CSE 7320 A12 Chunking

Student Name: Chenming Cui

Student ID: 47781790

Part One: Code

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

# create three sentences
sentence1 = "Cancers can be easier to cure if they are found in earlier stages."
sentence2 = "Jeff shared a video to Facebook named His Mansion House speech."
sentence3 = "China is still leading the fastest developing trends in this world."

# define a function to process all three sentences
def np_rule(sentence):
    tokens = nltk.word_tokenize(sentence)
    print (tokens)

    tags = nltk.pos_tag(tokens)
    print (tags)

    grammar = "NP: {<DT>?<PRP.*>*<JJ.*>*<VBG>*<NN.*>*}"
    cp = nltk.RegexpParser(grammar)

    result = cp.parse(tags)
    print (result)

np_rule(sentence1)
np_rule(sentence2)
np_rule(sentence3)
```

Part Two: Result

[25]:
```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

sentence1 = "Cancers can be easier to cure if they are found in earlier stages."
sentence2 = "Jeff shared a video to Facebook named His Mansion House speech."
sentence3 = "China is still leading the fastest developing trends in this world."

def np_rule(sentence):
    tokens = nltk.word_tokenize(sentence)
    print (tokens)

    tags = nltk.pos_tag(tokens)
    print (tags)

    grammar = "NP: {<DT>?<PRP.*>*<JJ.*>*<VBG>*<NN.*>*}"
    cp = nltk.RegexpParser(grammar)

    result = cp.parse(tags)
    print (result)

np_rule(sentence1)
np_rule(sentence2)
np_rule(sentence3)
```

```
['Cancers', 'can', 'be', 'easier', 'to', 'cure', 'if', 'they', 'are', 'found', 'in', 'earlier', 'stages', '.']
[('Cancers', 'NNS'), ('can', 'MD'), ('be', 'VB'), ('easier', 'JJR'), ('to', 'TO'), ('cure', 'VB'), ('if', 'IN'), ('they', 'PRP'), ('are', 'VBP'), ('found', 'VBN'), ('in', 'IN'), ('earlier', 'JJR'), ('stages', 'NNS'), ('.', '.')]
(S
  (NP Cancers/NNS)
  can/MD
  be/VB
  (NP easier/JJR)
  to/TO
  cure/VB
  if/IN
  (NP they/PRP)
  are/VBP
  found/VBN
  in/IN
  (NP earlier/JJR stages/NNS)
  ./.)
['Jeff', 'shared', 'a', 'video', 'to', 'Facebook', 'named', 'His', 'Mansion', 'House', 'speech', '.']
[('Jeff', 'NNP'), ('shared', 'VBD'), ('a', 'DT'), ('video', 'NN'), ('to', 'TO'), ('Facebook', 'NNP'), ('named', 'VBD'), ('His', 'PRP$'), ('Mansion', 'NNP'), ('House', 'NNP'), ('speech', 'NN'), ('.', '.')]
(S
  (NP Jeff/NNP)
  shared/VBD
  (NP a/DT video/NN)
  to/TO
  (NP Facebook/NNP)
  named/VBD
  (NP His/PRP$ Mansion/NNP House/NNP speech/NN)
  ./.)
['China', 'is', 'still', 'leading', 'the', 'fastest', 'developing', 'trends', 'in', 'this', 'world', '.']
[('China', 'NNP'), ('is', 'VBZ'), ('still', 'RB'), ('leading', 'VBG'), ('the', 'DT'), ('fastest', 'JJS'), ('developing', 'VBG'), ('trends', 'NNS'), ('in', 'IN'), ('this', 'DT'), ('world', 'NN'), ('.', '.')]
(S
  (NP China/NNP)
  is/VBZ
  still/RB
  (NP leading/VBG)
  (NP the/DT fastest/JJS developing/VBG trends/NNS)
  in/IN
  (NP this/DT world/NN)
  ./.)
```

Output Capture