SK_Rajendiran_Week4_Async

October 24, 2020

Name: Sathish Kumar Rajendiran Chapter : POS Tagging and Introduction to Machine Learning
Date: 10/23/2020 Week: 4

[1]:
```
#import libraries

# standard library
import os
import sys
from datetime import datetime
import time

# csv, xls, pandas & json
import pandas as pd
import json
import csv
import xlrd

# Language Processing
import nltk
from nltk import FreqDist

# web requests
from urllib import request


##  Regular Expression to match non-alphabetic characters
import re

os.getcwd()
```

[1]:
```
'/Users/sathishrajendiran/ist664-nlp'
```

1 Question 1: POS Tagging Probabilities: Tag Transitions

[2]:
```
text1 = '''
NNP/ Houston ,/ , NNP/ Monday ,/ , NNP/ July CD/ 21 :/ -- NN/ Men VBP/ have VBD/
 ↪ landed CC/ and VBD/ walked IN/ on DT/ the NN/ moon ./ . CD/ Two NNPS/␣
 ↪Americans ,/ , NNS/ astronauts IN/ of NNP/ Apollo CD/ 11 ,/ , VBD/ steered␣
 ↪PRP$/ their JJ/ fragile JJ/ four-legged NN/ lunar VB/ module RB/ safely CC/␣
 ↪and RB/ smoothly TO/ to DT/ the JJ/ historic NN/ landing NN/ yesterday IN/␣
 ↪at NN/ 4:17:40 NNP/ P.M. ,/ , NNP/ Eastern NN/ daylight NN/ time ./ . NNP/␣
 ↪Neil NNP/ A. NNP/ Armstrong ,/ , DT/ the JJ/ 38-year-old JJ/ civilian NN/␣
 ↪commander ,/ , VBD/ radioed TO/ to NN/ earth CC/ and DT/ the NN/ mission NN/␣
 ↪control NN/ room RB/ here :/ : ``/ " NNP/ Houston ,/ , NNP/ Tranquility NNP/␣
 ↪Base RB/ here :/ ; DT/ the NNP/ Eagle VBZ/ has VBN/ landed . . ''/ "
DT/ The JJ /first NNS/ men TO/ to VB/ reach DT/ the NN/ moon :/ -- NNP/ Mr. NNP/
 ↪ Armstrong CC/ and PRP$/ his NNS/ co-pilot ,/ , NNP/ Col. NNP/ Edwin NNP/ E.␣
 ↪NNP/ Aldrin ,/ , NNP/ Jr. IN/ of DT/ the NNP/ Air NNP/ Force :/ -- VBD/␣
 ↪brought PRP$/ their NN/ ship TO/ to VB/ rest IN/ on DT/ a NN/ level ,/ , NN/␣
 ↪rock-strewn NN/ plain IN/ near DT/ the JJ/ southwestern NN/ shore IN/ of DT/␣
 ↪the NN/ arid NNP/ Sea IN/ of NN/ Tranquility ./ . IN/ About CD/ six CC/ and␣
 ↪DT/ a JJ/ half NNS/ hours RB/ later ,/ , NNP/ Mr. NNP/ Armstrong VBD/ opened␣
 ↪DT/ the NN/ landing NN/ craft POS/ 's NN/ hatch ,/ , VBD/ stepped RB/ slowly␣
 ↪IN/ down DT/ the NN/ ladder CC/ and VBD/ declared IN/ as PRP/ he VBD/␣
 ↪planted DT /the JJ/ first NN/ human NN/ footprint IN/ on DT/ the NN/ lunar␣
 ↪NN/ crust :/ : ``/ " DT/ That VBZ/ 's CD/ one JJ/ small NN/ step IN/ for NN/␣
 ↪man ,/ , CD/ one JJ/ giant NN/ leap IN/ for NN/ mankind ./ . ''/ "
'''
```

1.0.1 1: P(VB|TO) and P(NN|TO)

[3]:
```
# Count (TO)
if re.search('[TO]',text1):
    re_count_TO = re.findall(r'\b(?:TO\b)',text1)
    print('Found a match:\n',re_count_TO,'\nNumber of matches:
 ↪\n',len(re_count_TO) )
else:
    print('No match.')
```

```
Found a match:
 ['TO', 'TO', 'TO', 'TO']
Number of matches:
 4
```

[4]:
```
# Count (TO VB)
if re.search('[TO]',text1):
    re_count_TO_VB = re.findall(r'(TO/(?: \w+) +(?:[VB]\w\b))',text1)
    print('Found a match:\n',re_count_TO_VB,'\nNumber of matches:
 ↪\n',len(re_count_TO_VB) )
```

```
else:
    print('No match.')
```

```
Found a match:
 ['TO/ to VB', 'TO/ to VB']
Number of matches:
 2
```

[5]:
```
# Count (TO NN)
if re.search('[TO]',text1):
    re_count_TO_NN = re.findall(r'(TO/(?: \w+) +(?:[NN]\w\b))',text1)
    print('Found a match:\n',re_count_TO_NN,'\nNumber of matches:
 ↪\n',len(re_count_TO_NN) )
else:
    print('No match.')
```

```
Found a match:
 ['TO/ to NN']
Number of matches:
 1
```

