Name : Asha Belcilda P. ¶

Rollno : 225229104

LAB_08 : EXPLORING POS OF LARGE TEXT FILES

Exercise-1

In [1]:
```
import pandas as pd
```

In [3]:
```
txt1 = open("12 Angry Men.txt", "r")
txt1 = txt1.read()
print(txt1)
```

```
Lumet's origins as a director of teledrama may well be obvious here in his fi
rst film, but there is no denying the suitability of his style - sweaty close
-ups, gritty monochrome 'realism', one-set claustrophobia - to his subject. S
cripted by Reginald Rose from his own teleplay, the story is pretty contrived
- during a murder trial, one man's doubts about the accused's guilt gradually
overcome the rather less-than-democratic prejudices of the other eleven membe
rs of the jury - but the treatment is tense, lucid, and admirably economical.
Fonda, though typecast as the bastion of liberalism, gives a nicely underplay
ed performance, while Cobb, Marshall and Begley in particular are highly effe
ctive in support. But what really transforms the piece from a rather talky de
monstration that a man is innocent until proven guilty, is the consistently t
aut, sweltering atmosphere, created largely by Boris Kaufman's excellent came
rawork. The result, however devoid of action, is a strangely realistic thrill
er.
```

In [4]:
```
import glob
import nltk
import pandas as pd
from nltk import *
from zipfile import ZipFile
from nltk.corpus import stopwords
```

In [5]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
stop_words = set(stopwords.words('english'))
```

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

A. How many sentence in the file??

In [6]:
```
from nltk.tokenize import sent_tokenize
sentences=sent_tokenize(txt1)
len(sentences)
```

Out[6]: 5

B. How many words in the file??

In [7]:
```
from nltk.tokenize import word_tokenize
words_in = nltk.tokenize.WhitespaceTokenizer()
words = words_in.tokenize(txt1)
len(words)
```

Out[7]: 155

C. What are the top 10 words and their counts??

In [8]:
```
top_10 = FreqDist(words)
top_10.most_common(10)
```

Out[8]:
```
[('the', 10),
 ('a', 6),
 ('of', 6),
 ('is', 6),
 ('his', 4),
 ('-', 4),
 ('in', 3),
 ('as', 2),
 ('but', 2),
 ('by', 2)]
```

D. How many different POS tags are represented in this

file??

In [9]:
```
nltk.download('averaged_perceptron_tagger')
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)
```

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

E. What are the top 10 POS tags and their counts??

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

Out[10]:
```
[(('-', ':'), 4),
 (('rather', 'RB'), 2),
 (("Lumet's", 'NNP'), 1),
 (('origins', 'VBZ'), 1),
 (('director', 'NN'), 1),
 (('teledrama', 'NN'), 1),
 (('may', 'MD'), 1),
 (('well', 'RB'), 1),
 (('obvious', 'VB'), 1),
 (('first', 'JJ'), 1)]
```

F. How many nouns in the file??

In [11]:
```
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)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
```

G. How many verbs in the file??

In [12]:
```
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)
```

```
1
2
3
4
5
6
7
8
9
10
```

H. How many adjectives in the file??

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]: 19

I. How many adverbs in the file??

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]: 13

J. What is the most frequent adverb??

In [18]:
```
adv = FreqDist(adv)
adv.most_common(1)
```

Out[18]:
```
[(('well', 'RB'), 1)]
```

K. What is the most frequent adjective??

In [20]:
```
adj = FreqDist(adj)
adj.most_common(1)
```

Out[20]:
```
[(('first', 'JJ'), 1)]
```

In [ ]:
```

```