**IR Assignment:**

**# Hari Sai Raghuram Veeramallu**

**# 1610110145**

import math

import numpy as np

def cal\_vector\_dot\_prod(doc1, doc2):

size = min(len(doc1), len(doc2))

f\_doc = 0

for i in range(0,size):

f\_doc = f\_doc + (doc1[i] \* doc2[i])

return f\_doc

def sim(doc1, doc2):

num = cal\_vector\_dot\_prod(doc1, doc2)

den = (math.sqrt(cal\_vector\_dot\_prod(doc1, doc1))) \* (math.sqrt(cal\_vector\_dot\_prod(doc2, doc2)))

return ((num/den))

def similarity\_score(i, k1, k2): # Single-link

return max(sim(i, k1), sim(i, k2)) # min for complete-link

def efficient\_hac(docs):

C = [[0 for x in range(len(docs))] for y in range(len(docs))]

act\_cls = [0 for x in range(len(docs))]

priority = [0 for x in range((len(docs) \* len(docs)))]

for n in range(0,len(docs)):

for i in range(0, len(docs)):

sim = cal\_vector\_dot\_prod(docs[n], docs[i])

ind = i

C[n][i] = (sim, ind)

act\_cls[n] = 1

priority[n] = sorted(C[n], key = lambda x: x[1])

if C[n][n] in priority:

priority.remove(C[n][n])

A = []

for k in range(1, (len(docs)-1)):

indices = [j for j,e in enumerate(act\_cls) if e == 1]

k1 = np.argmax([max(priority[x])[0] for x in indices])

k2 = max(priority[k1])[1]

A.append((k1,k2))

act\_cls[k2] = 1

priority[k1] = []

req\_indices = [j for j,e in enumerate(act\_cls) if ((e == 1) and (e != k1))]

for i in req\_indices:

priority[i].remove(C[i][k1])

priority[i].remove(C[i][k2])

C[i][k1][0] = similarity\_score(i, k1, k2)

priority[i].append(C[i][k1])

C[k1][i][0] = similarity\_score(i, k1, k2)

priority[k1].append(C[k1][i])

return A

x = np.array([[3,4],

[4,4],

[5,4],

[6,4],

[7,4],

[8,4],

[9,4],

[10,4],

[1,4],

[2,4],

[1,5],

[2,5],

[3,6],

[4,5],

[5,5],

[6,6],

[7,5],

[8,6],

[9,5],

[10,6],])

efficient\_hac(x)





