NLP & WORD CLOUD

NLP

```
In [1]: import os
        import nltk
```

```
In [2]: print("The nltk version is {}. ".format(nltk.__version__))
```

```
The nltk version is 3.9.1.
```

```
In [3]: nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
Out[3]: True
```

```
In [4]: import nltk.corpus
```

```
In [5]: from nltk.corpus import brown
        brown.words()
```

```
Out[5]: ['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

```
In [6]: nltk.corpus.brown.fileids()
```

```
Out[6]:  ['ca01',
          'ca02',
          'ca03',
          'ca04',
          'ca05',
          'ca06',
          'ca07',
          'ca08',
          'ca09',
          'ca10',
          'ca11',
          'ca12',
          'ca13',
          'ca14',
          'ca15',
          'ca16',
          'ca17',
          'ca18',
          'ca19',
          'ca20',
          'ca21',
          'ca22',
          'ca23',
          'ca24',
          'ca25',
          'ca26',
          'ca27',
          'ca28',
          'ca29',
          'ca30',
          'ca31',
          'ca32',
          'ca33',
          'ca34',
          'ca35',
          'ca36',
          'ca37',
          'ca38',
          'ca39',
          'ca40',
          'ca41',
          'ca42',
          'ca43',
          'ca44',
          'cb01',
          'cb02',
          'cb03',
          'cb04',
          'cb05',
          'cb06',
          'cb07',
          'cb08',
          'cb09',
          'cb10',
          'cb11',
          'cb12',
          'cb13',
          'cb14',
          'cb15',
          'cb16',
'cb17',
'cb18',
'cb19',
'cb20',
'cb21',
'cb22',
'cb23',
'cb24',
'cb25',
'cb26',
'cb27',
'cc01',
'cc02',
'cc03',
'cc04',
'cc05',
'cc06',
'cc07',
'cc08',
'cc09',
'cc10',
'cc11',
'cc12',
'cc13',
'cc14',
'cc15',
'cc16',
'cc17',
'cd01',
'cd02',
'cd03',
'cd04',
'cd05',
'cd06',
'cd07',
'cd08',
'cd09',
'cd10',
'cd11',
'cd12',
'cd13',
'cd14',
'cd15',
'cd16',
'cd17',
'ce01',
'ce02',
'ce03',
'ce04',
'ce05',
'ce06',
'ce07',
'ce08',
'ce09',
'ce10',
'ce11',
'ce12',
'ce13',
'ce14',
'ce15',
            'ce16',
            'ce17',
            'ce18',
            'ce19',
            'ce20',
            'ce21',
            'ce22',
            'ce23',
            'ce24',
            'ce25',
            'ce26',
            'ce27',
            'ce28',
            'ce29',
            'ce30',
            'ce31',
            'ce32',
            'ce33',
            'ce34',
            'ce35',
            'ce36',
            'cf01',
            'cf02',
            'cf03',
            'cf04',
            'cf05',
            'cf06',
            'cf07',
            'cf08',
            'cf09',
            'cf10',
            'cf11',
            'cf12',
            'cf13',
            'cf14',
            'cf15',
            'cf16',
            'cf17',
            'cf18',
            'cf19',
            'cf20',
            'cf21',
            'cf22',
            'cf23',
            'cf24',
            'cf25',
            'cf26',
            'cf27',
            'cf28',
            'cf29',
            'cf30',
            'cf31',
            'cf32',
            'cf33',
            'cf34',
            'cf35',
            'cf36',
            'cf37',
            'cf38',
            'cf39',
'cf40',
'cf41',
'cf42',
'cf43',
'cf44',
'cf45',
'cf46',
'cf47',
'cf48',
'cg01',
'cg02',
'cg03',
'cg04',
'cg05',
'cg06',
'cg07',
'cg08',
'cg09',
'cg10',
'cg11',
'cg12',
'cg13',
'cg14',
'cg15',
'cg16',
'cg17',
'cg18',
'cg19',
'cg20',
'cg21',
'cg22',
'cg23',
'cg24',
'cg25',
'cg26',
'cg27',
'cg28',
'cg29',
'cg30',
'cg31',
'cg32',
'cg33',
'cg34',
'cg35',
'cg36',
'cg37',
'cg38',
'cg39',
'cg40',
'cg41',
'cg42',
'cg43',
'cg44',
'cg45',
'cg46',
'cg47',
'cg48',
'cg49',
'cg50',
'cg51',
            'cg52',
            'cg53',
            'cg54',
            'cg55',
            'cg56',
            'cg57',
            'cg58',
            'cg59',
            'cg60',
            'cg61',
            'cg62',
            'cg63',
            'cg64',
            'cg65',
            'cg66',
            'cg67',
            'cg68',
            'cg69',
            'cg70',
            'cg71',
            'cg72',
            'cg73',
            'cg74',
            'cg75',
            'ch01',
            'ch02',
            'ch03',
            'ch04',
            'ch05',
            'ch06',
            'ch07',
            'ch08',
            'ch09',
            'ch10',
            'ch11',
            'ch12',
            'ch13',
            'ch14',
            'ch15',
            'ch16',
            'ch17',
            'ch18',
            'ch19',
            'ch20',
            'ch21',
            'ch22',
            'ch23',
            'ch24',
            'ch25',
            'ch26',
            'ch27',
            'ch28',
            'ch29',
            'ch30',
            'cj01',
            'cj02',
            'cj03',
            'cj04',
            'cj05',
            'cj06',
    'cj07',
    'cj08',
    'cj09',
    'cj10',
    'cj11',
    'cj12',
    'cj13',
    'cj14',
    'cj15',
    'cj16',
    'cj17',
    'cj18',
    'cj19',
    'cj20',
    'cj21',
    'cj22',
    'cj23',
    'cj24',
    'cj25',
    'cj26',
    'cj27',
    'cj28',
    'cj29',
    'cj30',
    'cj31',
    'cj32',
    'cj33',
    'cj34',
    'cj35',
    'cj36',
    'cj37',
    'cj38',
    'cj39',
    'cj40',
    'cj41',
    'cj42',
    'cj43',
    'cj44',
    'cj45',
    'cj46',
    'cj47',
    'cj48',
    'cj49',
    'cj50',
    'cj51',
    'cj52',
    'cj53',
    'cj54',
    'cj55',
    'cj56',
    'cj57',
    'cj58',
    'cj59',
    'cj60',
    'cj61',
    'cj62',
    'cj63',
    'cj64',
    'cj65',
    'cj66',
            'cj67',
            'cj68',
            'cj69',
            'cj70',
            'cj71',
            'cj72',
            'cj73',
            'cj74',
            'cj75',
            'cj76',
            'cj77',
            'cj78',
            'cj79',
            'cj80',
            'ck01',
            'ck02',
            'ck03',
            'ck04',
            'ck05',
            'ck06',
            'ck07',
            'ck08',
            'ck09',
            'ck10',
            'ck11',
            'ck12',
            'ck13',
            'ck14',
            'ck15',
            'ck16',
            'ck17',
            'ck18',
            'ck19',
            'ck20',
            'ck21',
            'ck22',
            'ck23',
            'ck24',
            'ck25',
            'ck26',
            'ck27',
            'ck28',
            'ck29',
            'cl01',
            'cl02',
            'cl03',
            'cl04',
            'cl05',
            'cl06',
            'cl07',
            'cl08',
            'cl09',
            'cl10',
            'cl11',
            'cl12',
            'cl13',
            'cl14',
            'cl15',
            'cl16',
            'cl17',
            'cl18',
            'cl19',
            'cl20',
            'cl21',
            'cl22',
            'cl23',
            'cl24',
            'cm01',
            'cm02',
            'cm03',
            'cm04',
            'cm05',
            'cm06',
            'cn01',
            'cn02',
            'cn03',
            'cn04',
            'cn05',
            'cn06',
            'cn07',
            'cn08',
            'cn09',
            'cn10',
            'cn11',
            'cn12',
            'cn13',
            'cn14',
            'cn15',
            'cn16',
            'cn17',
            'cn18',
            'cn19',
            'cn20',
            'cn21',
            'cn22',
            'cn23',
            'cn24',
            'cn25',
            'cn26',
            'cn27',
            'cn28',
            'cn29',
            'cp01',
            'cp02',
            'cp03',
            'cp04',
            'cp05',
            'cp06',
            'cp07',
            'cp08',
            'cp09',
            'cp10',
            'cp11',
            'cp12',
            'cp13',
            'cp14',
            'cp15',
            'cp16',
            'cp17',
            'cp18',
 'cp19',
 'cp20',
 'cp21',
 'cp22',
 'cp23',
 'cp24',
 'cp25',
 'cp26',
 'cp27',
 'cp28',
 'cp29',
 'cr01',
 'cr02',
 'cr03',
 'cr04',
 'cr05',
 'cr06',
 'cr07',
 'cr08',
 'cr09']
```

In [7]:
```
nltk.corpus.gutenberg
```

Out[7]:
```
<PlaintextCorpusReader in 'C:\\Users\\roy62\\AppData\\Roaming\\nltk_data\\corpo
ra\\gutenberg'>
```

In [8]:
```
nltk.corpus.gutenberg.fileids()
```

Out[8]:
```
['austen-emma.txt',
 'austen-persuasion.txt',
 'austen-sense.txt',
 'bible-kjv.txt',
 'blake-poems.txt',
 'bryant-stories.txt',
 'burgess-busterbrown.txt',
 'carroll-alice.txt',
 'chesterton-ball.txt',
 'chesterton-brown.txt',
 'chesterton-thursday.txt',
 'edgeworth-parents.txt',
 'melville-moby_dick.txt',
 'milton-paradise.txt',
 'shakespeare-caesar.txt',
 'shakespeare-hamlet.txt',
 'shakespeare-macbeth.txt',
 'whitman-leaves.txt']
```

In [9]:
```
AI = '''Artificial Intelligence refers to the intelligence of machines. This is
humans and animals. With Artificial Intelligence, machines perform functions suc
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of h
It is probably the fastest-growing development in the World of technology and in
AI could solve major challenges and crisis situations.'''
```

In [10]:
```
AI
```

```
Out[10]: 'Artificial Intelligence refers to the intelligence of machines. This is in con
         trast to the natural intelligence of \nhumans and animals. With Artificial Inte
         lligence, machines perform functions such as learning, planning, reasoning and
         \nproblem-solving. Most noteworthy, Artificial Intelligence is the simulation o
         f human intelligence by machines. \nIt is probably the fastest-growing developm
         ent in the World of technology and innovation. Furthermore, many experts believ
         e\nAI could solve major challenges and crisis situations.'
```

```
In [11]: type(AI)
```

```
Out[11]: str
```

```
In [12]: from nltk.tokenize import word_tokenize
```

```
In [13]: AI_tokens = word_tokenize(AI)
         AI_tokens
```

```
Out[13]: ['Artificial',
          'Intelligence',
          'refers',
          'to',
          'the',
          'intelligence',
          'of',
          'machines',
          '.',
          'This',
          'is',
          'in',
          'contrast',
          'to',
          'the',
          'natural',
          'intelligence',
          'of',
          'humans',
          'and',
          'animals',
          '.',
          'With',
          'Artificial',
          'Intelligence',
          ',',
          'machines',
          'perform',
          'functions',
          'such',
          'as',
          'learning',
          ',',
          'planning',
          ',',
          'reasoning',
          'and',
          'problem-solving',
          '.',
          'Most',
          'noteworthy',
          ',',
          'Artificial',
          'Intelligence',
          'is',
          'the',
          'simulation',
          'of',
          'human',
          'intelligence',
          'by',
          'machines',
          '.',
          'It',
          'is',
          'probably',
          'the',
          'fastest-growing',
          'development',
          'in',
          'the',
          'World',
          'of',
          'technology',
          'and',
          'innovation',
          '.',
          'Furthermore',
          ',',
          'many',
          'experts',
          'believe',
          'AI',
          'could',
          'solve',
          'major',
          'challenges',
          'and',
          'crisis',
          'situations',
          '.']
```

In [14]:
```
len(AI_tokens)
```

Out[14]:
```
81
```

In [15]:
```
from nltk.tokenize import sent_tokenize
```

In [16]:
```
AI_sent = sent_tokenize(AI)
AI_sent
```

Out[16]:
```
['Artificial Intelligence refers to the intelligence of machines.',
 'This is in contrast to the natural intelligence of \nhumans and animals.',
 'With Artificial Intelligence, machines perform functions such as learning, pl
anning, reasoning and \nproblem-solving.',
 'Most noteworthy, Artificial Intelligence is the simulation of human intellige
nce by machines.',
 'It is probably the fastest-growing development in the World of technology and
innovation.',
 'Furthermore, many experts believe\nAI could solve major challenges and crisis
situations.']
```

In [17]:
```
len(AI_sent)
```

Out[17]:
```
6
```

In [18]:
```
AI
```

Out[18]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of \nhumans and animals. With Artificial Inte
lligence, machines perform functions such as learning, planning, reasoning and
\nproblem-solving. Most noteworthy, Artificial Intelligence is the simulation o
f human intelligence by machines. \nIt is probably the fastest-growing developm
ent in the World of technology and innovation. Furthermore, many experts believ
e\nAI could solve major challenges and crisis situations.'
```

In [19]:
```
from nltk.tokenize import blankline_tokenize # GiVE YOU HOW MANY PARAGRAPH
AI_blank = blankline_tokenize(AI)
AI_blank
```

```
Out[19]: ['Artificial Intelligence refers to the intelligence of machines. This is in co
ntrast to the natural intelligence of \nhumans and animals. With Artificial Int
elligence, machines perform functions such as learning, planning, reasoning and
\nproblem-solving. Most noteworthy, Artificial Intelligence is the simulation o
f human intelligence by machines. \nIt is probably the fastest-growing developm
ent in the World of technology and innovation. Furthermore, many experts believ
e\nAI could solve major challenges and crisis situations.']
```

```
In [20]: len(AI_blank)
```

```
Out[20]: 1
```

```
In [21]: from nltk.util import bigrams,trigrams,ngrams
```

```
In [22]: string = 'the best and most beautifull thing in the world cannot be seen or even
         quotes_tokens = nltk.word_tokenize(string)
```

```
In [23]: quotes_tokens
```

```
Out[23]: ['the',
          'best',
          'and',
          'most',
          'beautifull',
          'thing',
          'in',
          'the',
          'world',
          'can',
          'not',
          'be',
          'seen',
          'or',
          'even',
          'touched',
          ',',
          'they',
          'must',
          'be',
          'felt',
          'with',
          'heart']
```

```
In [24]: len(quotes_tokens)
```

```
Out[24]: 23
```

```
In [25]: quotes_bigrams = list(nltk.bigrams(quotes_tokens))
         quotes_bigrams
```

```
Out[25]: [('the', 'best'),
          ('best', 'and'),
          ('and', 'most'),
          ('most', 'beautifull'),
          ('beautifull', 'thing'),
          ('thing', 'in'),
          ('in', 'the'),
          ('the', 'world'),
          ('world', 'can'),
          ('can', 'not'),
          ('not', 'be'),
          ('be', 'seen'),
          ('seen', 'or'),
          ('or', 'even'),
          ('even', 'touched'),
          ('touched', ','),
          (',', 'they'),
          ('they', 'must'),
          ('must', 'be'),
          ('be', 'felt'),
          ('felt', 'with'),
          ('with', 'heart')]
```

```
In [26]: quotes_tokens
```

```
Out[26]: ['the',
          'best',
          'and',
          'most',
          'beautifull',
          'thing',
          'in',
          'the',
          'world',
          'can',
          'not',
          'be',
          'seen',
          'or',
          'even',
          'touched',
          ',',
          'they',
          'must',
          'be',
          'felt',
          'with',
          'heart']
```

```
In [27]: quotes_trigrams = list(nltk.trigrams(quotes_tokens))
         quotes_trigrams
```

```
Out[27]: [('the', 'best', 'and'),
          ('best', 'and', 'most'),
          ('and', 'most', 'beautifull'),
          ('most', 'beautifull', 'thing'),
          ('beautifull', 'thing', 'in'),
          ('thing', 'in', 'the'),
          ('in', 'the', 'world'),
          ('the', 'world', 'can'),
          ('world', 'can', 'not'),
          ('can', 'not', 'be'),
          ('not', 'be', 'seen'),
          ('be', 'seen', 'or'),
          ('seen', 'or', 'even'),
          ('or', 'even', 'touched'),
          ('even', 'touched', ','),
          ('touched', ',', 'they'),
          (',', 'they', 'must'),
          ('they', 'must', 'be'),
          ('must', 'be', 'felt'),
          ('be', 'felt', 'with'),
          ('felt', 'with', 'heart')]
```

```
In [28]: quotes_trigrams = list(nltk.ngrams(quotes_tokens))
         quotes_trigrams
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 quotes_trigrams = list(nltk.ngrams(quotes_tokens))
      2 quotes_trigrams

TypeError: ngrams() missing 1 required positional argument: 'n'
```

```
In [29]: quotes_ngrams = list(nltk.ngrams(quotes_tokens, 4))
         quotes_ngrams
```

```
Out[29]: [('the', 'best', 'and', 'most'),
          ('best', 'and', 'most', 'beautifull'),
          ('and', 'most', 'beautifull', 'thing'),
          ('most', 'beautifull', 'thing', 'in'),
          ('beautifull', 'thing', 'in', 'the'),
          ('thing', 'in', 'the', 'world'),
          ('in', 'the', 'world', 'can'),
          ('the', 'world', 'can', 'not'),
          ('world', 'can', 'not', 'be'),
          ('can', 'not', 'be', 'seen'),
          ('not', 'be', 'seen', 'or'),
          ('be', 'seen', 'or', 'even'),
          ('seen', 'or', 'even', 'touched'),
          ('or', 'even', 'touched', ','),
          ('even', 'touched', ',', 'they'),
          ('touched', ',', 'they', 'must'),
          (',', 'they', 'must', 'be'),
          ('they', 'must', 'be', 'felt'),
          ('must', 'be', 'felt', 'with'),
          ('be', 'felt', 'with', 'heart')]
```

```
In [30]: len(quotes_tokens)
```

```
Out[30]: 23
```

```
In [31]: quotes_ngrams_1 = list(nltk.ngrams(quotes_tokens, 5))
         quotes_ngrams_1
```

```
Out[31]: [('the', 'best', 'and', 'most', 'beautifull'),
          ('best', 'and', 'most', 'beautifull', 'thing'),
          ('and', 'most', 'beautifull', 'thing', 'in'),
          ('most', 'beautifull', 'thing', 'in', 'the'),
          ('beautifull', 'thing', 'in', 'the', 'world'),
          ('thing', 'in', 'the', 'world', 'can'),
          ('in', 'the', 'world', 'can', 'not'),
          ('the', 'world', 'can', 'not', 'be'),
          ('world', 'can', 'not', 'be', 'seen'),
          ('can', 'not', 'be', 'seen', 'or'),
          ('not', 'be', 'seen', 'or', 'even'),
          ('be', 'seen', 'or', 'even', 'touched'),
          ('seen', 'or', 'even', 'touched', ','),
          ('or', 'even', 'touched', ',', 'they'),
          ('even', 'touched', ',', 'they', 'must'),
          ('touched', ',', 'they', 'must', 'be'),
          (',', 'they', 'must', 'be', 'felt'),
          ('they', 'must', 'be', 'felt', 'with'),
          ('must', 'be', 'felt', 'with', 'heart')]
```

```
In [32]: quotes_ngrams = list(nltk.ngrams(quotes_tokens, 9))
         quotes_ngrams
```

```
Out[32]: [('the', 'best', 'and', 'most', 'beautifull', 'thing', 'in', 'the', 'world'),
          ('best', 'and', 'most', 'beautifull', 'thing', 'in', 'the', 'world', 'can'),
          ('and', 'most', 'beautifull', 'thing', 'in', 'the', 'world', 'can', 'not'),
          ('most', 'beautifull', 'thing', 'in', 'the', 'world', 'can', 'not', 'be'),
          ('beautifull', 'thing', 'in', 'the', 'world', 'can', 'not', 'be', 'seen'),
          ('thing', 'in', 'the', 'world', 'can', 'not', 'be', 'seen', 'or'),
          ('in', 'the', 'world', 'can', 'not', 'be', 'seen', 'or', 'even'),
          ('the', 'world', 'can', 'not', 'be', 'seen', 'or', 'even', 'touched'),
          ('world', 'can', 'not', 'be', 'seen', 'or', 'even', 'touched', ','),
          ('can', 'not', 'be', 'seen', 'or', 'even', 'touched', ',', 'they'),
          ('not', 'be', 'seen', 'or', 'even', 'touched', ',', 'they', 'must'),
          ('be', 'seen', 'or', 'even', 'touched', ',', 'they', 'must', 'be'),
          ('seen', 'or', 'even', 'touched', ',', 'they', 'must', 'be', 'felt'),
          ('or', 'even', 'touched', ',', 'they', 'must', 'be', 'felt', 'with'),
          ('even', 'touched', ',', 'they', 'must', 'be', 'felt', 'with', 'heart')]
```

```
In [33]: #porter-stemmer
         from nltk.stem import PorterStemmer
         pst = PorterStemmer()
```

```
In [34]: pst.stem('having')
```

```
Out[34]: 'have'
```

```
In [35]: pst.stem('affection')
```

```
Out[35]: 'affect'
```

```
In [36]: pst.stem('playing')
```

Out[36]: 'play'

In [37]:
```
pst.stem('give')
```

Out[37]: 'give'

In [38]:
```
words_to_stem=['give','giving','given','gave']
for words in words_to_stem:
    print(words+  ':' + pst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
```

In [39]:
```
pst.stem('playing')
```

Out[39]: 'play'

In [40]:
```
words_to_stem=['give','giving','given','gave','thinking', 'loving', 'final', 'fi
# i am giving these different words to stem, using porter stemmer we get the out

for words in words_to_stem:
    print(words+ ':' +pst.stem(words))

#in porterstemmer removes ing and replaces with e
```

```
give:give
giving:give
given:given
gave:gave
thinking:think
loving:love
final:final
finalized:final
finally:final
```

In [41]:
```
#another stemmer known as lencastemmer stemmer and lets see what the different w
#stem the same thing using lencastemmer

from nltk.stem import LancasterStemmer
lst = LancasterStemmer()
for words in words_to_stem:
    print(words + ':' + lst.stem(words))

# lancasterstemmer is more aggresive then the porterstemmer
```

```
give:giv
giving:giv
given:giv
gave:gav
thinking:think
loving:lov
final:fin
finalized:fin
finally:fin
```

In [42]:
```
words_to_stem=['give','giving','given','gave','thinking', 'loving', 'final', 'fi
# i am giving these different words to stem, using porter stemmer we get the out
```

```
for words in words_to_stem:
    print(words+ ':' +pst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
thinking:think
loving:love
final:final
finalized:final
finally:final
```

In [43]:
```
#we have another stemmer called as snowball stemmer lets see about this snowball

from nltk.stem import SnowballStemmer
sbst = SnowballStemmer('english')
for words in words_to_stem:
    print(words+ ':' +sbst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
thinking:think
loving:love
final:final
finalized:final
finally:final
```

In [44]:
```
#sometime stemming does not work & lets say e.g - fish,fishes & fishing all of t
#one hand stemming will cut the end & lemmatization will take into the morpholog

from nltk.stem import wordnet
from nltk.stem import WordNetLemmatizer
word_lem = WordNetLemmatizer()

#Hear we are going to wordnet dictionary & we are going to import the wordnet le
```

In [45]:
```
words_to_stem
```

Out[45]:
```
['give',
 'giving',
 'given',
 'gave',
 'thinking',
 'loving',
 'final',
 'finalized',
 'finally']
```

