Named Entity Recognition(NER)

Lib

In [7]:
```
import nltk
import pandas as pd
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('tagsets')
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\student\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\student\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package tagsets to
[nltk_data]     C:\Users\student\AppData\Roaming\nltk_data...
[nltk_data]   Package tagsets is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     C:\Users\student\AppData\Roaming\nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\student\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\words.zip.
```

Out[7]: True

DATA

In [8]:
```
text = "Apple acquired Zoom in China on Wednesday 6th May 2020.\
This news has made Apple and Google stock jump by 5% on Dow Jones Index in the \
United States of America"
```

In [9]:
```
#tokenize to words Basic Named Entity (NE) tagging using NLTK - Word based
words = nltk.word_tokenize(text)
words
```

Out[9]:
```
['Apple',
 'acquired',
 'Zoom',
 'in',
 'China',
 'on',
 'Wednesday',
 '6th',
 'May',
 '2020.This',
 'news',
 'has',
 'made',
 'Apple',
 'and',
 'Google',
 'stock',
 'jump',
 'by',
 '5',
 '%',
 'on',
 'Dow',
 'Jones',
 'Index',
 'in',
 'the',
 'United',
 'States',
 'of',
 'America']
```

In [10]:
```
#Part of speech tagging
pos_tags = nltk.pos_tag(words)
pos_tags
```

Out[10]:
```
[('Apple', 'NNP'),
 ('acquired', 'VBD'),
 ('Zoom', 'NNP'),
 ('in', 'IN'),
 ('China', 'NNP'),
 ('on', 'IN'),
 ('Wednesday', 'NNP'),
 ('6th', 'CD'),
 ('May', 'NNP'),
 ('2020.This', 'CD'),
 ('news', 'NN'),
 ('has', 'VBZ'),
 ('made', 'VBN'),
 ('Apple', 'NNP'),
 ('and', 'CC'),
 ('Google', 'NNP'),
 ('stock', 'NN'),
 ('jump', 'NN'),
 ('by', 'IN'),
 ('5', 'CD'),
 ('%', 'NN'),
 ('on', 'IN'),
 ('Dow', 'NNP'),
 ('Jones', 'NNP'),
 ('Index', 'NNP'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('United', 'NNP'),
 ('States', 'NNPS'),
 ('of', 'IN'),
 ('America', 'NNP')]
```

In [11]:
```
#check nltk help for description of the tag
nltk.help.upenn_tagset('NNP')
```

```
NNP: noun, proper, singular
    Motown Venneboerger Czestochwa Ranzer Conchita Trumplane Christos
    Oceanside Escobar Kreisler Sawyer Cougar Yvette Ervin ODI Darryl CTCA
    Shannon A.K.C. Meltex Liverpool ...
```

In [12]:
```
chunks = nltk.ne_chunk(pos_tags, binary=True) #either NE or not NE
for chunk in chunks:
    print(chunk)
```

```
(NE Apple/NNP)
('acquired', 'VBD')
('Zoom', 'NNP')
('in', 'IN')
(NE China/NNP)
('on', 'IN')
('Wednesday', 'NNP')
('6th', 'CD')
('May', 'NNP')
('2020.This', 'CD')
('news', 'NN')
('has', 'VBZ')
('made', 'VBN')
(NE Apple/NNP)
('and', 'CC')
(NE Google/NNP)
('stock', 'NN')
('jump', 'NN')
('by', 'IN')
('5', 'CD')
('%', 'NN')
('on', 'IN')
('Dow', 'NNP')
('Jones', 'NNP')
('Index', 'NNP')
('in', 'IN')
('the', 'DT')
(NE United/NNP States/NNPS)
('of', 'IN')
(NE America/NNP)
```

In [13]:
```
entities =[]
labels =[]
for chunk in chunks:
    if hasattr(chunk,'label'):
        #print(chunk)
        entities.append(' '.join(c[0] for c in chunk))
        labels.append(chunk.label())

entities_labels = list(set(zip(entities, labels)))
entities_df = pd.DataFrame(entities_labels)
entities_df.columns = ["Entities","Labels"]
entities_df
```

