


SURIYA S_225229140 

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 [20]: files="Psycho.txt"
f=open(files,'r')
content=f.read()
f.close()
```

```
In [21]: from nltk.tokenize import sent_tokenize
sentences=sent_tokenize(content)
len(sentences)
```

Out[21]: 24

```
In [22]: word=nltk.tokenize.WhitespaceTokenizer()
words=word.tokenize(content)
len(words)
```

Out[22]: 612

```
In [23]: top10w=FreqDist(words)
top10w.most_common(10)
```

```
Out[23]: [('the', 50),
          ('of', 26),
          ('and', 20),
          ('a', 18),
          ('to', 14),
          ('is', 14),
          ('in', 12),
          ('as', 9),
          ('his', 7),
          ('Hitchcock', 5)]
```

```
In [24]: 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] Package averaged_perceptron_tagger is already up-to-
[nltk_data] date!
```

```
Out[24]: True
```

```
In [25]: 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[25]: 20
```

```
In [26]: top_pos=FreqDist(tagged)
top_pos.most_common(10)
```

```
Out[26]: (((('Hitchcock', 'NNP'), 5),
            (('The', 'DT'), 5),
            (('Paramount', 'NNP'), 3),
            (('murder', 'NN'), 3),
            (('John', 'NNP'), 3),
            (('June', 'NNP'), 2),
            (('Alfred', 'NNP'), 2),
            (('mystery', 'NN'), 2),
            (('Psycho', 'NNP'), 2),
            (('New', 'NNP'), 2)])
```

```
In [27]: 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)
```

161

```
In [28]: 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)
```

42

```
In [29]: 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[29]: 67

```
In [30]: 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[30]: 15

```
In [31]: adv = FreqDist(adv)
adv.most_common(1)
```

Out[31]: [(('prior', 'RB'), 1)]

```
In [32]: adv = FreqDist(adj)
adv.most_common(1)
```

Out[32]: [(('iconic', 'JJ'), 1)]

