In [1]:

```
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\dell\appdata\local\program
s\python\python311\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\dell\appdata\local\progra
ms\python\python311\lib\site-packages (from nltk) (8.1.3)
Requirement already satisfied: joblib in c:\users\dell\appdata\local\progr
ams\python\python311\lib\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\dell\appdata\lo
cal\programs\python\python311\lib\site-packages (from nltk) (2023.3.23)
Requirement already satisfied: tqdm in c:\users\dell\appdata\local\program
s\python\python311\lib\site-packages (from nltk) (4.65.0)
Requirement already satisfied: colorama in c:\users\dell\appdata\local\pro
grams\python\python311\lib\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip is available: 23.0.1 -> 23.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [2]:

```
import nltk
```

In [3]:

```
from nltk import word_tokenize , sent_tokenize
```

In [4]:

```
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!
```

Out[4]:

```
True
```

Tokenising and removing stopwords

In [5]:

```
sent = "I will walk 500 miles and I would walk 500 more , just to be the man who walks a
print(word_tokenize(sent))
print(sent_tokenize(sent))
```

```
['I', 'will', 'walk', '500', 'miles', 'and', 'I', 'would', 'walk', '500',
'more', ',', 'just', 'to', 'be', 'the', 'man', 'who', 'walks', 'a', 'thous
and', 'miles', 'to', 'fall', 'down', 'at', 'your', 'door']
['I will walk 500 miles and I would walk 500 more , just to be the man who
walks a thousand miles to fall down at your door']
```

In [6]:

```
from nltk.corpus import stopwords
```

In [7]:

```
nltk.download('stopwords')
```

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

Out[7]:

```
True
```

In [8]:

```
stopwords = stopwords.words('english')
```

In [9]:

```
print(stopwords)
```

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

In [10]:

```
tokens = word_tokenize(sent)

cleaned_token = []

for word in tokens:
  if word not in stopwords:
    cleaned_token.append(word)
```

In [11]:

```
print(cleaned_token)
```

```
['I', 'walk', '500', 'miles', 'I', 'would', 'walk', '500', ',', 'man', 'wa
lks', 'thousand', 'miles', 'fall', 'door']
```

Stemming

In [12]:

```
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
```

In [13]:

```
sent2 = "I played the play playfully as the players were playing in the play with playfu
tokens2 = word_tokenize(sent2)
stemmed = ""

for words in tokens2:

  stemmed += stemmer.stem(words) + " "

print(stemmed)
```

```
i play the play play as the player were play in the play with playful
```

Snow-ball Stemmer

In [14]:

```
from nltk.stem.snowball import SnowballStemmer

snow_stemmer = SnowballStemmer(language='english')
```

In [15]:

```
sent3 = "I played the play playfully as the players were playing in the play with playfu
tokens3 = word_tokenize(sent3)

stem_words = []

for words in tokens3:
  stem_words.append(snow_stemmer.stem(words))

print(stem_words)

for e1,e2 in zip(tokens3,stem_words):
  print(e1 + " ---> " + e2)
```

```
['i', 'play', 'the', 'play', 'play', 'as', 'the', 'player', 'were', 'pla
y', 'in', 'the', 'play', 'with', 'playful']
I ---> i
played ---> play
the ---> the
play ---> play
playfully ---> play
as ---> as
the ---> the
players ---> player
were ---> were
playing ---> play
in ---> in
the ---> the
play ---> play
with ---> with
playfullness ---> playful
```

Tagging Parts of Speech (POS)

In [16]:

```
from nltk import pos_tag
nltk.download('averaged_perceptron_tagger')
```

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

Out[16]:

```
True
```

In [17]:

```
tagged = pos_tag(cleaned_token)
print(tagged)
```

```
[('I', 'PRP'), ('walk', 'VBP'), ('500', 'CD'), ('miles', 'NNS'), ('I', 'PR
P'), ('would', 'MD'), ('walk', 'VB'), ('500', 'CD'), (',', ','), ('man',
'NN'), ('walks', 'NNS'), ('thousand', 'VBP'), ('miles', 'NNS'), ('fall',
'VB'), ('door', 'NN')]
```

Lemmatization

In [18]:

```
nltk.download('wordnet')
```

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

Out[18]:

```
True
```

In [19]:

```
from nltk.stem import WordNetLemmatizer

obj = WordNetLemmatizer()
```

In [20]:

```
for words in cleaned_token:

  print(words + " ---> " + obj.lemmatize(words))
```

```
I ---> I
walk ---> walk
500 ---> 500
miles ---> mile
I ---> I
would ---> would
walk ---> walk
500 ---> 500
, ---> ,
man ---> man
walks ---> walk
thousand ---> thousand
miles ---> mile
fall ---> fall
door ---> door
```

Sample example of lemmatization

In [21]:

```
list1 = ["kites" , "babies" , "dogs" , "flying" , "smiling" , "driving" , "died" , "trie

for words in list1:

  print(words + " ---> " + obj.lemmatize(words))
```

```
kites ---> kite
babies ---> baby
dogs ---> dog
flying ---> flying
smiling ---> smiling
driving ---> driving
died ---> died
tried ---> tried
feet ---> foot
```

In [ ]: