In [1]:
```
#A Objective-02 Steps involved in Natural Language Processing
```

In [ ]:
```
#B SYNOPSIS AND ALGORITHM
```

In [ ]:
```
#C CODE
```

In [1]:
```
#1 Import Libraries
```

In [2]:
```
import nltk.corpus
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.util import bigrams,trigrams,ngrams
from nltk.stem import PorterStemmer
from nltk.stem import LancasterStemmer
from nltk.stem import wordnet
from nltk.stem import WordNetLemmatizer
from nltk import ne_chunk
```

In [9]:
```
#2 Download
```

In [10]:
```
nltk.download("punkt")
```

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

Out[10]:
```
True
```

In [3]:
```
#3 Find Stopwords
```

In [6]:
```
print(stopwords.words("English"))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "yo
u've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him',
'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its',
'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who',
'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'we
re', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did',
'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'wh
ile', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'thr
ough', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',
'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'he
re', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few',
'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'sam
e', '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', "had
n't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "might
n't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

In [11]:
```
#4 Display the length of stop words
```

In [12]:
```
len(stopwords.words("English"))
```

Out[12]:
```
179
```

In [8]:
```
#5 Tokenization
```

In [9]:
```
myName="My name is PRASHANT THANVI. My passion is music."
```

In [10]:
```

```

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

Out[10]:
```
True
```

In [11]:
```

```

In [12]:
```
name_tokens=word_tokenize(myName)
name_tokens
```

Out[12]:
```
['My',
 'name',
 'is',
 'PRASHANT',
 'THANVI',
 '.',
 'My',
 'passion',
 'is',
 'music',
 '.']
```

In [13]:
```
len(name_tokens)
```

Out[13]:
```
11
```

In [14]:
```
#probability
from nltk.probability import FreqDist
fdist=FreqDist()
```

In [15]:
```
for word in name_tokens:
    fdist[word.lower()]+=1
fdist
```

Out[15]:
```
FreqDist({'my': 2, 'is': 2, '.': 2, 'name': 1, 'prashant': 1, 'thanvi': 1, 'passio
n': 1, 'music': 1})
```

In [16]:
```
fdist["my"]
```

Out[16]:
```
2
```

In [17]:
```
#distinct words
len(fdist)
```

Out[17]:
```
8
```

In [18]:
```
#paraphrasing
```

In [19]:
```
#bigrams trigrams ngrams
```

In [20]:
```

```

In [21]:
```
str_new="Hello! Good morning to all."
str1=word_tokenize(str_new)
```

In [22]:
```
name_bigrams=list(nltk.bigrams(str1))
```

In [23]:
```
print(name_bigrams)
```

```
[('Hello', '!'), ('!', 'Good'), ('Good', 'morning'), ('morning', 'to'), ('to', 'al
l'), ('all', '.')]
```

In [24]:
```
name_trigrams=list(nltk.trigrams(str1))
print(name_trigrams)
```

```
[('Hello', '!', 'Good'), ('!', 'Good', 'morning'), ('Good', 'morning', 'to'), ('mo
rning', 'to', 'all'), ('to', 'all', '.')]
```

In [25]:
```
name_ngrams=list(nltk.ngrams(str1,5))
print(name_ngrams)
```

```
[('Hello', '!', 'Good', 'morning', 'to'), ('!', 'Good', 'morning', 'to', 'all'),
('Good', 'morning', 'to', 'all', '.')]
```

In [26]:
```
#words into its base stemming , lemmatization
```

In [27]:
```
#PorterStemmer

sps=PorterStemmer()
sps.stem("Played")
```

Out[27]:
```
'play'
```

In [28]:
```
sps.stem("Giving")
```

Out[28]:
```
'give'
```

In [29]:
```
sps.stem("taking")
```

Out[29]:
```
'take'
```

In [30]:
```
sps.stem("took")
```

Out[30]:
```
'took'
```

In [31]:
```
#Lancaster Stemmer
```

In [32]:
```
lst=LancasterStemmer()
lst.stem("took")
```

Out[32]:
```
'took'
```

In [33]:
```
#checkpoint
lst.stem("taken")
```

Out[33]:
```
'tak'
```

In [34]:
```
#Lemmatizer
#wordnet

word_lem=WordNetLemmatizer()
```

In [35]:
```
nltk.download('wordnet')
```

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

Out[35]:
```
True
```

In [36]:
```
nltk.download('omw-1.4')
```

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

Out[36]:
```
True
```

In [37]:
```
word_lem.lemmatize("taken")
```

Out[37]:
```
'taken'
```

In [38]:
```
word_lem.lemmatize("took")
```

Out[38]:
```
'took'
```

In [39]:
```
#PARTS OF SPEECH TAGGING
myName1="My name is PRASHANT THANVI."
token1=word_tokenize(myName1)
```

In [40]:
```
nltk.download('averaged_perceptron_tagger')
```

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

Out[40]:
```
True
```

In [41]:
```
nltk.pos_tag(token1)
```

Out[41]:
```
[('My', 'PRP$'),
 ('name', 'NN'),
 ('is', 'VBZ'),
 ('PRASHANT', 'NNP'),
 ('THANVI', 'NNP'),
 ('.', '.')]
```

In [42]:
```
#named entity recognition
#chunking in nlp
```

In [43]:
```

```

In [44]:
```
nltk.download('maxent_ne_chunker')
```

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

Out[44]:
```
True
```

In [45]:
```
nltk.download('words')
```

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

Out[45]:
```
True
```

In [46]:
```
query1="I live in Jodhpur."
q_tok=word_tokenize(query1)
q_tok
```

Out[46]:
```
['I', 'live', 'in', 'Jodhpur', '.']
```

In [47]:
```
#pos tag
q_tag=nltk.pos_tag(q_tok)
```

In [48]:
```
#ner chunking
```

In [49]:
```
ner_tok=ne_chunk(q_tag)
print(ner_tok)
```

```
(S I/PRP live/VBP in/IN (GPE Jodhpur/NNP) ./.)
```

In [3]:
```
#D CONCLUSION AND DISCUSSION
```

In [ ]:
```
```

In [4]:
```
#E VIVA QUESTIONS
```

In [5]:
```
#1 Define NLP.
```

In [6]:
```
#2 Differentiate between Lemmatization and Stemming.
```

In [7]:
```
#3 Define n-grams.
```

In [ ]:
```
#4 Define tokenization.
```

In [ ]:
```
#5 Define chunking in NLP.
```