```
#cmudict has pronouncation words.
#wordnet gives all the synonyms for the word that is to be found with
its id's
#Stemming, getting rid of suffixes
# 3 types of stemmer, porter, lancaster, snowball
#Vectorizer
#Glovector for google,

import nltk

from nltk.corpus import stopwords

stopwords.words('english')

['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"]
```

```
entries = nltk.corpus.cmudict.entries()
len(entries)
for entry in entries[10000:10025]:
    print(entry)
    #this is a mapping, i.e there is no encoding availabe for the IBA
symbols

('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'])

from nltk.corpus import wordnet as wn
wn.synsets('school')

[Synset('school.n.01'),
 Synset('school.n.02'),
 Synset('school.n.03'),
 Synset('school.n.04'),
 Synset('school.n.05'),
 Synset('school.n.06'),
 Synset('school.n.07'),
 Synset('school.v.01'),
 Synset('educate.v.03'),
 Synset('school.v.03')]

from nltk.stem import PorterStemmer
stemmerporter = PorterStemmer()
stemmerporter.stem('happiness')

'happi'
```

```
from nltk.stem import LancasterStemmer
stemmerporter = LancasterStemmer()
stemmerporter.stem('happiness')
```

```
'happy'
```

```
from nltk.stem import RegexpStemmer
stemmerregexp = RegexpStemmer('ing')
stemmerregexp.stem('singing')
```

```
's'
```

```
from nltk.stem import SnowballStemmer
SnowballStemmer.languages
frenchstemmer = SnowballStemmer('french')
frenchstemmer.stem('manges')
```

```
'mang'
```

```
from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer(binary = True)
corpus = ["Tessaract is a good optical character recognition"]
vect.fit(corpus)

vocab = vect.vocabulary_
for key in sorted(vocab.keys()):
    print("{}:{}".format(key, vocab[key]))

print(vect.transform(["This is a good optical illusion"]).toarray())
```

```
character:0
good:1
is:2
optical:3
recognition:4
tessaract:5
[[0 1 1 1 0 0]]
```