[6]:
```
# 1: P(VB|TO) and P(NN|TO)

# P(VB|TO)
count_TO = len(re_count_TO) # count the number of occurrences of the tag TO
count_TO_VB = len(re_count_TO_VB) # count the number of occurrences of the tag␣
 ↪TO followed by Tag VBD

Prob_TO_VB = round(count_TO_VB/count_TO,2) # Transition Probability of P(VB|TO)
print('Count (TO) is :',count_TO )
print('Count (TO VB) :',count_TO_VB )
print('Transition Probability - P(VB|TO) is :',Prob_TO_VB )
print('\n-------------------------------------------------\n')

#P(NN|TO)
count_TO = len(re_count_TO) # count the number of occurrences of the tag TO
count_TO_NN = len(re_count_TO_NN) # count the number of occurrences of the tag␣
 ↪TO followed by Tag NN

Prob_TO_NN = round(count_TO_NN/count_TO,2) # Transition Probability of P(DT|VBD)
print('Count (TO) is :',count_TO )
print('Count (TO NN) :',count_TO_NN )
print('Transition Probability - P(NN|TO) is :',Prob_TO_NN )
print('\n-------------------------------------------------\n')
```

```
Count (TO) is : 4
Count (TO VB) : 2
```

```
Transition Probability - P(VB|TO) is : 0.5

------------------------------------------------

Count (TO) is : 4
Count (TO NN) : 1
Transition Probability - P(NN|TO) is : 0.25

------------------------------------------------
```

1.0.2 2: P(IN|VBD) and P(DT|VBD)

[7]:
```
# Count (VBD)
if re.search('[VBD]',text1):
    re_count_VBD = re.findall(r'\b(?:VBD\b)',text1)
    print('Found a match:\n',re_count_VBD,'\nNumber of matches:
 ↪\n',len(re_count_VBD) )
else:
    print('No match.')
```

```
Found a match:
 ['VBD', 'VBD', 'VBD', 'VBD', 'VBD', 'VBD', 'VBD', 'VBD', 'VBD']
Number of matches:
 9
```

[8]:
```
# Count (VBD IN)
if re.search('[VBD]',text1):
    re_count_VBD_IN = re.findall(r'(VBD/(?: \w+) +(?:[IN]\w\b))',text1)
    print('Found a match:\n',re_count_VBD_IN,'\nNumber of matches:
 ↪\n',len(re_count_VBD_IN) )
else:
    print('No match.')
```

```
Found a match:
 ['VBD/ walked IN', 'VBD/ declared IN']
Number of matches:
 2
```

[9]:
```
# Count (VBD DT)
if re.search('[VBD]',text1):
    re_count_VBD_DT = re.findall(r'(VBD/(?: \w+) +(?:[D]\w\b))',text1)
    print('Found a match:\n',re_count_VBD_DT,'\nNumber of matches:
 ↪\n',len(re_count_VBD_DT) )
else:
    print('No match.')
```

```
Found a match:
```

```
 ['VBD/ opened DT', 'VBD/ planted DT']
Number of matches:
 2
```

[10]:
```
# 2: P(IN|VBD) and P(DT|VBD)

# P(IN|VBD)
count_VBD = len(re_count_VBD) # count the number of occurrences of the tag VBD
count_IN_VBD = len(re_count_VBD_IN) # count the number of occurrences of the␣
 ↪tag VBD followed by Tag IN

Prob_IN_VBD = round(count_IN_VBD/count_VBD,2) # Transition Probability of␣
 ↪P(IN|VBD)
print('Count (VBD) is :',count_VBD )
print('Count (IN VBD) :',count_IN_VBD )
print('Transition Probability - P(IN|VBD) is :',Prob_IN_VBD )
print('\n------------------------------------------------\n')

#P(DT|VBD)
count_VBD = len(re_count_VBD) # count the number of occurrences of the tag VBD
count_DT_VBD = len(re_count_VBD_DT) # count the number of occurrences of the␣
 ↪tag VBD followed by Tag DT

Prob_DT_VBD = round(count_DT_VBD/count_VBD,2) # Transition Probability of␣
 ↪P(DT|VBD)
print('Count (VBD) is :',count_VBD )
print('Count (DT VBD) :',count_DT_VBD )
print('Transition Probability - P(DT|VBD) is :',Prob_DT_VBD )
print('\n------------------------------------------------\n')
```

```
Count (VBD) is : 9
Count (IN VBD) : 2
Transition Probability - P(IN|VBD) is : 0.22

------------------------------------------------

Count (VBD) is : 9
Count (DT VBD) : 2
Transition Probability - P(DT|VBD) is : 0.22

------------------------------------------------
```

1.0.3 3: P(NN|JJ) and P(JJ|JJ)

[11]:
```
# Count (JJ)
if re.search('[JJ]',text1):
    re_count_JJ = re.findall(r'\b(?:JJ\b)',text1)
    print('Found a match:\n',re_count_JJ,'\nNumber of matches:
 ↪\n',len(re_count_JJ) )
else:
    print('No match.')
```

```
Found a match:
 ['JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ']
Number of matches:
 11
```

