CSL7640 : Natural Language Understanding

Part-of-Speech Tagging Using Hidden Markov Model (HMM)

By - B22CS001, B22CS093, B22BB016

Abstract

Part-of-Speech (PoS) tagging is a fundamental task in natural language processing (NLP) that assigns grammatical categories to each word in a sentence. In this project, we develop a PoS tagger using the Hidden Markov Model (HMM) without relying on existing libraries. Our implementation employs the Viterbi algorithm to determine the most probable sequence of tags for a given sentence.

We utilize the English Penn Treebank (PTB) corpus, splitting it into an 80:20 ratio for training and testing. The performance of our model is evaluated in two configurations: (1) using all 36 Penn Treebank PoS tags and (2) collapsing these tags into four broad categories: N (nouns), V (verbs), A (adjectives and adverbs), and O (others). We experiment with three variations of HMM:

First-Order HMM assuming that the probability of a word depends only on the current tag. Second-Order HMM assuming that the probability of a word depends only on the current tag. First-Order HMM with Context assuming that the probability of a word depends on both the current tag and the previous word.

For handling unseen words, we assign the most frequent tag in the dataset as the default PoS tag. We compute both overall accuracy and tag-wise accuracy for all configurations. Additionally, we compare the performance of the 36-tag model with the 4-tag model and analyze why the latter might outperform the former based on transition and emission probability assumptions.

Through this study, we aim to understand the impact of tag granularity and different HMM configurations on PoS tagging accuracy, providing insights into probabilistic modeling for NLP tasks

Table of Contents

CODE LINK : **B22CS001_B22CS093_B22BB016.ipynb**

1 Introduction

Part-of-Speech (PoS) tagging is one of the foundational tasks in natural language processing (NLP). It involves assigning grammatical categories (tags) to words in a sentence. PoS tagging is crucial for many NLP applications such as syntactic parsing, machine translation, information extraction, and more. The Hidden Markov Model (HMM) is a statistical model widely used for PoS tagging due to its ability to capture sequential dependencies between words and their corresponding tags.

In this project, we implement an HMM-based PoS tagger from scratch using the Viterbi algorithm to determine the most probable sequence of PoS tags for a given input sentence. We evaluate the model using the English Penn Treebank (PTB) corpus, which provides a rich set of PoS tags and annotated text. Our study investigates the impact of different configurations of the HMM on the performance of the PoS tagger, exploring both fine-grained and coarse-grained PoS tags

2 Methodology

2.1 Hidden Markov Model (HMM)

The Hidden Markov Model (HMM) is a probabilistic model that assumes a sequence of hidden states generates the observed data. In the context of PoS tagging, the hidden states correspond to the PoS tags, and the observed data are the words in the sentence.

The model assumes the following:
- *The probability of a word depends on the current PoS tag (Markov property).*

• The probability of a tag depends only on the previous tag (transition probabilities).
• The probability of a word given a tag is called the emission probability.

We implement the HMM-based PoS tagger using three different configurations:
• First-Order HMM: The probability of a word depends only on the current PoS
tag.
• Second-Order HMM: The probability of a word depends on the current and
previous tags.
• First-Order HMM with Context: The probability of a word depends on both the current tag and the previous word in the sentence.

2.2 Viterbi Algorithm

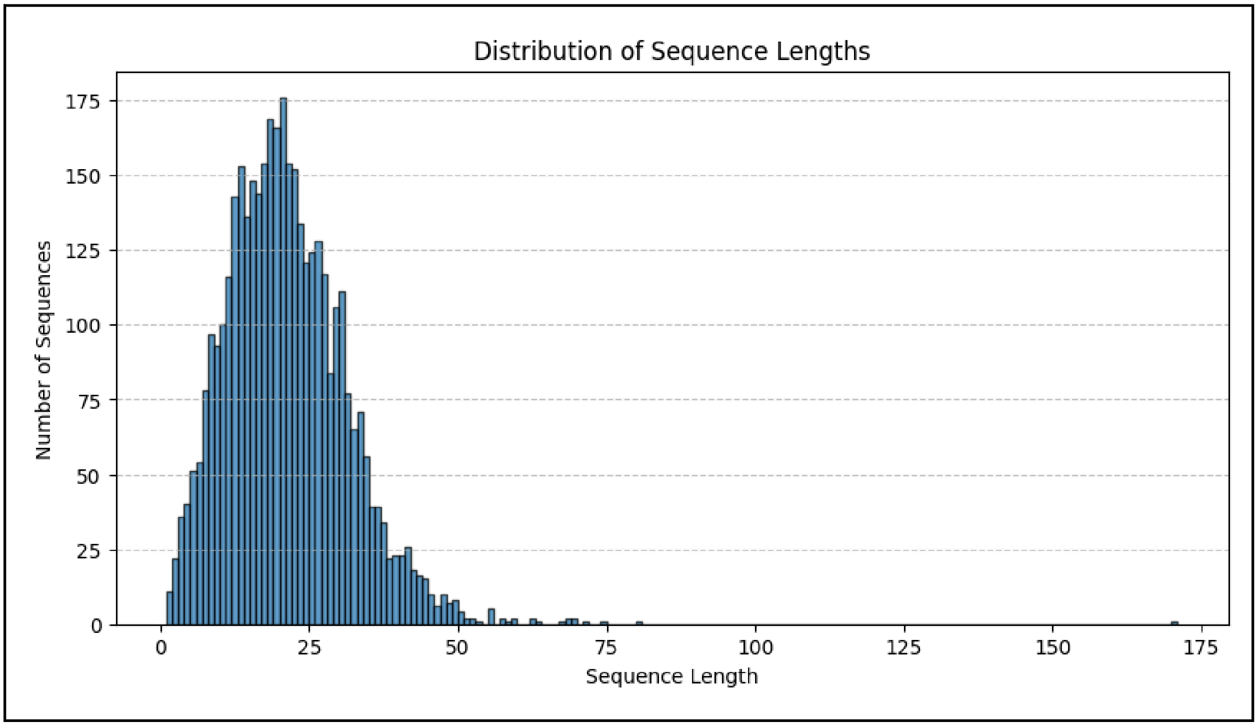
To determine the most probable sequence of PoS tags for a given sentence, we use the Viterbi algorithm. This dynamic programming algorithm efficiently computes the most likely sequence of hidden states (PoS tags) given a sequence of observations (words). The
algorithm maintains a table of probabilities and updates it as it processes each word in the sentence.

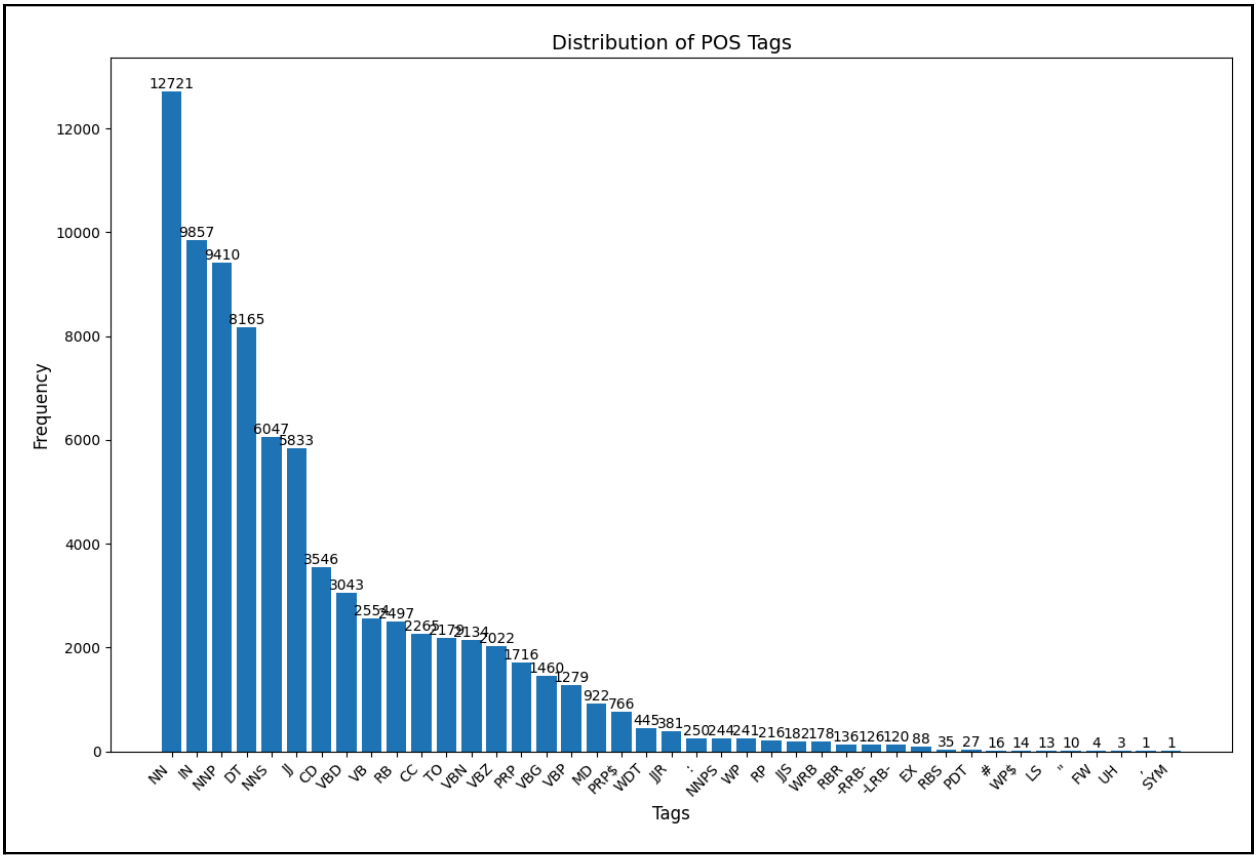
2.3 Data Preprocessing

We use the English Penn Treebank (PTB) corpus, which provides a set of annotated sentences with corresponding PoS tags. The corpus is split into training and test sets in
an 80:20 ratio. For handling unseen words during testing, we assign the most frequent tag in the training set as the default PoS tag.

The dataset is processed as follows:

- *The dataset is loaded from a JSON file (penn-data.json).*
- *Sentences are tokenized into (word, tag) pairs.*
- *The dataset is shuffled and split into 80% training and 20% testing.*
- *Computes sentence length statistics:*
  - *Total sentences.*
  - *Average, max, and min sentence lengths.*
- *Computes vocabulary size (unique words).*
- *Generates tag distribution analysis with a histogram.*





Data Statistics

Total sentences: 3914
Average sentence length: 20.73 words
Max sentence length: 171 words
Min sentence length: 1 words
Vocabulary size: 16458 unique words

Tag Distribution Analysis

Total words: 81147
Total unique tags: 41

3 Experiments and Results

3.1 Tag Granularity

We evaluate the performance of our PoS tagger in two configurations:
- *Using all 36 PoS tags from the Penn Treebank corpus.*
- *Collapsing these 36 tags into four broad categories:* N *(nouns),* V *(verbs),* A *(adjectives/adverbs), and* O *(others).*

3.2 Model Performance

The performance of each model is evaluated in terms of overall accuracy and tag-wise accuracy. We compare the performance of the 36-tag model with the 4-tag model, analyzing the trade-offs between model complexity and performance. We also investigate why the 4-tag model might outperform the 36-tag model due to assumptions in transition and emission probabilities.

3.3 Confusion Matrix and Error Analysis

We compute the confusion matrix to analyze errors in tagging. The most common misclassifications are identified, and error rates for each tag are calculated. The model performance is further analyzed using precision, recall, and F1-score for each PoS tag.

3.4 Visualization of Performance Metrics

We visualize F1-scores, precision, and recall for each tag using bar charts.

3.5 Results and Discussions

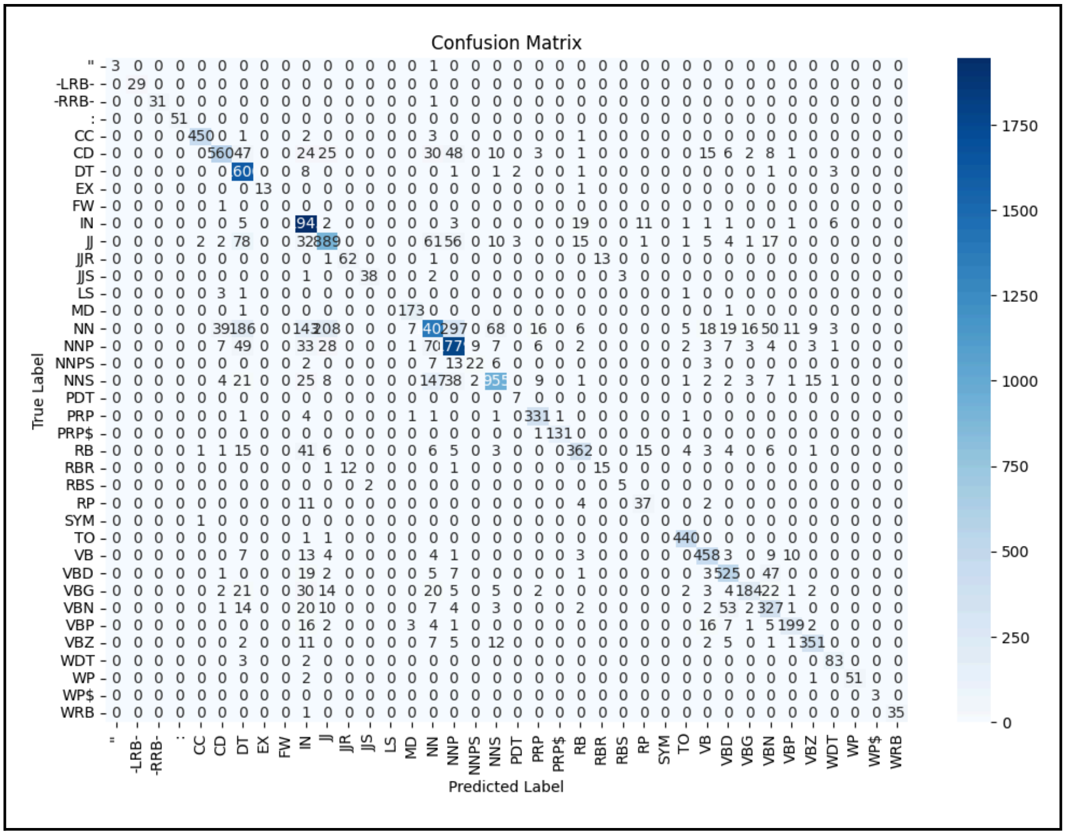
(A) 36 Tag Evaluation

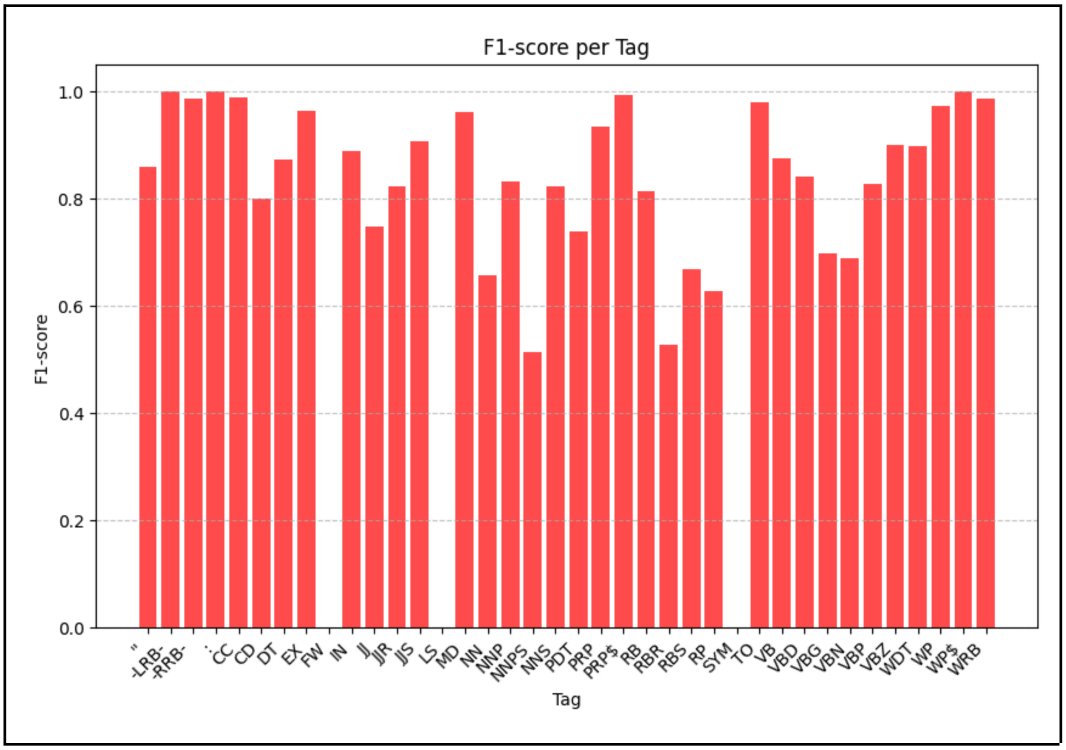
First-Order HMM (36-Tag Set) - Performance Analysis

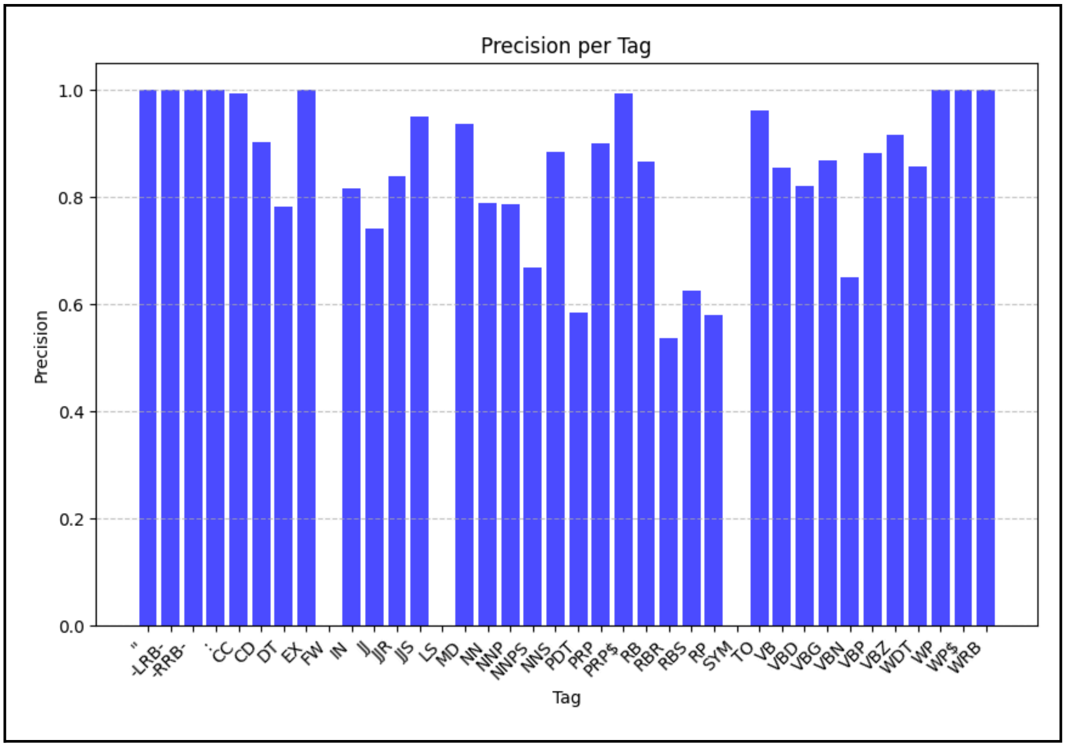
The First-Order Hidden Markov Model (HMM) achieved an overall accuracy of 82.22%, demonstrating its effectiveness in predicting part-of-speech tags based on the current tag. The model's precision (78.83%), recall (77.86%), and F1-score (77.78%) indicate a balanced performance, though there is room for improvement, particularly in handling ambiguous or less frequent tags.

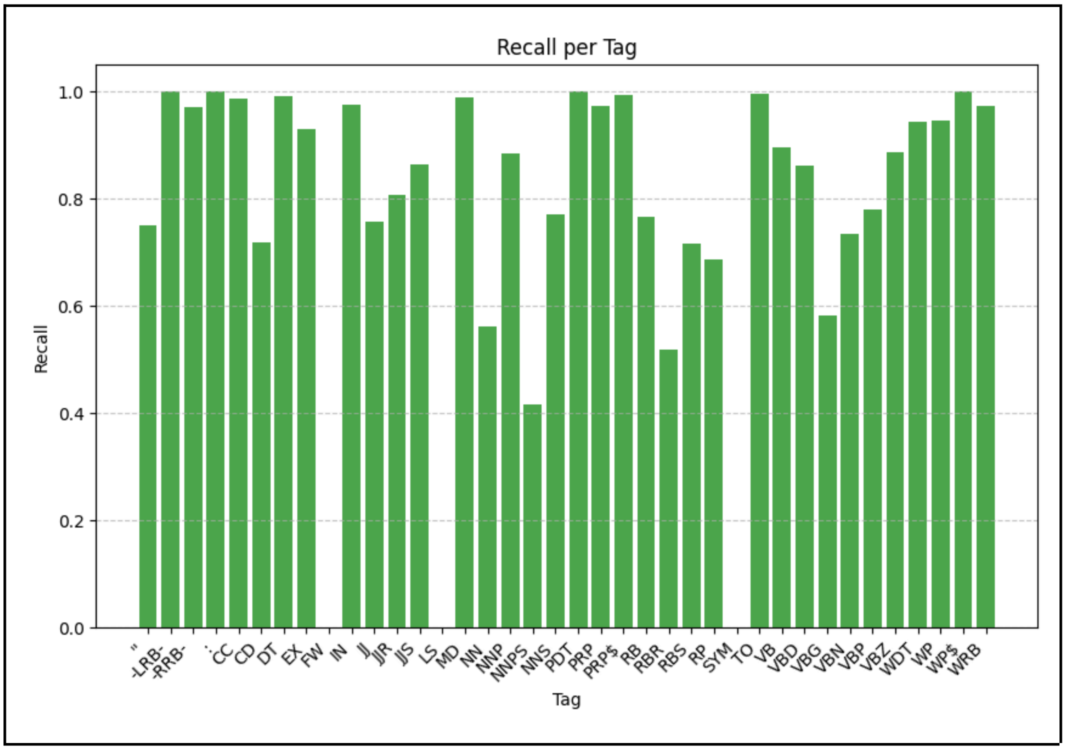
Tag-wise accuracy highlights variations in performance across different categories. The model performed exceptionally well on common function words such as determiners (DT: 98.95%), coordinating conjunctions (CC: 98.47%), and infinitive markers (TO: 99.55%). High accuracy was also observed for proper nouns (NNP: 88.33%), third-person singular verbs (VBZ: 88.41%), and modal verbs (MD: 98.86%). However, performance was notably lower for singular nouns (NN: 56.01%), present participles (VBG: 58.04%), and comparative adverbs (RBR: 51.72%), suggesting that the model struggles with distinguishing certain noun and verb forms.

Certain rare or specialized tags, such as foreign words (FW: 0.0%), list item markers (LS: 0.0%), and symbols (SYM: 0.0%), were not predicted accurately, likely due to their low frequency in the dataset. In contrast, punctuation marks such as parentheses (-LRB- and -RRB-) and colons (:) were predicted with perfect accuracy. The results suggest that while the model is effective in tagging frequent and structurally important words, improvements in handling less common words and ambiguous cases could enhance overall performance.


Confusion Matrix
True Label
Predicted Label
'' -LRB- -RRB- : CC CD DT EX FW IN JJ JJR JJS LS MD NN NNP NNPS NNS PDT PRP PRP$ RB RBR RBS RP SYM TO VB VBD VBG VBN VBP VBZ WDT WP WP$ WRB
0
250
500
750
1000
1250
1500
1750


F1-score per Tag
F1-score
Tag
0.0
0.2
0.4
0.6
0.8
1.0
'' -LRB- -RRB- : CC CD DT EX FW IN JJ JJR JJS LS MD NN NNP NNPS NNS PDT PRP PRP$ RB RBR RBS RP SYM TO VB VBD VBG VBN VBP VBZ WDT WP WP$ WRB





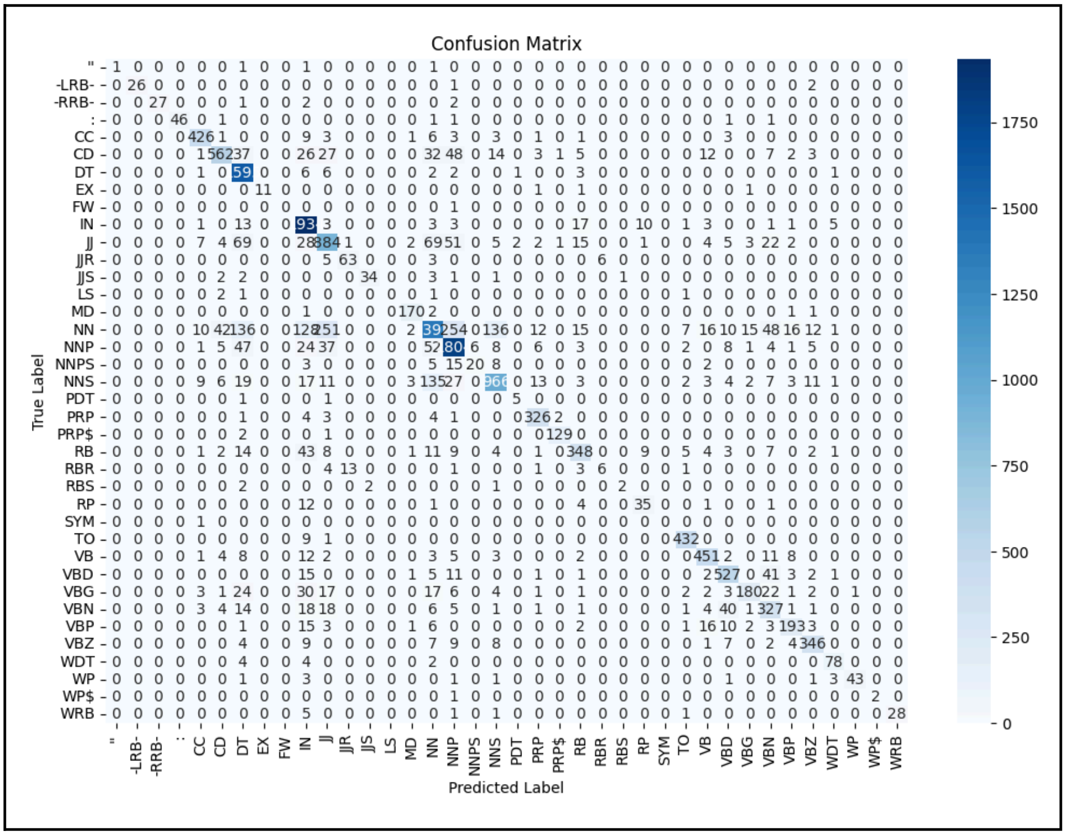
Second-Order HMM (36-Tag Set) - Performance Analysis

