AIM-829 NLP
Assignment 1 : Hidden Markov Model with Viterbi Algorithm for POS Tagging

Team Z

February 14, 2025

1 Introduction

Part-of-Speech (POS) tagging is the process of assigning POS labels to words in a sentence. The Hidden Markov Model (HMM) with the Viterbi algorithm is a probabilistic approach widely used for this task. This report details the implementation of an HMM for POS tagging, utilizing the Viterbi algorithm for decoding the most likely sequence of tags as taught in class.

2 Assumptions

- The corpus used for training provides a sufficiently large and balanced distribution of POS tags and is correct.
- The probability distributions remain constant over time since it's calculated using training corpus once.
- Each word's POS tag depends only on its previous tag (Markov assumption).
- Unknown words are handled by adding "UNKNOWN" tag in emission matrix and giving it equal probability to all pos tags.

3 Methodology

3.1 Preprocessing

- First, it converts all the words to lower case.
- It handles dates and currency by converting into integer and converting most integer into "1" as there can be infinite integers, hence reducing unique words.
- Then it convert the words into it's base form by using lemmetization.
- It then converts some meaningful words to unique words like:
  - Emails to `<EMAIL>`

  - URLs to `<URL>`
  - File extensions (like `.doc`, `.xlsx`, etc.) to `<FEXT>`
  - Abbreviations (like `u.s.s.r.`) to `<ABBREV>`
  - Ambiguous words (e.g., `a.`, `b.`) to `<AMBIG>`
  - And few more
- Since there can be various words that are not present in the corpus.
- It also add "start" in the beginning of each sentence to get initial probabilities.

3.2 Hidden Markov Model (HMM)

An HMM is defined as follows:

- **States or hidden states:** The possible POS tags.
- **Observations:** Words in the given sentence.
- **Transition Probabilities** $P(t_i|t_{i-1})$: Probability of transitioning from one tag to another and also for initial probability i.e. start tag.
- **Emission Probabilities** $P(w_i|t_i)$: Probability of a word occurring given a tag.

3.3 Viterbi Algorithm

The Viterbi algorithm efficiently computes the most likely sequence of POS tags for a given sentence using dynamic programming. The recurrence relation is given by:

$$V_k(j) = \max_i [V_{k-1}(i)P(t_j|t_i)P(w_k|t_j)] \tag{1}$$

where $V_k(j)$ represents the probability of the most likely path ending in state j at time step k.

3.4 Back Propogation

It uses back propogation to get the order of correct POS tag for the given sentence.

4 Results and Observations

The performance of the model was evaluated on a test dataset provided, and the following key results were observed:

- **Accuracy:** The model achieved an accuracy of **86.21%**.
- **Confusion Matrix:** A heatmap visualization of misclassified tags was generated to analyze errors.
- **Common Errors:** The most misclassified tags included nouns.

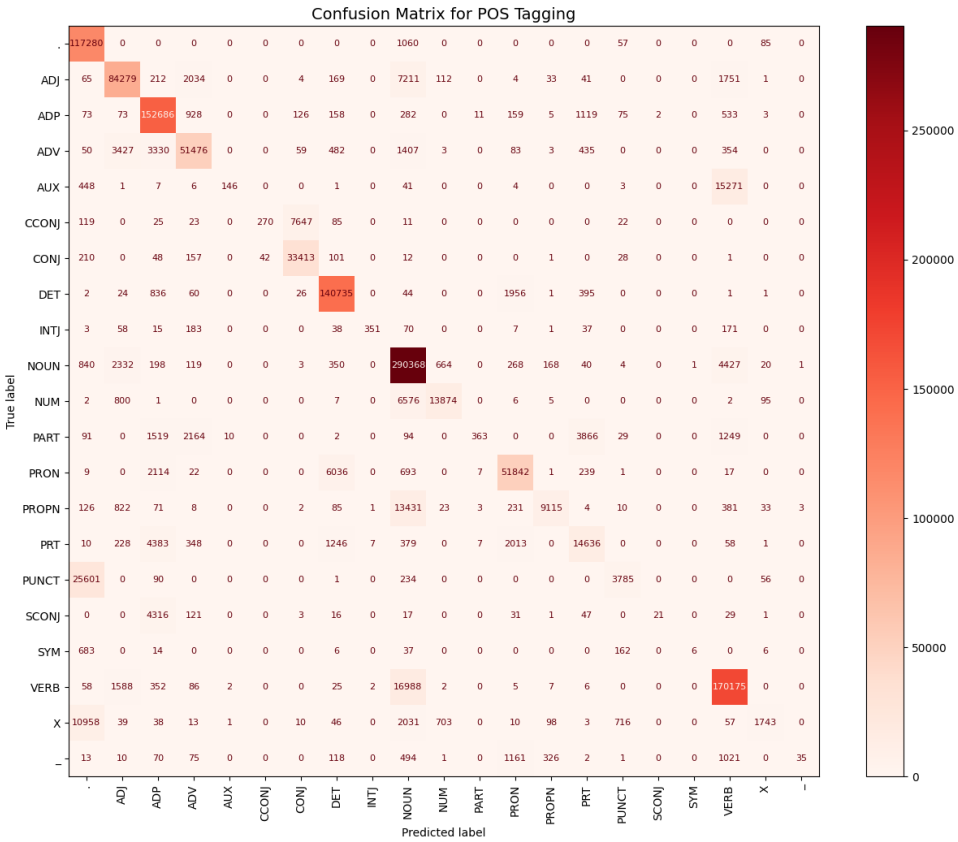

Figure 1: Confusion Matrix.

- **Visualization:** viterbi matrix, state transition graph and best tag path were plotted for some sentence.
- **Additional insights:** were that we could observe for "." mostly it's "." or "PUNCT" tag. It was also seen that some words were wrongly tagged in training corpus and all.

5 Conclusion

The HMM with the Viterbi algorithm is a powerful approach for POS tagging. The results indicate that performance is highly dependent on the quality of training data and handling of unknown words.

6 Team-wise Contribution

As a sole team member of the team, I individually handled all the work with some help of OpenAI for visualization part.