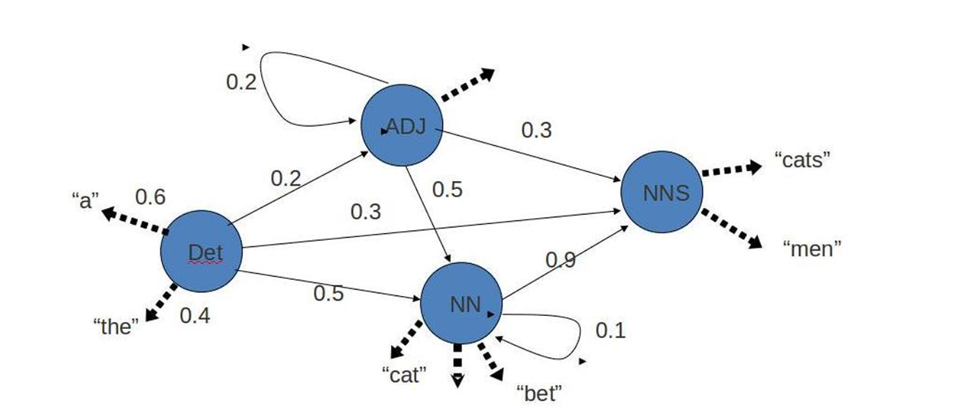
**EXPERIMENT - 4**

**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  , where  is an Observation. Informally, B is the probability that the output is given that the current state is 

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:**

| import nltk  from nltk.corpus import brown  nltk.download('brown') |
| --- |

| 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] |
| --- |

| [('SOS', 'START'),  ('AT', 'The'),  ('NP', 'Fulton'),  ('NN', 'County'),  ('JJ', 'Grand')] |
| --- |

| cfd\_tag\_words = nltk.ConditionalFreqDist(brown\_word\_tags)  cpd\_tag\_words = nltk.ConditionalProbDist(cfd\_tag\_words, nltk.LaplaceProbDist) |
| --- |

| 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 |
| --- |

| 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 |
| --- |

| brown\_tags = [tag for tag, words in brown\_word\_tags]  cfd\_tags = nltk.ConditionalFreqDist(nltk.bigrams(brown\_tags))  cpd\_tags = nltk.ConditionalProbDist(cfd\_tags, nltk.LaplaceProbDist)  print (f" The probability of DT occuring after NN is {cpd\_tags['NN'].prob('DT'):.6f}")  print (f" The probability of VB occuring after NN is {cpd\_tags['NN'].prob('VB'):.6f}") |
| --- |

| The probability of DT occuring after NN is 0.001839  The probability of VB occuring after NN is 0.064627 |
| --- |

| 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("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 |
| --- |

| 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("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 : 1.1313534426303036e-14 |
| --- |

**Conclusion:**

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