[12]:
```
# Count (JJ NN)
if re.search('[JJ]',text1):
    re_count_JJ_NN = re.findall(r'(JJ/(?: \w+) +(?:[NN]\w\b))',text1)
    print('Found a match:\n',re_count_JJ_NN,'\nNumber of matches:
 ↪\n',len(re_count_JJ_NN) )
else:
    print('No match.')
```

```
Found a match:
 ['JJ/ historic NN', 'JJ/ civilian NN', 'JJ/ southwestern NN', 'JJ/ first NN',
'JJ/ small NN', 'JJ/ giant NN']
Number of matches:
 6
```

[13]:
```
# Count (JJ JJ)
if re.search('[JJ]',text1):
    re_count_JJ_JJ = re.findall(r'(JJ/(?: \w+) +(?:[JJ]\w\b))',text1)
    print('Found a match:\n',re_count_JJ_JJ,'\nNumber of matches:
 ↪\n',len(re_count_JJ_JJ) )
else:
    print('No match.')
```

```
Found a match:
 ['JJ/ fragile JJ']
Number of matches:
 1
```

[14]:
```
# 3: P(NN|JJ) and P(JJ|JJ)

# P(NN|JJ)
count_JJ = len(re_count_JJ) # count the number of occurrences of the tag JJ
count_JJ_NN = len(re_count_JJ_NN) # count the number of occurrences of the tag␣
 ↪JJ followed by Tag NN
```

```
Prob_NN_JJ = round(count_JJ_NN/count_JJ,2) # Transition Probability of P(NN|JJ)
print('Count (JJ) is :',count_JJ )
print('Count (NN JJ) :',count_JJ_NN )
print('Transition Probability - P(NN|JJ) is :',Prob_NN_JJ )
print('\n-------------------------------------------------\n')

#P(JJ|JJ)
count_JJ = len(re_count_JJ) # count the number of occurrences of the tag JJ
count_JJ_JJ = len(re_count_JJ_JJ) # count the number of occurrences of the tag␣
 ↪JJ followed by Tag JJ

Prob_JJ_JJ = round(count_JJ_JJ/count_JJ,2) # Transition Probability of P(JJ|JJ)
print('Count (JJ) is :',count_JJ )
print('Count (JJ JJ) :',count_JJ_JJ )
print('Transition Probability - P(JJ|JJ) is :',Prob_JJ_JJ )
print('\n-------------------------------------------------\n')
```

```
Count (JJ) is : 11
Count (NN JJ) : 6
Transition Probability - P(NN|JJ) is : 0.55

-------------------------------------------------

Count (JJ) is : 11
Count (JJ JJ) : 1
Transition Probability - P(JJ|JJ) is : 0.09

-------------------------------------------------
```

1.0.4 4:P(NN|DT)

[15]:
```
# Count (DT)
if re.search('[DT]',text1):
    re_count_DT = re.findall(r'\b(?:DT\b)',text1)
    print('Found a match:\n',re_count_DT,'\nNumber of matches:
 ↪\n',len(re_count_DT) )
else:
    print('No match.')
```

```
Found a match:
 ['DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT', 'DT',
'DT', 'DT', 'DT', 'DT']
Number of matches:
 17
```

[16]:
```
# Count (DT NN)
if re.search('[DT]',text1):
    re_count_DT_NN = re.findall(r'(DT/(?: \w+) +(?:[NN]\w\b))',text1)
    print('Found a match:\n',re_count_DT_NN,'\nNumber of matches:
 ↪\n',len(re_count_DT_NN) )
else:
    print('No match.')
```

```
Found a match:
 ['DT/ the NN', 'DT/ the NN', 'DT/ the NN', 'DT/ a NN', 'DT/ the NN', 'DT/ the
NN', 'DT/ the NN', 'DT/ the NN']
Number of matches:
 8
```

[17]:
```
#4: P(NN|DT)

# P(NN|DT)
count_DT = len(re_count_DT) # count the number of occurrences of the tag DT
count_DT_NN = len(re_count_DT_NN) # count the number of occurrences of the tag␣
 ↪DT followed by Tag NN

Prob_NN_DT = round(count_DT_NN/count_DT,2) # Transition Probability of P(NN|DT)
print('Count (DT) is :',count_DT )
print('Count (NN DT) :',count_DT_NN )
print('Transition Probability - P(NN|DT)  is :',Prob_NN_DT )
print('\n-------------------------------------------------\n')
```

```
Count (DT) is : 17
Count (NN DT) : 8
Transition Probability - P(NN|DT)  is : 0.47

-------------------------------------------------
```

1.0.5 5: P(NN|IN)

[18]:
```
# Count (IN)
if re.search('[IN]',text1):
    re_count_IN = re.findall(r'\b(?:IN\b)',text1)
    print('Found a match:\n',re_count_IN,'\nNumber of matches:
 ↪\n',len(re_count_IN) )
else:
    print('No match.')
```

```
Found a match:
 ['IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN', 'IN',
'IN']
```

```
Number of matches:
 14
```

