K. J. Somaiya College of Engineering, Mumbai-77

Batch: B1 **Roll No.: 16010121045**

Experiment / assignment / tutorial No. 10

Title: Virtual Lab 2

Objective: Understand the Virtual Lab Experiment for tagging Parts of Speech using Hidden Markov Model.

Expected Outcome of Experiment:

Course Outcome	After successful completion of the course students should be able to
CO4	Analyse applications of AI and understand planning & learning processes in advanced AI applications

Books/ Journals/ Websites referred:

3. **"Artificial Intelligence: a Modern Approach" by Russel and Norving, Pearson education Publications**
4. **"Artificial Intelligence" By Rich and knight, Tata Mcgraw Hill Publications**

Pre Lab/ Prior Concepts: Natural Language Processing

Hidden Markov Models (HMMs) are statistical models used to model sequences of observable events that are assumed to depend on unobservable, or "hidden," states. One common application of HMMs is in part-of-speech (POS) tagging, where each word in a sentence is assigned a grammatical category (e.g., noun, verb, adjective) based on the context of the sentence.
Here's a general procedure for POS tagging using a Hidden Markov Model:

1. **Define States**: The first step is to define the states of the model. In POS tagging, each state typically represents a part-of-speech tag (e.g., noun, verb, adjective).

2. **Define Observations**: Next, define the observations. These are the observable events associated with each state. In POS tagging, the observations are the words in the sentence.

3. **Transition Probabilities**: Calculate the transition probabilities between states. These probabilities represent the likelihood of transitioning from one POS tag to another. They are typically estimated from a labeled corpus of text data.

4. **Emission Probabilities**: Calculate the emission probabilities of observing each word given each POS tag. These probabilities represent the likelihood of observing a particular word given a certain POS tag. They are also estimated from a labeled corpus of text data.

5. **Initial State Probabilities**: Calculate the initial probabilities of starting in each state. These probabilities represent the likelihood of starting a sentence with each POS tag.

6. **Forward Algorithm**: Use the forward algorithm (or other methods like Viterbi algorithm) to compute the probability of a sequence of observations given the model. This involves calculating the probability of being in a particular state at each step of the sequence and the probability of observing each observation given the current state.

7. **Backward Algorithm (Optional)**: In some cases, the backward algorithm is used to calculate the probability of ending a sequence with a particular state. This can be useful for certain applications or for refining the tagging process.

8. **Decoding**: Once the probabilities are calculated, decode the sequence of POS tags that best explains the observed sequence of words. This is typically done using the Viterbi algorithm, which finds the most likely sequence of states given the observations.

9. **Tagging**: Finally, assign the most likely POS tag to each word in the sentence based on the decoding results.

By following this procedure, a Hidden Markov Model can be trained and used for part-of-speech tagging, which is a fundamental task in natural language processing.

Reference Link: https://nlp-iiith.vlabs.ac.in/exp/markov-model/

Simulation Results:

Corpus B

EOS/eos **Book**/verb **a**/determiner **car**/noun EOS/eos **Park**/verb **a**/determiner **car**noun EOS/eos **The**/determiner **book**/noun **is**/verb **in**/preposition **the**/determiner **car**/noun EOS/eos **The**/determiner **car**/noun **is**/verb **in**/preposition **a**/determiner **park**/noun EOS/eos

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!!!

Feedback:

+

S.No	Discipline	Lab Name	Experiment Name	Delete
1	Computer Science & Engin	Natural Language Process	POS Tagging: Hidden Mark	

+

Please indicate your agreement with the following statements:

a. The degree to which the actual lab environment is simulated — Excellent / Very Good / Good / (selected) Fair / Poor

b. The manuals were to be found helpful — Excellent / Very Good / Good / (selected) Fair / Poor

Postlab Questions:

1. **How is Hidden Markov Model different from Markov Model?**

A Markov model is a stochastic model that describes a sequence of possible events where the probability of each event depends only on the state attained in the previous event. These models are memoryless, meaning that the future state depends only on the current state and not on the sequence of events that preceded it.

A Hidden Markov Model (HMM) extends the concept of a Markov model by introducing hidden states, which are not directly observable. Instead, the model emits observable symbols or events based on the hidden states. The transitions between hidden states and the emissions of symbols from each state are governed by probabilities. In HMMs, the goal is to infer the sequence of hidden states given the observed sequence of symbols.

2. **What is the basic design for HMM for finding out POS?**

- **States**: In a POS tagging HMM, the states represent the different parts of speech (e.g., noun, verb, adjective).
- **Observations**: The observations are the words in the text.
- **Transition Probabilities**: These represent the probability of transitioning from one part of speech to another. For example, the probability of transitioning from a noun to a verb or from a verb to an adjective.
- **Emission Probabilities**: These represent the probability of observing a word given a particular part of speech. For example, the probability of observing the word "dog" given that the current part of speech is a noun.
- **Initial State Probabilities**: These represent the probability of starting with each part of speech at the beginning of a sentence.
- **Decoding Algorithm**: Use algorithms like the Viterbi algorithm to find the most likely sequence of POS tags given the observed sequence of words.

The design of a POS tagging HMM involves estimating these probabilities from a labeled corpus of text data, where each word is annotated with its corresponding part of speech. These probabilities are then used to infer the most likely sequence of POS tags for unseen text data.