# DSBDA Practical 7 A

May 14, 2023

[3]:
```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\lalit\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

[16]:
```
#Initialize the text

text = "Tokenization is the first step in text analytics.The process of␣
 ↪breaking down a text paragraph into smaller chunks such as words or␣
 ↪sentences is called Tokenization."
```

[17]:
```
#Sentence Tokenization
from nltk.tokenize import sent_tokenize
tokenized_text= sent_tokenize(text)
print(tokenized_text)
```

```
['Tokenization is the first step in text analytics.The process of breaking down
a text paragraph into smaller chunks such as words or sentences is called
Tokenization.']
```

[18]:
```
# Word Tokenization
from nltk.tokenize import word_tokenize

text = "Tokenization is the first step in text analytics.The process of␣
 ↪breaking down a text paragraph into smaller chunks such as words or␣
 ↪sentences is called Tokenization."
tokens = word_tokenize(text)
print(tokens)
```

```
['Tokenization', 'is', 'the', 'first', 'step', 'in', 'text', 'analytics.The',
'process', 'of', 'breaking', 'down', 'a', 'text', 'paragraph', 'into',
'smaller', 'chunks', 'such', 'as', 'words', 'or', 'sentences', 'is', 'called',
'Tokenization', '.']
```

[19]:
```
#4: Removing Punctuations and Stop Word
```

```
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

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

[23]:
```
from nltk.corpus import stopwords
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("Filterd Sentence:",filtered_text)
```

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

[25]:
```
#perform Stemming

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
```

[32]:
```
#perofrm Lemmatization
import nltk
nltk.download('wordnet')

nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\lalit\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\lalit\AppData\Roaming\nltk_data…
```

[32]:
```
True
```

[33]:
```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
   print("Lemma for {} is {}".format(w,
wordnet_lemmatizer.lemmatize(w)))
```

```
Lemma for studies is study
Lemma for studying is studying
Lemma for cries is cry
Lemma for cry is cry
```

[35]:
```
#Apply POS Tagging to text
import nltk
nltk.download('averaged_perceptron_tagger')

from nltk.tokenize import word_tokenize
data="The pink sweater fit her perfectly"
words=word_tokenize(data)
for word in words:
  print(nltk.pos_tag([word]))
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\lalit\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

```
[('The', 'DT')]
[('pink', 'NN')]
[('sweater', 'NN')]
[('fit', 'NN')]
```

```
[('her', 'PRP$')]
[('perfectly', 'RB')]
```

[ ]: