

2/2/24 gayathri

NLP3

In [15]: `▶ nltk.download('stopwords')`

```
[nltk_data] Downloading package stopwords to  
[nltk_data] C:\Users\gayus\AppData\Roaming\nltk_data...  
[nltk_data] Unzipping corpora\stopwords.zip.
```

Out[15]: True

In [16]: `▶ import nltk
from nltk.corpus import stopwords
stopwords.words('english')`

```
['a',  
'an',  
'the',  
'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 [17]: `▶ nltk.download('cmudict')`

```
[nltk_data] Downloading package cmudict to  
[nltk_data] C:\Users\gayus\AppData\Roaming\nltk_data...  
[nltk_data] Unzipping corpora\cmudict.zip.
```

Out[17]: True

```
In [32]: ▶ entries = nltk.corpus.cmudict.entries()
print(len(entries))

# Now you can iterate over the entries
for entry in entries[10000:10025]:
    print(entry)

133737
('belford', ['B', 'EH1', 'L', 'F', 'ER0', 'D'])
('belfry', ['B', 'EH1', 'L', 'F', 'R', 'IY0'])
('belgacom', ['B', 'EH1', 'L', 'G', 'AH0', 'K', 'AA0', 'M'])
('belgacom', ['B', 'EH1', 'L', 'JH', 'AH0', 'K', 'AA0', 'M'])
('belgard', ['B', 'EH0', 'L', 'G', 'AA1', 'R', 'D'])
('belgarde', ['B', 'EH0', 'L', 'G', 'AA1', 'R', 'D', 'IY0'])
('belge', ['B', 'EH1', 'L', 'JH', 'IY0'])
('belger', ['B', 'EH1', 'L', 'G', 'ER0'])
('belgian', ['B', 'EH1', 'L', 'JH', 'AH0', 'N'])
('belgians', ['B', 'EH1', 'L', 'JH', 'AH0', 'N', 'Z'])
('belgique', ['B', 'EH0', 'L', 'ZH', 'IY1', 'K'])
('belgique's', ['B', 'EH0', 'L', 'JH', 'IY1', 'K', 'S'])
('belgium', ['B', 'EH1', 'L', 'JH', 'AH0', 'M'])
('belgium's', ['B', 'EH1', 'L', 'JH', 'AH0', 'M', 'Z'])
('belgo', ['B', 'EH1', 'L', 'G', 'OW2'])
('belgrade', ['B', 'EH1', 'L', 'G', 'R', 'EY0', 'D'])
('belgrade', ['B', 'EH1', 'L', 'G', 'R', 'AA2', 'D'])
('belgrade's', ['B', 'EH1', 'L', 'G', 'R', 'EY0', 'D', 'Z'])
('belgrade's', ['B', 'EH1', 'L', 'G', 'R', 'AA2', 'D', 'Z'])
('belgrave', ['B', 'EH1', 'L', 'G', 'R', 'EY2', 'V'])
('beli', ['B', 'EH1', 'L', 'IY0'])
('belich', ['B', 'EH1', 'L', 'IH0', 'K'])
('belie', ['B', 'IH0', 'L', 'AY1'])
('belied', ['B', 'IH0', 'L', 'AY1', 'D'])
('belief', ['B', 'IH0', 'L', 'IY1', 'F'])
```

```
In [28]: ▶ nltk.download('wordnet')

[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\gayus\AppData\Roaming\nltk_data...
```

Out[28]: True

```
In [29]: ▶ from nltk.corpus import wordnet as wn
wn.synsets('motorcar')
wn.synset('car.n.01').lemma_names()
```

Out[29]: ['car', 'auto', 'automobile', 'machine', 'motorcar']

```
In [21]: ▶ nltk.download('PorterStemmer')

[nltk_data] Error loading PorterStemmer: Package 'PorterStemmer' not
[nltk_data] found in index
```

Out[21]: False

In [31]: `from nltk.stem import PorterStemmer, LancasterStemmer`

```
# Create instances of the stemmers
stemmer_porter = PorterStemmer()
stemmer_lancaster = LancasterStemmer()

# Now you can use them
print(stemmer_porter.stem('happiness'))
print(stemmer_lancaster.stem('happiness'))
```

happi
happy

In [2]: `from nltk.stem import RegexpStemmer`
`stemmerregex=RegexpStemmer('learn')`
`stemmerregex.stem('learning')`

Out[2]: 'ing'

In [4]: `from nltk.stem import RegexpStemmer`
`stemmerregex=RegexpStemmer('ing')`
`stemmerregex.stem('singing')`

Out[4]: 's'

In [7]: `from nltk.stem import SnowballStemmer`
`SnowballStemmer.languages`
`frenchstemmer=SnowballStemmer('french')`
`frenchstemmer.stem('manges')`

Out[7]: 'mang'

In [22]: `nltk.download('punkt')`

```
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\gayus\AppData\Roaming\nltk_data...
[nltk_data] Unzipping tokenizers\punkt.zip.
```

Out[22]: True

In [25]: `nltk.download('averaged_perceptron_tagger')`

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] C:\Users\gayus\AppData\Roaming\nltk_data...
[nltk_data] Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[25]: True

```
In [26]: ► import nltk
          texts="In a quaint town nestled between rolling hills and babbling brook
          sentences=nltk.sent_tokenize(text)
          for text in texts:

              for sentence in sentences:
                  words=nltk.word_tokenize(sentence)
                  print(words)
                  tagged=nltk.pos_tag(words)
                  print(tagged)
```

[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']
[('I', 'PRP')]
['I']

In []: ►

In []: ►