Experiment-03

Name: Atharva Paliwal

Roll No.: 40

Aim: Write a program to find the keywords for a text document. Use TF-IDF (Term Frequency-Inverse Document Frequency) to find the importance of different words in the document.

Theory:

What is TF-IDF?

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

Applications of TF-IDF:

Determining how relevant a word is to a document, or TD-IDF, is useful in many ways, for example:

- **Information retrieval**

TF-IDF was invented for document search and can be used to deliver results that are most relevant to what you're searching for. Imagine you have a search engine and somebody looks for LeBron. The results will be displayed in order of relevance. That's to say the most relevant sports articles will be ranked higher because TF-IDF gives the word LeBron a higher score.

It's likely that every search engine you have ever encountered uses TF-IDF scores in its algorithm.

- **Keyword Extraction**

TF-IDF is also useful for extracting keywords from text. How? The highest scoring words of a document are the most relevant to that document, and therefore they can be considered keywords for that document. Pretty straightforward.

Code:

```
import pandas as pd
import math

with open('text1.txt') as f:
    first1 = f.read()
with open('text2.txt') as s:
    second1 = s.read()
with open('text3.txt') as t:
    third1 = t.read()
first1=first1.split(" ")
second1=second1.split(" ")
third1=third1.split(" ")
total= set(first1).union(set(second1)).union(set(third1))
# print(total)

wordDictA = dict.fromkeys(total, 0)
wordDictB = dict.fromkeys(total, 0)
wordDictC = dict.fromkeys(total, 0)

for word in first1:
    wordDictA[word]+=1

for word in second1:
    wordDictB[word]+=1

for word in third1:
    wordDictC[word]+=1
```

```
pd.DataFrame([wordDictA, wordDictB, wordDictC])

wordDictA = dict.fromkeys(total, 0)
wordDictB = dict.fromkeys(total, 0)
wordDictC = dict.fromkeys(total, 0)

for word in first1:
    wordDictA[word]+=1

for word in second1:
    wordDictB[word]+=1

for word in third1:
    wordDictC[word]+=1

pd.DataFrame([wordDictA, wordDictB, wordDictC])

# import nltk
# nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
filtered_sentence = [w for w in wordDictA if not w in stop_words]
# print(filtered_sentence)

def computeIDF(docList):
    idfDict = {}
    N = len(docList)

    idfDict = dict.fromkeys(docList[0].keys(), 0)
    for word, val in idfDict.items():
```

```
        idfDict[word] = math.log10(N / (float(val) + 1))

    return(idfDict)

#inputing our sentences in the log file
idfs = computeIDF([wordDictA, wordDictB, wordDictC])

def computeTFIDF(tfBow, idfs):
    tfidf = {}
    for word, val in tfBow.items():
        tfidf[word] = val*idfs[word]
    return(tfidf)

#running our two sentences through the IDF:
idfFirst = computeTFIDF(wordDictA, idfs)
idfSecond = computeTFIDF(wordDictB, idfs)
idfThird = computeTFIDF(wordDictC, idfs)

#putting it in a dataframe
idf= pd.DataFrame([idfFirst, idfSecond])
print(idf)
```

Output:

```
PS C:\Users\ACER\Desktop\Courses\College Courses\WIBD\Lab\EXPT1> & C:/Users/ACER/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/ACER/Desktop/Courses/
College Courses/WIBD/Lab/EXPT1/prac3.py"
    Kodagu.    former  agriculturists  northwest,     above      with     rises  ...       was    State,    metres       and        It  separate   located
0  0.000000  0.477121             0.0    0.000000  0.000000  0.000000  0.000000  ...  0.477121  0.477121  0.000000  0.000000  0.000000  0.477121  0.000000
1  0.477121  0.000000             0.0    0.477121  0.477121  0.954243  0.477121  ...  0.000000  0.000000  1.431364  1.431364  0.954243  0.000000  0.954243

[2 rows x 113 columns]
PS C:\Users\ACER\Desktop\Courses\College Courses\WIBD\Lab\EXPT1>
```

***** END *****