Information Retrieval – Assignment 16

# Code

# IR19B.py CS5154/6054 cheng 2022  
# twice k-means  
# confusion matrix  
# NMI, RI, and purity  
# Usage: python IR19B.py  
  
import numpy as np  
import random  
from sklearn.feature\_extraction.text import TfidfVectorizer  
from sklearn.cluster import KMeans  
from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, normalized\_mutual\_info\_score, rand\_score  
import matplotlib.pyplot as plt  
  
f = open("bible.txt", "r")  
docs = f.readlines()  
f.close()  
N = 1000  
firstk = docs[0:N]  
  
cv = TfidfVectorizer(max\_df=0.4, min\_df=3)  
X = cv.fit\_transform(firstk)  
  
model = KMeans(n\_init=1, max\_iter=10)  
model.fit\_predict(X)  
y1 = model.labels\_  
  
model.fit\_predict(X)  
y2 = model.labels\_  
  
cm = confusion\_matrix(y1, y2)  
  
# Confusion Matrix, Similarity Scores, NMI, Rand index, and purity  
disp = ConfusionMatrixDisplay(cm)  
disp.plot()  
plt.title('Confusion matrix of the two clusterings')  
plt.show()  
print('Similarity scores between the two clusterings')  
nmi = normalized\_mutual\_info\_score(y1,y2)  
print(f'NMI = {nmi}')  
rand\_index = rand\_score(y1, y2)  
print(f'Rand Index = {rand\_index}')  
purity\_clustering\_1 = sum(np.max(cm, axis=0))/N  
print(f'Purity for clustering 1 = {purity\_clustering\_1}')  
purity\_clustering\_2 = sum(np.max(cm, axis=1))/N  
print(f'Purity for clustering 2 = {purity\_clustering\_2}')  
print('-----------------------------------------')  
  
# Transpose of cm  
cm\_transpose = cm.transpose()  
disp = ConfusionMatrixDisplay(cm\_transpose)  
disp.plot()  
plt.title('Confusion matrix of the two clusterings')  
plt.show()  
print('Similarity scores between the two clusterings')  
nmi = normalized\_mutual\_info\_score(y2, y1)  
print(f'NMI = {nmi}')  
rand\_index = rand\_score(y2, y1)  
print(f'Rand Index = {rand\_index}')  
purity\_clustering\_1 = sum(np.max(cm, axis=0))/N  
print(f'Purity for clustering 1 = {purity\_clustering\_1}')  
purity\_clustering\_2 = sum(np.max(cm, axis=1))/N  
print(f'Purity for clustering 2 = {purity\_clustering\_2}')  
print('-----------------------------------------')  
  
# cm2  
cm2 = cm \* 2 # Exercise 16.3: replace every point d with two identical copies  
disp = ConfusionMatrixDisplay(cm2)  
disp.plot()  
plt.title('Confusion matrix of the two clusterings')  
plt.show()  
print('Similarity scores between the two clusterings')  
nmi = normalized\_mutual\_info\_score(y1,y2)  
print(f'NMI = {nmi}')  
rand\_index = rand\_score(y1, y2)  
print(f'Rand Index = {rand\_index}')  
purity\_clustering\_1 = sum(np.max(cm, axis=0))/N  
print(f'Purity for clustering 1 = {purity\_clustering\_1}')  
purity\_clustering\_2 = sum(np.max(cm, axis=1))/N  
print(f'Purity for clustering 2 = {purity\_clustering\_2}')  
print('-----------------------------------------')

# Results

## Original Confusion Matrix

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

## Transpose of Confusion Matrix

Graphical user interface

Description automatically generated

Text

Description automatically generated

## Original Confusion Matrix x 2

Graphical user interface, application

Description automatically generated

Text

Description automatically generated