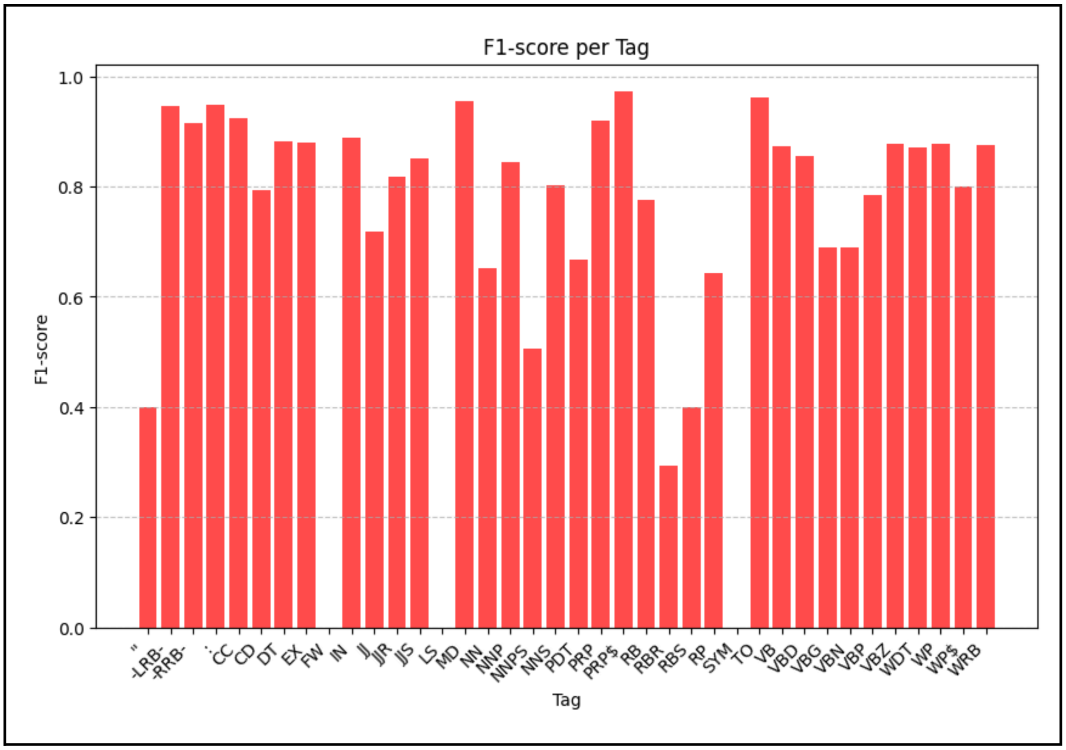
The Second-Order Hidden Markov Model (HMM), which takes into account the previous two tags for predicting the current tag, achieved an overall accuracy of 81.45%. The model's precision (78.46%), recall (69.75%), and F1-score (72.49%) indicate a drop in performance compared to the First-Order HMM, particularly in recall. This suggests that while the model maintains reasonable precision in correctly predicting frequent tags, it struggles more with generalizing across unseen sequences, likely due to the increased complexity introduced by the second-order dependencies.

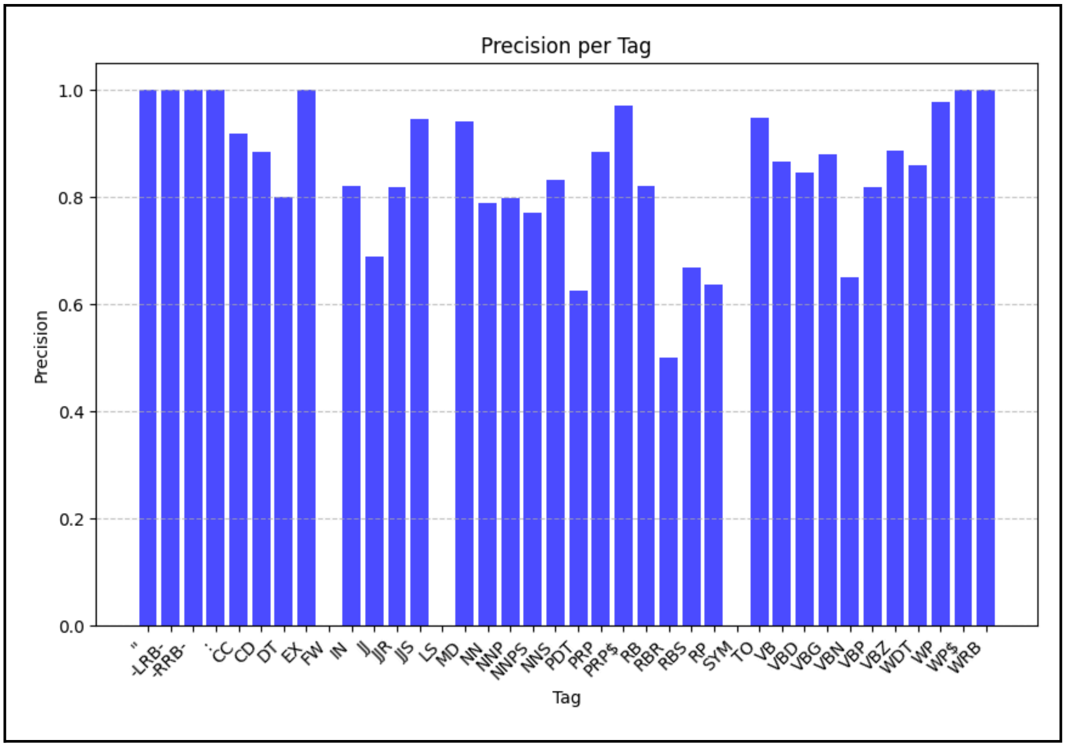
Tag-wise accuracy shows that the model performs well for proper nouns (NNP: 89.57%), infinitive markers (TO: 97.74%), and determiners (DT: 98.64%), indicating strong predictions for function words and structured sentence elements. Additionally, coordinating conjunctions (CC: 93.22%) and modal verbs (MD: 97.14%) were classified with high accuracy, suggesting that the model effectively handles grammatical connectors and auxiliary verbs.

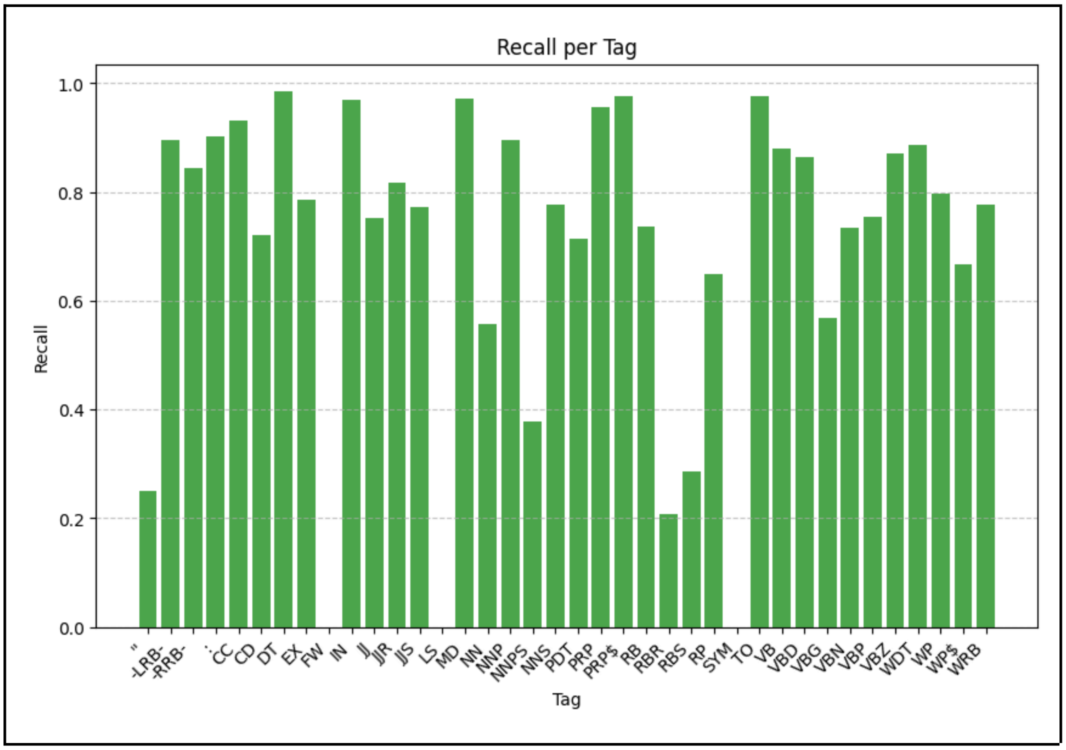
However, performance drops significantly for some categories, particularly comparative adverbs (RBR: 20.69%), superlative adverbs (RBS: 28.57%), and plural proper nouns (NNPS: 37.73%), which are more context-dependent and less frequent in the dataset. The model also struggles with certain verb forms, such as present participles (VBG: 56.78%), as well as infrequent tags like foreign words (FW: 0.0%), list item markers (LS: 0.0%), and symbols (SYM: 0.0%), which received no correct predictions. The lower performance on rare tags suggests that the model's ability to generalize to less frequent words is limited, possibly due to insufficient training data for those categories.

While the Second-Order HMM offers some advantages in modeling dependencies between multiple prior tags, its increased complexity does not necessarily lead to better performance across all tags. The results indicate that the added context is helpful for some categories but introduces greater challenges in recall, particularly for ambiguous or low-frequency words. Further optimizations, such as smoothing techniques or improved handling of rare words, could potentially enhance performance.


Confusion Matrix
True Label
Predicted Label
" - 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-LRB- - 0 26 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0
-RRB- - 0 0 27 0 0 0 1 0 0 2 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
: - 0 0 0 46 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
CC - 0 0 0 0 426 1 0 0 0 9 3 0 0 0 1 6 3 0 3 0 1 0 1 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0
CD - 0 0 0 0 1562 37 0 0 26 27 0 0 0 0 32 48 0 14 0 3 1 5 0 0 0 0 0 12 0 0 7 2 3 0 0 0 0
DT - 0 0 0 0 1 0 59 0 0 6 6 0 0 0 0 2 2 0 0 1 0 0 3 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
EX - 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
FW - 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
IN - 0 0 0 0 1 0 13 0 0 93 3 0 0 0 0 3 3 0 0 0 0 0 17 0 0 10 0 1 3 0 0 1 1 0 5 0 0 0
JJ - 0 0 0 0 7 4 69 0 0 28 384 1 0 0 2 69 51 0 5 2 2 1 15 0 0 1 0 0 4 5 3 22 2 0 0 0 0 0
JJR - 0 0 0 0 0 0 0 0 0 0 5 63 0 0 0 3 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0
JJS - 0 0 0 0 0 2 2 0 0 0 0 0 34 0 0 3 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
LS - 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
MD - 0 0 0 0 0 0 0 0 0 1 0 0 0 0 170 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
NN - 0 0 0 0 10 42 136 0 0 128 251 0 0 0 2 39 254 0 136 0 12 0 15 0 0 0 0 7 16 10 15 48 16 12 1 0 0 0
NNP - 0 0 0 0 1 5 47 0 0 24 37 0 0 0 0 52 80 6 8 0 6 0 3 0 0 0 0 2 0 8 1 4 1 5 0 0 0 0
NNPS - 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 5 15 20 8 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0
NNS - 0 0 0 0 9 6 19 0 0 17 11 0 0 0 3 135 27 0 966 0 13 0 3 0 0 0 0 2 3 4 2 7 3 11 1 0 0 0
PDT - 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PRP - 0 0 0 0 0 0 1 0 0 4 3 0 0 0 0 4 1 0 0 0 326 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PRP$ - 0 0 0 0 0 0 2 0 0 0 1 0 0 0 0 0 0 0 0 0 0 129 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
RB - 0 0 0 0 1 2 14 0 0 43 8 0 0 0 1 11 9 0 4 0 1 0 348 0 0 9 0 5 4 3 0 7 0 2 1 0 0 0
RBR - 0 0 0 0 0 0 0 0 0 0 4 13 0 0 0 0 1 0 0 0 1 0 3 6 0 0 0 1 0 0 0 0 0 0 0 0 0 0
RBS - 0 0 0 0 0 0 2 0 0 0 0 0 2 0 0 0 0 0 1 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0
RP - 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 1 0 0 0 0 0 0 4 0 0 35 0 0 1 0 0 1 0 0 0 0 0 0
SYM - 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
TO - 0 0 0 0 0 0 0 0 0 9 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 432 0 0 0 0 0 0 0 0 0 0
VB - 0 0 0 0 1 4 8 0 0 12 2 0 0 0 0 3 5 0 3 0 0 0 2 0 0 0 0 0 451 2 0 11 8 0 0 0 0 0
VBD - 0 0 0 0 0 0 0 0 0 15 0 0 0 0 1 5 11 0 0 0 1 0 1 0 0 0 0 0 2 527 0 41 3 2 1 0 0 0
VBG - 0 0 0 0 3 1 24 0 0 30 17 0 0 0 0 17 6 0 4 0 1 0 1 0 0 0 0 2 2 3 180 22 1 2 0 1 0 0
VBN - 0 0 0 0 3 4 14 0 0 18 18 0 0 0 0 6 5 0 1 0 1 0 1 0 0 0 0 1 4 40 1 327 1 1 0 0 0 0
VBP - 0 0 0 0 0 0 1 0 0 15 3 0 0 0 1 6 0 0 0 0 0 0 2 0 0 0 0 1 16 10 2 3 193 3 0 0 0 0
VBZ - 0 0 0 0 0 0 4 0 0 9 0 0 0 0 0 7 9 0 8 0 0 0 0 0 0 0 0 0 1 7 0 2 4 346 0 0 0 0
WDT - 0 0 0 0 0 0 4 0 0 4 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 78 0 0 0
WP - 0 0 0 0 0 0 1 0 0 3 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 3 43 0 0
WP$ - 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0
WRB - 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 28
"
-LRB-
-RRB-
:
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
LS
MD
NN
NNP
NNPS
NNS
PDT
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB
1750
1500
1250
1000
750
500
250
0


