Experiment No : 4

```
import re

import numpy

import nltk

from nltk.corpus import stopwords

set(stopwords.words('english'))


from nltk.tokenize import sent_tokenize, word_tokenize


f = open("C:/Users/MY/Desktop/New folder/HMM.txt", "r")


if f.mode == 'r':

	text = f.read()


list_of_sentences = sent_tokenize(text)

print()

print(list_of_sentences)


token_list = []

for text in list_of_sentences:

	text = text.lower()

	text = re.sub('[^a-zA-Z]', ' ', text)

	text = re.sub("&lt;/?.*?&gt;"," &lt;&gt; ",text)

	word_list = word_tokenize(text)
```

```
        token_list.append(word_list)



tagged_list = []

alt_tagged_list = []


for word_list in token_list:

        tagged = nltk.pos_tag(word_list)

        for x in tagged:

                tagged_list.append(x)


        alt_tagged_list.append(tagged)


print()

print(alt_tagged_list)


data = alt_tagged_list

#data = [[("mary", "N"), ("jane", "N"), ("can", "M"), ("see", "V"), ("will", "N")], [("spot", "N"),
("will", "M"), ("see", "V"), ("mary", "N")], [("will", "M"), ("jane", "N"), ("spot", "V"), ("mary",
"N")], [("mary", "N"), ("will", "M"), ("pat", "V"), ("spot", "N")]]


tags = set()

words = set()


for i in data:

        for j in i:
                words.add(j[0])
                tags.add(j[1])

taglist = list(tags)
wordlist = list(words)

print("\nTag List: ")
print(taglist)

print("\nWord List: ")
print(wordlist)

rows = len(wordlist)
cols = len(taglist)

emission = numpy.zeros(shape=(rows, cols))

for i in range(rows):
        for j in range(cols):
                x = (wordlist[i], taglist[j])

                for k in data:
                        if(x in k):
                                emission[i][j] += 1
```

```
#print(emission)

for j in range(cols):
    count=0
    for i in range(rows):
        count += emission[i][j]
    for i in range(rows):
        emission[i][j] /= count

print()
print("Emission Probability: ")
print(emission)

tagseq = []
for i in data:
    temp = []
    temp.append("START")
    for j in i:
        temp.append(j[1])
    temp.append("END")

    tagseq.append(temp)


row = []
```

```
row.append("START")

for i in taglist:
    row.append(i)

col = []

for i in taglist:
    col.append(i)

col.append("END")

#print(row)
#print(col)

size = len(row)

transmission = numpy.zeros(shape=(size, size))

for i in range(size):
    for j in range(size):
        count = 0
        for a in tagseq:
            for b in range(len(a)-1):
                if(row[i]==a[b] and col[j]==a[b+1]):
                    count += 1
        transmission[i][j] = count

#print(transmission)

for i in range(size):
    count=0
    for j in range(size):
        count += transmission[i][j]
    for j in range(size):
        transmission[i][j] /= count

print()
print("Transmission Probability:")
print(transmission)

string = ["jane", "will", "see", "kevin"]

print()
print("Test sequence: ")
print(string)

final = []
size = len(taglist)
```

```
j = 0;

temp = []

for i in range(size):

    k = wordlist.index(string[j])

    x = emission[k][i]

    y = transmission[0][i]

    temp.append((x*y, -1))


final.append(temp)




for j in range(1, len(string)):

    temp = []


    tup = final[j-1]


    for i in range(size):

        k = wordlist.index(string[j])

        x = emission[k][i]


        arr = [x]*size


        for n in range(size):

            arr[n] *= transmission[n+1][i]
        for n in range(size):
            arr[n] *= tup[n][0]

        y = max(arr)
        temp.append((y, arr.index(y)))

    final.append(temp)

#print(final)

j = len(string);
tup = final[j-1]
temp = []
arr = [1]*size

for i in range(size):

    x = tup[i][0]
    y = transmission[i+1][size]
    arr[i] = arr[i]*x*y

y = max(arr)
index = arr.index(y)
```

```
for i in range(size):
    temp.append((arr[i], index))




fseq = []
fprob = temp[index][0]

for i in range(len(string)):
    fseq.append(taglist[index])
    tup = final[len(string)-i-1]
    nexthop = tup[index][1]
    index = nexthop




print()
for x in final:
    print(x)

print()
fseq = fseq[::-1]
print("Tag sequence: ",fseq)
print()

print("Probability of the tag sequence: ", fprob)
```

Output:

```
C:\Users\MY\Desktop>python exp1.py
```

Input :

Jane can see Kevin.

Spot will see Jane.

Jane will see Spot.

Kevin can see Jane.

Jane and Spot will see Kevin.

```
C:\Users\MY\Desktop>python exp1.py


['Jane can see Kevin.', 'Spot will see Jane.', 'Jane will see Spot.', 'Kevin can see Jane.', 'Jane and
Spot will see Kevin.']


[[('jane', 'NN'), ('can', 'MD'), ('see', 'VB'), ('kevin', 'VB')], [('spot', 'NN'), ('will', 'MD'), ('see',
'VB'), ('jane', 'NN')], [('jane', 'NN'), ('will', 'MD'), ('see', 'VB'), ('spot', 'NN')], [('kevin', 'NN'),
('can', 'MD'), ('see', 'VB'), ('jane', 'NN')], [('jane', 'NN'), ('and', 'CC'), ('spot', 'NN'), ('will', 'MD'),
('see', 'VB'), ('kevin', 'VB')]]


Tag List:

['MD', 'VB', 'NN', 'CC']


Word List:
```

```
['spot', 'jane', 'can', 'and', 'see', 'will', 'kevin']

Emission Probability:
[[0.         0.         0.33333333 0.        ]
 [0.         0.         0.55555556 0.        ]
 [0.4        0.         0.         0.        ]
 [0.         0.         0.         1.        ]
 [0.         0.71428571 0.         0.        ]
 [0.6        0.         0.         0.        ]
 [0.         0.28571429 0.11111111 0.        ]]

Transmission Probability:
[[0.         0.         1.         0.         0.        ]
 [0.         1.         0.         0.         0.        ]
 [0.         0.28571429 0.42857143 0.         0.28571429]
 [0.55555556 0.         0.         0.11111111 0.33333333]
 [0.         0.         1.         0.         0.        ]]

Test sequence:
['jane', 'will', 'see', 'kevin']

[(0.0, -1), (0.0, -1), (0.5555555555555556, -1), (0.0, -1)]
[(0.18518518518518517, 2), (0.0, 0), (0.0, 0), (0.0, 0)]
[(0.0, 0), (0.13227513227513227, 0), (0.0, 0), (0.0, 0)]
[(0.0, 0), (0.01079796998164345, 1), (0.006298815822625346, 1), (0.0, 0)]
```

```
Tag sequence:  ['NN', 'MD', 'VB', 'VB']


Probability of the tag sequence:  0.003085134280469557
```