In [46]:
```
#word_lem.lemmatize('corpora') #we get output as corpus

#refers to a collection of texts. Such collections may be formed of a single lan

for words in words_to_stem:
    print(words+ ':' +word_lem.lemmatize(words))
```

```
give:give
giving:giving
given:given
gave:gave
thinking:thinking
loving:loving
final:final
finalized:finalized
finally:finally
```

```
In [47]: pst.stem('final')

Out[47]: 'final'

In [48]: lst.stem('finally')

Out[48]: 'fin'

In [49]: sbst.stem('finalized')

Out[49]: 'final'

In [50]: lst.stem('final')

Out[50]: 'fin'

In [51]: lst.stem('finalized')

Out[51]: 'fin'

In [52]: from nltk.corpus import stopwords

In [53]: stopwords.words('english')
```

```
Out[53]: ['i',
          'me',
          'my',
          'myself',
          'we',
          'our',
          'ours',
          'ourselves',
          'you',
          "you're",
          "you've",
          "you'll",
          "you'd",
          'your',
          'yours',
          'yourself',
          'yourselves',
          'he',
          'him',
          'his',
          'himself',
          'she',
          "she's",
          'her',
          'hers',
          'herself',
          'it',
          "it's",
          'its',
          'itself',
          'they',
          'them',
          'their',
          'theirs',
          'themselves',
          'what',
          'which',
          'who',
          'whom',
          'this',
          'that',
          "that'll",
          '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',
'while',
'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',
'there',
'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',
"don't",
'should',
"should've",
'now',
'd',
'll',
'm',
'o',
're',
've',
'y',
'ain',
'aren',
"aren't",
'couldn',
"couldn't",
'didn',
"didn't",
'doesn',
"doesn't",
'hadn',
"hadn't",
'hasn',
"hasn't",
'haven',
"haven't",
'isn',
"isn't",
'ma',
'mightn',
"mightn't",
'mustn',
"mustn't",
'needn',
"needn't",
'shan',
"shan't",
'shouldn',
"shouldn't",
'wasn',
"wasn't",
'weren',
"weren't",
'won',
"won't",
'wouldn',
"wouldn't"]
```

```
In [54]: len(stopwords.words('english'))
```

```
Out[54]: 179
```

```
In [55]: stopwords.words('spanish')
```

```
Out[55]: ['de',
          'la',
          'que',
          'el',
          'en',
          'y',
          'a',
          'los',
          'del',
          'se',
          'las',
          'por',
          'un',
          'para',
          'con',
          'no',
          'una',
          'su',
          'al',
          'lo',
          'como',
          'más',
          'pero',
          'sus',
          'le',
          'ya',
          'o',
          'este',
          'sí',
          'porque',
          'esta',
          'entre',
          'cuando',
          'muy',
          'sin',
          'sobre',
          'también',
          'me',
          'hasta',
          'hay',
          'donde',
          'quien',
          'desde',
          'todo',
          'nos',
          'durante',
          'todos',
          'uno',
          'les',
          'ni',
          'contra',
          'otros',
          'ese',
          'eso',
          'ante',
          'ellos',
          'e',
          'esto',
          'mí',
          'antes',
'algunos',
'qué',
'unos',
'yo',
'otro',
'otras',
'otra',
'él',
'tanto',
'esa',
'estos',
'mucho',
'quienes',
'nada',
'muchos',
'cual',
'poco',
'ella',
'estar',
'estas',
'algunas',
'algo',
'nosotros',
'mi',
'mis',
'tú',
'te',
'ti',
'tu',
'tus',
'ellas',
'nosotras',
'vosotros',
'vosotras',
'os',
'mío',
'mía',
'míos',
'mías',
'tuyo',
'tuya',
'tuyos',
'tuyas',
'suyo',
'suya',
'suyos',
'suyas',
'nuestro',
'nuestra',
'nuestros',
'nuestras',
'vuestro',
'vuestra',
'vuestros',
'vuestras',
'esos',
'esas',
'estoy',
'estás',
'está',
'estamos',
'estáis',
'están',
'esté',
'estés',
'estemos',
'estéis',
'estén',
'estaré',
'estarás',
'estará',
'estaremos',
'estaréis',
'estarán',
'estaría',
'estarías',
'estaríamos',
'estaríais',
'estarían',
'estaba',
'estabas',
'estábamos',
'estabais',
'estaban',
'estuve',
'estuviste',
'estuvo',
'estuvimos',
'estuvisteis',
'estuvieron',
'estuviera',
'estuvieras',
'estuviéramos',
'estuvierais',
'estuvieran',
'estuviese',
'estuvieses',
'estuviésemos',
'estuvieseis',
'estuviesen',
'estando',
'estado',
'estada',
'estados',
'estadas',
'estad',
'he',
'has',
'ha',
'hemos',
'habéis',
'han',
'haya',
'hayas',
'hayamos',
'hayáis',
'hayan',
'habré',
'habrás',
'habrá',
'habremos',
'habréis',
'habrán',
'habría',
'habrías',
'habríamos',
'habríais',
'habrían',
'había',
'habías',
'habíamos',
'habíais',
'habían',
'hube',
'hubiste',
'hubo',
'hubimos',
'hubisteis',
'hubieron',
'hubiera',
'hubieras',
'hubiéramos',
'hubierais',
'hubieran',
'hubiese',
'hubieses',
'hubiésemos',
'hubieseis',
'hubiesen',
'habiendo',
'habido',
'habida',
'habidos',
'habidas',
'soy',
'eres',
'es',
'somos',
'sois',
'son',
'sea',
'seas',
'seamos',
'seáis',
'sean',
'seré',
'serás',
'será',
'seremos',
'seréis',
'serán',
'sería',
'serías',
'seríamos',
'seríais',
'serían',
'era',
'eras',
'éramos',
'erais',
'eran',
'fui',
'fuiste',
'fue',
'fuimos',
'fuisteis',
'fueron',
'fuera',
'fueras',
'fuéramos',
'fuerais',
'fueran',
'fuese',
'fueses',
'fuésemos',
'fueseis',
'fuesen',
'sintiendo',
'sentido',
'sentida',
'sentidos',
'sentidas',
'siente',
'sentid',
'tengo',
'tienes',
'tiene',
'tenemos',
'tenéis',
'tienen',
'tenga',
'tengas',
'tengamos',
'tengáis',
'tengan',
'tendré',
'tendrás',
'tendrá',
'tendremos',
'tendréis',
'tendrán',
'tendría',
'tendrías',
'tendríamos',
'tendríais',
'tendrían',
'tenía',
'tenías',
'teníamos',
'teníais',
'tenían',
'tuve',
'tuviste',
'tuvo',
'tuvimos',
'tuvisteis',
'tuvieron',
'tuviera',
'tuvieras',
'tuviéramos',
 'tuvierais',
 'tuvieran',
 'tuviese',
 'tuvieses',
 'tuviésemos',
 'tuvieseis',
 'tuviesen',
 'teniendo',
 'tenido',
 'tenida',
 'tenidos',
 'tenidas',
 'tened']
```

```
In [56]: len(stopwords.words('spanish'))

Out[56]: 313

In [57]: stopwords.words('french')
```

```
Out[57]: ['au',
          'aux',
          'avec',
          'ce',
          'ces',
          'dans',
          'de',
          'des',
          'du',
          'elle',
          'en',
          'et',
          'eux',
          'il',
          'ils',
          'je',
          'la',
          'le',
          'les',
          'leur',
          'lui',
          'ma',
          'mais',
          'me',
          'même',
          'mes',
          'moi',
          'mon',
          'ne',
          'nos',
          'notre',
          'nous',
          'on',
          'ou',
          'par',
          'pas',
          'pour',
          'qu',
          'que',
          'qui',
          'sa',
          'se',
          'ses',
          'son',
          'sur',
          'ta',
          'te',
          'tes',
          'toi',
          'ton',
          'tu',
          'un',
          'une',
          'vos',
          'votre',
          'vous',
          'c',
          'd',
          'j',
          'l',
          'à',
          'm',
          'n',
          's',
          't',
          'y',
          'été',
          'étée',
          'étées',
          'étés',
          'étant',
          'étante',
          'étants',
          'étantes',
          'suis',
          'es',
          'est',
          'sommes',
          'êtes',
          'sont',
          'serai',
          'seras',
          'sera',
          'serons',
          'serez',
          'seront',
          'serais',
          'serait',
          'serions',
          'seriez',
          'seraient',
          'étais',
          'était',
          'étions',
          'étiez',
          'étaient',
          'fus',
          'fut',
          'fûmes',
          'fûtes',
          'furent',
          'sois',
          'soit',
          'soyons',
          'soyez',
          'soient',
          'fusse',
          'fusses',
          'fût',
          'fussions',
          'fussiez',
          'fussent',
          'ayant',
          'ayante',
          'ayantes',
          'ayants',
          'eu',
          'eue',
          'eues',
          'eus',
           'ai',
           'as',
           'avons',
           'avez',
           'ont',
           'aurai',
           'auras',
           'aura',
           'aurons',
           'aurez',
           'auront',
           'aurais',
           'aurait',
           'aurions',
           'auriez',
           'auraient',
           'avais',
           'avait',
           'avions',
           'aviez',
           'avaient',
           'eut',
           'eûmes',
           'eûtes',
           'eurent',
           'aie',
           'aies',
           'ait',
           'ayons',
           'ayez',
           'aient',
           'eusse',
           'eusses',
           'eût',
           'eussions',
           'eussiez',
           'eussent']
```

```
In [58]: len(stopwords.words('french'))
```

```
Out[58]: 157
```

```
In [59]: stopwords.words('german')
```

```
Out[59]:  ['aber',
           'alle',
           'allem',
           'allen',
           'aller',
           'alles',
           'als',
           'also',
           'am',
           'an',
           'ander',
           'andere',
           'anderem',
           'anderen',
           'anderer',
           'anderes',
           'anderm',
           'andern',
           'anderr',
           'anders',
           'auch',
           'auf',
           'aus',
           'bei',
           'bin',
           'bis',
           'bist',
           'da',
           'damit',
           'dann',
           'der',
           'den',
           'des',
           'dem',
           'die',
           'das',
           'dass',
           'daß',
           'derselbe',
           'derselben',
           'denselben',
           'desselben',
           'demselben',
           'dieselbe',
           'dieselben',
           'dasselbe',
           'dazu',
           'dein',
           'deine',
           'deinem',
           'deinen',
           'deiner',
           'deines',
           'denn',
           'derer',
           'dessen',
           'dich',
           'dir',
           'du',
           'dies',
'diese',
'diesem',
'diesen',
'dieser',
'dieses',
'doch',
'dort',
'durch',
'ein',
'eine',
'einem',
'einen',
'einer',
'eines',
'einig',
'einige',
'einigem',
'einigen',
'einiger',
'einiges',
'einmal',
'er',
'ihn',
'ihm',
'es',
'etwas',
'euer',
'eure',
'eurem',
'euren',
'eurer',
'eures',
'für',
'gegen',
'gewesen',
'hab',
'habe',
'haben',
'hat',
'hatte',
'hatten',
'hier',
'hin',
'hinter',
'ich',
'mich',
'mir',
'ihr',
'ihre',
'ihrem',
'ihren',
'ihrer',
'ihres',
'euch',
'im',
'in',
'indem',
'ins',
'ist',
'jede',
                    'jedem',
                    'jeden',
                    'jeder',
                    'jedes',
                    'jene',
                    'jenem',
                    'jenen',
                    'jener',
                    'jenes',
                    'jetzt',
                    'kann',
                    'kein',
                    'keine',
                    'keinem',
                    'keinen',
                    'keiner',
                    'keines',
                    'können',
                    'könnte',
                    'machen',
                    'man',
                    'manche',
                    'manchem',
                    'manchen',
                    'mancher',
                    'manches',
                    'mein',
                    'meine',
                    'meinem',
                    'meinen',
                    'meiner',
                    'meines',
                    'mit',
                    'muss',
                    'musste',
                    'nach',
                    'nicht',
                    'nichts',
                    'noch',
                    'nun',
                    'nur',
                    'ob',
                    'oder',
                    'ohne',
                    'sehr',
                    'sein',
                    'seine',
                    'seinem',
                    'seinen',
                    'seiner',
                    'seines',
                    'selbst',
                    'sich',
                    'sie',
                    'ihnen',
                    'sind',
                    'so',
                    'solche',
                    'solchem',
                    'solchen',
 'solcher',
 'solches',
 'soll',
 'sollte',
 'sondern',
 'sonst',
 'über',
 'um',
 'und',
 'uns',
 'unsere',
 'unserem',
 'unseren',
 'unser',
 'unseres',
 'unter',
 'viel',
 'vom',
 'von',
 'vor',
 'während',
 'war',
 'waren',
 'warst',
 'was',
 'weg',
 'weil',
 'weiter',
 'welche',
 'welchem',
 'welchen',
 'welcher',
 'welches',
 'wenn',
 'werde',
 'werden',
 'wie',
 'wieder',
 'will',
 'wir',
 'wird',
 'wirst',
 'wo',
 'wollen',
 'wollte',
 'würde',
 'würden',
 'zu',
 'zum',
 'zur',
 'zwar',
 'zwischen']
```

```
In [60]: len(stopwords.words('german'))
```

```
Out[60]: 232
```

```
In [61]: stopwords.words('hindi') # research phase
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
Cell In[61], line 1
----> 1 stopwords.words('hindi')

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\wordlis
t.py:21, in WordListCorpusReader.words(self, fileids, ignore_lines_startswith)
     18 def words(self, fileids=None, ignore_lines_startswith="\n"):
     19     return [
     20         line
---> 21         for line in line_tokenize(self.raw(fileids))
     22         if not line.startswith(ignore_lines_startswith)
     23     ]

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\api.py:
218, in CorpusReader.raw(self, fileids)
    216 contents = []
    217 for f in fileids:
--> 218     with self.open(f) as fp:
    219         contents.append(fp.read())
    220 return concat(contents)

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\api.py:
231, in CorpusReader.open(self, file)
    223 """
    224 Return an open stream that can be used to read the given file.
    225 If the file's encoding is not None, then the stream will
   (...)
    228 :param file: The file identifier of the file to read.
    229 """
    230 encoding = self.encoding(file)
--> 231 stream = self._root.join(file).open(encoding)
    232 return stream

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\data.py:333, in FileS
ystemPathPointer.join(self, fileid)
    331 def join(self, fileid):
    332     _path = os.path.join(self._path, fileid)
--> 333     return FileSystemPathPointer(_path)

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\data.py:311, in FileS
ystemPathPointer.__init__(self, _path)
    309 _path = os.path.abspath(_path)
    310 if not os.path.exists(_path):
--> 311     raise OSError("No such file or directory: %r" % _path)
    312 self._path = _path

OSError: No such file or directory: 'C:\\Users\\roy62\\AppData\\Roaming\\nltk_dat
a\\corpora\\stopwords\\hindi'
```

```
In [62]: stopwords.words('marathi')
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
Cell In[62], line 1
----> 1 stopwords.words('marathi')

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\wordlis
t.py:21, in WordListCorpusReader.words(self, fileids, ignore_lines_startswith)
     18 def words(self, fileids=None, ignore_lines_startswith="\n"):
     19     return [
     20         line
---> 21         for line in line_tokenize(self.raw(fileids))
     22         if not line.startswith(ignore_lines_startswith)
     23     ]

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\api.py:
218, in CorpusReader.raw(self, fileids)
    216 contents = []
    217 for f in fileids:
--> 218     with self.open(f) as fp:
    219         contents.append(fp.read())
    220 return concat(contents)

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\api.py:
231, in CorpusReader.open(self, file)
    223 """
    224 Return an open stream that can be used to read the given file.
    225 If the file's encoding is not None, then the stream will
   (...)
    228 :param file: The file identifier of the file to read.
    229 """
    230 encoding = self.encoding(file)
--> 231 stream = self._root.join(file).open(encoding)
    232 return stream

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\data.py:333, in FileS
ystemPathPointer.join(self, fileid)
    331 def join(self, fileid):
    332     _path = os.path.join(self._path, fileid)
--> 333     return FileSystemPathPointer(_path)

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\data.py:311, in FileS
ystemPathPointer.__init__(self, _path)
    309 _path = os.path.abspath(_path)
    310 if not os.path.exists(_path):
--> 311     raise OSError("No such file or directory: %r" % _path)
    312 self._path = _path

OSError: No such file or directory: 'C:\\Users\\roy62\\AppData\\Roaming\\nltk_dat
a\\corpora\\stopwords\\marathi'
```

```
In [63]: stopwords.words('telugu')
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
Cell In[63], line 1
----> 1 stopwords.words('telugu')

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\wordlis
t.py:21, in WordListCorpusReader.words(self, fileids, ignore_lines_startswith)
     18 def words(self, fileids=None, ignore_lines_startswith="\n"):
     19     return [
     20         line
---> 21         for line in line_tokenize(self.raw(fileids))
     22         if not line.startswith(ignore_lines_startswith)
     23     ]

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\api.py:
218, in CorpusReader.raw(self, fileids)
    216 contents = []
    217 for f in fileids:
--> 218     with self.open(f) as fp:
    219         contents.append(fp.read())
    220 return concat(contents)

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\corpus\reader\api.py:
231, in CorpusReader.open(self, file)
    223 """
    224 Return an open stream that can be used to read the given file.
    225 If the file's encoding is not None, then the stream will
   (...)
    228 :param file: The file identifier of the file to read.
    229 """
    230 encoding = self.encoding(file)
--> 231 stream = self._root.join(file).open(encoding)
    232 return stream

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\data.py:333, in FileS
ystemPathPointer.join(self, fileid)
    331 def join(self, fileid):
    332     _path = os.path.join(self._path, fileid)
--> 333     return FileSystemPathPointer(_path)

File ~\anaconda3\envs\tensorflow_env\lib\site-packages\nltk\data.py:311, in FileS
ystemPathPointer.__init__(self, _path)
    309 _path = os.path.abspath(_path)
    310 if not os.path.exists(_path):
--> 311     raise OSError("No such file or directory: %r" % _path)
    312 self._path = _path

OSError: No such file or directory: 'C:\\Users\\roy62\\AppData\\Roaming\\nltk_dat
a\\corpora\\stopwords\\telugu'
```

```
In [64]: stopwords.words('spanish')
```

```
Out[64]: ['de',
          'la',
          'que',
          'el',
          'en',
          'y',
          'a',
          'los',
          'del',
          'se',
          'las',
          'por',
          'un',
          'para',
          'con',
          'no',
          'una',
          'su',
          'al',
          'lo',
          'como',
          'más',
          'pero',
          'sus',
          'le',
          'ya',
          'o',
          'este',
          'sí',
          'porque',
          'esta',
          'entre',
          'cuando',
          'muy',
          'sin',
          'sobre',
          'también',
          'me',
          'hasta',
          'hay',
          'donde',
          'quien',
          'desde',
          'todo',
          'nos',
          'durante',
          'todos',
          'uno',
          'les',
          'ni',
          'contra',
          'otros',
          'ese',
          'eso',
          'ante',
          'ellos',
          'e',
          'esto',
          'mí',
          'antes',
'algunos',
'qué',
'unos',
'yo',
'otro',
'otras',
'otra',
'él',
'tanto',
'esa',
'estos',
'mucho',
'quienes',
'nada',
'muchos',
'cual',
'poco',
'ella',
'estar',
'estas',
'algunas',
'algo',
'nosotros',
'mi',
'mis',
'tú',
'te',
'ti',
'tu',
'tus',
'ellas',
'nosotras',
'vosotros',
'vosotras',
'os',
'mío',
'mía',
'míos',
'mías',
'tuyo',
'tuya',
'tuyos',
'tuyas',
'suyo',
'suya',
'suyos',
'suyas',
'nuestro',
'nuestra',
'nuestros',
'nuestras',
'vuestro',
'vuestra',
'vuestros',
'vuestras',
'esos',
'esas',
'estoy',
'estás',
'está',
'estamos',
'estáis',
'están',
'esté',
'estés',
'estemos',
'estéis',
'estén',
'estaré',
'estarás',
'estará',
'estaremos',
'estaréis',
'estarán',
'estaría',
'estarías',
'estaríamos',
'estaríais',
'estarían',
'estaba',
'estabas',
'estábamos',
'estabais',
'estaban',
'estuve',
'estuviste',
'estuvo',
'estuvimos',
'estuvisteis',
'estuvieron',
'estuviera',
'estuvieras',
'estuviéramos',
'estuvierais',
'estuvieran',
'estuviese',
'estuvieses',
'estuviésemos',
'estuvieseis',
'estuviesen',
'estando',
'estado',
'estada',
'estados',
'estadas',
'estad',
'he',
'has',
'ha',
'hemos',
'habéis',
'han',
'haya',
'hayas',
'hayamos',
'hayáis',
'hayan',
'habré',
'habrás',
'habrá',
'habremos',
'habréis',
'habrán',
'habría',
'habrías',
'habríamos',
'habríais',
'habrían',
'había',
'habías',
'habíamos',
'habíais',
'habían',
'hube',
'hubiste',
'hubo',
'hubimos',
'hubisteis',
'hubieron',
'hubiera',
'hubieras',
'hubiéramos',
'hubierais',
'hubieran',
'hubiese',
'hubieses',
'hubiésemos',
'hubieseis',
'hubiesen',
'habiendo',
'habido',
'habida',
'habidos',
'habidas',
'soy',
'eres',
'es',
'somos',
'sois',
'son',
'sea',
'seas',
'seamos',
'seáis',
'sean',
'seré',
'serás',
'será',
'seremos',
'seréis',
'serán',
'sería',
'serías',
'seríamos',
'seríais',
'serían',
'era',
'eras',
'éramos',
'erais',
'eran',
'fui',
'fuiste',
'fue',
'fuimos',
'fuisteis',
'fueron',
'fuera',
'fueras',
'fuéramos',
'fuerais',
'fueran',
'fuese',
'fueses',
'fuésemos',
'fueseis',
'fuesen',
'sintiendo',
'sentido',
'sentida',
'sentidos',
'sentidas',
'siente',
'sentid',
'tengo',
'tienes',
'tiene',
'tenemos',
'tenéis',
'tienen',
'tenga',
'tengas',
'tengamos',
'tengáis',
'tengan',
'tendré',
'tendrás',
'tendrá',
'tendremos',
'tendréis',
'tendrán',
'tendría',
'tendrías',
'tendríamos',
'tendríais',
'tendrían',
'tenía',
'tenías',
'teníamos',
'teníais',
'tenían',
'tuve',
'tuviste',
'tuvo',
'tuvimos',
'tuvisteis',
'tuvieron',
'tuviera',
'tuvieras',
'tuviéramos',
 'tuvierais',
 'tuvieran',
 'tuviese',
 'tuvieses',
 'tuviésemos',
 'tuvieseis',
 'tuviesen',
 'teniendo',
 'tenido',
 'tenida',
 'tenidos',
 'tenidas',
 'tened']
```