F1-score per Tag
F1-score
Tag
1.0
0.8
0.6
0.4
0.2
0.0
"
-LRB-
-RRB-
:
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
LS
MD
NN
NNP
NNPS
NNS
PDT
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB


Precision per Tag
Precision
1.0
0.8
0.6
0.4
0.2
0.0
Tag
''
-LRB-
-RRB-
.
:
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
LS
MD
NN
NNP
NNPS
NNS
PDT
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB


Recall per Tag
Recall
1.0
0.8
0.6
0.4
0.2
0.0
Tag
''
-LRB-
-RRB-
.
:
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
LS
MD
NN
NNP
NNPS
NNS
PDT
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB

First-Order HMM with Previous Word Dependency (36-Tag Set) - Performance Analysis

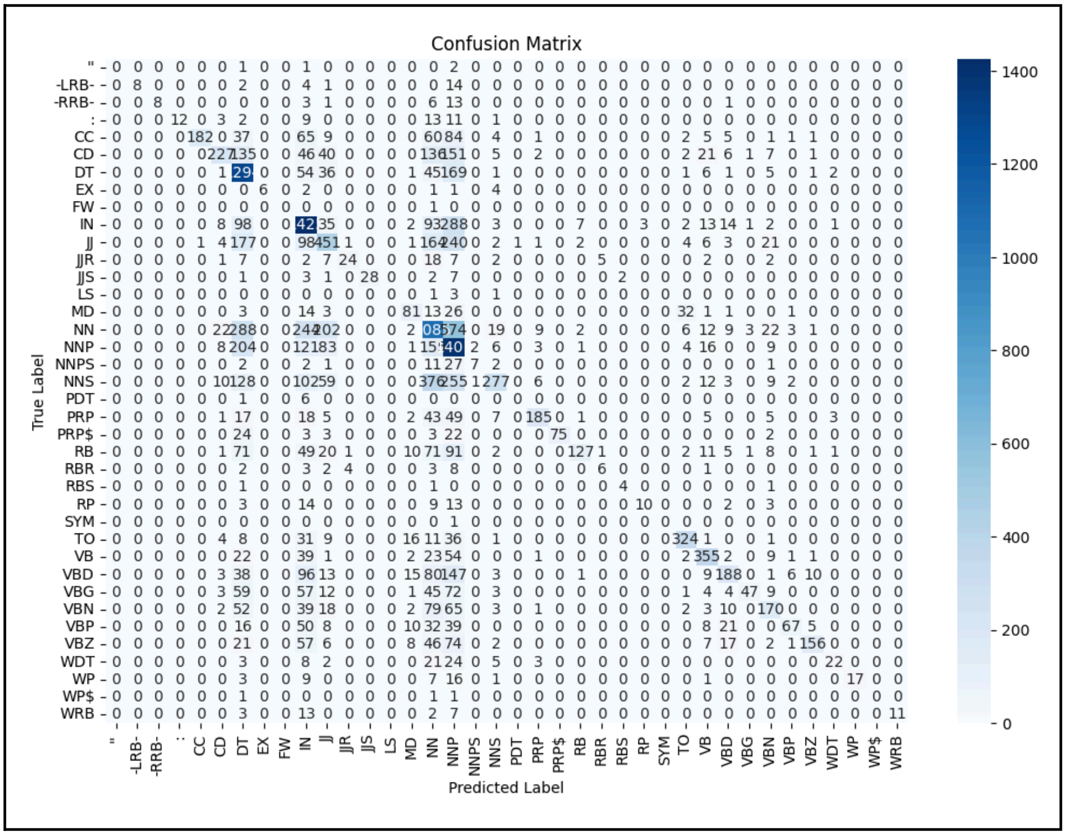
The First-Order Hidden Markov Model (HMM) *with Previous Word Dependency, which assumes that the probability of a word depends on both the current Part-of-Speech* (PoS) *tag and the preceding word, achieves an overall accuracy of* 50.26%. *The model's precision* (64.74%) *is relatively high, but its recall* (33.69%) *and F1-score* (40.76%) *are significantly lower, indicating that while it correctly identifies some tags with high confidence, it struggles to generalize effectively. The sharp decline in recall suggests that the model misses many correct predictions, leading to poor overall tagging performance.*

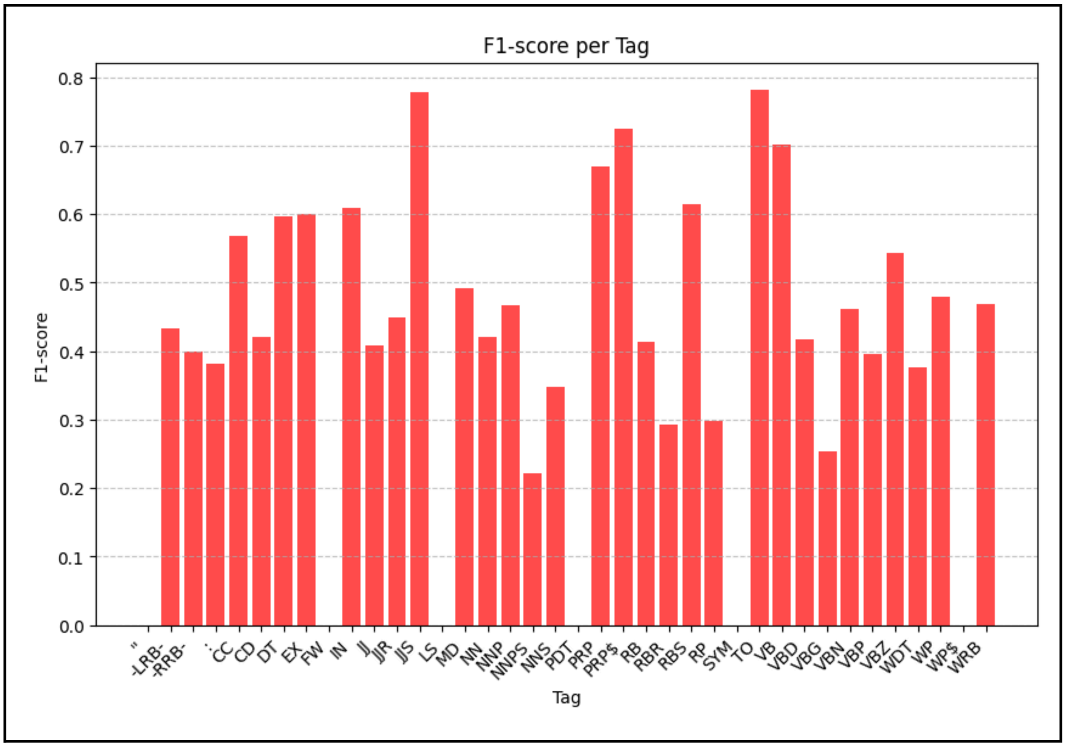
Examining tag-wise accuracy reveals that certain word categories perform reasonably well, such as proper nouns (NNP: 69.56%), *base verbs* (VB: 69.34%), *and determiners* (DT: 80.02%), *which are more predictable based on previous word dependencies. However, other crucial categories, like plural nouns* (NNS: 22.30%), *prepositions* (IN: 71.42%), *and coordinating conjunctions* (CC: 39.82%), *show reduced accuracy, highlighting the model's difficulty in consistently identifying function words.*

