DSBDA practical 7B

May 14, 2023

[1]:

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
```

[2]:

```
#Initialize the Documents.

documentA = 'Jupiter is the largest Planet'
documentB = 'Mars is the fourth planet from the Sun'
```

[3]:

```
#Create BagofWords (BoW) for Document A and B.

bagOfWordsA = documentA.split(' ')
bagOfWordsB = documentB.split(' ')
```

[4]:

```
#Create Collection of Unique words from Document A and B.

uniqueWords = set(bagOfWordsA).union(set(bagOfWordsB))
```

[9]:

```
# Create a dictionary of words and their occurrence for each document in the
#corpus

numOfWordsA = dict.fromkeys(uniqueWords, 0)
for word in bagOfWordsA:
 numOfWordsA[word] += 1
 numOfWordsB = dict.fromkeys(uniqueWords, 0)
for word in bagOfWordsB:
  numOfWordsB[word] += 1
```

[11]:

```
# Print the frequency dictionaries
print( numOfWordsA)
print( numOfWordsB)
```

```
{'from': 0, 'Mars': 0, 'is': 1, 'Sun': 0, 'Planet': 1, 'planet': 0, 'Jupiter':
1, 'the': 1, 'fourth': 0, 'largest': 1}
{'from': 1, 'Mars': 1, 'is': 1, 'Sun': 1, 'Planet': 0, 'planet': 1, 'Jupiter':
0, 'the': 2, 'fourth': 1, 'largest': 0}
```

[17]:

```
#Compute the term frequency for each of our documents.
```

```python
def computeTF(wordDict, bagOfWords):
    tfDict = {}
    bagOfWordsCount = len(bagOfWords)
    for word, count in wordDict.items():
        tfDict[word] = count / float(bagOfWordsCount)
    return tfDict

tfA = computeTF(numOfWordsA, bagOfWordsA)
tfB = computeTF(numOfWordsB, bagOfWordsB)
```

[18]:

```python
import math

def computeIDF(documents):
    N = len(documents)
    idfDict = dict.fromkeys(documents[0].keys(), 0)
    for document in documents:
        for word, val in document.items():
            if val > 0:
                idfDict[word] += 1
    for word, val in idfDict.items():
        idfDict[word] = math.log(N / float(val))
    return idfDict

idfs = computeIDF([numOfWordsA, numOfWordsB])
print(idfs)
```

```
{'from': 0.6931471805599453, 'Mars': 0.6931471805599453, 'is': 0.0, 'Sun':
0.6931471805599453, 'Planet': 0.6931471805599453, 'planet': 0.6931471805599453,
'Jupiter': 0.6931471805599453, 'the': 0.0, 'fourth': 0.6931471805599453,
'largest': 0.6931471805599453}
```

[36]:

```python
#Compute the term TF/IDF for all words.



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

tfidfA = computeTFIDF(numOfWordsA, idfs)
tfidfB = computeTFIDF(numOfWordsB, idfs)
df = pd.DataFrame([tfidfA, tfidfB])
print(df)
```

```
       from      Mars   is       Sun    Planet    planet   Jupiter  the
0  0.000000  0.000000  0.0  0.000000  0.693147  0.000000  0.693147  0.0  \
1  0.693147  0.693147  0.0  0.693147  0.000000  0.693147  0.000000  0.0
```

```
     fourth   largest
0  0.000000  0.693147
1  0.693147  0.000000
```

[ ]:

[ ]: