In [22]:

```
import nltk
```

In [23]:

```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Out[23]:

```
True
```

In [24]:

```
text= "Tokenization is the first step in text analytics. The process of breaking down a
```

In [25]:

```
#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)
```

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

In [26]:

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

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

In [27]:

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

example_sent = """This is a sample sentence,
                  showing off the stop words filtration."""

stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example_sent)

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

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print(word_tokens)
print(filtered_sentence)
```

```
['This', 'is', 'a', 'sample', 'sentence', ',', 'showing', 'off', 'the', 's
top', 'words', 'filtration', '.']
['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtratio
n', '.']
```

In [28]:

```
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
wait
wait
wait
```

In [31]:

```
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
```

In [34]:

```
import nltk
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]))
```

```
[('The', 'DT')]
[('pink', 'NN')]
[('sweater', 'NN')]
[('fit', 'NN')]
[('her', 'PRP$')]
[('perfectly', 'RB')]
```