R12725049 徐尚淵 作業一

1. 執行環境：Jupyter Notebook
2. 程式語言：Python (版本 3.11.49/24/2023)

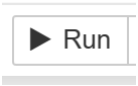
```
In [1]: from platform import python_version

        print(python_version())

        3.11.4
```

3. 執行方式：

   直接使用 Jupyter Notebook GUI Run code

4. 作業處理邏輯說明

   1. 先讀取網址中的文字內容

      Read Text Data

```
In [2]: import requests

        url = "https://ceiba.ntu.edu.tw/course/35d27d/content/28.txt"

        try:
            response = requests.get(url)
            response.raise_for_status()  # 檢查是否有錯誤的HTTP requests

            # 獲取網頁內容
            raw_data = response.text

            print(raw_data)

        except requests.exceptions.RequestException as e:
            print(f"發生錯誤: {e}")

        And Yugoslav authorities are planning the arrest of eleven coal miners
        and two opposition politicians on suspicion of sabotage, that's in
        connection with strike action against President Slobodan Milosevic.
        You are listening to BBC news for The World.
```

   2. Lowercasing everything：於字串狀態時就處理大小寫

      Lowercasing everything

```
# 將文字轉換為小寫
lower_data = raw_data.lower()
print(lower_data)

and yugoslav authorities are planning the arrest of eleven coal miners
and two opposition politicians on suspicion of sabotage, that's in
connection with strike action against president slobodan milosevic.
you are listening to bbc news for the world.
```

3. Tokenization：將單字切分成為 token

Tokenization

```
def tokenize_text(text):
    # empty List 用以儲存Tokens
    tokens = []

    # empty String 用以儲存單字
    current_token = ""

    # 追蹤每個字母
    for char in text:
        # 如果字母是空格或標點符號，並且current string is not empty，則將其添加到tokens List
        if char.isspace() or char in ('.', ',', '!', '?', ';', ':'):
            if current_token:
                tokens.append(current_token)
                current_token = ""
        else:
            # 如果字母不是空格或標點符號，則將其添加到current token
            current_token += char

    # 將最後一個word添加到tokens列表中
    if current_token:
        tokens.append(current_token)

    return tokens

text = lower_data
tokens = tokenize_text(text)
print(tokens)
```

```
['and', 'yugoslav', 'authorities', 'are', 'planning', 'the', 'arrest', 'of', 'eleven', 'coal', 'miners', 'and', 'two', 'opposit
ion', 'politicians', 'on', 'suspicion', 'of', 'sabotage', "that's", 'in', 'connection', 'with', 'strike', 'action', 'against',
'president', 'slobodan', 'milosevic', 'you', 'are', 'listening', 'to', 'bbc', 'news', 'for', 'the', 'world']
```

4. Stopword removal：移除沒有明顯語意的 stopword

Stopword removal

```
# 讀取stopwords
stopwords_file = open("NLTK's list of english stopwords.txt", "r")
stopwords = stopwords_file.read()
print(stopwords)
```

```
stopwords_list = stopwords.splitlines()
print(stopwords_list)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'h
is', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'wha
t', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have',
'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'wh
ile', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above',
'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'th
ere', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor',
'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now']
```

5. Stemming using Porters algorithm：最後才執行 Stemming

Stemming using Porter's algorithm

```
# import package
from nltk.stem.porter import PorterStemmer
import nltk

# stemmer
ps=PorterStemmer()
```

```
# stemming
result = []
for t in filtered_tokens:
    print(t, " : ", ps.stem(t))
    result.append(ps.stem(t))
```