from sipplife inter sipplife import glob import pandes as pd. gill. troping from . Sklearn. feature - extraction. Lext import . 1209d Vactorizer drom skleann. Nedrics. pairwise import Lineau terne! sollk. Copus import Stopwords Import warnings. Wannings. Allerwannings ('Postore') Exe :1 O fle-name = "movles. zlp" with applied the name, (x') as zip: zip. print dir () Liles = [Lile for file in glob. glob("movies ("")) fles. nitk. daonload ('punkt') MIK. download (1 stopwoods) Stope words = set (stopwords, words ('english')) tokenizer = nltk. bokenize. Whitespace tokenizer () from Mith. stem import porter stemmer. PS = portexStemmer () from Mtb. stem import Lancaster stemmer. ls = Lan Caster Schemmer () from nikk. Istern import word Nethermatizer Lemmatiter = Word Nettommatiter ()

# Natural Language Processing Lab Lab5. Stemming and Lemmatization on Movie Dataset

## **EXERCISE-1**

The file movie.zip contains 20 files about various movies. For each of the files in movies.zip, you will have to do the following:

- A. How many sentences in each file?
- B. How many tokens in each file?
- C. How many tokens excluding stop words in each file?
- D. How many unique stems (ie., stemming) in each file? (Use PorterStemmer)
- E. How many unique stems (ie., stemming) in each file? (Use LancasterStemmer)
- F. How many unique words (ie., lemmatization) in each file? (Use WordNetLemmatizer)
- G. Pretty Printing: Print the details of A to E in the following order

File Name Sentences Tokens Tokens-Only StemsPorter StemsLancaseter Lemmas

### **EXERCISE-2**

In this exercise, you will build your Term-Document Matrix for this movie collection of 20 movies. In order to improve the similarity search experience, you will use only lemmatized terms for creating the matrix.

#### Step-1

For each movie:

- Tokenize terms and build list of tokens
- Find lemmatized words from the tokens

#### Step-2

Build Term-Document matrix using TfIdfVectorizer

#### Step-3

Take vectors of any two movies and compute cosine similarity

#### **EXERCISE-3**

Will lemmatized matrix help to achieve better similarity search or not?. Please comment.

DEPTIOF DATA SCIENCE (BISHOP HEBER COLLEGE ) TRICHY

DR K RAIKUMAR

. for fle Pn files: with open (tile, r, encoding: (cp1252) as b: Contents: f. readlines() print (contents) print /11 A- How many · Senterces Pn each file? 13. How many tokens in each file.) C. How many tokens excluding stop woods in each file) files: [tiles for file in glob. glob (1 maries | \*") ] los file in fles; With open (bile, 'r', encoding = (cp1252) as f: for you in Contents: Sent-text=nltk. Sent-toboriae (row) Print (" centence toporite", bus (sent-text)) for now in contents words = nltk. wood fotente (row 1) Print ("Word tokenfre", bn (words)) Altered gentences = [ W for w in words if not w Tebrow-gols of Print ("Stopwords", lonfültered - Senten (03)) Frant ( + x xx xx) def fort stemsentence (sendence)? tope fopenfor. Cokenfre (sentence) fil tened sentence = [w for w in fok if not w stem\_santaro arrord) Stem\_ Senfence = [] for word an filtered - senter e: | return (en/stem-sentence)

```
NOTES
```

tox fles . In fles! With open (tile, 'Y', encoding = 1 Cp12501) as for. contents: f. readline() print (" porter\_ Stemming") print (port\_ stem Sentence (contents)) Print (" \* \* K \* \* \* K \* K \* K \* ") F). for lan\_ dom Sentence (sentence): tok = tokenizes. tokenize (sentence). Bleesed\_Sentence = [w for w in tok if not w in step\_woods] Stem\_sentence = () for word in filtered\_sentence. ster-sertence. append (18. stem (word))
return by (stem-sentence) for file. in files: with open (file, 'r', encoding = (cp (258') as 6: contents = freed line() Point (" (ancaster - a temming") Print (lan-Sem Senten O (contents)) Print(" x x x x x x x x x x x x x) F) alst. lemmsentence (sentence): tok = fokenizer. tokenize (sentence). filtered - senten a = [ w for w for tok if not w Isbrow got Lemm\_sentences=[] for word in piltered sentence. lemm-sentence. append (bmmatizer. bmmatizer return long (lomm- Sentence) DR: K. RAJKUMAR DEPT OF DATA SCIENCE (BISHOP HEBER COLLEGE BURICH)

```
for the
        In files!
     With open (fib, 1 r), encoding: 12 p12601) as to
            Condends: for readline ()
              print ("lemmatration")
            point ( lemm Sentence (contento))
            Cxosoze-E
           tok=[]
        for fle in fles:
            with open (file, 1r', encoding = (cp1252) as b:
               Content = 1. read()
                let = tokenizer, tokenize (contents)
                tok. append(let)
       GOK.
tor-lemo IJ
   for ? . In tok:
         for 3 Pai:
              fox lems lemmatizer, lemmatize (i)
              tok-lem. append (to-lem)
        tok-lem
 etep!a
        for the hafter:
        with open (tile, 'r', enading: (cp1252)) as b'.
          Contents = fired()
           fox; tokeniser. fokenise (contents)
       di ltered centence = [ w for win tok if not win
                                       Stop_words 7
                                                        Gid = Tol Vectorizer (min-d)=2, max-db=0.5,
       fathures > With fit - transform (filtered - dentence)
db = pd. Data frame (features. footinger), ldums = thirty get feature.
```

Scanned with CamScanner

NOTES

1000

With open (free [to], (x', 2nading = 'cq 10 na') on d: Contents = fixed () bot: Coronizex. Coronize (Contants) belowed sendence - [w for w in tok if not win filply: They voicion ? zex (min-de = 2, max. de = 0.5) movie 1: (fitt. fit - tranform (fittered .. sentence) polat ( mor lo 1 With open (tiles [10], "", encoding = 'cp 1260') as for (ortens=fored() tok = fokenfren. Expenfrencontens) for ferred, semence = (w for w in tok if not win stop-words) tofield: The vectorizer (Min. d):2, max. di =0.6, ngram rarge= (1/2)) marie > = flid f. fit - transform (filtered, sentence) point (movie 2) doc1 = marie 1 [0310] docs = movie 2 [:] George & Great - Kennel (doc't doce) print (score)

POEPT OF DATA SCIENCE, BISHOP HEBER COLLEGES TRICHY

DR. K. RAJKUMAR