02/02/2024

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

In [ ]:
```
import nltk
from nltk.corpus import stopwords
# list of words with no meaning
stopwords.words('english') #at last we remove the. But for sequencial we do not discard
```

In [ ]:
```
from nltk.corpus import cmudict
entries=nltk.corpus.cmudict.entries()
len(entries)
for entry in entries[10000:10025]:
    print(entry)
```

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

In [ ]:
```
# STEMMING
from  nltk.stem import PorterStemmer
from  nltk.stem import LancasterStemmer
stemmerporter=PorterStemmer()
stemmerporter1=LancasterStemmer()
```

In [ ]:
```
stemmerporter.stem('happiness')
stemmerporter.stem('Thinking')
stemmerporter.stem('Laughing')
stemmerporter1.stem('happiness')
```

In [ ]:
```
from nltk.stem import RegexpStemmer
stemmerregexp=RegexpStemmer('learn')
stemmerregexp.stem('learning')
stemmerregexp1=RegexpStemmer('ing')
stemmerregexp1.stem('singing')
```

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

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

In [ ]:
```
texts="Over the years, advancements in the work have been incorporated towards professor relationships and data c
for text in texts:
    sentences=nltk.sent_tokenization(text) #list of sentences
    for sentance in sentances:
        words=nltk.word_tokenize(sentance)
        print(words)
        tagged=nltk.pos_tag(words)
        print(tagged)
```

In [ ]:
```

```