Experiment No 05: Part of Speech Tagging

Rebecca Dias

Roll no: 18 BE CMPN A2

Pid: 182027

Aim : To implement the Part of Speech Tagging with Hidden Markov Models

```
string = "<s> Mary (N), Jane (N) can (M) see (V) Will (N) </s> <s> Spot (N) will (M) see (V) Mary (N) </s> <s> Will (M) Jane (N) spot (V) Ma
string
```

```
['<s>',
 'Mary',
 '(N)',
 'Jane',
 '(N)',
 'can',
 '(M)',
 'see',
 '(V)',
 'Will',
 '(N)',
 '</s>',
 '<s>',
 'Spot',
 '(N)',
 'will',
 '(M)',
 'see',
 '(V)',
 'Mary',
 '(N)',
 '</s>',
 '<s>',
 'Will',
 '(M)',
 'Jane',
 '(N)',
 'spot',
 '(V)',
 'Mary',
 '(N)',
 '</s>',
 '<s>',
 'Mary',
 '(N)',
 'will',
 '(M)',
 'see',
 '(V)',
 'Spot',
 '(N)',
 '</s>']
```

```
newString = "<s> Will will see Spot </s>".replace(",", "").lower().split(" ")[:-1]
# <s> Can Mary see Jane </s>
newString
```

```
['<s>', 'will', 'will', 'see', 'spot']
```

```
tags = ['<s>', 'N', 'M', 'V', '</s>']
tags
```

```
['<s>', 'N', 'M', 'V', '</s>']
```

```
l1 = [['<s>', '<s>']]
transition_string = "<s>"
for i  in range(1, len(string)-1, 2):
  transition_string+=" "+string[i+1][1] if (string[i] != '</s>') else " "+string[i]
  l1.append([string[i].lower(), string[i+1][1] if (string[i] != '</s>') else string[i]])
  if string[i] == "</s>":
    transition_string+=" <s>"
    l1.append(['<s>', '<s>'])

transition_string+=" </s>"
l1.append(['</s>', '</s>'])

l1
```

```
[['<s>', '<s>'],
 ['mary', 'N'],
 ['jane', 'N'],
       ['can', 'M'],
       ['see', 'V'],
       ['will', 'N'],
       ['</s>', '</s>'],
       ['<s>', '<s>'],
       ['spot', 'N'],
       ['will', 'M'],
       ['see', 'V'],
       ['mary', 'N'],
       ['</s>', '</s>'],
       ['<s>', '<s>'],
       ['will', 'M'],
       ['jane', 'N'],
       ['spot', 'V'],
       ['mary', 'N'],
       ['</s>', '</s>'],
       ['<s>', '<s>'],
       ['mary', 'N'],
       ['will', 'M'],
       ['see', 'V'],
       ['spot', 'N'],
       ['</s>', '</s>']]
```

```
transition_string = transition_string.split()
```

```
transition_string
```

```
      ['<s>',
       'N',
       'N',
       'M',
       'V',
       'N',
       '</s>',
       '<s>',
       'N',
       'M',
       'V',
       'N',
       '</s>',
       '<s>',
       'M',
       'N',
       'V',
       'N',
       '</s>',
       '<s>',
       'N',
       'M',
       'V',
       'N',
       '</s>']
```

```
def calcProb(firstWord, secondWord):
  countFollowedBy = 0
  for i in range(len(l1)-1):
    if transition_string[i] == firstWord and transition_string[i+1] == secondWord:
      countFollowedBy += 1

  return countFollowedBy/transition_string.count(firstWord)
```

```
transitions = {}
pairs = []

for i in range(len(tags)):
  for j in range(i, len(tags)):
    transitions[tags[i]+", "+tags[j]] = 0.0
# print(transitions)

for i in range(len(transition_string) - 1):
  pair = transition_string[i]+", "+transition_string[i+1]
  if pair == "</s>, <s>":
    continue

  # if pair not in transitions:
  pairs.append(pair)
  transitions[pair] = calcProb(transition_string[i], transition_string[i+1])
```

```
transitions
```

```
      {'</s>, </s>': 0.0,
       '<s>, </s>': 0.0,
       '<s>, <s>': 0.0,
       '<s>, M': 0.25,
