NLP PROGRAM-2

A program to count word frequency and to remove stop words

Soundarya G_ 2048057

Import necessary libraries

```
import nltk
nltk.download("all")
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /root/nltk_data...
[nltk_data]    |   Package abc is already up-to-date!
[nltk_data]    | Downloading package alpino to /root/nltk_data...
[nltk_data]    |   Package alpino is already up-to-date!
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package biocreative_ppi is already up-to-date!
[nltk_data]    | Downloading package brown to /root/nltk_data...
[nltk_data]    |   Package brown is already up-to-date!
[nltk_data]    | Downloading package brown_tei to /root/nltk_data...
[nltk_data]    |   Package brown_tei is already up-to-date!
[nltk_data]    | Downloading package cess_cat to /root/nltk_data...
[nltk_data]    |   Package cess_cat is already up-to-date!
[nltk_data]    | Downloading package cess_esp to /root/nltk_data...
[nltk_data]    |   Package cess_esp is already up-to-date!
[nltk_data]    | Downloading package chat80 to /root/nltk_data...
[nltk_data]    |   Package chat80 is already up-to-date!
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package city_database is already up-to-date!
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package comparative_sentences to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package comparative_sentences is already up-to-
[nltk_data]    |       date!
[nltk_data]    | Downloading package comtrans to /root/nltk_data...
[nltk_data]    |   Package comtrans is already up-to-date!
[nltk_data]    | Downloading package conll2000 to /root/nltk_data...
[nltk_data]    |   Package conll2000 is already up-to-date!
[nltk_data]    | Downloading package conll2002 to /root/nltk_data...
[nltk_data]    |   Package conll2002 is already up-to-date!
[nltk_data]    | Downloading package conll2007 to /root/nltk_data...
[nltk_data]    |   Package conll2007 is already up-to-date!
[nltk_data]    | Downloading package crubadan to /root/nltk_data...
[nltk_data]    |   Package crubadan is already up-to-date!
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package dependency_treebank is already up-to-date!
[nltk_data]    | Downloading package dolch to /root/nltk_data...
[nltk_data]    |   Package dolch is already up-to-date!
[nltk_data]    | Downloading package europarl_raw to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package europarl_raw is already up-to-date!
[nltk_data]    | Downloading package floresta to /root/nltk_data...
[nltk_data]    |   Package floresta is already up-to-date!
[nltk_data]    | Downloading package framenet_v15 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package framenet_v15 is already up-to-date!
[nltk_data]    | Downloading package framenet_v17 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package framenet_v17 is already up-to-date!
```

```
[nltk_data]    | Downloading package gazetteers to /root/nltk_data...
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to /root/nltk_data...
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to /root/nltk_data...
```

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk import FreqDist
```

List of stop words.

```
print(set(stopwords.words('english')))
```

```
{'on', "don't", 'be', 'than', 'because', 'doesn', 'mightn', 'which', "needn't", 'where',
```

Shows how stop words are removed

```
example_sent = "This is a sample sentence,showing off the stop words filtration."

stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)

filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]
filtered_sentence = []

for w in word_tokens:
  if w not in stop_words:
    filtered_sentence.append(w)

print('Tokenized sentence:\n\t',word_tokens)
print('Removed Stop words:\n\t',filtered_sentence)
```

```
Tokenized sentence:
         ['This', 'is', 'a', 'sample', 'sentence', ',', 'showing', 'off', 'the', 'stop',
Removed Stop words:
         ['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration',
```

Frequency of words after tokenization

```
text = "Learn and practice and learn to practice"
words = text.split()
fdist1 = FreqDist(words)
print(fdist1)
print()
fdist1.most_common()
```

```
<FreqDist with 5 samples and 7 outcomes>

[('and', 2), ('practice', 2), ('Learn', 1), ('learn', 1), ('to', 1)]
```

Parts of Speech for tokens

```
def parts_of_speech(txt):
  import spacy
  sp = spacy.load('en_core_web_sm')
```

```
  sentence = sp(txt)
  print("Parts of speech with tokens:")
  print("==========================")
  for word in sentence:
    print("\t",word.pos_,'\t-', word.text)
```

```
parts_of_speech(text)
```

```
     Parts of speech with tokens:
     ==========================
              VERB   - Learn
              CCONJ  - and
              VERB   - practice
              CCONJ  - and
              VERB   - learn
              PART   - to
              VERB   - practice
```

Main Code

```
text = '''Celebrate Independence Day every year on 15 August as a national holiday in India t
```

```
def word_count(text):
  from nltk import FreqDist
  from nltk.corpus import stopwords
  from nltk.tokenize import word_tokenize

  stop_words = set(stopwords.words('english'))
  word_tokens = word_tokenize(text)

  filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]

  for w in word_tokens:
    if w not in stop_words:
      filtered_sentence.append(w)

  fdist1 = FreqDist(filtered_sentence)
  print(fdist1)
  list1 = fdist1.most_common()
  print("\n\n\t Count",'- Tokens')
  for i in list1:
     print("\t",i[1],'-', i[0])
```

```
def token():
  print('PROGRAM 2')
  print('=========')
  option = 'yes'
  while(option=='yes'):

    print('\t1. Count \n\t2. Parts of Speech')
    op = input('Select your option:')
    print('\nEnter the sentence to be tokenized:')
    text = input('\t')
    if op == '1':
      word_count(text)
    elif op == '2':
      parts_of_speech(text)
    else:
      print('Invalid input')

    option=input('\n Do you want to continue[yes/no]:\t')
```

```
token()
```

```
PROPN  - India
PART   - to
VERB   - commemorate
DET    - the
NOUN   - independence
ADP    - of
DET    - the
NOUN   - nation
ADP    - from
DET    - the
PROPN  - United
PROPN  - Kingdom
ADP    - on
NUM    - 15
PROPN  - August
NUM    - 1947
PUNCT  - .
NOUN   - Day
ADP    - on
DET    - which
DET    - the
NOUN   - provisions
ADP    - of
DET    - the
PROPN  - Indian
PROPN  - Independence
PROPN  - Act
ADP    - of
NUM    - 1947
VERB   - came
ADP    - into
NOUN   - effect
PUNCT  - ,
DET    - which
VERB   - transferred
ADJ    - legislative
NOUN   - sovereignty
ADP    - to
DET    - the
PROPN  - Indian
PROPN  - Constituent
PROPN  - Assembly
PUNCT  - .
NOUN   - Independence
VERB   - corresponded
ADP    - with
PROPN  - India
PART   - 's
NOUN   - partition
PUNCT  - ,
ADV    - wherein
PROPN  - British
PROPN  - India
AUX    - had
SPACE  -
AUX    - been
VERB   - divided
ADP    - into
DET    - the
```