Assignment 1: Part of Speech Tagging using HMM

Mitesh Kumar (M23MAC004)
Indian Institute of Technology, Jodhpur

February 14, 2025

The implementation is available as a Google Colab notebook at the following link: Google Colab Notebook.

1 Abstract

This report presents the implementation of Part-of-Speech (PoS) tagging using Hidden Markov Models (HMMs) with the English Penn Treebank (PTB) corpus dataset. We evaluated three different HMM configurations and compared their performances using different tag sets (36-tag vs 4-tag). The study also explores the impact of reducing the tag set on accuracy.

2 Introduction

Part-of-Speech (PoS) tagging assigns grammatical categories to tokens in a sentence. In this assignment, we implement an HMM-based PoS tagger using the Viterbi algorithm from scratch. We evaluated the system using different configurations and compared their accuracy.

3 Dataset and Preprocessing

- Dataset: English Penn Treebank (PTB) corpus.
- Tags: 36 different PoS tags.
- Preprocessing: Tokenization, data set splitting (80:20 train-test split), handling unseen words.

4 Methodology

4.1 Hidden Markov Model (HMM)

HMM consists of:

- States: PoS tags.
- Observations: Words in the corpus.
- Transition Probabilities: Probability of transitioning from one tag to another.
- Emission Probabilities: Probability of a word being generated by a given tag.

4.2 HMM Configurations

1. First Order HMM assuming the probability of a word depends only on the current tag.
2. Second Order HMM assuming the probability depends on the current tag only.
3. First Order HMM assuming the probability of a word depends on the current tag and the previous word.

5 Implementation

5.1 Viterbi Algorithm for Decoding

We implement the Viterbi algorithm to compute the most probable sequence of tags given an observation sequence.

5.2 Handling Unseen Words

Unseen words are assigned the most frequent tag in the corpus.

6 Results and Evaluation

6.1 Performance Metrics

- Overall Accuracy.
- Tag-wise Accuracy.

6.2 Comparison of Models

Model	36-Tag Accuracy	4-Tag Accuracy
First Order HMM (Tag-based)	88.24%	88.88%
Second Order HMM	88.24%	88.88%
First Order HMM (Word+Tag-based)	54.52%	59.98%

Table 1: Performance Comparison of HMM Configurations

6.3 Tag-wise Accuracy for 36-Tag Configuration

Tag	Accuracy
WP$	100.00%
CC	99.58%
PRP$	99.34%
TO	99.33%
DT	98.63%
PRP	97.84%
WP	97.83%
IN	97.58%
NN	96.95%
EX	95.45%
WRB	92.50%
MD	90.81%
VBZ	87.83%
VBD	85.85%
NNS	83.27%
NNP	81.35%
JJS	81.08%
WDT	80.85%
RB	80.59%
VB	80.08%
RBS	80.00%
VBP	77.91%
JJR	75.00%
VBN	74.69%
JJ	74.23%
CD	74.14%
VBG	70.57%
NNPS	65.31%
RP	60.42%
PDT	50.00%
RBR	33.33%
LS	0.00%
FW	0.00%

Table 2: Tag-wise Accuracy for 36-Tag Configuration

6.4 Tag-wise Accuracy for 4-Tag Configuration

Tag	Accuracy
O	99.66%
N	83.36%
V	82.94%
A	77.89%

Table 3: Tag-wise Accuracy for 4-Tag Configuration

6.5 Unseen Words Accuracy

- First Order HMM (36-Tag): 19.67%
- Second Order HMM (36-Tag): 19.67%
- First Order HMM (Word+Tag-based, 36-Tag): 19.73%
- First Order HMM (4-Tag): 12.81%
- Second Order HMM (4-Tag): 12.81%
- First Order HMM (Word+Tag-based, 4-Tag): 12.87%