[19]:
```
# Count (IN NN)
if re.search('[IN]',text1):
    re_count_IN_NN = re.findall(r'(IN/(?: \w+) +(?:[NN]\w\b))',text1)
    print('Found a match:\n',re_count_IN_NN,'\nNumber of matches:
 ↪\n',len(re_count_IN_NN) )
else:
    print('No match.')
```

```
Found a match:
 ['IN/ at NN', 'IN/ of NN', 'IN/ for NN', 'IN/ for NN']
Number of matches:
 4
```

[20]:
```
#5: P(NN|IN)

count_IN = len(re_count_IN) # count the number of occurrences of the tag IN
count_IN_NN = len(re_count_IN_NN) # count the number of occurrences of the tag␣
 ↪IN followed by Tag NN

Prob_NN_IN = round(count_IN_NN/count_IN,2) # Transition Probability of P(NN|DT)
print('Count (IN) is :',count_IN )
print('Count (IN NN) :',count_IN_NN )
print('Transition Probability - P(NN|IN)  is :',Prob_NN_IN )
print('\n-------------------------------------------------\n')
```

```
Count (IN) is : 14
Count (IN NN) : 4
Transition Probability - P(NN|IN)  is : 0.29

-------------------------------------------------
```

2 Question 2: POS Tagging Probabilities: Word Likelihoods

[21]:
```
text2 ='''
```

```
NNP/ Houston ,/ , NNP/ Monday ,/ , NNP/ July CD/ 21 :/ -- NN/ Men VBP/ have VBD/
 ↪ landed CC/ and VBD/ walked IN/ on DT/ the NN/ moon ./ . CD/ Two NNPS/␣
 ↪Americans ,/ , NNS/ astronauts IN/ of NNP/ Apollo CD/ 11 ,/ , VBD/ steered␣
 ↪PRP$/ their JJ/ fragile JJ/ four-legged NN/ lunar VB/ module RB/ safely CC/␣
 ↪and RB/ smoothly TO/ to DT/ the JJ/ historic NN/ landing NN/ yesterday IN/␣
 ↪at NN/ 4:17:40 NNP/ P.M. ,/ , NNP/ Eastern NN/ daylight NN/ time ./ . NNP/␣
 ↪Neil NNP/ A. NNP/ Armstrong ,/ , DT/ the JJ/ 38-year-old JJ/ civilian NN/␣
 ↪commander ,/ , VBD/ radioed TO/ to NN/ earth CC/ and DT/ the NN/ mission NN/␣
 ↪control NN/ room RB/ here :/ : ``/ " NNP/ Houston ,/ , NNP/ Tranquility NNP/␣
 ↪Base RB/ here :/ ; DT/ the NNP/ Eagle VBZ/ has VBN/ landed . . ''/ "
DT/ The JJ /first NNS/ men TO/ to VB/ reach DT/ the NN/ moon :/ -- NNP/ Mr. NNP/
 ↪ Armstrong CC/ and PRP$/ his NNS/ co-pilot ,/ , NNP/ Col. NNP/ Edwin NNP/ E.␣
 ↪NNP/ Aldrin ,/ , NNP/ Jr. IN/ of DT/ the NNP/ Air NNP/ Force :/ -- VBD/␣
 ↪brought PRP$/ their NN/ ship TO/ to VB/ rest IN/ on DT/ a NN/ level ,/ , NN/␣
 ↪rock-strewn NN/ plain IN/ near DT/ the JJ/ southwestern NN/ shore IN/ of DT/␣
 ↪the NN/ arid NNP/ Sea IN/ of NN/ Tranquility ./ . IN/ About CD/ six CC/ and␣
 ↪DT/ a JJ/ half NNS/ hours RB/ later ,/ , NNP/ Mr. NNP/ Armstrong VBD/ opened␣
 ↪DT/ the NN/ landing NN/ craft POS/ 's NN/ hatch ,/ , VBD/ stepped RB/ slowly␣
 ↪IN/ down DT/ the NN/ ladder CC/ and VBD/ declared IN/ as PRP/ he VBD/␣
 ↪planted DT /the JJ/ first NN/ human NN/ footprint IN/ on DT/ the NN/ lunar␣
 ↪NN/ crust :/ : ``/ " DT/ That VBZ/ 's CD/ one JJ/ small NN/ step IN/ for NN/␣
 ↪man ,/ , CD/ one JJ/ giant NN/ leap IN/ for NN/ mankind ./ . ''/ "

'''
```

