# CSDL7013 Natural Language Processing Lab 8

- NOTE: Prepend your Roll Number to the name of this file

- YET ANOTHER NOTE: Take print out of the relevant code cells only.

## Cosine Distance Lab

1. **Title:** Cosine Distance

2. **Objective/Aim:** To illustrate Cosine Distance between Documents

3. **Due Date:** Friday September 30, 2022

### Name : Harshvardhan Singh

### Roll Number : 1019161

## Use the following helper code snippets to solve the exercises of this lab.

## Load the packages

```
#Uncomment the following lines

from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pandas as pd
```

## Q.1. Create a corpus of the following documents.

The world is full of obvious things which nobody by any chance ever observes.

I never guess. It is a shocking habit,destructive to the logical faculty.'

Eliminate all other factors, and the one which remains must be the truth.

How often have I said to you that when you have eliminated the impossible, whatever remains,however improbable, must be the truth?

It is an old maxim of mine that when you have excluded the impossible, whatever remains, however improbable, must be the truth.

```
# Uncomment the following lines

doc_sherlock_1 = "The world is full of obvious things which nobody by
```

```
any chance ever observes."
doc_sherlock_2 = "I never guess. It is a shocking habit, destructive
to the logical faculty."
doc_sherlock_3 = "Eliminate all other factors, and the one which
remains must be the truth."
doc_sherlock_4 = "How often have I said to you that when you have
eliminated the impossible, whatever remains,however improbable, must
be the truth?"
doc_sherlock_5 = "It is an old maxim of mine that when you have
excluded the impossible, whatever remains, however improbable, must be
the truth."

corpus = [doc_sherlock_1, doc_sherlock_2, doc_sherlock_3,
doc_sherlock_4, doc_sherlock_5]
```

## Q.2. Build a vector of unique terms. [Try bigrams and trigrams also]

```
# Check with each of unigram, bigrams and trigrams

terms = CountVectorizer(stop_words='english')  ### By default unigram
terms are returned.
terms = CountVectorizer(stop_words='english',ngram_range=(2,2))  ###
For bigrams.
terms = CountVectorizer(stop_words='english',ngram_range=(3,3))  ###
For trigrams.
vector = terms.fit_transform(corpus)
elements = terms.get_feature_names()
elements
```

```
c:\Users\admin\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in
1.0 and will be removed in 1.2. Please use get_feature_names_out
instead.
  warnings.warn(msg, category=FutureWarning)
```

```
['destructive logical faculty',
 'eliminate factors remains',
 'eliminated impossible remains',
 'excluded impossible remains',
 'factors remains truth',
 'guess shocking habit',
 'habit destructive logical',
 'impossible remains improbable',
 'maxim excluded impossible',
 'obvious things chance',
 'old maxim excluded',
 'remains improbable truth',
 'said eliminated impossible',
 'shocking habit destructive',
 'things chance observes',
 'world obvious things']
```

### Q.3. Display the frequency of these terms.

```
# Uncomment the following lines.

row_head = ['Sherlock 1','Sherlock 2','Sherlock 3','Sherlock
4','Sherlock 5']
term_matrix = vector.todense()
df = pd.DataFrame(term_matrix, columns=elements, index=row_head)
df
```

```
            destructive logical faculty  eliminate factors remains  \
Sherlock 1                            0                          0   
Sherlock 2                            1                          0   
Sherlock 3                            0                          1   
Sherlock 4                            0                          0   
Sherlock 5                            0                          0   

            eliminated impossible remains  excluded impossible remains 
\
Sherlock 1                              0                            0 

Sherlock 2                              0                            0 

Sherlock 3                              0                            0 

Sherlock 4                              1                            0 

Sherlock 5                              0                            1 


            factors remains truth  guess shocking habit  \
Sherlock 1                      0                     0   
Sherlock 2                      0                     1   
Sherlock 3                      1                     0   
Sherlock 4                      0                     0   
Sherlock 5                      0                     0   

            habit destructive logical  impossible remains 
improbable  \
Sherlock 1                          0                              0 

Sherlock 2                          1                              0 

Sherlock 3                          0                              0 

Sherlock 4                          0                              1 
Sherlock 5                          0                              1

            maxim excluded impossible  obvious things chance  \
Sherlock 1                          0                      1
Sherlock 2                          0                      0
Sherlock 3                          0                      0
Sherlock 4                          0                      0
Sherlock 5                          1                      0

            old maxim excluded  remains improbable truth  \
Sherlock 1                   0                        0
Sherlock 2                   0                        0
Sherlock 3                   0                        0
Sherlock 4                   0                        1
Sherlock 5                   1                        1

            said eliminated impossible  shocking habit destructive  \
Sherlock 1                           0                           0
Sherlock 2                           0                           1
Sherlock 3                           0                           0
Sherlock 4                           1                           0
Sherlock 5                           0                           0

            things chance observes  world obvious things
Sherlock 1                       1                     1
Sherlock 2                       0                     0
Sherlock 3                       0                     0
Sherlock 4                       0                     0
Sherlock 5                       0                     0
```

## Q.4. Display the tf-idf value of these terms.

```
# In place of CountVectorizer in solution to question 2 use this...
# Againg check with each of unigram, bigrams and trigrams.
terms = TfidfVectorizer(stop_words='english')
```

## Q.5. Find the cosine distance between the documents.

```
#Uncomment the following lines. Repeat the exercise with other
sentences. Satisfy yourself.

dist_matrix = cosine_similarity(df, df)
dist_matrix
```

```
array([[1.        , 0.        , 0.        , 0.        , 0.        ],
       [0.        , 1.        , 0.        , 0.        , 0.        ],
       [0.        , 0.        , 1.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 1.        , 0.4472136],
       [0.        , 0.        , 0.        , 0.4472136, 1.       ]])
```