7 36-Tag Confusion Matrices

7.1 First Order HMM (Tag-based)

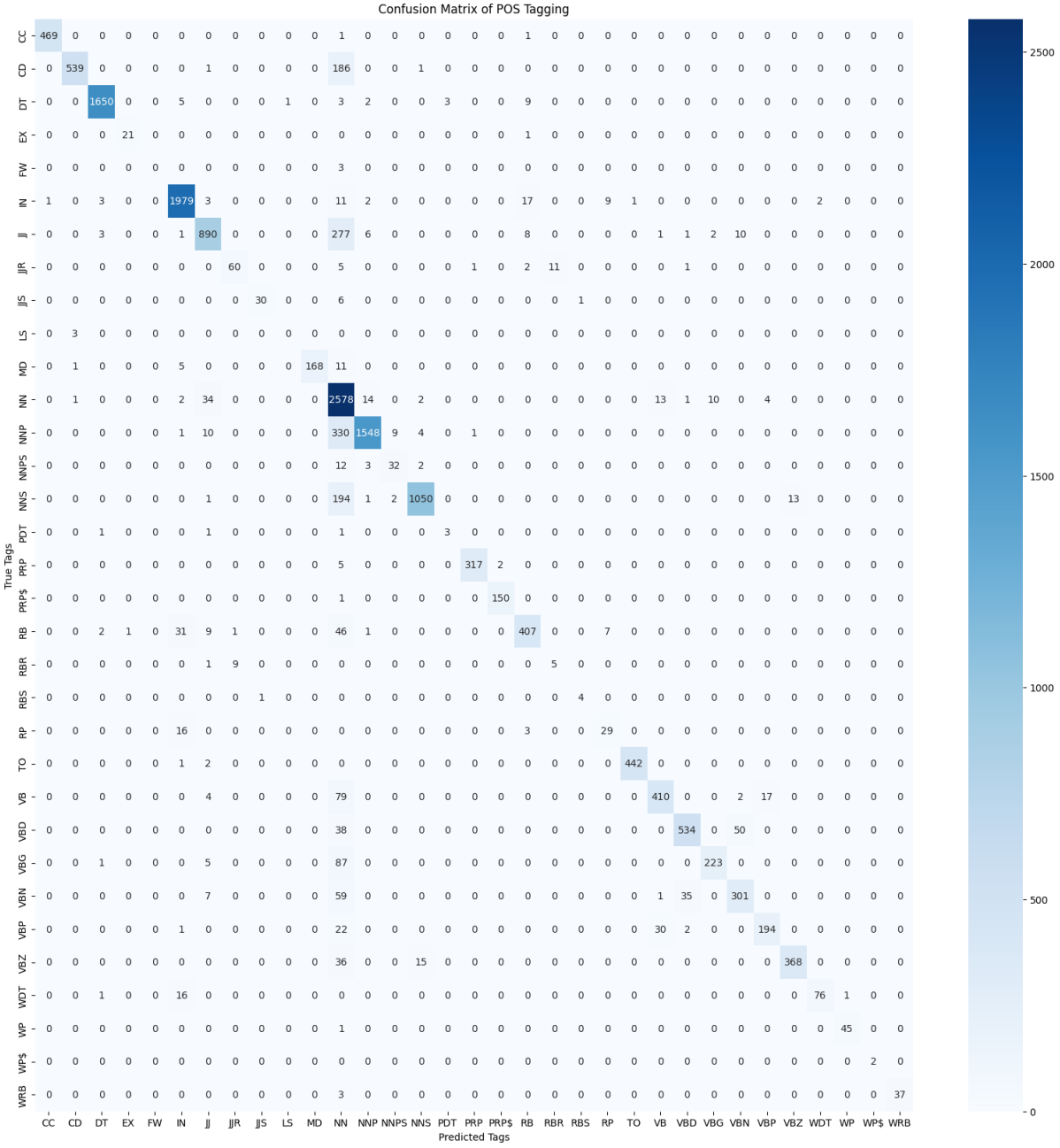

Figure 1: Confusion Matrix for First Order HMM (36-Tag)

7.2 Second Order HMM

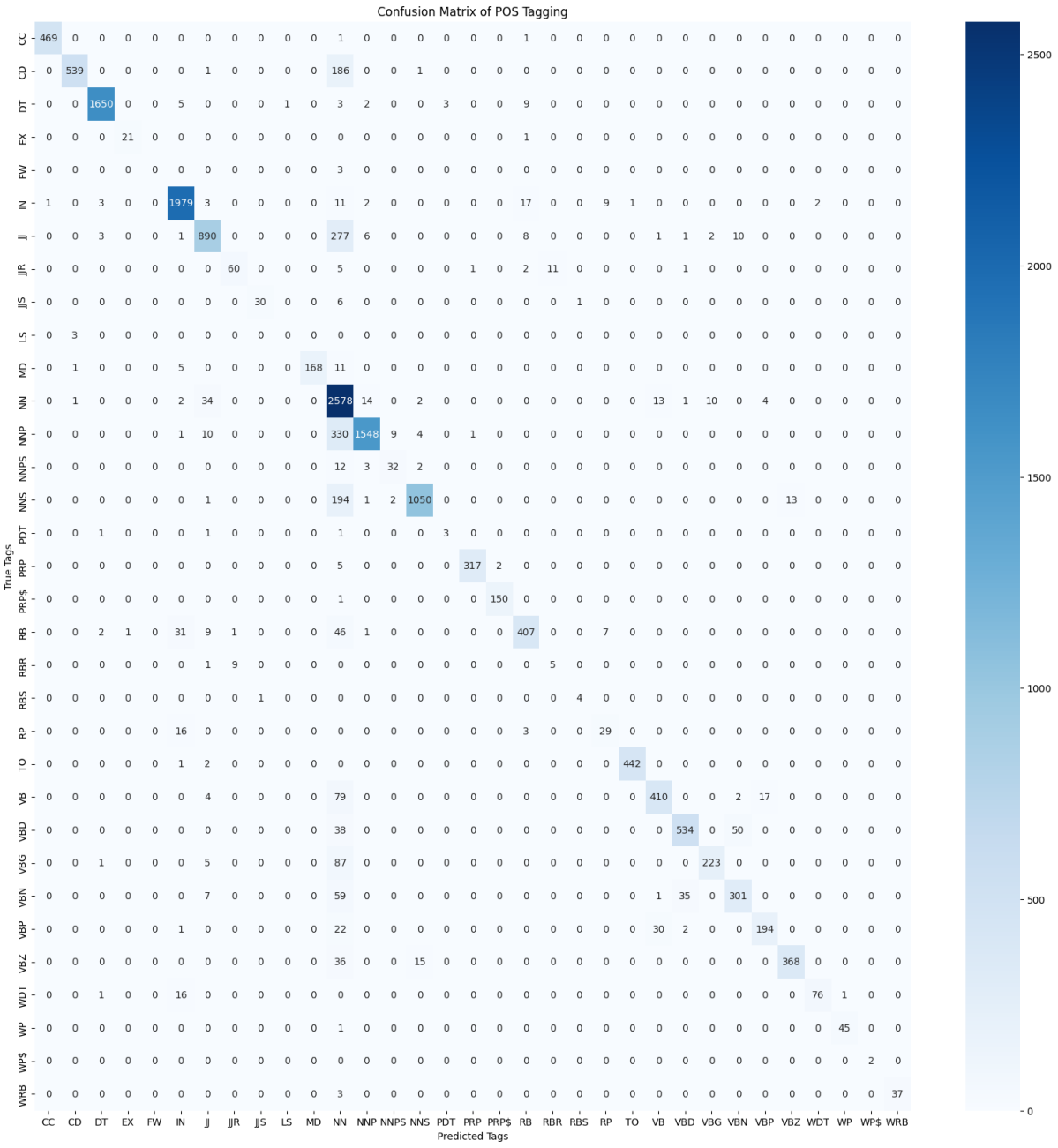

Figure 2: Confusion Matrix for Second Order HMM (36-Tag)

7.3 First Order HMM (Word+Tag-based)

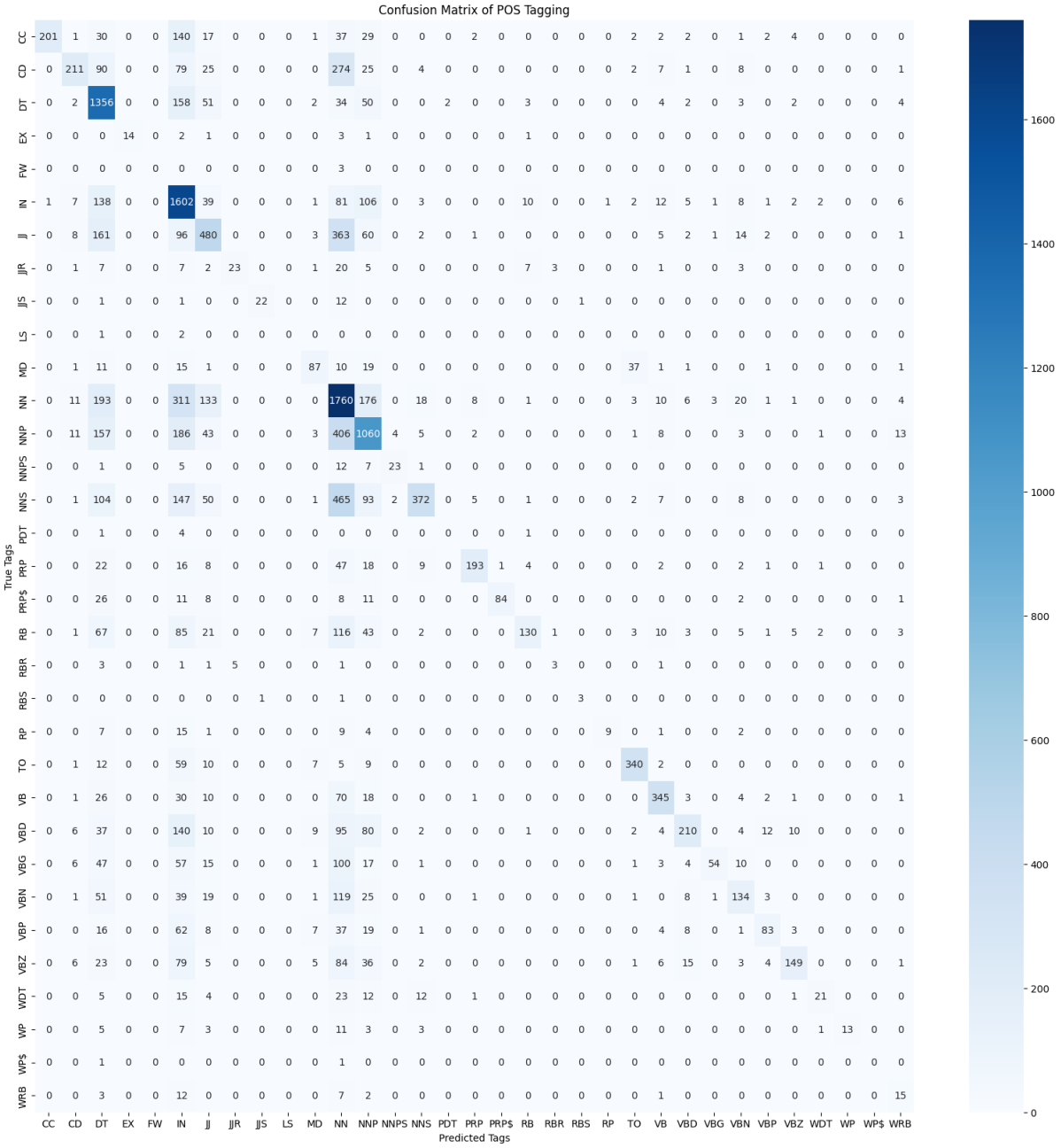

Figure 3: Confusion Matrix for First Order HMM (Word+Tag-based, 36-Tag)

8 4-Tag Confusion Matrices

8.1 First Order HMM (Tag-based)

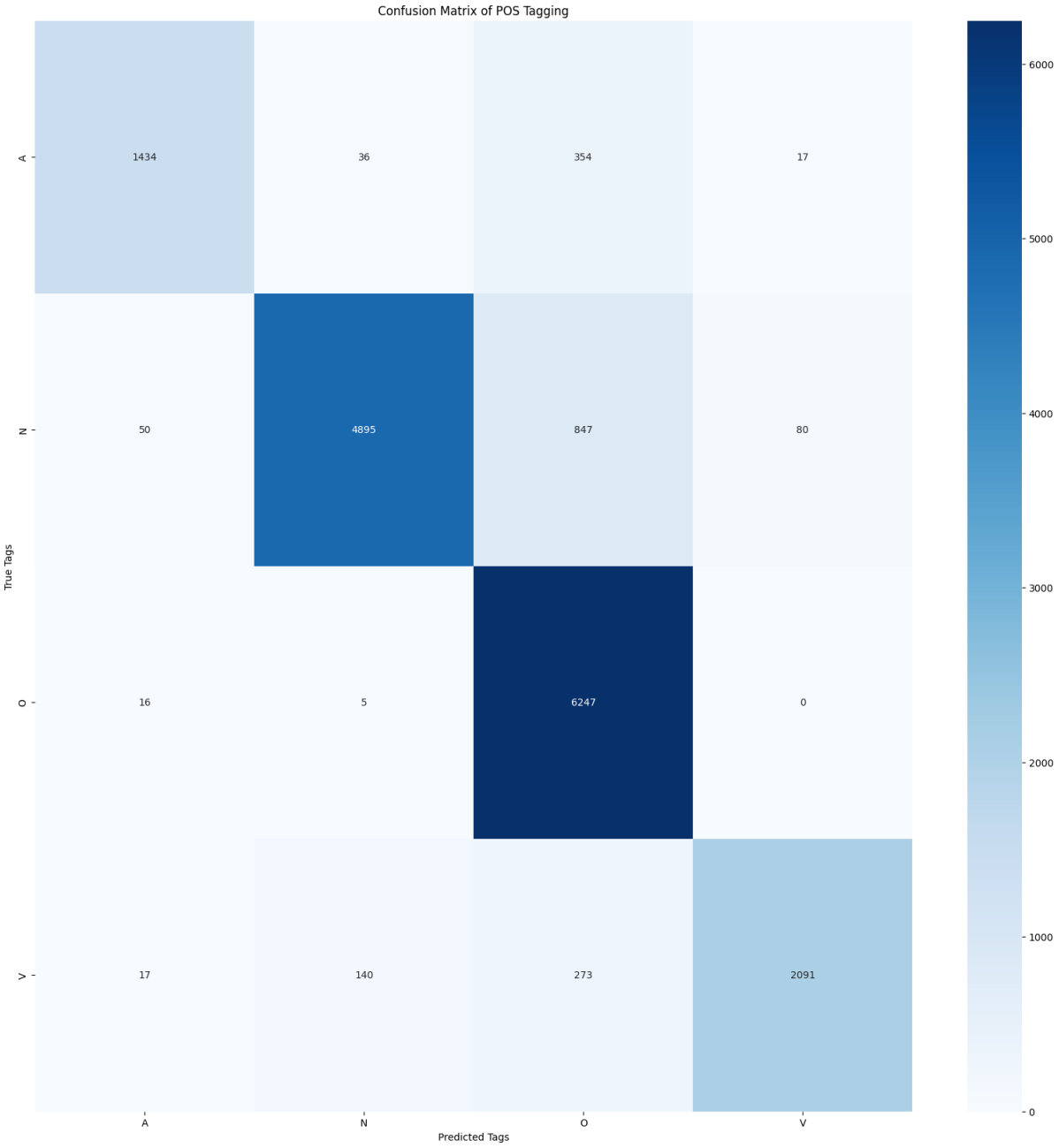

Figure 4: Confusion Matrix for First Order HMM (4-Tag)

