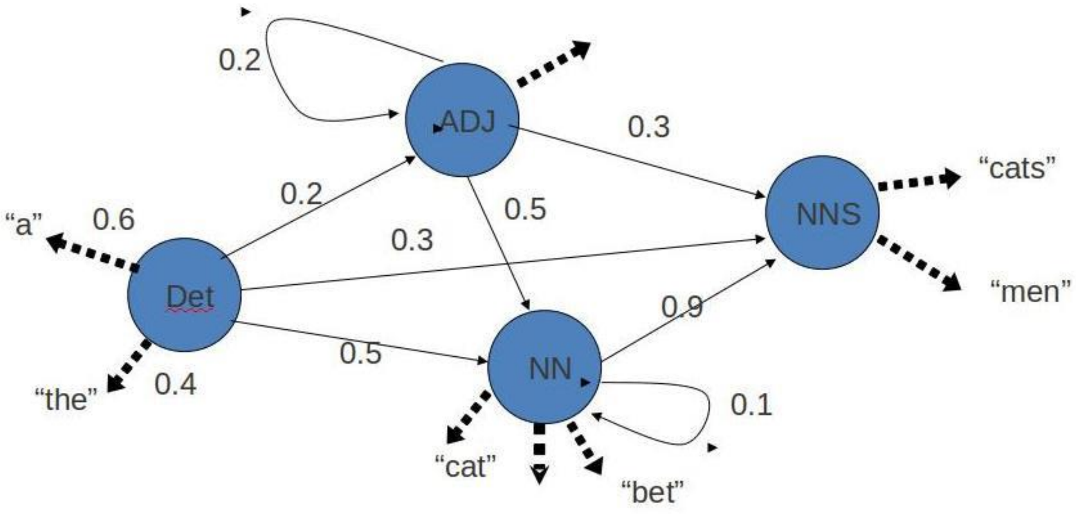
Experiment 4

Aim

To study and implement Hidden Markov Models (HMM) to calculate the probability of a sequence of tags using NLTK

Theory

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible.

