Experiment 3

Aim

To perform

1) Chunking for verb/prepositional phrases using NLTK and Spacy.

2) POS Tagging, Count the POS tags (counter function) and visualization of Frequency distribution of each word in the graph (take a sample from corpus)

3) Named Entity Recognition

Theory

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify named entity mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

Chunking is a process of extracting phrases from unstructured text. Instead of just simple tokens which may not represent the actual meaning of the text, it's advisable to use phrases such as "South Africa" as a single word instead of 'South 'and 'Africa 'separate words.
Chunking works on top of POS tagging, it uses POS-tags as input and provides chunks as output. Similar to POS tags, there are a standard set of Chunk tags like Noun Phrase (NP), Verb Phrase (VP), etc.
Chunking is very important when you want to extract information from text such as Locations, Person Names etc. In NLP called Named Entity Extraction.

There are a lot of libraries which gives phrases out-of-box such as Spacy or TextBlob. NLTK just provides a mechanism using regular expressions to generate chunks.

Steps to be performed:
To perform Noun Phrase Chunking, search for chunks corresponding to an individual noun phrase.
In order to create NP chunk, define the chunk grammar using POS tags.
Define this using a single regular expression rule ('''

```
NP: {<DT>?<JJ>*<NN>} # NP
''')
```

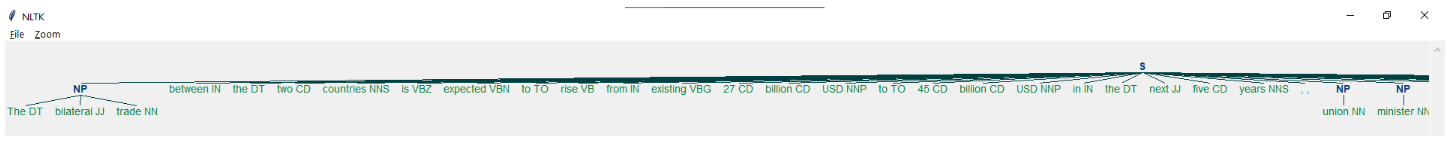
The rule states that whenever the chunk finds an optional determiner (DT) followed by any number of adjectives (JJ) and then a noun (NN) then the Noun Phrase (NP) chunk should be formed. Then perform chunking using the following:

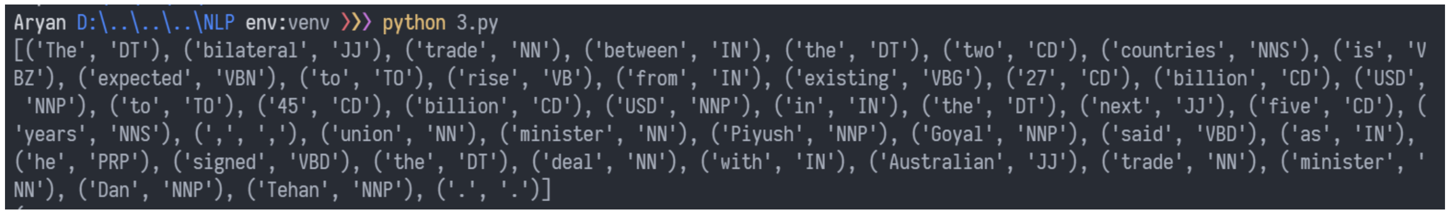
```
chunkParser = nltk.RegexpParser(grammar)
tagged = nltk.pos_tag(nltk.word_tokenize(sentence))
```

Code

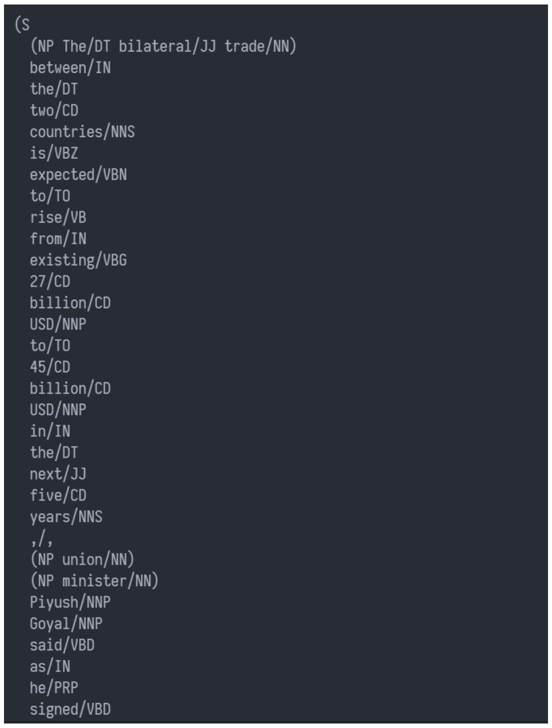
```python
from pprint import pprint

import en_core_web_sm

nlp = en_core_web_sm.load()

import nltk

ex = "The bilateral trade between the two countries is expected to rise from existing 27 billion
USD to 45 billion USD in the next five years, union minister Piyush Goyal said as he signed the
deal with Australian trade minister Dan Tehan."


def preprocess(sent):
    sent = nltk.word_tokenize(sent)
    sent = nltk.pos_tag(sent)
    return sent


sent = preprocess(ex)
print(sent)
pattern = "NP: {<DT>?<JJ>*<NN>}"
cp = nltk.RegexpParser(pattern)
tree = cp.parse(sent)
for subtree in tree.subtrees():
    print(subtree)

doc = nlp(ex)
pprint([(X.text, X.label_) for X in doc.ents])
tree.draw()
```