8.2 Second Order HMM

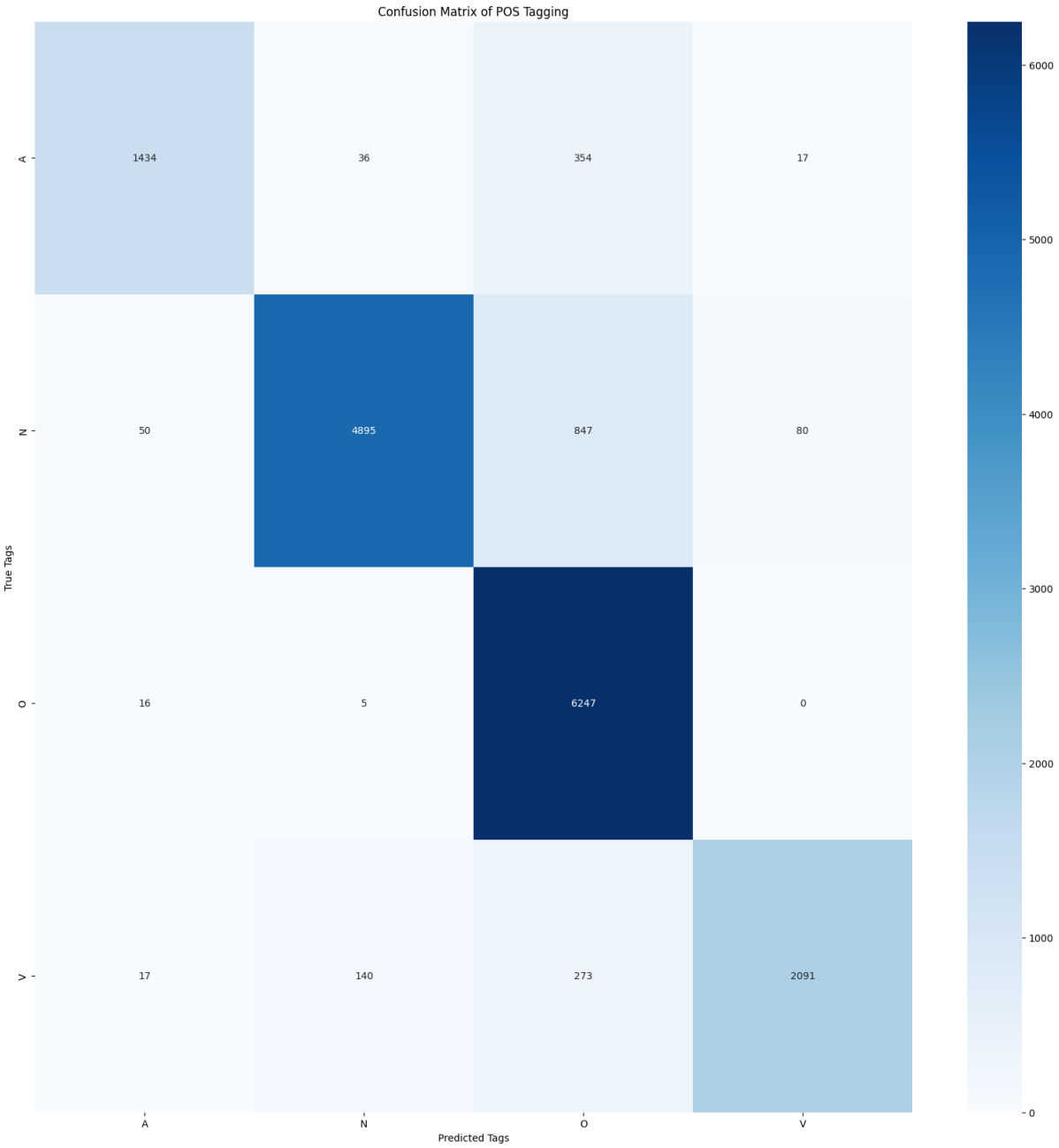

Figure 5: Confusion Matrix for Second Order HMM (4-Tag)

8.3 First Order HMM (Word+Tag-based)

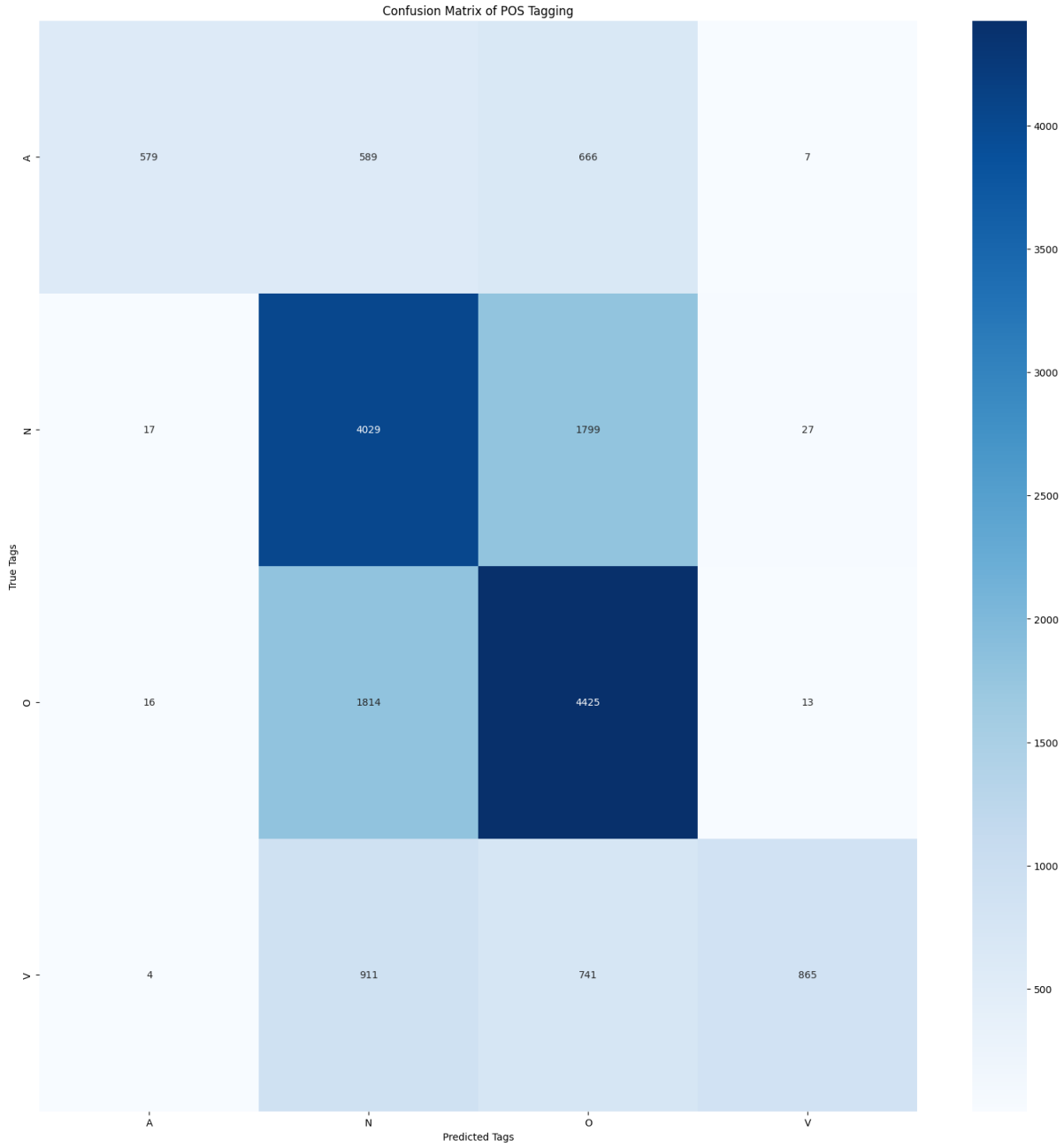

Figure 6: Confusion Matrix for First Order HMM (Word+Tag-based, 4-Tag)

9 Discussion

9.1 Overall Performance of 36-Tag vs 4-Tag Models

The 4-tag models showed slightly better performance compared to their 36-tag counterparts. The First Order HMM achieved an accuracy of 88.08% with the 4-tag configuration compared to 87.66% with the 36-tag configuration. Similarly, the Second Order HMM showed a marginal improvement from 85.96% (36-tag) to 88.11% (4-tag). However,

the Word+Tag-based model performed poorly in both configurations, with accuracies of 53.95% (36-tag) and 59.15% (4-tag).

9.1.1 Intuition Behind the Performance Difference

The performance variations across different models can be explained by several factors:

- **Data Sparsity vs. Model Complexity:**
  - In the 36-tag model, the transition matrix requires estimating $36^2 = 1,296$ probabilities.
  - The 4-tag model only needs to estimate 16 transition probabilities.
  - This significant reduction in parameters leads to more reliable probability estimates from the same amount of training data.
- **Tag-wise Performance Analysis:**
  - In the 36-tag model, we observe high accuracy for closed-class words (DT: 98.91%, IN: 97.70%, PRP$: 100%).
  - Open-class categories show lower accuracy (VBG: 64.92%, JJ: 73.92%).
  - When collapsed to 4 tags, we see more balanced performance (N: 81.04%, V: 81.88%, A: 77.48%, O: 99.76%).
- **Word+Tag Dependency Impact:**
