practice-assignment-7

May 4, 2025

[1]:
```python
import nltk
```

[3]:
```python
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger_eng')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     C:\Users\Varad\AppData\Roaming\nltk_data…
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
```

[3]:
```
True
```

[5]:
```python
text="India are the winners of ICC T20 World Cup 2024.In the ODI format, India␣
↪are the winners of the ICC Champions Trophy 2025."
```

```
[7]: #Sentence Tokenization
from nltk.tokenize import sent_tokenize
tokenized_text= sent_tokenize(text)
print(tokenized_text)

#Word Tokenization
from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

```
['India are the winners of ICC T20 World Cup 2024.In the ODI format, India are
the winners of the ICC Champions Trophy 2025.']
['India', 'are', 'the', 'winners', 'of', 'ICC', 'T20', 'World', 'Cup',
'2024.In', 'the', 'ODI', 'format', ',', 'India', 'are', 'the', 'winners', 'of',
'the', 'ICC', 'Champions', 'Trophy', '2025', '.']
```

```
[9]: # print stop words of English
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```
{'can', 'own', "you've", "they'd", 'for', 'its', 'will', "needn't", 'out',
'himself', 'if', 'such', 'down', 'which', 'only', "hadn't", 'no', 'your', 'don',
'further', 'all', 'she', 'ain', 'each', 'an', 'because', 'isn', 'these',
'myself', 'above', 'hadn', "he'd", 'be', 'wasn', 'more', 'that', 'over',
'before', 'has', "hasn't", 'itself', 'up', 'then', "i'll", 'wouldn', "shan't",
'shouldn', "you're", 'a', 'some', 'off', "they've", 'their', 'while', 'aren',
"it'll", 'both', 'we', 'ours', 'and', "mustn't", 'they', 'nor', 'mightn', 'had',
'hasn', "i'm", "weren't", 'than', 'hers', 've', "i'd", 'me', 'been', "you'd",
'are', 'you', "wouldn't", "isn't", 'my', 'until', "we'll", "we've", "don't",
"shouldn't", 'didn', 'them', 'being', 'needn', 'too', 'yours', 'theirs', 'so',
"they're", 'to', 'against', "that'll", 'what', "she's", "you'll", 'about',
'there', 'those', 'who', 'how', 'below', 'on', 'after', 'shan', "doesn't",
"he'll", "won't", 'where', 'ma', 's', 'few', 'd', 'm', 'do', 'am', 'it', 'why',
'themselves', 're', 'under', 'most', 'him', 'at', 'from', "it'd", "i've",
'ourselves', 'again', 'once', 'whom', 'll', 'doesn', 'not', 'y', "aren't", 'of',
'during', "didn't", 'won', "we're", 'doing', "couldn't", 'in', 'other', 'our',
'couldn', "should've", 'haven', 'were', 'same', 'herself', 'through', 'very',
'yourself', 'mustn', "mightn't", 'when', 'is', 'her', 'yourselves', 'as',
'here', 'by', 'should', "wasn't", 'i', 'does', 'into', 'any', 'he', "she'll",
'o', "haven't", 'this', 'his', 't', 'having', 'have', 'but', 'or', 'was', 'the',
'now', 'did', "it's", "they'll", 'just', "we'd", 'between', 'weren', 'with',
"she'd", "he's"}
```

```
[11]: import re
text= "How to remove stop words with NLTK library in Python?"
text= re.sub('[^a-zA-Z]', ' ',text)
```

```
tokens = word_tokenize(text.lower())
filtered_text=[]
for w in tokens:
  if w not in stop_words:
    filtered_text.append(w)
print("Tokenized Sentence:",tokens)
print("Filtered Sentence:",filtered_text)
```

```
Tokenized Sentence: ['how', 'to', 'remove', 'stop', 'words', 'with', 'nltk',
'library', 'in', 'python']
Filtered Sentence: ['remove', 'stop', 'words', 'nltk', 'library', 'python']
```

[13]:
```
from nltk.stem import PorterStemmer
e_words= ["wait", "waiting", "waited", "waits"]
ps =PorterStemmer()
for w in e_words:
  rootWord=ps.stem(w)
print(rootWord)
```

```
wait
```

[15]:
```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "Pune is a sprawling city of Maharashtra."
tokenization = nltk.word_tokenize(text)
for w in tokenization:
  print("Lemma for {} is {}".format(w,
wordnet_lemmatizer.lemmatize(w)))
```

```
Lemma for Pune is Pune
Lemma for is is is
Lemma for a is a
Lemma for sprawling is sprawling
Lemma for city is city
Lemma for of is of
Lemma for Maharashtra is Maharashtra
Lemma for . is .
```

[17]:
```
import nltk
from nltk.tokenize import word_tokenize
data="Virat and Rohit chased down a daunting total in yesterday's match."
words=word_tokenize(data)
for word in words:
  print(nltk.pos_tag([word]))
```

```
[('Virat', 'NNP')]
[('and', 'CC')]
[('Rohit', 'NN')]
```

```
[('chased', 'VBN')]
[('down', 'RB')]
[('a', 'DT')]
[('daunting', 'VBG')]
[('total', 'JJ')]
[('in', 'IN')]
[('yesterday', 'NN')]
[("'s", 'POS')]
[('match', 'NN')]
[('.', '.')]
```

[19]:
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
```

[21]:
```
documentA = "FC Barcelona is a professional football club based in Barcelona"
documentB = "FC Barcelona is one of the world's most decorated football clubs␣
  ↪and compete in La Liga."
```

[23]:
```
bagOfWordsA = documentA.split(' ')
bagOfWordsB = documentB.split(' ')
```

[25]:
```
uniqueWords = set(bagOfWordsA).union(set(bagOfWordsB))
uniqueWords
```

[25]:
```
{'Barcelona',
 'FC',
 'La',
 'Liga.',
 'a',
 'and',
 'based',
 'club',
 'clubs',
 'compete',
 'decorated',
 'football',
 'in',
 'is',
 'most',
 'of',
 'one',
 'professional',
 'the',
 "world's"}
```

[27]:
```
numOfWordsA = dict.fromkeys(uniqueWords, 0)
for word in bagOfWordsA:
```

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

[29]:
```python
def computeTF(wordDict, bagOfWords):
  tfDict = {}
  bagOfWordsCount = len(bagOfWords) # bagOfWords is defined as a parameter
  for word, count in wordDict.items():
    tfDict[word] = count / float(bagOfWordsCount)
  return tfDict
tfA = computeTF(numOfWordsA, bagOfWordsA) # bagOfWordsA is passed to bagOfWords
tfB = computeTF(numOfWordsB, bagOfWordsB) # bagOfWordsB is passed to bagOfWords
```

[31]:
```python
def computeIDF(documents):
  import math
  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])
```

[33]:
```python
def computeTFIDF(tfBagOfWords, idfs):
  tfidf = {}
  for word, val in tfBagOfWords.items():
    tfidf[word] = val * idfs[word]
  return tfidf

# Call the function with tfA or tfB as the first argument
# to calculate the TF-IDF for document A or B, respectively.
tfidfA = computeTFIDF(tfA, idfs)
tfidfB = computeTFIDF(tfB, idfs)

print("TF-IDF for document A:", tfidfA)
print("TF-IDF for document B:", tfidfB)
print("----------------")
df = pd.DataFrame([tfidfA, tfidfB])
df
```

```
TF-IDF for document A: {'based': 0.06931471805599453, 'most': 0.0, 'FC': 0.0,
'the': 0.0, 'decorated': 0.0, 'a': 0.06931471805599453, 'one': 0.0, 'of': 0.0,
'Liga.': 0.0, "world's": 0.0, 'is': 0.0, 'club': 0.06931471805599453, 'in': 0.0,
'compete': 0.0, 'clubs': 0.0, 'Barcelona': 0.0, 'football': 0.0, 'and': 0.0,
'La': 0.0, 'professional': 0.06931471805599453}
TF-IDF for document B: {'based': 0.0, 'most': 0.04332169878499658, 'FC': 0.0,
'the': 0.04332169878499658, 'decorated': 0.04332169878499658, 'a': 0.0, 'one':
0.04332169878499658, 'of': 0.04332169878499658, 'Liga.': 0.04332169878499658,
"world's": 0.04332169878499658, 'is': 0.0, 'club': 0.0, 'in': 0.0, 'compete':
0.04332169878499658, 'clubs': 0.04332169878499658, 'Barcelona': 0.0, 'football':
0.0, 'and': 0.04332169878499658, 'La': 0.04332169878499658, 'professional': 0.0}
----------------
```

[33]:
```
      based      most   FC       the  decorated         a       one        of  \
0  0.069315  0.000000  0.0  0.000000   0.000000  0.069315  0.000000  0.000000   
1  0.000000  0.043322  0.0  0.043322   0.043322  0.000000  0.043322  0.043322   

      Liga.   world's   is      club   in   compete     clubs  Barcelona  \
0  0.000000  0.000000  0.0  0.069315  0.0  0.000000  0.000000        0.0   
1  0.043322  0.043322  0.0  0.000000  0.0  0.043322  0.043322        0.0   

   football       and        La  professional  
0       0.0  0.000000  0.000000      0.069315  
1       0.0  0.043322  0.043322      0.000000  
```

[ ]: