NAME : RETHINAGIRI G

ROLL NO : 225229130

COURSE TITLE : NATURAL LANGUAGE PRE-PROCESSING LAB

LAB_05. Stemming and Lemmatization on Movie Dataset

In [1]:

```
from zipfile import ZipFile
import glob
import pandas as pd
import nltk
```

In [2]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
from nltk.corpus import stopwords
```

In [3]:

```
import warnings
warnings.filterwarnings('ignore')
```

Exercise 1

In [4]:

```
f="movies.zip"
with ZipFile(f,'r') as zip:
    zip.printdir()
```

```
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 [13]:

```
nltk.download('punkt')
nltk.download('stopwords')
stop_words=set(stopwords.words("english"))
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\1mscdsa39\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\1mscdsa39\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [14]:

```
from nltk.stem import PorterStemmer
ps=PorterStemmer()
tokenizer=nltk.tokenize.WhitespaceTokenizer()
from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()
from nltk.stem import LancasterStemmer
ls=LancasterStemmer()
```

In [15]:

```
files = [file for file in glob.glob("movies/*")]
for file in files:
    with open(file, 'r', encoding='cp1252') as f:
        contents=f.readlines()
        print(contents)
        print("-----------------------------------------------------------------------------------------------------------------------------------------------------")
        print("")
        print("The above is content end")
        print("")
        print("-----------------------------------------------------------------------------------------------------------------------------------------------------")
```

```
["Lumet's origins as a director of teledrama may well be obvious here in his first film, but there is no denying the suit
ability of his style - sweaty close-ups, gritty monochrome 'realism', one-set claustrophobia - to his subject. Scripted b
y 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 members of the jury - b
ut the treatment is tense, lucid, and admirably economical. Fonda, though typecast as the bastion of liberalism, gives a
nicely underplayed performance, while Cobb, Marshall and Begley in particular are highly effective in support. But what r
eally transforms the piece from a rather talky demonstration that a man is innocent until proven guilty, is the consisten
tly taut, sweltering atmosphere, created largely by Boris Kaufman's excellent camerawork. The result, however devoid of a
ction, is a strangely realistic thriller."]
-----------------------------------------------------------------------------------------------------------------------------------------------------
---

The above is content end

-----------------------------------------------------------------------------------------------------------------------------------------------------
---
['There are movies to which the critical response lags far behind the emotional one. Two days after seeing 12 Years a Sla
ve, British director Steve McQueen’s adaptation of the 1853 memoir of a free black man kidnapped into slavery, I’m still
awaiting delivery of the apparatus that would permit me to analyze it. So overpowering is this film’s simple, horrible, a
```

In [16]:

```
files = [file for file in glob.glob("movies/*")]
for file in files:
    with open(file, 'r', encoding='cp1252') as f:
        contents = f.readlines()
        for row in contents:
            sent_text = nltk.sent_tokenize (row)
        print("sentence tokenize",len(sent_text))
        for row1 in contents:
            words= nltk.word_tokenize(row1)
        print("word tokenize ", len (words))
        for row2 in contents:
            filtered_sentence = [w for w in words if not w in stop_words]
        print("stopwords", len(filtered_sentence))
        print("-----------------------------------")
```

```
sentence tokenize 5
word tokenize  181
stopwords 122
---------------------------------
sentence tokenize 2
word tokenize  119
stopwords 68
---------------------------------
sentence tokenize 1
word tokenize  20
stopwords 11
---------------------------------
sentence tokenize 7
word tokenize  276
stopwords 178
---------------------------------
sentence tokenize 1
word tokenize  9
stopwords 7
---------------------------------
sentence tokenize 4
word tokenize  70
stopwords 45
---------------------------------
sentence tokenize 2
word tokenize  49
stopwords 25
---------------------------------
sentence tokenize 3
word tokenize  98
stopwords 52
---------------------------------
sentence tokenize 6
word tokenize  242
stopwords 157
---------------------------------
sentence tokenize 4
word tokenize  67
stopwords 46
---------------------------------
sentence tokenize 6
word tokenize  131
stopwords 81
---------------------------------
sentence tokenize 5
word tokenize  157
stopwords 101
---------------------------------
sentence tokenize 4
word tokenize  69
stopwords 43
---------------------------------
sentence tokenize 2
word tokenize  66
stopwords 41
---------------------------------
sentence tokenize 3
word tokenize  39
stopwords 28
---------------------------------
sentence tokenize 1
word tokenize  25
stopwords 18
---------------------------------
sentence tokenize 2
word tokenize  50
stopwords 33
---------------------------------
sentence tokenize 8
word tokenize  208
stopwords 129
---------------------------------
sentence tokenize 6
word tokenize  100
stopwords 64
---------------------------------
sentence tokenize 19
word tokenize  569
stopwords 365
---------------------------------
```

In [17]:

```
def port_stemSentence (sentence):
    tokenizer = nltk.tokenize.WhitespaceTokenizer()
    tok = tokenizer.tokenize(sentence)
    filtered_sentence = [w for w in tok if not w in stop_words]
    stem_sentence = []
    for word in filtered_sentence:
        stem_sentence.append(ps.stem (word))
    return len(stem_sentence)
```

In [22]:

```
files = [file for file in glob.glob("movies/*")]
for file in files:
        with open(file, 'r', encoding='cp1252') as f:
            contents = f.readline()
            print("porter_stemming")
            print(port_stemSentence (contents))
        print("------------------------------")
```

```
porter_stemming
96
------------------------------
porter_stemming
83
------------------------------
porter_stemming
20
------------------------------
porter_stemming
138
------------------------------
porter_stemming
63
------------------------------
porter_stemming
64
------------------------------
porter_stemming
20
------------------------------
porter_stemming
51
------------------------------
porter_stemming
131
------------------------------
porter_stemming
27
------------------------------
porter_stemming
53
------------------------------
porter_stemming
87
------------------------------
porter_stemming
35
------------------------------
porter_stemming
93
------------------------------
porter_stemming
23
------------------------------
porter_stemming
34
------------------------------
porter_stemming
52
------------------------------
porter_stemming
38
------------------------------
porter_stemming
33
------------------------------
porter_stemming
282
------------------------------
```

In [23]:

```
def lan_stemSentence (sentence):
    tokenizer = nltk.tokenize. WhitespaceTokenizer()
    tok = tokenizer.tokenize (sentence)
    filtered_sentence = [w for w in tok if not w in stop_words]
    stem_sentence = []
    for word in filtered_sentence:
        stem_sentence.append(ls.stem(word))
        return len(stem_sentence)
```

In [27]:

```
files = [file for file in glob.glob("movies/*")]
for file in files:
    with open(file, 'r', encoding='cp1252') as f:
        contents=f.readline()
        print("lancaster_stemming ")
        print(port_stemSentence (contents))
        print('-------------------------------------------------')
```

```
lancaster_stemming
96
-------------------------------------------------
lancaster_stemming
83
-------------------------------------------------
lancaster_stemming
20
-------------------------------------------------
lancaster_stemming
138
-------------------------------------------------
lancaster_stemming
63
-------------------------------------------------
lancaster_stemming
64
-------------------------------------------------
lancaster_stemming
20
-------------------------------------------------
lancaster_stemming
51
-------------------------------------------------
lancaster_stemming
131
-------------------------------------------------
lancaster_stemming
27
-------------------------------------------------
lancaster_stemming
53
-------------------------------------------------
lancaster_stemming
87
-------------------------------------------------
lancaster_stemming
35
-------------------------------------------------
lancaster_stemming
93
-------------------------------------------------
lancaster_stemming
23
-------------------------------------------------
lancaster_stemming
34
-------------------------------------------------
lancaster_stemming
52
-------------------------------------------------
lancaster_stemming
38
-------------------------------------------------
lancaster_stemming
33
-------------------------------------------------
lancaster_stemming
282
-------------------------------------------------
```

In [30]:

```
def lemmSentence (sentence):
    tokenizer = nltk.tokenize. WhitespaceTokenizer()
    tok = tokenizer.tokenize (sentence)
    filtered_sentence=[w for w in tok if not w in stop_words]
    lemm_sentence = []
    for word in filtered_sentence:
        lemm_sentence.append(lemmatizer.lemmatize (word))
    return len(lemm_sentence)

for file in files:
    with open(file, 'r', encoding='cp1252') as f:
        contents = f.readline()
        print("lemmatization ")
        print(lemmSentence (contents))
    print("---------------------------------------")
```

```
lemmatization
96
---------------------------------------
lemmatization
83
---------------------------------------
lemmatization
20
---------------------------------------
lemmatization
138
---------------------------------------
lemmatization
63
---------------------------------------
lemmatization
64
---------------------------------------
lemmatization
20
---------------------------------------
lemmatization
51
---------------------------------------
lemmatization
131
---------------------------------------
lemmatization
27
---------------------------------------
lemmatization
53
---------------------------------------
lemmatization
87
---------------------------------------
lemmatization
35
---------------------------------------
lemmatization
93
---------------------------------------
lemmatization
23
---------------------------------------
lemmatization
34
---------------------------------------
lemmatization
52
---------------------------------------
lemmatization
38
---------------------------------------
lemmatization
33
---------------------------------------
lemmatization
282
---------------------------------------
```

Exercise -II

Step 1

In [32]:

```
tok = []
for file in files:
    with open(file, 'r', encoding='cp1252') as f:
        contents = f.read()
        let=tokenizer.tokenize(contents)
        tok.append(let)
tok
```

Out[32]:

```
[["Lumet's",
  'origins',
  'as',
  'a',
  'director',
  'of',
  'teledrama',
  'may',
  'well',
  'be',
  'obvious',
  'here',
  'in',
  'his',
  'first',
  'film,',
  'but',
  'there',
```

In [33]:

```
tok_lem =[]
for i in tok:
    for j in i:
        to_lem=lemmatizer.lemmatize(j)
        tok_lem.append(to_lem)
tok_lem
```

```
 'his',
 'first',
 'film,',
 'but',
 'there',
 'is',
 'no',
 'denying',
 'the',
 'suitability',
 'of',
 'his',
 'style',
 '-',
 'sweaty',
 'close-ups,',
 'gritty',
 'monochrome',
 "'realism',",
```

Step 2

In [35]:

```
for file in files:
    with open(file, 'r', encoding='cp1252') as f:
        contents=f.read()
        tok=tokenizer.tokenize(contents)
        filtered_sentence=[w for w in tok if not w in stop_words]
        tfidf=TfidfVectorizer (min_df=2, max_df=0.5,ngram_range=(1,2))
        features=tfidf.fit_transform(filtered_sentence)
        df=pd.DataFrame(features.todense(),columns=tfidf.get_feature_names())
        print(df)
print("----------------------------------------")
```

```
    man  one  rather
0   0.0  0.0     0.0
1   0.0  0.0     0.0
2   0.0  0.0     0.0
3   0.0  0.0     0.0
4   0.0  0.0     0.0
5   0.0  0.0     0.0
6   0.0  0.0     0.0
7   0.0  0.0     0.0
8   0.0  0.0     0.0
9   0.0  0.0     0.0
10  0.0  0.0     0.0
11  0.0  0.0     0.0
12  0.0  0.0     0.0
13  0.0  0.0     0.0
14  0.0  0.0     0.0
15  0.0  0.0     0.0
16  0.0  0.0     0.0
17  0.0  0.0     0.0
```

Step 3

In [38]:

```
with open(files [9], 'r', encoding='cp1252') as f:
    contents=f.read()
    tok=tokenizer.tokenize(contents)
    filtered_sentence=[w for w in tok if not w in stop_words]
    tfidf=TfidfVectorizer (min_df=2, max_df=0.5,ngram_range=(1,2))
    movie1=tfidf.fit_transform(filtered_sentence)
    print(movie1)
```

```
  (1, 39)       1.0
  (2, 73)       1.0
  (3, 57)       1.0
  (4, 32)       1.0
  (6, 43)       1.0
  (7, 64)       1.0
  (9, 125)      1.0
  (11, 125)     1.0
  (12, 76)      1.0
  (13, 125)     1.0
  (19, 125)     1.0
  (20, 64)      1.0
  (22, 125)     1.0
  (25, 71)      1.0
  (27, 57)      1.0
  (28, 32)      1.0
  (30, 85)      1.0
  (31, 70)      1.0
  (32, 30)      1.0
  (33, 123)     1.0
  (34, 4)       1.0
  (35, 68)      1.0
  (37, 133)     1.0
  (38, 46)      1.0
  (39, 8)       1.0
  :     :
  (724, 40)     1.0
  (733, 39)     1.0
  (736, 1)      1.0
  (738, 105)    1.0
  (741, 74)     1.0
  (742, 123)    1.0
  (745, 43)     1.0
  (746, 15)     1.0
  (750, 71)     1.0
  (751, 103)    1.0
  (754, 58)     1.0
  (763, 38)     1.0
  (766, 71)     1.0
  (768, 120)    1.0
  (772, 101)    1.0
  (776, 83)     1.0
  (783, 38)     1.0
  (784, 57)     1.0
  (785, 32)     1.0
  (787, 30)     1.0
  (788, 64)     1.0
  (789, 2)      1.0
  (796, 64)     1.0
  (797, 85)     1.0
  (798, 52)     1.0
```

In [37]:

```
with open(files [6], 'r', encoding='cp1252') as f:
    contents=f.read()
    tok=tokenizer.tokenize(contents)
    filtered_sentence=[w for w in tok if not w in stop_words]
    tfidf=TfidfVectorizer (min_df=2, max_df=0.5,ngram_range=(1,2))
    movie1=tfidf.fit_transform(filtered_sentence)
    print(movie1)
```

```
  (4, 11)       1.0
  (8, 3)        1.0
  (20, 9)       1.0
  (22, 13)      1.0
  (28, 2)       1.0
  (38, 12)      1.0
  (44, 9)       1.0
  (45, 13)      1.0
  (47, 10)      1.0
  (48, 5)       1.0
  (55, 14)      1.0
  (58, 1)       1.0
  (59, 0)       1.0
  (60, 2)       1.0
  (71, 0)       1.0
  (72, 14)      1.0
  (74, 3)       1.0
  (76, 4)       1.0
  (107, 11)     1.0
  (116, 10)     1.0
  (117, 5)      1.0
  (127, 7)      1.0
  (130, 7)      1.0
  (147, 15)     0.7071067811865476
  (147, 8)      0.7071067811865476
  (154, 12)     1.0
  (155, 2)      1.0
  (173, 1)      1.0
  (179, 6)      1.0
  (183, 4)      1.0
  (190, 8)      1.0
  (194, 6)      1.0
  (195, 14)     1.0
  (199, 15)     1.0
```

In [39]:

```
doc1=movie1 [0:10]
doc2=movie1[:]
score=linear_kernel(doc1, doc2)
print(score)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Exercise-III

```
Lemmatization has higher accuracy than stemming.
Lemmatization is preferred for context analysis, whereas stemming is recommended when the context is not important.
```