Out[13]:

	Entities	Labels
0	China	NE
1	Google	NE
2	America	NE
3	Apple	NE
4	United States	NE

In [14]:
```
chunks = nltk.ne_chunk(pos_tags, binary=False) #either NE or not NE
for chunk in chunks:
    print(chunk)

entities =[]
labels =[]
for chunk in chunks:
    if hasattr(chunk,'label'):
        #print(chunk)
        entities.append(' '.join(c[0] for c in chunk))
        labels.append(chunk.label())

entities_labels = list(set(zip(entities, labels)))
entities_df = pd.DataFrame(entities_labels)
entities_df.columns = ["Entities","Labels"]
entities_df
```

```
(PERSON Apple/NNP)
('acquired', 'VBD')
(PERSON Zoom/NNP)
('in', 'IN')
(GPE China/NNP)
('on', 'IN')
('Wednesday', 'NNP')
('6th', 'CD')
('May', 'NNP')
('2020.This', 'CD')
('news', 'NN')
('has', 'VBZ')
('made', 'VBN')
(PERSON Apple/NNP)
('and', 'CC')
(ORGANIZATION Google/NNP)
('stock', 'NN')
('jump', 'NN')
('by', 'IN')
('5', 'CD')
('%', 'NN')
('on', 'IN')
(PERSON Dow/NNP Jones/NNP Index/NNP)
('in', 'IN')
('the', 'DT')
(GPE United/NNP States/NNPS)
('of', 'IN')
(GPE America/NNP)
```

Out[14]:

	Entities	Labels
0	America	GPE
1	Apple	PERSON
2	Dow Jones Index	PERSON
3	United States	GPE
4	Google	ORGANIZATION
5	China	GPE

	Entities	Labels
6	Zoom	PERSON

In [15]:
```
#Basic Named Entity (NE) tagging using NLTK - Sentence based
entities = []
labels = []

sentence = nltk.sent_tokenize(text)
for sent in sentence:
    for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent)),binary=Fals
        if hasattr(chunk,'label'):
            entities.append(' '.join(c[0] for c in chunk))
            labels.append(chunk.label())

entities_labels = list(set(zip(entities,labels)))

entities_df = pd.DataFrame(entities_labels)
entities_df.columns = ["Entities","Labels"]
entities_df
```

Out[15]:

	Entities	Labels
0	America	GPE
1	Apple	PERSON
2	Dow Jones Index	PERSON
3	United States	GPE
4	Google	ORGANIZATION
5	China	GPE
6	Zoom	PERSON

In [17]:
```
import spacy
from spacy import displacy
#SpaCy 2.x brough significant speed and accuracy improvements
spacy.__version__
```

Out[17]: '3.4.1'

In [24]:
```
# Load SpaCy model
nlp = spacy.load("en_core_web_sm")
#nlp = spacy.load("en_core_web_md")
#nlp = spacy.load("en_core_web_lg")
```

In [25]:
```
doc = nlp(text)

entities = []
labels = []
position_start = []
position_end = []

for ent in doc.ents:
    entities.append(ent)
    labels.append(ent.label_)
    position_start.append(ent.start_char)
    position_end.append(ent.end_char)

df = pd.DataFrame({'Entities':entities,'Labels':labels,'Position_Start':position_

df
```

Out[25]:

	Entities	Labels	Position_Start	Position_End
0	(Apple)	ORG	0	5
1	(Zoom)	ORG	15	19
2	(China)	GPE	23	28
3	(Wednesday, 6th)	DATE	32	45
4	(Apple)	ORG	74	79
5	(5, %)	PERCENT	105	107
6	(Dow, Jones)	ORG	111	120
7	(the, United, States, of, America)	GPE	130	158

In [ ]:
```
```