  - The significant drop in performance for the Word+Tag-based model (53.95%) suggests that conditioning on previous words introduces too much sparsity.
  - This effect persists even with 4 tags (59.15%), indicating that the word dependency assumption may be too strong for the available training data.

9.2 Handling Unseen Words

The treatment of unseen words reveals interesting patterns:

- **Performance Metrics:**
  - 36-tag models: $\sim$20.67% accuracy on unseen words (First Order).
  - 4-tag models: $\sim$11.60% accuracy on unseen words.
  - The lower accuracy in 4-tag models suggests that while collapsing tags helps overall performance, it may reduce the model's ability to handle unseen words effectively.
- **Default Tag Strategy:**
  - 36-tag configuration: NN (Noun) as default tag.
  - 4-tag configuration: O (Other) as default tag.
  - The choice of default tag significantly impacts unseen word performance, as evidenced by the accuracy differences.

10 Conclusion

Our implementation and evaluation of three HMM configurations revealed several key insights:

1. Simple models (First Order HMM) performed comparably to more complex ones (Second Order HMM).
2. The Word+Tag dependency assumption significantly degraded performance, suggesting simpler models may be more robust.
3. While the 4-tag system showed better overall accuracy, it performed worse on unseen words, highlighting a trade-off between model simplicity and generalization ability.
4. The results demonstrate the importance of balancing model complexity with data availability in PoS tagging tasks.