Output

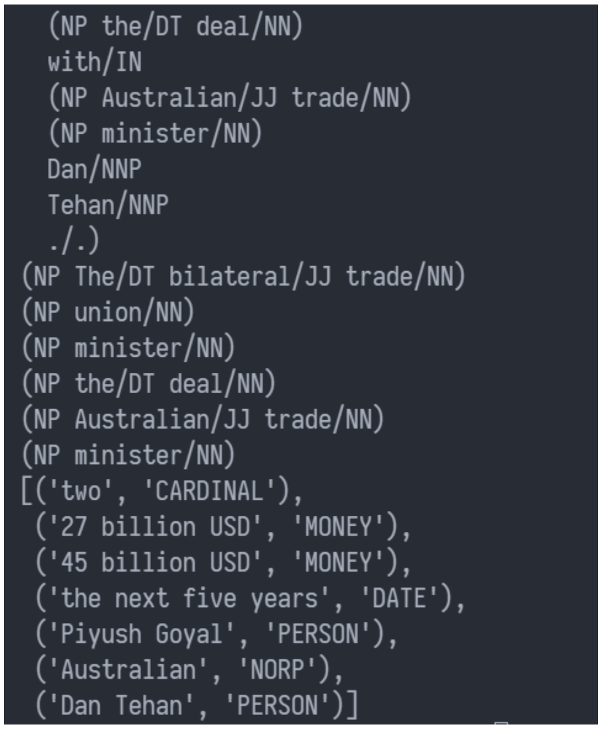




```
Aryan D:\..\..\..\NLP env:venv ❯❯❯ python 3.py
[('The', 'DT'), ('bilateral', 'JJ'), ('trade', 'NN'), ('between', 'IN'), ('the', 'DT'), ('two', 'CD'), ('countries', 'NNS'), ('is', 'V
BZ'), ('expected', 'VBN'), ('to', 'TO'), ('rise', 'VB'), ('from', 'IN'), ('existing', 'VBG'), ('27', 'CD'), ('billion', 'CD'), ('USD',
 'NNP'), ('to', 'TO'), ('45', 'CD'), ('billion', 'CD'), ('USD', 'NNP'), ('in', 'IN'), ('the', 'DT'), ('next', 'JJ'), ('five', 'CD'), (
'years', 'NNS'), (',', ','), ('union', 'NN'), ('minister', 'NN'), ('Piyush', 'NNP'), ('Goyal', 'NNP'), ('said', 'VBD'), ('as', 'IN'),
('he', 'PRP'), ('signed', 'VBD'), ('the', 'DT'), ('deal', 'NN'), ('with', 'IN'), ('Australian', 'JJ'), ('trade', 'NN'), ('minister', '
NN'), ('Dan', 'NNP'), ('Tehan', 'NNP'), ('.', '.')]
```



```
(S
  (NP The/DT bilateral/JJ trade/NN)
  between/IN
  the/DT
  two/CD
  countries/NNS
  is/VBZ
  expected/VBN
  to/TO
  rise/VB
  from/IN
  existing/VBG
  27/CD
  billion/CD
  USD/NNP
  to/TO
  45/CD
  billion/CD
  USD/NNP
  in/IN
  the/DT
  next/JJ
  five/CD
  years/NNS
  ,/,
  (NP union/NN)
  (NP minister/NN)
  Piyush/NNP
  Goyal/NNP
  said/VBD
  as/IN
  he/PRP
  signed/VBD
```



```
  (NP the/DT deal/NN)
  with/IN
  (NP Australian/JJ trade/NN)
  (NP minister/NN)
  Dan/NNP
  Tehan/NNP
  ./.)
(NP The/DT bilateral/JJ trade/NN)
(NP union/NN)
(NP minister/NN)
(NP the/DT deal/NN)
(NP Australian/JJ trade/NN)
(NP minister/NN)
[('two', 'CARDINAL'),
 ('27 billion USD', 'MONEY'),
 ('45 billion USD', 'MONEY'),
 ('the next five years', 'DATE'),
 ('Piyush Goyal', 'PERSON'),
 ('Australian', 'NORP'),
 ('Dan Tehan', 'PERSON')]
```

Conclusion

Thus, we have successfully performed chunking, POS tagging and Named Entity Recognition.