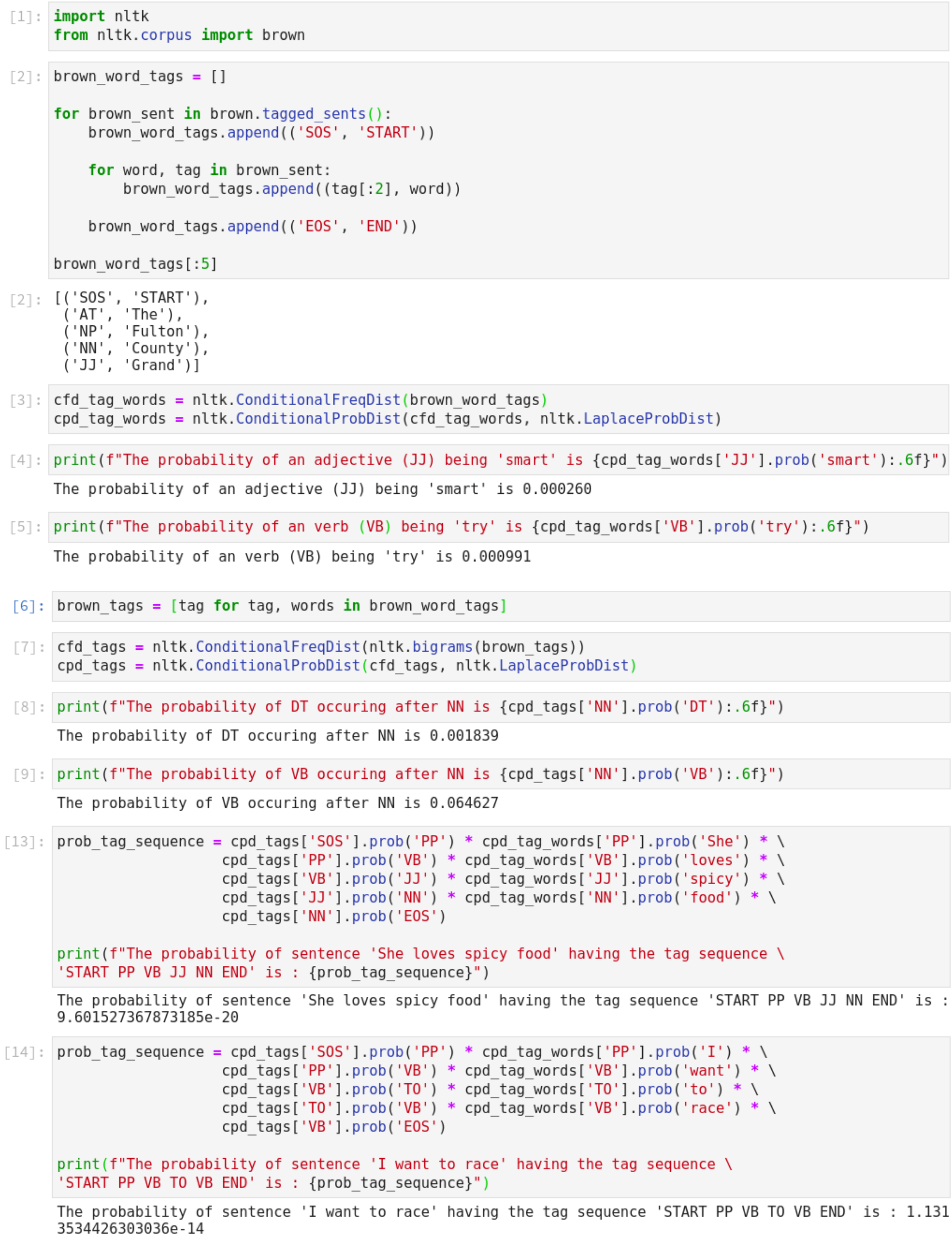


Hidden Markov Model has two important components:

1. Transition Probabilities: The one-step transition probability is the probability of transitioning from one state to another in a single step.
2. Emission Probabilities: The output probabilities for an observation from state. Emission probabilities $B = \{b_{i,k} = b_i(o_k) = P(o_k|q_i)\}$, where okis an Observation. Informally, B is the probability that the output is o_k given that the current state is q_i

For POS tagging; it is assumed that POS are generated as random processes, and each process randomly generates a word. Hence, transition matrix denotes the transition probability from one POS to another and emission matrix denotes the probability that a given word can have a particular POS.

Code



```
[1]: import nltk
     from nltk.corpus import brown
```

```
[2]: brown_word_tags = []

     for brown_sent in brown.tagged_sents():
         brown_word_tags.append(('SOS', 'START'))

         for word, tag in brown_sent:
             brown_word_tags.append((tag[:2], word))

         brown_word_tags.append(('EOS', 'END'))

     brown_word_tags[:5]
```

```
[2]: [('SOS', 'START'),
      ('AT', 'The'),
      ('NP', 'Fulton'),
      ('NN', 'County'),
      ('JJ', 'Grand')]
```

```
[3]: cfd_tag_words = nltk.ConditionalFreqDist(brown_word_tags)
     cpd_tag_words = nltk.ConditionalProbDist(cfd_tag_words, nltk.LaplaceProbDist)
```

```
[4]: print(f"The probability of an adjective (JJ) being 'smart' is {cpd_tag_words['JJ'].prob('smart'):.6f}")
```

```
     The probability of an adjective (JJ) being 'smart' is 0.000260
```

```
[5]: print(f"The probability of an verb (VB) being 'try' is {cpd_tag_words['VB'].prob('try'):.6f}")
```

```
     The probability of an verb (VB) being 'try' is 0.000991
```

```
[6]: brown_tags = [tag for tag, words in brown_word_tags]
```

```
[7]: cfd_tags = nltk.ConditionalFreqDist(nltk.bigrams(brown_tags))
     cpd_tags = nltk.ConditionalProbDist(cfd_tags, nltk.LaplaceProbDist)
```

```
[8]: print(f"The probability of DT occuring after NN is {cpd_tags['NN'].prob('DT'):.6f}")
```

```
     The probability of DT occuring after NN is 0.001839
```

```
[9]: print(f"The probability of VB occuring after NN is {cpd_tags['NN'].prob('VB'):.6f}")
```

```
     The probability of VB occuring after NN is 0.064627
```

```
[13]: prob_tag_sequence = cpd_tags['SOS'].prob('PP') * cpd_tag_words['PP'].prob('She') * \
                          cpd_tags['PP'].prob('VB') * cpd_tag_words['VB'].prob('loves') * \
                          cpd_tags['VB'].prob('JJ') * cpd_tag_words['JJ'].prob('spicy') * \
                          cpd_tags['JJ'].prob('NN') * cpd_tag_words['NN'].prob('food') * \
                          cpd_tags['NN'].prob('EOS')

      print(f"The probability of sentence 'She loves spicy food' having the tag sequence \
      'START PP VB JJ NN END' is : {prob_tag_sequence}")
```

```
      The probability of sentence 'She loves spicy food' having the tag sequence 'START PP VB JJ NN END' is :
      9.601527367873185e-20
```

```
[14]: prob_tag_sequence = cpd_tags['SOS'].prob('PP') * cpd_tag_words['PP'].prob('I') * \
                          cpd_tags['PP'].prob('VB') * cpd_tag_words['VB'].prob('want') * \
                          cpd_tags['VB'].prob('TO') * cpd_tag_words['TO'].prob('to') * \
                          cpd_tags['TO'].prob('VB') * cpd_tag_words['VB'].prob('race') * \
                          cpd_tags['VB'].prob('EOS')

      print(f"The probability of sentence 'I want to race' having the tag sequence \
      'START PP VB TO VB END' is : {prob_tag_sequence}")
```

```
      The probability of sentence 'I want to race' having the tag sequence 'START PP VB TO VB END' is : 1.131
      3534426303036e-14
```

Conclusion

Thus, studied the Hidden Markov Model and computed the probability tag sequence using HMMs.