```
!pip install nltk
```

POS TAG

```
# POS TAGGING

import nltk
from nltk import word_tokenize


nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('maxent_ne_chunker')
nltk.download('maxent_ne_chunker_tab')
nltk.download('words')
nltk.download('averaged_perceptron_tagger')
nltk.download('averaged_perceptron_tagger_eng')
sent= "The quick brown fox jumps over the lazy dog."
tokens=word_tokenize(sent)
pos_tags=nltk.pos_tag(tokens)
print(pos_tags)
```

```
[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN'), ('jumps', 'VBZ'), ('over', 'IN'), ('the', 'DT'), ('lazy', 'JJ'), ('dog'
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker_tab to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker_tab is already up-to-date!
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
```

NER

```
import nltk
from nltk import word_tokenize, pos_tag,ne_chunk

nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('maxent_ne_chunker')
nltk.download('maxent_ne_chunker_tab')
nltk.download('words')
nltk.download('averaged_perceptron_tagger')
nltk.download('averaged_perceptron_tagger_eng')
sent= "Barack Obama was born in Hawaii in 1961 and was the president of the United States."
tokens=word_tokenize(sent)
pos_tags = pos_tag(tokens)
named_entities=ne_chunk(pos_tags)
print(named_entities)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker_tab to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping chunkers/maxent_ne_chunker_tab.zip.
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger_eng.zip.
(S
  (PERSON Barack/NNP)
  (PERSON Obama/NNP)
  was/VBD
  born/VBN
  in/IN
  (GPE Hawaii/NNP)
  in/IN
  1961/CD
  and/CC
  was/VBD
  the/DT
  president/NN
  of/IN
  the/DT
  (GPE United/NNP States/NNPS)
  ./.)
```

Token

```
t="""Hello Welcome,to NLP Tutorials.
Please do watch the entire tutorial to become expert in NLP.
"""
import nltk
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
from nltk.tokenize import wordpunct_tokenize
from nltk.tokenize import TreebankWordTokenizer
print(t)
nltk.download('punkt')
documents=sent_tokenize(t)
type(documents)
for sentence in documents:
    print(sentence)
word_tokenize(t)
for sentence in documents:
  print(word_tokenize(sentence))
wordpunct_tokenize(t)
tokenizer=TreebankWordTokenizer()
tokenizer.tokenize(t)
```

```
Hello Welcome,to NLP Tutorials.
Please do watch the entire tutorial to become expert in NLP.

Hello Welcome,to NLP Tutorials.
Please do watch the entire tutorial to become expert in NLP.
['Hello', 'Welcome', ',', 'to', 'NLP', 'Tutorials', '.']
['Please', 'do', 'watch', 'the', 'entire', 'tutorial', 'to', 'become', 'expert', 'in', 'NLP', '.']
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
['Hello',
 'Welcome',
 ',',
 'to',
 'NLP',
 'Tutorials.',
 'Please',
 'do',
 'watch',
 'the',
 'entire',
 'tutorial',
 'to',
 'become',
 'expert',
 'in',
 'NLP',
 '.']
```

Stemming

```
words=["eating","eats","eaten","writing","writes","programming","programs","history","finally","finalized"]
```

porter steeming

```
from nltk.stem import PorterStemmer
stemming=PorterStemmer()
for word in words:
    print(word+"---->"+stemming.stem(word))
stemming.stem('congratulations')
stemming.stem("sitting")
```

```
eating---->eat
eats---->eat
eaten---->eaten
writing---->write
writes---->write
programming---->program
programs---->program
history---->histori
finally---->final
finalized---->final
```

RegexpStemmer class

```
from nltk.stem import RegexpStemmer
reg_stemmer=RegexpStemmer('ing$|s$|e$|able$', min=4)
reg_stemmer.stem('eating')
for word in words:
    print(word+"---->"+reg_stemmer.stem(word))
```

```
eating---->eat
eats---->eat
eaten---->eaten
writing---->writ
writes---->write
programming---->programm
programs---->program
history---->history
finally---->finally
finalized---->finalized
```

Snowball Stemmer

```
from nltk.stem import SnowballStemmer
snowballsstemmer=SnowballStemmer('english')
for word in words:
    print(word+"---->"+snowballsstemmer.stem(word))
stemming.stem("fairly"),stemming.stem("sportingly")
snowballsstemmer.stem("fairly"),snowballsstemmer.stem("sportingly")
snowballsstemmer.stem('goes')
stemming.stem('goes')
```

```
eating---->eat
eats---->eat
eaten---->eaten
writing---->write
writes---->write
programming---->program
programs---->program
history---->histori
finally---->final
finalized---->final
```

stopword

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
nltk.download('punkt')
nltk.download('stopwords')
```

```
sentence = "This is a simple example showing how to remove stop words in NLP."
words = word_tokenize(sentence)
stop_words = set(stopwords.words('english'))
filtered_sentence = [word for word in words if word.lower() not in stop_words]
print("Original Sentence:", sentence)
print("Filtered Sentence:", ' '.join(filtered_sentence))
```

```
Original Sentence: This is a simple example showing how to remove stop words in NLP.
Filtered Sentence: simple example showing remove stop words NLP .
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

lemmatizers

```
## Q&A,chatbots,text summarization
import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
lemmatizer=WordNetLemmatizer()
'''
POS- Noun-n
verb-v
adjective-a
adverb-r
'''
lemmatizer.lemmatize("going",pos='v')
words=["eating","eats","eaten","writing","writes","programming","programs","history","finally","finalized"]
for word in words:
    print(word+"---->"+lemmatizer.lemmatize(word,pos='v'))
lemmatizer.lemmatize("goes",pos='v')
lemmatizer.lemmatize("fairly",pos='v'),lemmatizer.lemmatize("sportingly")
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
eating---->eat
eats---->eat
eaten---->eat
writing---->write
writes---->write
programming---->program
programs---->program
history---->history
finally---->finally
finalized---->finalize
('fairly', 'sportingly')
```