In [65]:
```
len(stopwords.words('spanish') )
```

Out[65]:
```
313
```

In [71]:
```
stopwords.words('chinese')
```

```
Out[71]: ['一',
          '一下',
          '一些',
          '一切',
          '一则',
          '一天',
          '一定',
          '一方面',
          '一旦',
          '一时',
          '一来',
          '一样',
          '一次',
          '一片',
          '一直',
          '一致',
          '一般',
          '一起',
          '一边',
          '一面',
          '万一',
          '上下',
          '上升',
          '上去',
          '上来',
          '上述',
          '上面',
          '下列',
          '下去',
          '下来',
          '下面',
          '不一',
          '不久',
          '不仅',
          '不会',
          '不但',
          '不光',
          '不单',
          '不变',
          '不只',
          '不可',
          '不同',
          '不够',
          '不如',
          '不得',
          '不怕',
          '不惟',
          '不成',
          '不拘',
          '不敢',
          '不断',
          '不是',
          '不比',
          '不然',
          '不特',
          '不独',
          '不管',
          '不能',
          '不要',
          '不论',
    '不足',
    '不过',
    '不问',
    '与',
    '与其',
    '与否',
    '与此同时',
    '专门',
    '且',
    '两者',
    '严格',
    '严重',
    '个',
    '个人',
    '个别',
    '中小',
    '中间',
    '丰富',
    '临',
    '为',
    '为主',
    '为了',
    '为什么',
    '为什麽',
    '为何',
    '为着',
    '主张',
    '主要',
    '举行',
    '乃',
    '乃至',
    '么',
    '之',
    '之一',
    '之前',
    '之后',
    '之後',
    '之所以',
    '之类',
    '乌乎',
    '乎',
    '乘',
    '也',
    '也好',
    '也是',
    '也罢',
    '了',
    '了解',
    '争取',
    '于',
    '于是',
    '于是乎',
    '云云',
    '互相',
    '产生',
    '人们',
    '人家',
    '什么',
    '什么样',
    '什麽',
'今后',
'今天',
'今年',
'今後',
'仍然',
'从',
'从事',
'从而',
'他',
'他人',
'他们',
'他的',
'代替',
'以',
'以上',
'以下',
'以为',
'以便',
'以免',
'以前',
'以及',
'以后',
'以外',
'以後',
'以来',
'以至',
'以至于',
'以致',
'们',
'任',
'任何',
'任凭',
'任务',
'企图',
'伟大',
'似乎',
'似的',
'但',
'但是',
'何',
'何况',
'何处',
'何时',
'作为',
'你',
'你们',
'你的',
'使得',
'使用',
'例如',
'依',
'依照',
'依靠',
'促进',
'保持',
'俺',
'俺们',
'倘',
'倘使',
'倘或',
'倘然',
'倘若',
'假使',
'假如',
'假若',
'做到',
'像',
'允许',
'充分',
'先后',
'先後',
'先生',
'全部',
'全面',
'兮',
'共同',
'关于',
'其',
'其一',
'其中',
'其二',
'其他',
'其余',
'其它',
'其实',
'其次',
'具体',
'具体地说',
'具体说来',
'具有',
'再者',
'再说',
'冒',
'冲',
'决定',
'况且',
'准备',
'几',
'几乎',
'几时',
'凭',
'凭借',
'出去',
'出来',
'出现',
'分别',
'则',
'别',
'别的',
'别说',
'到',
'前后',
'前者',
'前进',
'前面',
'加之',
'加以',
'加入',
'加强',
'十分',
'即',
'即令',
'即使',
'即便',
'即或',
'即若',
'却不',
'原来',
'又',
'及',
'及其',
'及时',
'及至',
'双方',
'反之',
'反应',
'反映',
'反过来',
'反过来说',
'取得',
'受到',
'变成',
'另',
'另一方面',
'另外',
'只是',
'只有',
'只要',
'只限',
'叫',
'叫做',
'召开',
'叮咚',
'可',
'可以',
'可是',
'可能',
'可见',
'各',
'各个',
'各人',
'各位',
'各地',
'各种',
'各级',
'各自',
'合理',
'同',
'同一',
'同时',
'同样',
'后来',
'后面',
'向',
'向着',
'吓',
'吗',
'否则',
'吧',
'吧哒',
'吱',
'呀',
'呃',
'呕',
'呗',
'呜',
'呜呼',
'呢',
'周围',
'呵',
'呸',
'呼哧',
'咋',
'和',
'咚',
'咦',
'咱',
'咱们',
'咳',
'哇',
'哈',
'哈哈',
'哉',
'哎',
'哎呀',
'哎哟',
'哗',
'哟',
'哦',
'哩',
'哪',
'哪个',
'哪些',
'哪儿',
'哪天',
'哪年',
'哪怕',
'哪样',
'哪边',
'哪里',
'哼',
'哼唷',
'唉',
'啊',
'啐',
'啥',
'啦',
'啪达',
'喂',
'喏',
'喔唷',
'嗡嗡',
'嗬',
'嗯',
'嗳',
'嘎',
'嘎登',
'嘘',
'嘛',
'嘻',
'嘿',
'因',
'因为',
'因此',
'因而',
'固然',
'在',
'在下',
'地',
'坚决',
'坚持',
'基本',
'处理',
'复杂',
'多',
'多少',
'多数',
'多次',
'大力',
'大多数',
'大大',
'大家',
'大批',
'大约',
'大量',
'失去',
'她',
'她们',
'她的',
'好的',
'好象',
'如',
'如上所述',
'如下',
'如何',
'如其',
'如果',
'如此',
'如若',
'存在',
'宁',
'宁可',
'宁愿',
'宁肯',
'它',
'它们',
'它们的',
'它的',
'安全',
'完全',
'完成',
'实现',
'实际',
'宣布',
'容易',
'密切',
'对',
'对于',
'对应',
'将',
'少数',
'尔后',
'尚且',
'尤其',
'就',
'就是',
'就是说',
'尽',
'尽管',
'属于',
'岂但',
'左右',
'巨大',
'巩固',
'己',
'已经',
'帮助',
'常常',
'并',
'并不',
'并不是',
'并且',
'并没有',
'广大',
'广泛',
'应当',
'应用',
'应该',
'开外',
'开始',
'开展',
'引起',
'强烈',
'强调',
'归',
'当',
'当前',
'当时',
'当然',
'当着',
'形成',
'彻底',
'彼',
'彼此',
'往',
'往往',
'待',
'後来',
'後面',
'得',
'得出',
'得到',
'心里',
'必然',
'必要',
'必须',
'怎',
'怎么',
'怎么办',
'怎么样',
'怎样',
'怎麼',
'总之',
'总是',
'总的来看',
'总的来说',
'总的说来',
'总结',
'总而言之',
'恰恰相反',
'您',
'意思',
'愿意',
'慢说',
'成为',
'我',
'我们',
'我的',
'或',
'或是',
'或者',
'战斗',
'所',
'所以',
'所有',
'所谓',
'打',
'扩大',
'把',
'抑或',
'拿',
'按',
'按照',
'换句话说',
'换言之',
'据',
'掌握',
'接着',
'接著',
'故',
'故此',
'整个',
'方便',
'方面',
'旁人',
'无宁',
'无法',
'无论',
'既',
'既是',
'既然',
'时候',
'明显',
'明确',
'是',
'是否',
'是的',
'显然',
'显著',
'普通',
'普遍',
'更加',
'曾经',
'替',
'最后',
'最大',
'最好',
'最後',
'最近',
'最高',
'有',
'有些',
'有关',
'有利',
'有力',
'有所',
'有效',
'有时',
'有点',
'有的',
'有着',
'有著',
'望',
'朝',
'朝着',
'本',
'本着',
'来',
'来着',
'极了',
'构成',
'果然',
'果真',
'某',
'某个',
'某些',
'根据',
'根本',
'欢迎',
'正在',
'正如',
'正常',
'此',
'此外',
'此时',
'此间',
'毋宁',
'每',
'每个',
'每天',
'每年',
'每当',
'比',
'比如',
'比方',
'比较',
'毫不',
'没有',
'沿',
'沿着',
'注意',
'深入',
'清楚',
'满足',
'漫说',
'焉',
'然则',
'然后',
'然後',
'然而',
'照',
'照着',
'特别是',
'特殊',
'特点',
'现代',
'现在',
'甚么',
'甚而',
'甚至',
'用',
'由',
'由于',
'由此可见',
'的',
'的话',
'目前',
'直到',
'直接',
'相似',
'相信',
'相反',
'相同',
'相对',
'相对而言',
'相应',
'相当',
'相等',
'省得',
'看出',
'看到',
'看来',
'看看',
'看见',
'真是',
'真正',
'着',
'着呢',
'矣',
'知道',
'确定',
'离',
'积极',
'移动',
'突出',
'突然',
'立即',
'第',
'等',
'等等',
'管',
'紧接着',
'纵',
'纵令',
'纵使',
'纵然',
'练习',
'组成',
'经',
'经常',
'经过',
'结合',
'结果',
'给',
'绝对',
'继续',
'继而',
'维持',
'综上所述',
'罢了',
'考虑',
'者',
'而',
'而且',
'而况',
'而外',
'而已',
'而是',
'而言',
'联系',
'能',
'能否',
'能够',
'腾',
'自',
'自个儿',
'自从',
'自各儿',
'自家',
'自己',
'自身',
'至',
'至于',
'良好',
'若',
'若是',
'若非',
'范围',
'莫若',
'获得',
'虽',
'虽则',
'虽然',
'虽说',
'行为',
'行动',
'表明',
'表示',
'被',
'要',
'要不',
'要不是',
'要不然',
'要么',
'要是',
'要求',
'规定',
'觉得',
'认为',
'认真',
'认识',
'让',
'许多',
'论',
'设使',
'设若',
'该',
'说明',
'诸位',
'谁',
'谁知',
'赶',
'起',
'起来',
'起见',
'趁',
'趁着',
'越是',
'跟',
'转动',
'转变',
'转贴',
'较',
'较之',
'边',
'达到',
'迅速',
'过',
'过去',
'过来',
'运用',
'还是',
'还有',
'这',
'这个',
'这么',
'这么些',
'这么样',
'这么点儿',
'这些',
'这会儿',
'这儿',
'这就是说',
'这时',
'这样',
'这点',
'这种',
'这边',
'这里',
'这麽',
'进入',
'进步',
'进而',
'进行',
'连',
'连同',
'适应',
'适当',
'适用',
'逐步',
'逐渐',
'通常',
'通过',
'造成',
'遇到',
'遭到',
'避免',
'那',
'那个',
'那么',
'那么些',
'那么样',
'那些',
'那会儿',
'那儿',
'那时',
'那样',
'那边',
'那里',
'那麽',
'部分',
'鄙人',
'采取',
'里面',
'重大',
'重新',
'重要',
'鉴于',
'问题',
'防止',
'阿',
'附近',
'限制',
'除',
'除了',
'除此之外',
'除非',
'随',
'随着',
'随著',
'集中',
'需要',
'非但',
'非常',
'非徒',
'靠',
'顺',
'顺着',
'首先',
 '高兴',
 '是不是']
```

In [72]:
```
len(stopwords.words('chinese') )
```

Out[72]:
```
841
```

In [73]:
```
stopwords.words('german')
```

```
Out[73]:  ['aber',
           'alle',
           'allem',
           'allen',
           'aller',
           'alles',
           'als',
           'also',
           'am',
           'an',
           'ander',
           'andere',
           'anderem',
           'anderen',
           'anderer',
           'anderes',
           'anderm',
           'andern',
           'anderr',
           'anders',
           'auch',
           'auf',
           'aus',
           'bei',
           'bin',
           'bis',
           'bist',
           'da',
           'damit',
           'dann',
           'der',
           'den',
           'des',
           'dem',
           'die',
           'das',
           'dass',
           'daß',
           'derselbe',
           'derselben',
           'denselben',
           'desselben',
           'demselben',
           'dieselbe',
           'dieselben',
           'dasselbe',
           'dazu',
           'dein',
           'deine',
           'deinem',
           'deinen',
           'deiner',
           'deines',
           'denn',
           'derer',
           'dessen',
           'dich',
           'dir',
           'du',
           'dies',
'diese',
'diesem',
'diesen',
'dieser',
'dieses',
'doch',
'dort',
'durch',
'ein',
'eine',
'einem',
'einen',
'einer',
'eines',
'einig',
'einige',
'einigem',
'einigen',
'einiger',
'einiges',
'einmal',
'er',
'ihn',
'ihm',
'es',
'etwas',
'euer',
'eure',
'eurem',
'euren',
'eurer',
'eures',
'für',
'gegen',
'gewesen',
'hab',
'habe',
'haben',
'hat',
'hatte',
'hatten',
'hier',
'hin',
'hinter',
'ich',
'mich',
'mir',
'ihr',
'ihre',
'ihrem',
'ihren',
'ihrer',
'ihres',
'euch',
'im',
'in',
'indem',
'ins',
'ist',
'jede',
'jedem',
'jeden',
'jeder',
'jedes',
'jene',
'jenem',
'jenen',
'jener',
'jenes',
'jetzt',
'kann',
'kein',
'keine',
'keinem',
'keinen',
'keiner',
'keines',
'können',
'könnte',
'machen',
'man',
'manche',
'manchem',
'manchen',
'mancher',
'manches',
'mein',
'meine',
'meinem',
'meinen',
'meiner',
'meines',
'mit',
'muss',
'musste',
'nach',
'nicht',
'nichts',
'noch',
'nun',
'nur',
'ob',
'oder',
'ohne',
'sehr',
'sein',
'seine',
'seinem',
'seinen',
'seiner',
'seines',
'selbst',
'sich',
'sie',
'ihnen',
'sind',
'so',
'solche',
'solchem',
'solchen',
'solcher',
'solches',
'soll',
'sollte',
'sondern',
'sonst',
'über',
'um',
'und',
'uns',
'unsere',
'unserem',
'unseren',
'unser',
'unseres',
'unter',
'viel',
'vom',
'von',
'vor',
'während',
'war',
'waren',
'warst',
'was',
'weg',
'weil',
'weiter',
'welche',
'welchem',
'welchen',
'welcher',
'welches',
'wenn',
'werde',
'werden',
'wie',
'wieder',
'will',
'wir',
'wird',
'wirst',
'wo',
'wollen',
'wollte',
'würde',
'würden',
'zu',
'zum',
'zur',
'zwar',
'zwischen']
```

In [74]:
```
len(stopwords.words('german') )
```

Out[74]:
```
232
```

In [75]:
```
# first we need to compile from re module to create string that matched any digi
import re
punctuation = re.compile(r'[-.?!,:;()|0-9]')
```

```
#now i am going to create to empty list and append the word without any punctuat
```

In [76]:
```
punctuation
```

Out[76]:
```
re.compile(r'[-.?!,:;()|0-9]', re.UNICODE)
```

In [77]:
```
AI
```

Out[77]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of \nhumans and animals. With Artificial Inte
lligence, machines perform functions such as learning, planning, reasoning and
\nproblem-solving. Most noteworthy, Artificial Intelligence is the simulation o
f human intelligence by machines. \nIt is probably the fastest-growing developm
ent in the World of technology and innovation. Furthermore, many experts believ
e\nAI could solve major challenges and crisis situations.'
```

In [78]:
```
AI_tokens
```

```
Out[78]: ['Artificial',
          'Intelligence',
          'refers',
          'to',
          'the',
          'intelligence',
          'of',
          'machines',
          '.',
          'This',
          'is',
          'in',
          'contrast',
          'to',
          'the',
          'natural',
          'intelligence',
          'of',
          'humans',
          'and',
          'animals',
          '.',
          'With',
          'Artificial',
          'Intelligence',
          ',',
          'machines',
          'perform',
          'functions',
          'such',
          'as',
          'learning',
          ',',
          'planning',
          ',',
          'reasoning',
          'and',
          'problem-solving',
          '.',
          'Most',
          'noteworthy',
          ',',
          'Artificial',
          'Intelligence',
          'is',
          'the',
          'simulation',
          'of',
          'human',
          'intelligence',
          'by',
          'machines',
          '.',
          'It',
          'is',
          'probably',
          'the',
          'fastest-growing',
          'development',
          'in',
          'the',
          'World',
          'of',
          'technology',
          'and',
          'innovation',
          '.',
          'Furthermore',
          ',',
          'many',
          'experts',
          'believe',
          'AI',
          'could',
          'solve',
          'major',
          'challenges',
          'and',
          'crisis',
          'situations',
          '.']
```

In [79]:
```
len(AI_tokens)
```

Out[79]:
```
81
```

In [80]:
```
# we will see how to work in POS using NLTK library

sent = 'kathy is a natural when it comes to drawing'
sent_tokens = word_tokenize(sent)
sent_tokens

# first we will tokenize usning word_tokenize & then we will use pos_tag on all
```

Out[80]:
```
['kathy', 'is', 'a', 'natural', 'when', 'it', 'comes', 'to', 'drawing']
```

In [81]:
```
for token in sent_tokens:
    print(nltk.pos_tag([token]))
```

```
[('kathy', 'NN')]
[('is', 'VBZ')]
[('a', 'DT')]
[('natural', 'JJ')]
[('when', 'WRB')]
[('it', 'PRP')]
[('comes', 'VBZ')]
[('to', 'TO')]
[('drawing', 'VBG')]
```

In [82]:
```
sent2 = 'john is eating a delicious cake'
sent2_tokens = word_tokenize(sent2)

for token in sent2_tokens:
    print(nltk.pos_tag([token]))
```

```
[('john', 'NN')]
[('is', 'VBZ')]
[('eating', 'VBG')]
[('a', 'DT')]
[('delicious', 'JJ')]
[('cake', 'NN')]
```

In [83]:
```
from nltk import ne_chunk
```

In [84]:
```
NE_sent = 'The US president stays in the WHITEHOUSE '
```

In [85]:
```
NE_tokens = word_tokenize(NE_sent)

#after tokenize need to add the pos tags
NE_tokens
```

Out[85]:
```
['The', 'US', 'president', 'stays', 'in', 'the', 'WHITEHOUSE']
```

In [86]:
```
NE_tags = nltk.pos_tag(NE_tokens)
NE_tags
```

Out[86]:
```
[('The', 'DT'),
 ('US', 'NNP'),
 ('president', 'NN'),
 ('stays', 'NNS'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('WHITEHOUSE', 'NNP')]
```

In [87]:
```
#we are passin the NE_NER into ne_chunks function and lets see the outputs

NE_NER = ne_chunk(NE_tags)
print(NE_NER)
```

```
(S
  The/DT
  (GSP US/NNP)
  president/NN
  stays/NNS
  in/IN
  the/DT
  (ORGANIZATION WHITEHOUSE/NNP))
```

In [88]:
```
new = 'the big cat ate the little mouse who was after fresh cheese'
new_tokens = nltk.pos_tag(word_tokenize(new))
new_tokens

# tokenize done and lets add the pos tags also
```

```
Out[88]: [('the', 'DT'),
           ('big', 'JJ'),
           ('cat', 'NN'),
           ('ate', 'VBD'),
           ('the', 'DT'),
           ('little', 'JJ'),
           ('mouse', 'NN'),
           ('who', 'WP'),
           ('was', 'VBD'),
           ('after', 'IN'),
           ('fresh', 'JJ'),
           ('cheese', 'NN')]
```

```
In [ ]:
```

Word Cloud

```
In [89]: # Libraries
         from wordcloud import WordCloud
         import matplotlib.pyplot as plt
```

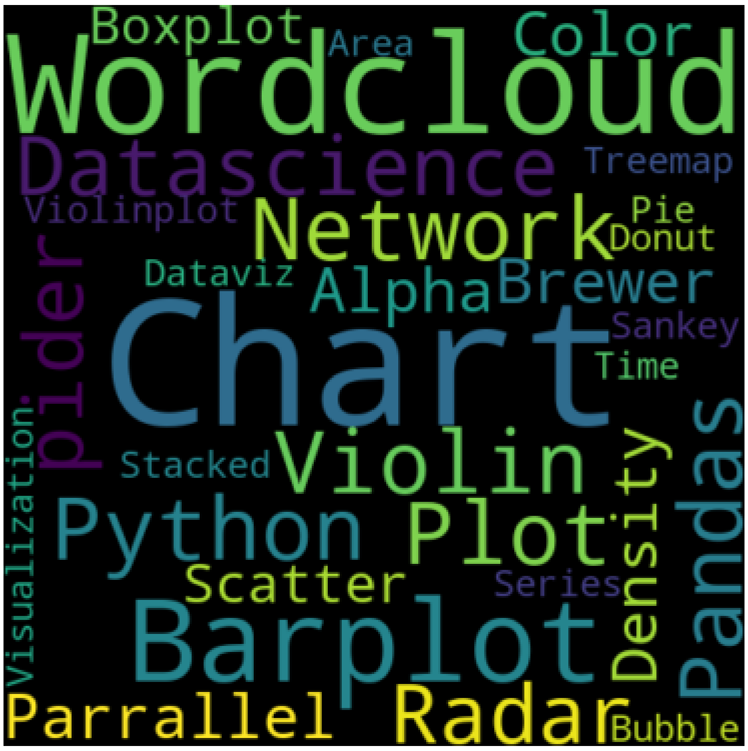
```
In [90]: # Create a list of word
         text=("Python Network Plot Violin Chart Pandas Datascience Wordcloud pider Radar
```

```
In [91]: text
```

```
Out[91]: 'Python Network Plot Violin Chart Pandas Datascience Wordcloud pider Radar Parr
         allel Alpha Color Brewer Density Scatter Barplot Barplot Boxplot Violinplot Tre
         emap Stacked Area Chart Chart Visualization Dataviz Donut Pie Time-Series Wordc
         loud Wordcloud Sankey Bubble'
```

```
In [92]: # Create the wordcloud object
         wordcloud = WordCloud(width=480, height=480, margin=0).generate(text)
```

```
In [93]: # Display the generated image:
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis("off")
         plt.margins(x=0, y=0)
         plt.show()
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```