       '<s>, N': 0.75,
       '<s>, V': 0.0,
       'M, </s>': 0.0,
       'M, M': 0.0,
 'M, N': 0.25,
 'M, V': 0.75,
 'N, </s>': 0.4444444444444444,
 'N, M': 0.3333333333333333,
 'N, N': 0.1111111111111111,
 'N, V': 0.1111111111111111,
 'V, </s>': 0.0,
 'V, N': 1.0,
 'V, V': 0.0}
```

```
pairs
```

```
['<s>, N',
 'N, N',
 'N, M',
 'M, V',
 'V, N',
 'N, </s>',
 '<s>, N',
 'N, M',
 'M, V',
 'V, N',
 'N, </s>',
 '<s>, M',
 'M, N',
 'N, V',
 'V, N',
 'N, </s>',
 '<s>, N',
 'N, M',
 'M, V',
 'V, N',
 'N, </s>']
```

```
for item in transitions:
  print("P("+item+") = "+str(transitions[item]))
```

```
P(<s>, <s>) = 0.0
P(<s>, N) = 0.75
P(<s>, M) = 0.25
P(<s>, V) = 0.0
P(<s>, </s>) = 0.0
P(N, N) = 0.1111111111111111
P(N, M) = 0.3333333333333333
P(N, V) = 0.1111111111111111
P(N, </s>) = 0.4444444444444444
P(M, M) = 0.0
P(M, V) = 0.75
P(M, </s>) = 0.0
P(V, V) = 0.0
P(V, </s>) = 0.0
P(</s>, </s>) = 0.0
P(V, N) = 1.0
P(M, N) = 0.25
```

```
emissions = {}
for i in range(len(l1)):
  numerator, denominator = 0, 0
  emission_pair = l1[i][0] + ", " + l1[i][1]
  if emission_pair in emissions:
    continue

  for j in range(len(l1)):
    if l1[i] == l1[j]:
      numerator+=1
    if l1[i][0] == l1[j][0]:
      denominator+=1
  emissions[emission_pair] = numerator / denominator
```

```
emissions
```

```
{'</s>, </s>': 1.0,
 '<s>, <s>': 1.0,
 'can, M': 1.0,
 'jane, N': 1.0,
 'mary, N': 1.0,
 'see, V': 1.0,
 'spot, N': 0.6666666666666666,
 'spot, V': 0.3333333333333333,
 'will, M': 0.75,
 'will, N': 0.25}
```

```
viterbi = {}
for word in newString:
  for pair in emissions:
    if word in pair:
      if pair == '</s>, </s>':
        continue
      elif pair == '<s>, <s>':
        viterbi[pair] = 1.0
      else:
        viterbi[pair] = 0.0

# viterbi


viterbiMat = []
for i in range(len(newString)):
  viterbiMat.append([])
  for j in range(len(tags)):
      viterbiMat[i].append(0)
viterbiMat[0][0] = 1
# viterbiMat


def calcValue(col, pair, currTag):
  global viterbiMat, transitions, emissions
  for row in range(len(viterbiMat)):
    prevViterbi = viterbiMat[row][col-1]
    if prevViterbi != 0:
      value = prevViterbi * emissions[pair]  * transitions[tags[row]+", "+currTag]
      viterbiMat[currTagIndex][col] = max(viterbiMat[currTagIndex][col], round(value, 4))

for col in range(1, len(newString)):
  for pair in viterbi:
    if newString[col] in pair:
      currTag = pair.split(", ")[1]
      currTagIndex = tags.index(currTag)
      calcValue(col, pair, currTag)

  calcValue(col, pair, currTag)



print("\t",end="")
for word in newString:
  print(word,end="\t")

print()
for i in range(len(viterbiMat)):
  for j in range(len(viterbiMat[0])):
    # if i == 0 and j == 0:
    if j == 0:
      print(tags[i],end="\t")
    print(viterbiMat[i][j],end="\t")
  print()
```

```
        <s>     will    will    see     spot
<s>     1       0       0       0       0
N       0       0.1875  0.0117  0       0.0235
M       0       0.1875  0.0469  0       0
V       0       0       0       0.0352  0
</s>    0       0       0       0       0
```

Conclusion:

In this experiment we implemented the Part of Speech Tagging with Hidden Markov Models.