NAME :RETHINAGIRI G

ROLL NO : 225229130

COURSE TITLE : NATURAL LANGUAGE PRE-PROCESSING LAB

LAB.08 Exploring Part of Speech Tagging on Large Text Files ¶

1. Open any movie file from your movies sub directory

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

In [3]:
```
import glob
import nltk
```

In [4]:
```
f="movies.zip"
with ZipFile(f,"r") as file:
    file.printdir()
    file.extractall('.')
```

```
File Name                                             Modified             Size
movies/                                        2018-01-19 08:32:38            0
movies/12 Angry Men.txt                        2018-01-17 20:40:42         1007
movies/12 Years a Slave.txt                    2018-01-17 20:42:50         6451
movies/4 Months, 3 Weeks and 2 Days.txt        2018-01-17 20:37:10         1151
movies/All About Eve.txt                       2018-01-17 20:33:18         1346
movies/American Graffiti.txt                   2018-01-17 20:44:30         3417
movies/Boyhood.txt                             2018-01-17 20:27:14         1970
movies/Casablanca.txt                          2018-01-17 20:26:26         1896
movies/Citizen Kane.txt                        2018-01-17 20:23:56         1483
movies/Gone with the Wind.txt                  2018-01-17 20:38:10         1318
movies/Hoop Dreams.txt                         2018-01-17 20:34:12         7909
movies/Manchester by the Sea.txt               2018-01-17 20:40:06         3674
movies/Moonlight.txt                           2018-01-17 20:31:42         2323
movies/My Left Foot.txt                        2018-01-17 20:38:50         1115
movies/Pan's Labyrinth.txt                     2018-01-17 20:32:18         4431
movies/Psycho.txt                              2018-01-17 20:34:46         3727
movies/Ran.txt                                 2018-01-17 20:43:48         2207
movies/Singin' in the Rain.txt                 2018-01-17 20:29:42          782
movies/Some Like It Hot.txt                    2018-01-17 20:35:40         7489
movies/The Godfather.txt                       2018-01-17 20:25:32         4293
movies/Three Colors Red.txt                    2018-01-17 20:28:22         2892
```

In [5]:
```
files=[file for file in glob.glob("movies/*")]
with open (files[19],'r',encoding="cp1252") as f:
 count =f.read()
 print(count)
```

```
In one second, the time it takes to look away from the road to tune a car radio, drop a book to the street,
or bump into a stranger, the path of one's life can be drastically altered, as well as the lives of people n
ot even known. Red, a masterful film of chance encounters, parallel lives, mysterious coincidence, and mutua
l wonderment, is the latest from director Krzysztof Kieslowski. The final and best chapter of his “Three Col
ors” trilogy, based on the colors of the French flag (blue/liberty, white/equality, red/fraternity), Red see
ms at first more about isolation than fraternity. The young model Valentine (Jacob, from Kieslowski's The Do
uble Life of Veronique) leads a mundane existence, prolonging a brittle long-distance relationship. She has
never met Auguste (Lorit), a law student who lives right around the corner. The retired judge (Trintignant)
spends his days eavesdropping on others, seemingly having given up on a life of his own. When Valentine appe
ars at the judge's door with his injured dog, she is rudely dismissed. Repulsed by his indifference and his
eavesdropping, Valentine nonetheless returns to him, and slowly, like absorbing the warmth of the sun, they
are drawn to each other, opening up in completely unexpected ways. Together with longtime collaborator Krzys
ztof Piesiewicz, Kieslowski has devised an utterly contemporary fable, imbued with a richness and complexit
y, yet so understated that the visuals often carry the story. Haunting imagery, swathed in red, permeates th
e screen. Red holds on to its mysteries tenaciously, revealing them slowly, if ever. Kieslowski's detractors
label him pretentious and murky, and they are not likely to change their minds here. Yet Kieslowski understa
nds that life's beauty is not in spite of the unexpected and unexplained, but because of it. The improbabili
ties of chance are endless; what might have happened if you'd been looking the other way, spoken to the pers
on sitting next to you, been born 20 years earlier… as in life, you never know where this story is going. Th
e film courses with vitality -- and makes you glad to be alive. Kieslowski's deft touch gives Red its real m
agic; in the end, the subtle nuances are what stay with you: a ringing phone just out of reach, late afterno
on sunlight brushing across a face, a shattered beer glass in a bowling alley. Despite the neat tie-in at th
e film's conclusion, you need not have seen the other two films in this series to follow what is going on he
re (though both Blue and White are on video and recommended). Kieslowski, who finds the filmmaking process a
rduous, has announced his retirement. At age 53, with only a handful of remarkable features to his name (inc
luding the epic Decalogue), it remains to be seen whether he will keep his word, but either way, he could no
t have chosen a more memorable testament to his singular talent and vision than Red.
```

a. How many sentences in the file?

In [6]:
```
nltk.download('punkt')
```

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

Out[6]:
```
True
```

In [7]:
```
from nltk.tokenize import sent_tokenize
sentences=sent_tokenize(count)
len(sentences)
```

Out[7]:
```
19
```

b. How many words in the file?

In [8]:
```
from nltk.tokenize import word_tokenize
word=word_tokenize(count)
len(word)
```

Out[8]:
```
569
```

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

In [9]:
```
from nltk import *
freq_word=FreqDist(word)
freq_word
```

Out[9]:
```
FreqDist({',': 46, 'the': 25, '.': 19, 'a': 15, 'to': 14, 'of': 13, 'and': 12, 'his': 10, "'s": 7, 'Kieslowsk
i': 7, ...})
```

In [10]:
```
freq_word.most_common(10)
```

Out[10]:
```
[(',', 46),
 ('the', 25),
 ('.', 19),
 ('a', 15),
 ('to', 14),
 ('of', 13),
 ('and', 12),
 ('his', 10),
 ("'s", 7),
 ('Kieslowski', 7)]
```

d. How many different POS tags are represented in this file?

In [11]:
```
from nltk.corpus import stopwords
stop_words=set(stopwords.words('english'))
```

In [12]:
```
nltk.download('averaged_perceptron_tagger')
```

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

Out[12]:
```
True
```

In [13]:
```
tag=[]
d_tags=[]
word=[w for w in word if not w in stop_words]
tagged=nltk.pos_tag(word)
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[13]:
```
26
```

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

In [14]:
```
top_pos=FreqDist(tagged)
top_pos.most_common(10)
```

Out[14]:
```
[((',', ','), 46),
 (('.', '.'), 19),
 (("'s", 'POS'), 7),
 (('Kieslowski', 'NNP'), 7),
 (('The', 'DT'), 6),
 (('(', '('), 6),
 ((')', ')'), 6),
 (('life', 'NN'), 4),
 (('Red', 'NNP'), 4),
 (('film', 'NN'), 3)]
```

f. How many nouns in the file?

In [17]:
```
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)
```

```
114
```

g. How many verbs in the file?

In [18]:
```
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 == 'VBZ':
        verbs+=1
print(verbs)
```

```
41
```

h. How many adjectives in the file?

In [19]:
```
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[19]:
```
40
```

i. How many adverbs in the file?

In [20]:
```
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[20]:
```
15
```

j. What is the most frequent adverb ?

In [21]:
```
adv = FreqDist(adv)
adv.most_common(1)
```

Out[21]:
```
[(('away', 'RB'), 1)]
```

k. What is the most frequent adjective?

In [22]:
```
adj = FreqDist(adj)
adj.most_common(1)
```

Out[22]:
```
[(('masterful', 'JJ'), 1)]
```