CL-II 1 IR

July 15, 2025

0.0.1 1. Write a program for pre-processing of a text document such as stop word removal, stemming.

[ ]:
```
'''NAME:Aher Swami Sandip
ROLL NO.01
COURSE: AI&DS
CLASS: BE
SUB:Computer Laboratory-II (Information Retrival)'''
```

[ ]:
```
#stop words removal
```

[65]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[65]:
```
True
```

[67]:
```
import nltk
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt_tab is already up-to-date!
```

[67]:
```
True
```

[69]:
```
from nltk.corpus import stopwords
```

[71]:
```
nltk.download('stopwords')
print(stopwords.words('english'))
```

```
['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an',
'and', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be', 'because', 'been',
'before', 'being', 'below', 'between', 'both', 'but', 'by', 'can', 'couldn',
"couldn't", 'd', 'did', 'didn', "didn't", 'do', 'does', 'doesn', "doesn't",
'doing', 'don', "don't", 'down', 'during', 'each', 'few', 'for', 'from',
'further', 'had', 'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven',
```

```
"haven't", 'having', 'he', "he'd", "he'll", 'her', 'here', 'hers', 'herself',
"he's", 'him', 'himself', 'his', 'how', 'i', "i'd", 'if', "i'll", "i'm", 'in',
'into', 'is', 'isn', "isn't", 'it', "it'd", "it'll", "it's", 'its', 'itself',
"i've", 'just', 'll', 'm', 'ma', 'me', 'mightn', "mightn't", 'more', 'most',
'mustn', "mustn't", 'my', 'myself', 'needn', "needn't", 'no', 'nor', 'not',
'now', 'o', 'of', 'off', 'on', 'once', 'only', 'or', 'other', 'our', 'ours',
'ourselves', 'out', 'over', 'own', 're', 's', 'same', 'shan', "shan't", 'she',
"she'd", "she'll", "she's", 'should', 'shouldn', "shouldn't", "should've", 'so',
'some', 'such', 't', 'than', 'that', "that'll", 'the', 'their', 'theirs',
'them', 'themselves', 'then', 'there', 'these', 'they', "they'd", "they'll",
"they're", "they've", 'this', 'those', 'through', 'to', 'too', 'under', 'until',
'up', 've', 'very', 'was', 'wasn', "wasn't", 'we', "we'd", "we'll", "we're",
'were', 'weren', "weren't", "we've", 'what', 'when', 'where', 'which', 'while',
'who', 'whom', 'why', 'will', 'with', 'won', "won't", 'wouldn', "wouldn't", 'y',
'you', "you'd", "you'll", 'your', "you're", 'yours', 'yourself', 'yourselves',
"you've"]
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[73]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

[75]:
```
example_sent = "my name is Swami just don't forget it"
```

[77]:
```
stop_words = set(stopwords.words('english'))
```

[79]:
```
word_tokens = word_tokenize(example_sent)
```

[81]:
```
filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]
```

[83]:
```
filtered_sentence= []
```

[87]:
```
print(w)
```

```
it
```

[89]:
```
for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)
```

[91]:
```
print(word_tokens)
```

```
['my', 'name', 'is', 'Swami', 'just', 'do', "n't", 'forget', 'it']
```

[117]:
```
print(filtered_sentence)
```

```
['name', 'Swami', "n't", 'forget']
```

[57]:
```
# performing stopwords operation in file
```

[119]:
```
import io
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import os
os.getcwd()
```

[119]:
```
'C:\\Users\\Dell\\Computer Laboratory 2'
```

[121]:
```
stop_words = set(stopwords.words('english'))
```

[127]:
```
file1=open("text.txt",'r')
```

[129]:
```
line = file1.read()
words = line.split()
```

[131]:
```
file1 = open("text.txt", 'w')
```

[133]:
```
print(words)
```

```
[]
```

[135]:
```
for r in words:
    if not r in stop_words:
        appendFile=open('text.txt','a')
        appendFile.write(" "+r)
        appendFile.close()
```

[23]:
```
# stemming
```

[137]:
```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
```

[139]:
```
words = ["program", "programs", "programmer" ,"progamming" ,"programmers"]
```

[141]:
```
for w in words:
    print(w,":", ps.stem(w))
```

```
program : program
programs : program
programmer : programm
progamming : progam
programmers : programm
```

[113]:
```
#code 2 (stemming words from sentences)
```

[143]:
```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
sentence = "Programmer program with programming languages"
words = word_tokenize(sentence)
for w in words :
    print(w,":" , ps.stem(w))
```

```
Programmer : programm
program : program
with : with
programming : program
languages : languag
```

[ ]:
```

```