1.Parts-Of-Speech taggeing in NLP

Using NLTK library

In [ ]:

```python
import nltk
nltk.download("punkt")
nltk.download('averaged_perceptron_tagger')
text="Ram ate the delicious chocolate"
from nltk.tokenize import word_tokenize
res=word_tokenize(text)
nltk.pos_tag(res)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

Out[6]:

```
[('Ram', 'NNP'),
 ('ate', 'VBP'),
 ('the', 'DT'),
 ('delicious', 'JJ'),
 ('chocolate', 'NN')]
```

Using Spacy library

In [ ]:

```python
import spacy
nlp=spacy.load("en_core_web_sm")
doc=nlp(text)
for word in doc:
  print(word,"|",word.pos_,"|",spacy.explain(word.pos_))
```

```
Ram | NOUN | noun
ate | VERB | verb
the | DET | determiner
delicious | ADJ | adjective
chocolate | NOUN | noun
```

In [ ]:

```
```

2.Perform Word Tokenization

In [1]:
```
import nltk
nltk.download("punkt")
from nltk.tokenize import word_tokenize
text='''He said "i need delicious chocolate." shopkeeper replied "That is not
word_tokenize(text)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

Out[1]:
```
['He',
 'said',
 '``',
 'i',
 'need',
 'delicious',
 'chocolate',
 '.',
 "''",
 'shopkeeper',
 'replied',
 '``',
 'That',
 'is',
 'not',
 'good',
 ',',
 'This',
 'is',
 'better',
 '!',
 "''"]
```

3. Perform Sentence Tokenization

In [2]:
```
from nltk.tokenize import sent_tokenize
sent_tokenize(text)
textnew="Mr. Anil is talking lot. He will change his place."
sent_tokenize(textnew)
```

Out[2]:
```
['Mr. Anil is talking lot.', 'He will change his place.']
```

In [3]:
```
texth='''आर्टिफिशियल इंटेलिजेंस की शुरुआत साल 1950 में हुई थी। यह ह्यूमन कंप्यूटर इंटरेक्शन
sent_tokenize(texth)
```

Out[3]:
```
['आर्टिफिशियल इंटेलिजेंस की शुरुआत साल 1950 में हुई थी। यह ह्यूमन कंप्यूटर इंटरेक्शन है।']
```

Using Spacy

In [4]:

```
import spacy
from spacy.lang.en import English
from spacy.lang.hi import Hindi
texth='''आर्टिफिशियल इंटेलिजेंस की शुरुआत साल 1950 में हुई थी। यह ह्यूमन कंप्यूटर इंटरेक्शन
textnew="Mr. Anil is talking lot. He will change his place."
nlp=English()
doc=nlp(text)
nlp.add_pipe('sentencizer')
doc_1=nlp(texth)
for i in doc_1.sents:
  print(i.text)
```

```
आर्टिफिशियल इंटेलिजेंस की शुरुआत साल 1950 में हुई थी।
यह ह्यूमन कंप्यूटर इंटरेक्शन है।
```

4. Perform Stemming

Using Porter Stemmer

In [13]:

```
import nltk
stemmer = nltk.PorterStemmer()
def stemming(data):
  d=dict()
  for w in data.split():
    d[w]=stemmer.stem(w)
  return d
```

In [14]:

```
stemmng=stemming("computing world is computed compulsive")
print(stemmng)
```

```
{'computing': 'comput', 'world': 'world', 'is': 'is', 'computed': 'comput',
'compulsive': 'compuls'}
```

Using snowball stemmer

In [6]:

```
from nltk.stem import SnowballStemmer
snow=SnowballStemmer('english')
for i in l:
  print(snow.stem(i))
```

```
eat
eat
eaten
eat
ate
talk
talk
meet
```

Using Lancaster stemmer

In [7]:
```
from nltk.stem import LancasterStemmer
lanc=LancasterStemmer()
for i in l:
  print(lanc.stem(i))
```

```
eat
eat
eat
eat
at
talk
talk
meet
```

5. Perform Lemmatization

In [8]:
```
from nltk.stem import WordNetLemmatizer
st=WordNetLemmatizer()
nltk.download('wordnet')
nltk.download('omw-1.4')
for i in l:
  print(i, " | ", st.lemmatize(i))
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...

Eat  |  Eat
Eats  |  Eats
Eaten  |  Eaten
Eating  |  Eating
ate  |  ate
Talk  |  Talk
Talking  |  Talking
meeting  |  meeting
```

Lemmatization Using Spacy

In [9]:
```
nlp=spacy.load("en_core_web_sm")
doc=nlp("Eat Eats Eaten Eating ate Talk Talking meeting")
for i in doc:
  print(i, " | ", i.lemma_)
```

```
Eat  |  eat
Eats  |  eat
Eaten  |  Eaten
Eating  |  Eating
ate  |  eat
Talk  |  talk
Talking  |  talk
meeting  |  meet
```

6. Perform n grams

In [15]:

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[15]:

```
True
```

In [16]:

```
from nltk.tokenize import word_tokenize
```

In [17]:

```
text="i ate chocolate cake"
r=word_tokenize(text)
```

In [18]:

```
from nltk import bigrams
list(bigrams(r))
```

Out[18]:

```
[('i', 'ate'), ('ate', 'chocolate'), ('chocolate', 'cake')]
```

In [19]:

```
from nltk import trigrams
list(trigrams(r))
```

Out[19]:

```
[('i', 'ate', 'chocolate'), ('ate', 'chocolate', 'cake')]
```

In [20]:

```
from nltk import everygrams
list(everygrams(r,4,4))
```

Out[20]:

```
[('i', 'ate', 'chocolate', 'cake')]
```

7. Remove stopwords

In [10]:

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words=stopwords.words('english')
#DataFrame.apply(Function_to_apply_to_each_row)
def rem_stopword(data):
  li=[]
  for w in data.split():
    if w not in stop_words:
      li.append(w)
  return " ".join(li)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

In [12]:

```
data="The course i am studying is AI ML "
print(rem_stopword(data.lower()))
```

```
course studying ai ml
```

In [ ]:

```
```