In [1]:
```
import nltk
from nltk.corpus import stopwords
stopwords.words('english')
```

```
 '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 [4]:
```
entries=nltk.corpus.cmudict.entries()
len(entries)
for entry in entries[10000:10025]:
    print(entry)
```

```
('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 [6]:
```
from nltk.corpus import wordnet as wn
wn.synsets('motocar')
wn.synset('car.n.01').lemma_names()
```

Out[6]:
```
['car', 'auto', 'automobile', 'machine', 'motorcar']
```

In [7]:
```
from nltk.corpus import wordnet as wn
wn.synsets('good')
```

Out[7]:
```
[Synset('good.n.01'),
 Synset('good.n.02'),
 Synset('good.n.03'),
 Synset('commodity.n.01'),
 Synset('good.a.01'),
 Synset('full.s.06'),
 Synset('good.a.03'),
 Synset('estimable.s.02'),
 Synset('beneficial.s.01'),
 Synset('good.s.06'),
 Synset('good.s.07'),
 Synset('adept.s.01'),
 Synset('good.s.09'),
 Synset('dear.s.02'),
 Synset('dependable.s.04'),
 Synset('good.s.12'),
 Synset('good.s.13'),
 Synset('effective.s.04'),
 Synset('good.s.15'),
 Synset('good.s.16'),
 Synset('good.s.17'),
 Synset('good.s.18'),
 Synset('good.s.19'),
 Synset('good.s.20'),
 Synset('good.s.21'),
 Synset('well.r.01'),
 Synset('thoroughly.r.02')]
```

In [8]:
```
from nltk.stem import PorterStemmer
stemmerporter=PorterStemmer()
stemmerporter.stem('happiness')
```

Out[8]:
```
'happi'
```

In [9]:
```
from nltk.stem import LancasterStemmer
stemmerporter=LancasterStemmer()
stemmerporter.stem('happiness')
```

Out[9]:
```
'happy'
```

In [10]:
```
from nltk.stem import RegexpStemmer
stemmerregexp=RegexpStemmer('learn')
stemmerregexp.stem('learning')
```

Out[10]:
```
'ing'
```

In [11]:
```
from nltk.stem import SnowballStemmer
SnowballStemmer.languages
frenchstemmer=SnowballStemmer('french')
frenchstemmer.stem('manges')
```

Out[11]:
```
'mang'
```

In [12]:
```
sent="Become an expert in NLP"
words=nltk.word_tokenize(sent)
print(words)
```

```
['Become', 'an', 'expert', 'in', 'NLP']
```

In [21]:
```
texts="Nature is that natural and physical world that surrounds us and makes l
sentences=nltk.sent_tokenize(texts)
print(sentences)
for text in texts:
    #sentences = nltk.sent_tokenize(text)
    #print(sentences)
    for sentence in sentences:
        words=nltk.word_tokenize(sentence)
        print(words)
        tagged=nltk.pos_tag(words)
        print(tagged)
```

```
['Nature is that natural and physical world that surrounds us and makes li
fe possible on earth.', 'Nature is the heart of earth.', 'Nature heals us
and helps build connection with our freedom, authenticity and our souls.',
'Simply connecting and feeling nature gives us a divine pleasure.', 'We ha
ve a strong bond and emotional connection with nature.']
['Nature', 'is', 'that', 'natural', 'and', 'physical', 'world', 'that', 's
urrounds', 'us', 'and', 'makes', 'life', 'possible', 'on', 'earth', '.']
[('Nature', 'NN'), ('is', 'VBZ'), ('that', 'IN'), ('natural', 'JJ'), ('an
d', 'CC'), ('physical', 'JJ'), ('world', 'NN'), ('that', 'WDT'), ('surroun
ds', 'VBZ'), ('us', 'PRP'), ('and', 'CC'), ('makes', 'VBZ'), ('life', 'N
N'), ('possible', 'JJ'), ('on', 'IN'), ('earth', 'NN'), ('.', '.')]
['Nature', 'is', 'the', 'heart', 'of', 'earth', '.']
[('Nature', 'NN'), ('is', 'VBZ'), ('the', 'DT'), ('heart', 'NN'), ('of',
'IN'), ('earth', 'NN'), ('.', '.')]
['Nature', 'heals', 'us', 'and', 'helps', 'build', 'connection', 'with',
'our', 'freedom', ',', 'authenticity', 'and', 'our', 'souls', '.']
[('Nature', 'NN'), ('heals', 'NNS'), ('us', 'PRP'), ('and', 'CC'), ('help
s', 'VBZ'), ('build', 'VB'), ('connection', 'NN'), ('with', 'IN'), ('our',
'PRP$'), ('freedom', 'NN'), (',', ','), ('authenticity', 'NN'), ('and', 'C
```

In [ ]: