Department of Computer Engineering
Academic Term: July-November 2023

Rubrics for Lab Experiments

Class : B.E. Computer
Semester : VII
Subject Name : NLP
Subject Code : CSDC7023

Practical No:	5
Title:	Hidden Markov Model
Date of Performance:	30/08/2023
Roll No:	9426
Name of the Student:	Atharva Prashant Pawar

Evaluation:

Performance Indicator	Below average	Average	Good	Excellent	Marks
On time Submission (2)	Not submitted(0)	Submitted after deadline (1)	Early or on time submission(2)	---	
Test cases and output (4)	Incorrect output (1)	The expected output is verified only a for few test cases (2)	The expected output is Verified for all test cases but is not presentable (3)	Expected output is obtained for all test cases. Presentable and easy to follow (4)	
Coding efficiency (2)	The code is not structured at all (0)	The code is structured but not efficient (1)	The code is structured and efficient. (2)	-	
Knowledge(2)	Basic concepts not clear (0)	Understood the basic concepts (1)	Could explain the concept with suitable example (1.5)	Could relate the theory with real world application(2)	
Total					

Experiment – 5
Hidden Markov Model

Aim: To implement theHidden Markov model

Task 1:

Emission Matrix							
	book	park	car	is	in	a	the
determiner	0	0	0	0	0	1	1
noun	0.5	0.5	1	0	0	0	0
verb	0.5	0.5	0	1	0	0	0
preposition	0	0	0	0	1	0	0

Transition Matrix					
	eos	determiner	noun	verb	preposition
eos	0	0.33	0	0.5	0
determiner	0	0	1	0	0
noun	1	0	0	0.5	0
verb	0	0.33	0	0	1
preposition	0	0.33	0	0	0

Check

Right answer!!!

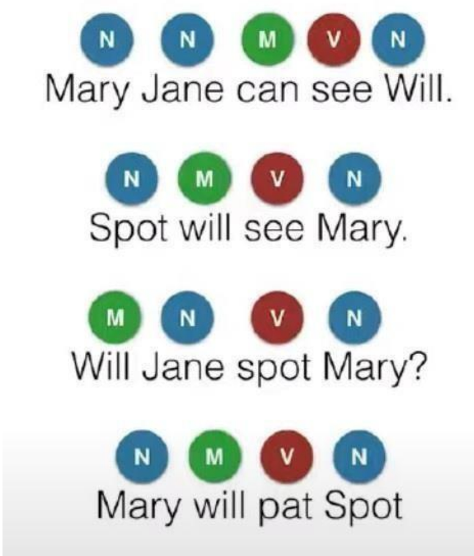
Task 2: Implement Calculation of emission probability and transition probability matrix using python code.

```python
from collections import defaultdict

# Sample tagged sentences
tagged_sentences = [
    [('<start>', 'start'), ('Mary', 'Noun'), ('Jane', 'Noun'), ('can', 'Modal'), ('see', 'Verb'), ('will', 'Noun'), ('<end>', 'end')],
    [('<start>', 'start'), ('Spot', 'Noun'), ('will', 'Modal'), ('see', 'Verb'), ('Mary', 'Noun'), ('<end>', 'end')],
    [('<start>', 'start'), ('Will', 'Modal'), ('Jane', 'Noun'), ('Spot', 'Verb'), ('Mary', 'Noun'), ('<end>', 'end')],
    [('<start>', 'start'), ('Mary', 'Noun'), ('will', 'Modal'), ('pat', 'Verb'), ('Spot', 'Noun'), ('<end>', 'end')]
]

# Calculate emission probability matrix
emission_probabilities = defaultdict(lambda: defaultdict(int))
tag_counts = defaultdict(int)

for sentence in tagged_sentences:
    for i in range(len(sentence)):
        word, tag = sentence[i]
        emission_probabilities[tag][word] += 1
        tag_counts[tag] += 1

emission_matrix = {}
for tag, word_counts in emission_probabilities.items():
    emission_matrix[tag] = {word: count / (tag_counts[tag]) for word, count in word_counts.items()}

# Calculate transition probability matrix
transition_probabilities = defaultdict(lambda: defaultdict(int))
tag_pair_counts = defaultdict(int)

for sentence in tagged_sentences:
    for i in range(len(sentence) - 1):
        current_tag, next_tag = sentence[i][1], sentence[i + 1][1]
        transition_probabilities[current_tag][next_tag] += 1
        tag_pair_counts[current_tag] += 1

transition_matrix = {}
for current_tag, next_tag_counts in transition_probabilities.items():
    transition_matrix[current_tag] = {next_tag: count / (tag_pair_counts[current_tag]) for next_tag, count in next_tag_counts.items()}

# Print emission and transition matrices
print("Emission Probability Matrix:\n")
for tag, word_probabilities in emission_matrix.items():
    print(tag, word_probabilities)

print("\n\nTransition Probability Matrix:\n")
for current_tag, next_tag_probabilities in transition_matrix.items():
    print(current_tag, next_tag_probabilities)
```

```
Emission Probability Matrix:

start {'<start>': 1.0}
Noun {'Mary': 0.4444444444444444, 'Jane': 0.2222222222222222, 'will': 0.1111111111111111, 'Spot': 0.2222222222222222}
Modal {'can': 0.25, 'will': 0.5, 'Will': 0.25}
Verb {'see': 0.5, 'Spot': 0.25, 'pat': 0.25}
end {'<end>': 1.0}


Transition Probability Matrix:

start {'Noun': 0.75, 'Modal': 0.25}
Noun {'Noun': 0.1111111111111111, 'Modal': 0.3333333333333333, 'end': 0.4444444444444444, 'Verb': 0.1111111111111111}
Modal {'Verb': 0.75, 'Noun': 0.25}
Verb {'Noun': 1.0}
```

Post Lab questions:

1) Compute emission and transition matrix for the following example. Consider the following three Tags – Noun (N), Verb (V), Modal (M),



And check whether the following tagging is correct or not using HMM model.

Will Marry Spot Jane?

Will	Marry	Spot	Jane
Noun	Noun	Verb	Noun

Atharva Prashant Pawar (9427) - Compe - A [Batch D]

NLP - Postlab - 5

Q1)

	N	N	M	V	N	
<S>	Mary	good	can	see	will	<E>
	(N)	(M)	(V)	(N)		
<S>	spot	will	see	mary		<E>
	(M)	(N)	(V)	(M)		
<S>	will	Jone	spot	mary?		<E>
	(N)	(M)	(V)	(N)		
<S>	mary	will	~~pat~~ pat	spot		<E>

* Emission Matrix:

words	Noun	modal	verb
mary	4/9	0/4	0/4
Jone	2/9	0/4	0/4
can	0/9	1/4	0/4
see	0/9	0/4	2/4
will	1/9	3/4	0/4
spot	2/9	0/4	1/4
pat	0/9	0/4	1/4

* Transition matrix

	Noun	Modal	Verb	<E>
<S>	3/4	1/4	0	0
Noun	1/9	3/9	1/9	4/9
modal	1/4	0/4	3/4	0
verb	4/4	0	0	0

N N V N
will marry spot Jone ?

$$\langle S\rangle \xrightarrow{3/4} \underset{\substack{\downarrow \\ \text{will}}}{\text{Noun}} \xrightarrow{3/4} \text{modal} \xrightarrow{3/4} \text{verb} \xrightarrow{0} \langle E\rangle$$

$$\langle S\rangle \xrightarrow{3/4} \underset{\substack{1/9\,\downarrow \\ \text{will}}}{\text{Noun}} \xrightarrow{1/9} \underset{\substack{4/9\,\downarrow \\ \text{marry}}}{\text{Noun}} \xrightarrow{1/9} \underset{\substack{1/4\,\downarrow \\ \text{spot}}}{\text{verb}} \xrightarrow{1} \underset{\substack{2/9\,\downarrow \\ \text{jone}}}{\text{Noun}} \xrightarrow{4/9} \langle E\rangle$$

$$\frac{3}{4} \times \frac{1}{9} \times \frac{1}{9} \times 1 \times \frac{4}{9} \times \frac{1}{9} \times \frac{4}{9} \times \frac{1}{4} \times \frac{2}{9}$$

$$= 0.000001129$$

The given tagging is correct.

2) What are the limitations of HMM model? How do you overcome these limitations?

Q2) limitation:

① Independence Assumption:
one of the primary limitations of Hmms is the assumption of the marker property. which states that the feature state of the sys. only depends on current state.

② Fixed state space:
Hmms assume a fixed finite set of hidden states. In reality the number of state might not be known beforehand.

③ Overcoming:
to address this, you can build more flexible models like Dynamic Bayseian n/ws.

④ Limited Expressiveness:
Hmms have limited capacity to capture complex relationships b/w observations & states.

⑤ Overcoming:
To capture riches dependence, you can be more advanced models like Recurrent Neural N/ws or LSTMs.

⑥ Difficulty in determining Hidden states:
In some cases, determining the appropriate no. of hidden states can be challenging.
Overcoming:
model selection techniques ~~~~ such as Cross-Validation, etc.