NLP Experiment 8

Aim: To implement Part of Speech- tagging using HMM.

Theory:

Part of Speech (POS) tagging is a fundamental task in natural language processing (NLP) that involves assigning parts of speech to each word in a sentence, such as noun, verb, adjective, etc. It's a crucial step in linguistic analysis for various NLP tasks, including text summarization, sentiment analysis, and machine translation.

One common approach to POS tagging is through **Hidden Markov Models (HMMs)**, a probabilistic model that is well-suited for sequential data like text. HMM is often employed because of its efficiency in modelling time series and sequences where an underlying sequence (hidden states) governs the observed data.

Components of HMM:

1. **States (POS Tags)**: The hidden states in the HMM correspond to the POS tags we want to assign. Common tags include Nouns (NN), Verbs (VB), Adjectives (JJ), etc.
2. **Observations (Words)**: The words in the sentence are the observations. These are known data points in the sequence, but their corresponding POS tags are unknown (hidden states).
3. **Transition Probabilities**: These are the probabilities of moving from one hidden state (POS tag) to another. For example, the probability of a noun being followed by a verb.
4. **Emission Probabilities**: These represent the probability of a word (observation) being generated from a particular state (POS tag). For instance, the probability of the word "dog" being a noun.
5. **Initial Probabilities**: These probabilities define the likelihood of starting in each state (POS tag) at the beginning of the sentence.

Steps in POS Tagging Using HMM:

1. **Data Preprocessing**: Prepare a tagged corpus of sentences (like the Penn Treebank), where each word in the sentences has a corresponding POS tag.
2. **Training**:
   - Compute the **initial probabilities** by counting how often each POS tag appears at the start of a sentence.
   - Calculate the **transition probabilities** by counting how often one POS tag follows another.
   - Calculate the **emission probabilities** by counting how often a word is associated with a specific POS tag.
3. **Decoding (Viterbi Algorithm)**: After training, the HMM can be used to predict the sequence of POS tags for a new, unseen sentence. This prediction is done using the

Viterbi algorithm, which finds the most probable sequence of hidden states (POS tags) given the observed sequence of words.

  - The Viterbi algorithm is a dynamic programming algorithm that efficiently computes the most likely sequence of hidden states by combining both transition and emission probabilities at each step.

Why Use HMM for POS Tagging?

- **Efficiency**: HMM efficiently handles sequences of words and considers the likelihood of transitions between POS tags, which helps capture linguistic structures like noun-verb agreements.
- **Data Sparsity**: Despite limited training data, HMM performs well due to the probabilistic approach, allowing for generalizations from sparse data.
- **Markov Assumption**: The HMM assumes that the current state (POS tag) depends only on the previous state, simplifying the computation of sequences while still providing accurate predictions.

Limitations:

- **Independence Assumptions**: HMM assumes that the probability of a word depends only on its corresponding POS tag and that the current POS tag depends only on the previous tag, which might oversimplify language dependencies.
- **Limited Context**: Since HMM only looks at one preceding word, it may struggle with more complex dependencies that span across multiple words or phrases.