed_file_path, 'w') as stemmed_file:
97     stemmed_file.write('\n'.join(stemmed_tokens))
98 print(f'Stemmed_output for {file_name} has been saved \n')
```

The sample output for 'all-exchanges-strings.lc.txt' as following

```
Stemmed File:all-exchanges-strings.lc.txt
['amex', 'ase', 'asx', 'biffex', 'bse', 'cboe', 'cvt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse', 'liff',
'lme', 'lse', 'mase', 'mise', 'mnse', 'mose', 'nasdaq', 'nyce', 'nycsc', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex', 'sse', 'stse',
'tose', 'tse', 'wce', 'zse']

Stemmed_output for all-exchanges-strings.lc.txt has been saved
```

As we can see compared to the tokens file, words 'liffe' has changed to 'liff', words 'nycsce' has changed into 'nycsc'.

## F. Stopword Removal

Removes stopwords from the tokens using a custom list and prints the result.

```
100 # ==== print the tokens after remove the stopwords
101 stopword_token = No_Stopword_output(file_path, stop_words, stopwords_file_path)
102 print(f'No Stopwords File:{file_name}')
103 print(stopword_token)
104 print('\n\n')
```

And then save the removed stopwords output into new txt files

```
106 # ==== save the output of the tokens by using Porter stemmer file in to new txt file with name of 'ste
107 No_stopwords_file_path = f'./No_stopword_output/No_stopword_output_{file_name}.txt'
108 with open(No_stopwords_file_path, 'w') as No_stopwords_file:
109     No_stopwords_file.write('\n'.join(stopword_token))
110 print(f'No_stopword_output for {file_name} has been saved \n')
```

The sample output for 'all-exchanges-strings.lc.txt' as following

```
No Stopwords File:all-exchanges-strings.lc.txt
['cboe', 'cvt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse', 'liffe', 'lme', 'lse', 'mase', 'mise',
'mnse', 'mose', 'nasdaq', 'nyce', 'nycsce', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex', 'sse']

No_stopword_output for all-exchanges-strings.lc.txt has been saved
```

As we can see the first and the last 5 words has been taken out of the list.



## Key Strengths

- Modular and organized code structure for different preprocessing tasks.

```
11 def Tokenizer_output(file_path):...
16
17 def Lowercased_output(file_path):...
22
23 def Stemmed_output(file_path):...
29
30 def No_Stopword_output(file_path, stop_words, stopwords_file_path):...
38
39 def Stopword_read(stopwords_file_path):...
```

- Detailed and clear output for each processing step.

```
D:\python\comp479_P1\env\Scripts\python.exe D:\python\comp479_P1\main.py
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\69509\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\69509\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Tokenized File:all-exchanges-strings.lc.txt
['amex', 'ase', 'asx', 'biffex', 'bse', 'cboe', 'cbt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse',
'liffe', 'lme', 'lse', 'mase', 'mise', 'mnse', 'mose', 'nasdaq', 'nyce', 'nycsce', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex',
'sse', 'stse', 'tose', 'tse', 'wce', 'zse']

Tokenizer_output for all-exchanges-strings.lc.txt has been saved

Lowercase File:all-exchanges-strings.lc.txt
['amex', 'ase', 'asx', 'biffex', 'bse', 'cboe', 'cbt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse',
'liffe', 'lme', 'lse', 'mase', 'mise', 'mnse', 'mose', 'nasdaq', 'nyce', 'nycsce', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex',
'sse', 'stse', 'tose', 'tse', 'wce', 'zse']

Lowercased_output for all-exchanges-strings.lc.txt has been saved

Stemmed File:all-exchanges-strings.lc.txt
['amex', 'ase', 'asx', 'biffex', 'bse', 'cboe', 'cbt', 'cme', 'comex', 'cse', 'fox', 'fse', 'hkse', 'ipe', 'jse', 'klce', 'klse',
'liffe', 'lme', 'lse', 'mase', 'mise', 'mnse', 'mose', 'nasdaq', 'nyce', 'nycsce', 'nymex', 'nyse', 'ose', 'pse', 'set', 'simex',
'sse', 'stse', 'tose', 'tse', 'wce', 'zse']
```

- Output saved in separate files for each step.