[22]:
```
text2
```

[22]:
```
'\n\nNNP/ Houston ,/ , NNP/ Monday ,/ , NNP/ July CD/ 21 :/ -- NN/ Men VBP/ have
VBD/ landed CC/ and VBD/ walked IN/ on DT/ the NN/ moon ./ . CD/ Two NNPS/
Americans ,/ , NNS/ astronauts IN/ of NNP/ Apollo CD/ 11 ,/ , VBD/ steered PRP$/
their JJ/ fragile JJ/ four-legged NN/ lunar VB/ module RB/ safely CC/ and RB/
smoothly TO/ to DT/ the JJ/ historic NN/ landing NN/ yesterday IN/ at NN/
4:17:40 NNP/ P.M. ,/ , NNP/ Eastern NN/ daylight NN/ time ./ . NNP/ Neil NNP/ A.
NNP/ Armstrong ,/ , DT/ the JJ/ 38-year-old JJ/ civilian NN/ commander ,/ , VBD/
radioed TO/ to NN/ earth CC/ and DT/ the NN/ mission NN/ control NN/ room RB/
here :/ : ``/ " NNP/ Houston ,/ , NNP/ Tranquility NNP/ Base RB/ here :/ ; DT/
the NNP/ Eagle VBZ/ has VBN/ landed . . \'\'/ "\nDT/ The JJ /first NNS/ men TO/
to VB/ reach DT/ the NN/ moon :/ -- NNP/ Mr. NNP/ Armstrong CC/ and PRP$/ his
NNS/ co-pilot ,/ , NNP/ Col. NNP/ Edwin NNP/ E. NNP/ Aldrin ,/ , NNP/ Jr. IN/ of
DT/ the NNP/ Air NNP/ Force :/ -- VBD/ brought PRP$/ their NN/ ship TO/ to VB/
rest IN/ on DT/ a NN/ level ,/ , NN/ rock-strewn NN/ plain IN/ near DT/ the JJ/
southwestern NN/ shore IN/ of DT/ the NN/ arid NNP/ Sea IN/ of NN/ Tranquility
./ . IN/ About CD/ six CC/ and DT/ a JJ/ half NNS/ hours RB/ later ,/ , NNP/ Mr.
NNP/ Armstrong VBD/ opened DT/ the NN/ landing NN/ craft POS/ \'s NN/ hatch ,/ ,
VBD/ stepped RB/ slowly IN/ down DT/ the NN/ ladder CC/ and VBD/ declared IN/ as
PRP/ he VBD/ planted DT /the JJ/ first NN/ human NN/ footprint IN/ on DT/ the
NN/ lunar NN/ crust :/ : ``/ " DT/ That VBZ/ \'s CD/ one JJ/ small NN/ step IN/
```

```
for NN/ man ,/ , CD/ one JJ/ giant NN/ leap IN/ for NN/ mankind ./ . \'\'/ "
\n\n'
```

2.0.1 1: P(is|VBZ) and P(has|VBZ)

[23]:
```
#P(is|VBZ) - Begins


# Count (VBZ)
if re.search('[VBZ]',text2):
    re_count_VBZ = re.findall(r'\b(?:VBZ\b)',text2)
    print('Found a match:\n',re_count_VBZ,'\nNumber of matches:
 ↪\n',len(re_count_VBZ) )
else:
    print('No match.')


# Count (VBZ is)
if re.search('[VBZ]',text2):
    re_count_VBZ_is = re.findall(r'(VBZ/(?: )+(?:[is]\w+))',text2)
    print('Found a match:\n',re_count_VBZ_is,'\nNumber of matches:
 ↪\n',len(re_count_VBZ_is) )
else:
    print('No match.')

#PP(is|VBZ)

count_VBZ = len(re_count_VBZ) # count the number of occurrences of the tag VBZ
count_VBZ_is = len(re_count_VBZ_is) # count the number of occurrences of the␣
 ↪tag VBZ followed by Tag is

if count_VBZ_is > 0:
    Prob_is_VBZ = round(count_VBZ_is/count_VBZ,2) # Transition Probability of␣
 ↪P(is|VBZ)
    print('Count (VBZ) is :',count_VBZ )
    print('Count (is VBZ) :',count_VBZ_is )
    print('Transition Probability - P(is|VBZ)  is :',Prob_is_VBZ )
else:
    print('Transition Probability - P(is|VBZ)  is : 0' )
print('\n-------------------------------------------------\n')
```

```
Found a match:
 ['VBZ', 'VBZ']
Number of matches:
 2
Found a match:
 []
```

```
Number of matches:
 0
Transition Probability - P(is|VBZ)  is : 0

------------------------------------------------
```

[24]:
```
#P(has|VBZ) - Begins

# Count (VBZ)
if re.search('[VBZ]',text2):
    re_count_VBZ = re.findall(r'\b(?:VBZ\b)',text2)
    print('Found a match:\n',re_count_VBZ,'\nNumber of matches:
 ↪\n',len(re_count_VBZ) )
else:
    print('No match.')


# Count (VBZ has)
if re.search('[VBZ]',text2):
    re_count_VBZ_has = re.findall(r'(VBZ/(?: )+(?:[has]\w+))',text2)
    print('Found a match:\n',re_count_VBZ_has,'\nNumber of matches:
 ↪\n',len(re_count_VBZ_has) )
else:
    print('No match.')

#PP(has|VBZ)

count_VBZ = len(re_count_VBZ) # count the number of occurrences of the tag VBZ
count_VBZ_has = len(re_count_VBZ_has) # count the number of occurrences of the␣
 ↪tag VBZ followed by Tag has

if count_VBZ_has > 0:
    Prob_has_VBZ = round(count_VBZ_has/count_VBZ,2) # Transition Probability of␣
 ↪P(has|VBZ)
    print('Count (VBZ) is :',count_VBZ )
    print('Count (has VBZ) :',count_VBZ_has )
    print('Transition Probability - P(has|VBZ)  is :',Prob_has_VBZ )
else:
    print('Transition Probability - P(has|VBZ)  is : 0' )
print('\n------------------------------------------------\n')
```

```
Found a match:
 ['VBZ', 'VBZ']
Number of matches:
 2
Found a match:
```

```
 ['VBZ/ has']
Number of matches:
 1
Count (VBZ) is : 2
Count (has VBZ) : 1
Transition Probability - P(has|VBZ)  is : 0.5

------------------------------------------------
```

2.1 2: P(moon|NN) and P(earth|NN)

[33]:
```
#P(moon|NN) - Begins

# Count (NN)
if re.search('[NN]',text2):
    re_count_NN = re.findall(r'\b(?:NN\b)',text2)
    print('Found a match:\n',re_count_NN,'\nNumber of matches:
 ↪',len(re_count_NN) )
else:
    print('No match.')


# Count (NN moon)
if re.search('[NN]',text2):
    re_count_NN_moon = re.findall(r'(NN/ moon\b)',text2)
    print('Found a match:\n',re_count_NN_moon,'\nNumber of matches:
 ↪',len(re_count_NN_moon) )
else:
    print('No match.')

#PP(moon|NN)

count_NN = len(re_count_NN) # count the number of occurrences of the tag NN
count_NN_moon = len(re_count_NN_moon) # count the number of occurrences of the␣
 ↪tag NN followed by Tag has

if count_NN_moon > 0:
    Prob_moon_NN = round(count_NN_moon/count_NN,2) # Transition Probability of␣
 ↪P(moon|NN)
    print('Count (NN) is :',count_NN )
    print('Count (has NN) :',count_NN_moon )
    print('Transition Probability - P(moon|NN)  is :',Prob_moon_NN )
else:
    print('Transition Probability - P(moon|NN)  is : 0' )
print('\n------------------------------------------------\n')
```

```
Found a match:
 ['NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN',
'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN',
'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN']
Number of matches: 33
Found a match:
 ['NN/ moon', 'NN/ moon']
Number of matches: 2
Count (NN) is : 33
Count (has NN) : 2
Transition Probability - P(moon|NN)  is : 0.06

------------------------------------------------
```

[26]:
```
#P(earth|NN) - Begins

# Count (NN)
if re.search('[NN]',text2):
    re_count_NN = re.findall(r'\b(?:NN\b)',text2)
    print('Found a match:\n',re_count_NN,'\nNumber of matches:
 ↪\n',len(re_count_NN) )
else:
    print('No match.')


# Count (NN earth)
if re.search('[NN]',text2):
    re_count_NN_earth = re.findall(r'(NN/ earth\b)',text2)
    print('Found a match:\n',re_count_NN_earth,'\nNumber of matches:
 ↪\n',len(re_count_NN_earth) )
else:
    print('No match.')

#PP(earth|NN)

count_NN = len(re_count_NN) # count the number of occurrences of the tag NN
count_NN_earth = len(re_count_NN_earth) # count the number of occurrences of␣
 ↪the tag NN followed by Tag earth

if count_NN_earth > 0:
    Prob_earth_NN = round(count_NN_earth/count_NN,2) # Transition Probability␣
 ↪of P(earth|NN)
    print('Count (NN) is :',count_NN )
    print('Count (earth NN) :',count_NN_earth )
    print('Transition Probability - P(earth|NN)  is :',Prob_earth_NN )
else:
```

```
    print('Transition Probability - P(earth|NN)  is : 0' )
print('\n-----------------------------------------------\n')
```

```
Found a match:
 ['NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN',
'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN',
'NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'NN']
Number of matches:
 33
Found a match:
 ['NN/ earth']
Number of matches:
 1
Count (NN) is : 33
Count (earth NN) : 1
Transition Probability - P(earth|NN)  is : 0.03

-----------------------------------------------
```

2.2 3: P(small|JJ) and P(fragile|JJ)

[27]:
```
#P(small|JJ) - Begins

# Count (JJ)
if re.search('[JJ]',text2):
    re_count_JJ = re.findall(r'\b(?:JJ\b)',text2)
    print('Found a match:\n',re_count_JJ,'\nNumber of matches:
 ↪\n',len(re_count_JJ) )
else:
    print('No match.')


# Count (JJ small)
if re.search('[JJ]',text2):
    re_count_JJ_small = re.findall(r'(JJ/ small\b)',text2)
    print('Found a match:\n',re_count_JJ_small,'\nNumber of matches:
 ↪\n',len(re_count_JJ_small) )
else:
    print('No match.')

#PP(small|JJ)

count_JJ = len(re_count_JJ) # count the number of occurrences of the tag JJ
count_JJ_small = len(re_count_JJ_small) # count the number of occurrences of␣
 ↪the tag JJ followed by Tag small
```

```
if count_JJ_small > 0:
    Prob_small_JJ = round(count_JJ_small/count_JJ,2) # Transition Probability␣
 ↪of P(small|JJ)
    print('Count (JJ) is :',count_JJ )
    print('Count (small JJ) :',count_JJ_small )
    print('Transition Probability - P(small|JJ)  is :',Prob_small_JJ )
else:
    print('Transition Probability - P(small|JJ)  is : 0' )
print('\n-------------------------------------------------\n')
```

```
Found a match:
 ['JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ']
Number of matches:
 11
Found a match:
 ['JJ/ small']
Number of matches:
 1
Count (JJ) is : 11
Count (small JJ) : 1
Transition Probability - P(small|JJ)  is : 0.09

-------------------------------------------------
```

[28]:
```
#P(fragile|JJ) - Begins

# Count (JJ)
if re.search('[JJ]',text2):
    re_count_JJ = re.findall(r'\b(?:JJ\b)',text2)
    print('Found a match:\n',re_count_JJ,'\nNumber of matches:
 ↪\n',len(re_count_JJ) )
else:
    print('No match.')


# Count (JJ fragile)
if re.search('[JJ]',text2):
    re_count_JJ_fragile = re.findall(r'(JJ/ fragile\b)',text2)
#     re_count_JJ_fragile = re.findall(r'(JJ/(?: )+(?:[fragile]\w+))',text2)
    print('Found a match:\n',re_count_JJ_fragile,'\nNumber of matches:
 ↪\n',len(re_count_JJ_fragile) )
else:
    print('No match.')

#PP(fragile|JJ)
```

```
count_JJ = len(re_count_JJ) # count the number of occurrences of the tag JJ
count_JJ_fragile = len(re_count_JJ_fragile) # count the number of occurrences␣
↪of the tag JJ followed by Tag fragile

if count_JJ_fragile > 0:
    Prob_fragile_JJ = round(count_JJ_fragile/count_JJ,2) # Transition␣
↪Probability of P(fragile|JJ)
    print('Count (JJ) is :',count_JJ )
    print('Count (fragile JJ) :',count_JJ_fragile )
    print('Transition Probability - P(fragile|JJ)  is :',Prob_fragile_JJ )
else:
    print('Transition Probability - P(fragile|JJ)  is : 0' )
print('\n----------------------------------------------\n')
```

```
Found a match:
 ['JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ', 'JJ']
Number of matches:
 11
Found a match:
 ['JJ/ fragile']
Number of matches:
 1
Count (JJ) is : 11
Count (fragile JJ) : 1
Transition Probability - P(fragile|JJ)  is : 0.09

----------------------------------------------
```

2.3 4: P(one|CD) and P(six|CD)

[29]:
```
#P(one|CD) - Begins

# Count (CD)
if re.search('[CD]',text2):
    re_count_CD = re.findall(r'\b(?:CD\b)',text2)
    print('Found a match:\n',re_count_CD,'\nNumber of matches:
↪\n',len(re_count_CD) )
else:
    print('No match.')


# Count (CD one)
if re.search('[CD]',text2):
    re_count_CD_one = re.findall(r'(CD/ one\b)',text2)
#     re_count_CD_one= re.findall(r'(CD/(?: )+(?:[one]\w+))',text2)
```

```
    print('Found a match:\n',re_count_CD_one,'\nNumber of matches:
 ↪\n',len(re_count_CD_one) )
else:
    print('No match.')

#PP(one|CD)

count_CD = len(re_count_CD) # count the number of occurrences of the tag CD
count_CD_one = len(re_count_CD_one) # count the number of occurrences of the␣
 ↪tag CD followed by Tag one

if count_CD_one > 0:
    Prob_one_CD = round(count_CD_one/count_CD,2) # Transition Probability of␣
 ↪P(one|CD)
    print('Count (CD) is :',count_CD )
    print('Count (one CD) :',count_CD_one )
    print('Transition Probability - P(one|CD)  is :',Prob_one_CD )
else:
    print('Transition Probability - P(one|CD)  is : 0' )
print('\n----------------------------------------------\n')
```

```
Found a match:
 ['CD', 'CD', 'CD', 'CD', 'CD', 'CD']
Number of matches:
 6
Found a match:
 ['CD/ one', 'CD/ one']
Number of matches:
 2
Count (CD) is : 6
Count (one CD) : 2
Transition Probability - P(one|CD)  is : 0.33

----------------------------------------------
```