## Exercise

## Do the same with the following sentences

```
doc_1 = "A graph is a non-linear data structure consisting of nodes 
and edges. The nodes are sometimes also referred to as vertices and 
the edges are lines or arcs that connect any two nodes in the graph."
doc_2 = "A graph can be defined as group of vertices and edges that 
are used to connect these vertices."
doc_3 = "A graph data structure consists of a finite (and possibly 
mutable) set of vertices (also called nodes or points), together with
a set of unordered pairs of these vertices for an undirected graph or
a set of ordered pairs for a directed graph."
doc_4 = "A tree is one of the data structures that represent 
hierarchical data."
doc_5 = "A tree data structure can be defined recursively as a 
collection of nodes, where each node is a data structure consisting of
a value and a list of references to nodes."

corpus = [doc_1,doc_2,doc_3,doc_4,doc_5]

terms = CountVectorizer(stop_words='english')  ### By default unigram
terms are returned.
terms = CountVectorizer(stop_words='english',ngram_range=(2,2))  ### 
For bigrams.
terms = CountVectorizer(stop_words='english',ngram_range=(3,3))  ### 
For trigrams.
vector = terms.fit_transform(corpus)
elements = terms.get_feature_names()
elements
```

```
c:\Users\admin\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function 
get_feature_names is deprecated; get_feature_names is deprecated in 
1.0 and will be removed in 1.2. Please use get_feature_names_out 
instead.
  warnings.warn(msg, category=FutureWarning)

['arcs connect nodes',
 'called nodes points',
 'collection nodes node',
 'connect nodes graph',
 'consisting nodes edges',
 'consisting value list',
 'consists finite possibly',
 'data structure consisting',
 'data structure consists',
 'data structure defined',
 'data structures represent',
 'defined group vertices',
 'defined recursively collection',
 'edges lines arcs',
 'edges nodes referred',
 'edges used connect',
 'finite possibly mutable',
 'graph data structure',
 'graph defined group',
 'graph non linear',
 'graph set ordered',
 'group vertices edges',
 'linear data structure',
 'lines arcs connect',
 'list references nodes',
 'mutable set vertices',
 'node data structure',
 'nodes edges nodes',
 'nodes node data',
 'nodes points set',
 'nodes referred vertices',
 'non linear data',
 'ordered pairs directed',
 'pairs directed graph',
 'pairs vertices undirected',
 'points set unordered',
 'possibly mutable set',
 'recursively collection nodes',
 'referred vertices edges',
 'represent hierarchical data',
 'set ordered pairs',
 'set unordered pairs',
 'set vertices called',
 'structure consisting nodes',
 'structure consisting value',
 'structure consists finite',
 'structure defined recursively',
 'structures represent hierarchical',
 'tree data structure',
 'tree data structures',
 'undirected graph set',
 'unordered pairs vertices',
 'used connect vertices',
 'value list references',
 'vertices called nodes',
 'vertices edges lines',
 'vertices edges used',
 'vertices undirected graph']
```

```
row_head = ['doc_1','doc_2','doc_3','doc_4','doc_5']
term_matrix = vector.todense()
df = pd.DataFrame(term_matrix, columns=elements, index=row_head)
df
```

```
       arcs connect nodes  called nodes points  collection nodes node
\
doc_1                   1                    0                       0

doc_2                   0                    0                       0

doc_3                   0                    1                       0

doc_4                   0                    0                       0

doc_5                   0                    0                       1


       connect nodes graph  consisting nodes edges  consisting value
list  \
doc_1                    1                       1
0
doc_2                    0                       0
0
doc_3                    0                       0
0
doc_4                    0                       0
0
doc_5                    0                       0
1

       consists finite possibly  data structure consisting  \
doc_1                         0                          1
doc_2                         0                          0
doc_3                         1                          0
doc_4                         0                          0
doc_5                         0                          1

       data structure consists  data structure defined  ...  \
doc_1                        0                       0  ...
doc_2                        0                       0  ...
doc_3                        1                       0  ...
doc_4                        0                       0  ...
doc_5                        0                       1  ...

       tree data structure  tree data structures  undirected graph set
\
doc_1                    0                     0                     0
```

```
doc_2                    0                     0                     0

doc_3                    0                     0                     1

doc_4                    0                     1                     0

doc_5                    1                     0                     0


       unordered pairs vertices  used connect vertices  value list
references  \
doc_1                         0                      0
0
doc_2                         0                      1
0
doc_3                         1                      0
0
doc_4                         0                      0
0
doc_5                         0                      0
1

       vertices called nodes  vertices edges lines  vertices edges
used  \
doc_1                      0                     1
0
doc_2                      0                     0
1
doc_3                      1                     0
0
doc_4                      0                     0
0
doc_5                      0                     0
0

       vertices undirected graph
doc_1                          0
doc_2                          0
doc_3                          1
doc_4                          0
doc_5                          0

[5 rows x 58 columns]
```

```
terms = TfidfVectorizer(stop_words='english')
```

```
dist_matrix = cosine_similarity(df, df)
dist_matrix
```

```
array([[1.        , 0.        , 0.        , 0.        , 0.07161149],
       [0.        , 1.        , 0.        , 0.        , 0.        ],
       [0.        , 0.        , 1.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 1.        , 0.        ],
       [0.07161149, 0.        , 0.        , 0.        , 1.        ]])
```