In [ ]:

```
#SWETHA JENIFER_28-2-23
```

NLP_LAB8_Exploring Part of Speech Tagging on Large Text Files

In [1]:

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\sweth\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[1]:

```
True
```

In [2]:

```
import glob
import nltk
import pandas as pd
from nltk import *
import zipfile
from nltk.corpus import stopwords
stop_words = set (stopwords.words('english'))
```

In [3]:

```
files="All About Eve.txt"
f=open(files,'r')
content=f.read()
f.close()
```

In [4]:

```
from nltk.tokenize import sent_tokenize
sentences=sent_tokenize(content)
len(sentences)
```

Out[4]:

```
7
```

In [5]:

```
word=nltk.tokenize.WhitespaceTokenizer()
words=word.tokenize(content)
len(words)
```

Out[5]:

```
224
```

In [6]:

```
top10w=FreqDist(words)
top10w.most_common(10)
```

Out[6]:

```
[('the', 12),
 ('of', 8),
 ('and', 8),
 ('in', 6),
 ('for', 6),
 ('Best', 5),
 ('Mankiewicz', 4),
 ('from', 3),
 ('“All', 3),
 ('About', 3)]
```

In [10]:

```
import nltk
nltk.download('averaged_perceptron_tagger')
```

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

Out[10]:

```
True
```

In [11]:

```
tag = []
d_tags = []
words = [w for w in words if not w in stop_words]
tagged = nltk.pos_tag(words)
for i in tagged:
    (word,pos)=i
    tag.append(pos)
for j in tag:
    if j not in d_tags:
        d_tags.append(j)
len(d_tags)
```

Out[11]:

```
19
```

In [12]:

```
top_pos=FreqDist(tagged)
top_pos.most_common(10)
```

Out[12]:

```
[(('Best', 'NNP'), 5),
 (('Mankiewicz', 'NNP'), 4),
 (('“All', 'NN'), 3),
 (('About', 'IN'), 3),
 (('retrospective', 'JJ'), 2),
 (('two', 'CD'), 2),
 (('Actress', 'NNP'), 2),
 (('Eve”', 'NNP'), 2),
 (('greatest', 'JJS'), 2),
 (('Some', 'DT'), 1)]
```

In [14]:

```
noun=0
for i in top_pos.keys():
    (word,pos)=i
    if pos == 'NN' or pos == 'NNS' or pos == 'NNP' or pos == 'NNPS':
        noun+=1
print(noun)
```

```
67
```

In [15]:

```
verbs=0
for i in top_pos.keys():
    (word,pos)=i
    if pos == 'VB' or pos == 'VBD' or pos == 'VBN' or pos == 'VBP' or pos ==
        verbs+=1
print(verbs)
```

```
17
```

In [16]:

```
adj = []
for i in top_pos.keys():
    (word,pos)=i
    if pos == 'JJ' or pos == 'JJR' or pos == 'JJS':
        adj.append(i)
len(adj)
```

Out[16]:

```
17
```

In [17]:

```
adv=[]
for i in top_pos.keys():
    (word,pos)=i
    if pos == 'RB' or pos == 'RBR' or pos == 'RBS' or pos == 'BP':
        adv.append(i)
len(adv)
```

Out[17]:

```
4
```

In [18]:

```
adv = FreqDist(adv)
adv.most_common(1)
```

Out[18]:

```
[(('Not', 'RB'), 1)]
```

In [19]:

```
adv = FreqDist(adj)
adv.most_common(1)
```

Out[19]:

```
[(('best', 'JJS'), 1)]
```