Basics of Natural LAnguage Processing

```
text = "I am learning AI Subject with the module of Natural Language Processing"
text
```

```
'I am learning AI Subject with the module of Natural Language Processing'
```

```
text.split()
```

```
['I',
 'am',
 'learning',
 'AI',
 'Subject',
 'with',
 'the',
 'module',
 'of',
 'Natural',
 'Language',
 'Processing']
```

```
'NaturalP' in text
```

```
False
```

```
words = text.split()
```

```
word = [w.lower() for w in words]
print(word)
```

```
['i', 'am', 'learning', 'ai', 'subject', 'with', 'the', 'module', 'of', 'natural', 'language', 'processing']
```

```
" ".join(word)
```

```
'i am learning ai subject with the module of natural language processing'
```

```
word = [w.upper() for w in words]
print(word)
```

```
['I', 'AM', 'LEARNING', 'AI', 'SUBJECT', 'WITH', 'THE', 'MODULE', 'OF', 'NATURAL', 'LANGUAGE', 'PROCESSING']
```

```
" ".join(word)
```

```
'I AM LEARNING AI SUBJECT WITH THE MODULE OF NATURAL LANGUAGE PROCESSING'
```

Punctuation Marks : !@#$%%^&*()":{}><>

```
import string
string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
text[5]
```

```
'l'
```

```
type(text)
```

```
str
```

```
import spacy
```

```
nlp = spacy.load('en_core_web_sm')
```

```
doc = nlp(text)
```

```
type(doc)
```

```
spacy.tokens.doc.Doc
```

```
text1 = "My name is Mukesh, learning Artificial Intelligence in MA112"
```

NER : Named Entity Recognition

```
doc = nlp(text1)
```

```
for token in doc.ents:
    print(token.text, token.label_)
```

```
Mukesh PERSON
Artificial Intelligence ORG
MA112 PRODUCT
Marwadi University ORG
Rajkot GPE
India GPE
11:00AM CARDINAL
```

```
doc = nlp("Red cars do not carry heigher insurance rate")
for chunk in doc.noun_chunks:
    print(chunk.text)
```

```
Red cars
heigher insurance rate
```

POS Tags : Part of speech tags

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

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\prash\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
True
```

```
doc = nlp(text1)
```

```
for token in doc:
    print(token.text, ":" , token.pos_)
```

```
My : PRON
name : NOUN
is : AUX
Mukesh : PROPN
, : PUNCT
learning : VERB
Artificial : PROPN
Intelligence : PROPN
in : ADP
MA112 : PROPN
of : ADP
Marwadi : PROPN
University : PROPN
, : PUNCT
Rajkot : PROPN
, : PUNCT
India : PROPN
from : ADP
11:00AM : PROPN
```

```
#stopWords
```

Start coding or generate with AI.

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

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

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
```

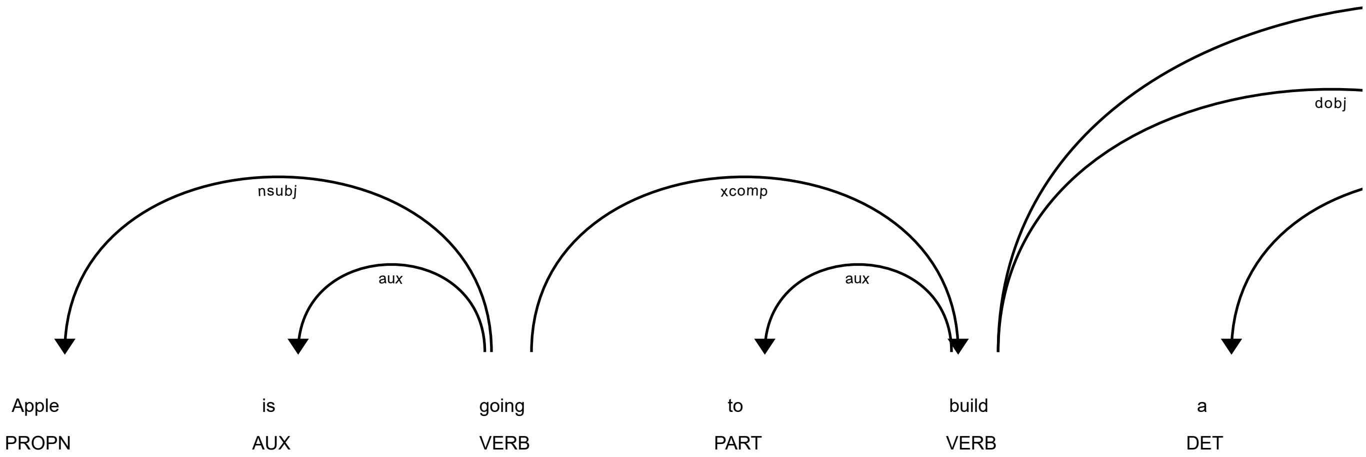
```
stop_words
```

```
{'a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'ain',
 'all',
 'am',
 'an',
 'and',
 'any',
 'are',
 'aren',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 'couldn',
 "couldn't",
 'd',
 'did',
 'didn',
 "didn't",
 'do',
 'does',
 'doesn',
 "doesn't",
 'doing',
 'don',
 "don't",
 'down',
 'during',
 'each',
 'few',
 'for',
 'from',
 'further',
 'had',
 'hadn',
 "hadn't",
 'has',
 'hasn',
 "hasn't",
 'have',
 'haven',
 "haven't",
 'having',
```

```
#Dependancy Parcing
```

```
from spacy import displacy
doc = nlp("Apple is going to build a U.K. Factory for $5 Million")
```

```
displacy.render(doc, style="dep", jupyter=True)
```



Stemming and Lemmetization

Stemming follows rule based approach Lemmetization is a word mapped corpus which is trained

```
text = "I studied artificial intelligence and then meeting Mr.Virat Tommorrow in a meeting"
```

```
from nltk.stem.porter import *
```

```
stemmer = PorterStemmer()
for word in text.split():
    print(word,": ",stemmer.stem(word))
```

```
I :  i
studied :  studi
artificial :  artifici
intelligence :  intellig
and :  and
then :  then
meeting :  meet
Mr.Virat :  mr.virat
Tommorrow :  tommorrow
in :  in
a :  a
meeting :  meet
```

```
doc = nlp(text)
```

```
for token in doc:
    print(token,":",token.pos_ ," : " , token.lemma_)
```

```
I : PRON  :  I
studied : VERB  :  study
artificial : ADJ  :  artificial
intelligence : NOUN  :  intelligence
and : CCONJ  :  and
then : ADV  :  then
meeting : VERB  :  meet
Mr. : PROPN  :  Mr.
Virat : PROPN  :  Virat
Tommorrow : PROPN  :  Tommorrow
in : ADP  :  in
a : DET  :  a
meeting : NOUN  :  meeting
```