[30]:
```
#P(six|CD) - Begins

# Count (CD)
if re.search('[CD]',text2):
    re_count_CD = re.findall(r'\b(?:CD\b)',text2)
    print('Found a match:\n',re_count_CD,'\nNumber of matches:
 ↪\n',len(re_count_CD) )
else:
    print('No match.')
```

```
# Count (CD six)
if re.search('[CD]',text2):
    re_count_CD_six = re.findall(r'(CD/ six\b)',text2)
#     re_count_CD_six= re.findall(r'(CD/(?: )+(?:[six]\w+))',text2)
    print('Found a match:\n',re_count_CD_six,'\nNumber of matches:
 ↪\n',len(re_count_CD_six) )
else:
    print('No match.')

#PP(six|CD)

count_CD = len(re_count_CD) # count the number of occurrences of the tag CD
count_CD_six = len(re_count_CD_six) # count the number of occurrences of the␣
 ↪tag CD followed by Tag six

if count_CD_six > 0:
    Prob_six_CD = round(count_CD_six/count_CD,2) # Transition Probability of␣
 ↪P(six|CD)
    print('Count (CD) is :',count_CD )
    print('Count (six CD) :',count_CD_six )
    print('Transition Probability - P(six|CD)  is :',Prob_six_CD )
else:
    print('Transition Probability - P(six|CD)  is : 0' )
print('\n----------------------------------------------\n')
```

```
Found a match:
 ['CD', 'CD', 'CD', 'CD', 'CD', 'CD']
Number of matches:
 6
Found a match:
 ['CD/ six']
Number of matches:
 1
Count (CD) is : 6
Count (six CD) : 1
Transition Probability - P(six|CD)  is : 0.17

----------------------------------------------
```

2.4 5: P(Mr.|NNP) and P(Eagle|NNP)

[31]:
```
#P(Mr.|NNP) - Begins

# Count (NNP)
if re.search('[NNP]',text2):
    re_count_NNP = re.findall(r'\b(?:NNP\b)',text2)
```

```
    print('Found a match:\n',re_count_NNP,'\nNumber of matches:
 ↪\n',len(re_count_NNP) )
else:
    print('No match.')


# Count (NNP Mr)
if re.search('[NNP]',text2):
    re_count_NNP_Mr = re.findall(r'(NNP/ Mr\b)',text2)
#     re_count_NNP_Mr= re.findall(r'(NNP/(?: )+(?:[Mr]\w+))',text2)
    print('Found a match:\n',re_count_NNP_Mr,'\nNumber of matches:
 ↪\n',len(re_count_NNP_Mr) )
else:
    print('No match.')

#PP(Mr|NNP)

count_NNP = len(re_count_NNP) # count the number of occurrences of the tag NNP
count_NNP_Mr = len(re_count_NNP_Mr) # count the number of occurrences of the␣
 ↪tag NNP followed by Tag Mr

if count_NNP_Mr > 0:
    Prob_Mr_NNP = round(count_NNP_Mr/count_NNP,2) # Transition Probability of␣
 ↪P(Mr|NNP)
    print('Count (NNP) is :',count_NNP )
    print('Count (Mr NNP) :',count_NNP_Mr )
    print('Transition Probability - P(Mr|NNP)  is :',Prob_Mr_NNP )
else:
    print('Transition Probability - P(Mr|NNP)  is : 0' )
print('\n-----------------------------------------------\n')
```

```
Found a match:
 ['NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP',
'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP',
'NNP', 'NNP', 'NNP']
Number of matches:
 25
Found a match:
 ['NNP/ Mr', 'NNP/ Mr']
Number of matches:
 2
Count (NNP) is : 25
Count (Mr NNP) : 2
Transition Probability - P(Mr|NNP)  is : 0.08

-----------------------------------------------
```

[32]:
```
#P(Eagle|NNP) - Begins

# Count (NNP)
if re.search('[NNP]',text2):
    re_count_NNP = re.findall(r'\b(?:NNP\b)',text2)
    print('Found a match:\n',re_count_NNP,'\nNumber of matches:
 ↪\n',len(re_count_NNP) )
else:
    print('No match.')


# Count (NNP Eagle)
if re.search('[NNP]',text2):
    re_count_NNP_Eagle = re.findall(r'(NNP/ Eagle\b)',text2)
#     re_count_NNP_Eagle= re.findall(r'(NNP/(?: )+(?:[Eagle]\w+))',text2)
    print('Found a match:\n',re_count_NNP_Eagle,'\nNumber of matches:
 ↪\n',len(re_count_NNP_Eagle) )
else:
    print('No match.')

#PP(Eagle|NNP)

count_NNP = len(re_count_NNP) # count the number of occurrences of the tag NNP
count_NNP_Eagle = len(re_count_NNP_Eagle) # count the number of occurrences of␣
 ↪the tag NNP followed by Tag Eagle

if count_NNP_Eagle > 0:
    Prob_Eagle_NNP = round(count_NNP_Eagle/count_NNP,2) # Transition␣
 ↪Probability of P(Eagle|NNP)
    print('Count (NNP) is :',count_NNP )
    print('Count (Eagle NNP) :',count_NNP_Eagle )
    print('Transition Probability - P(Eagle|NNP)  is :',Prob_Eagle_NNP )
else:
    print('Transition Probability - P(Eagle|NNP)  is : 0' )
print('\n----------------------------------------------------\n')
```

```
Found a match:
 ['NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP',
'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP', 'NNP',
'NNP', 'NNP', 'NNP']
Number of matches:
 25
Found a match:
 ['NNP/ Eagle']
Number of matches:
 1
Count (NNP) is : 25
```

```
Count (Eagle NNP) : 1
Transition Probability - P(Eagle|NNP)  is : 0.04

-----------------------------------------------
```