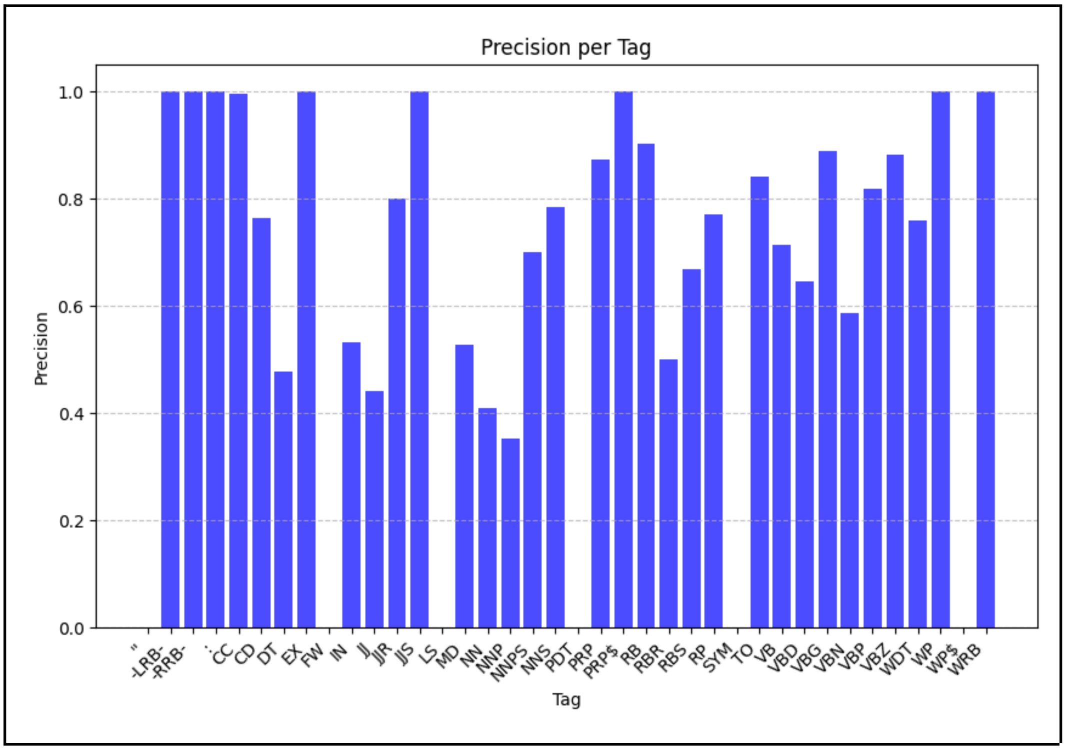
A *notable issue is the poor performance on low-frequency and context-sensitive tags, such as comparative adjectives* (JJR: 31.16%), *modal verbs* (MD: 46.28%), *and relative pronouns* (WDT: 25.00%). *Furthermore, several tags, including foreign words* (FW: 0.0%), *list item markers* (LS: 0.0%), *and symbols* (SYM: 0.0%), *received no correct predictions, demonstrating the model's struggle with rare words and specialized categories.*

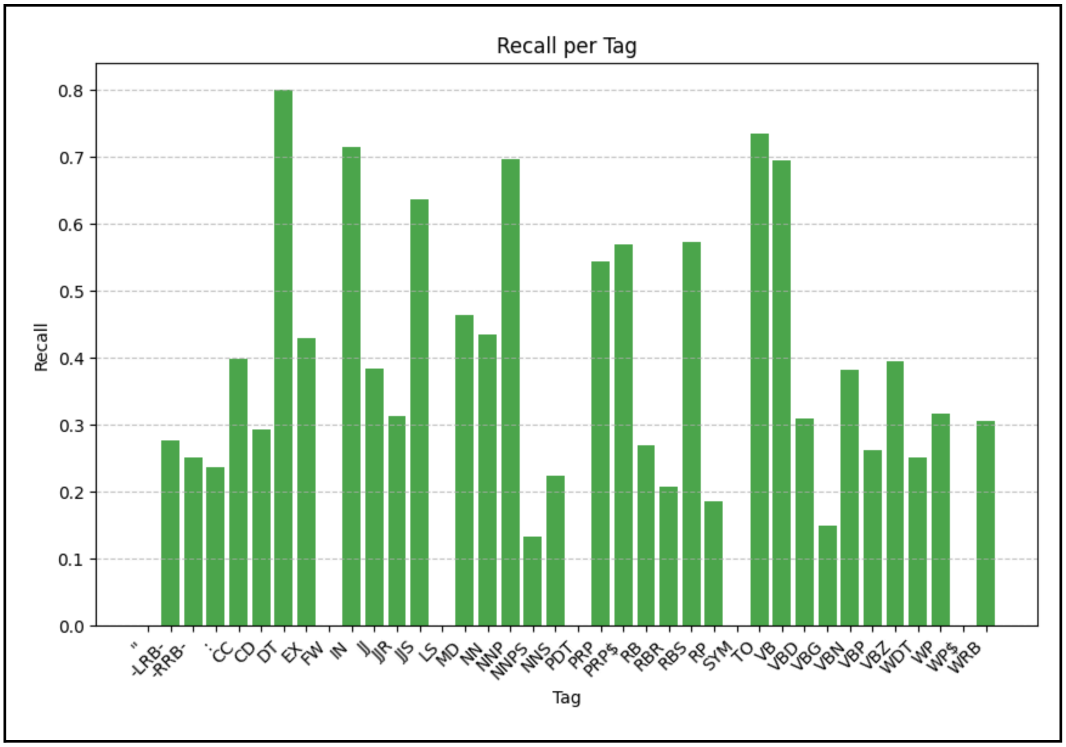
This drastic drop in accuracy compared to other HMM variations suggests that relying solely on the previous word is not sufficient for accurate PoS tagging. Many PoS categories require broader contextual information, which this model fails to capture effectively. The high uncertainty in rare words and structurally ambiguous cases leads to a significantly weaker recall.

While this model provides a different perspective on word dependencies in PoS tagging, its limited contextual awareness results in increased misclassification *rates and lower overall performance. Future improvements could involve backoff strategies, smoothing techniques, or additional contextual features to better handle ambiguous and infrequent tags.*




F1-score per Tag
F1-score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Tag
''
-LRB-
-RRB-
:
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
LS
MD
NN
NNP
NNPS
NNS
PDT
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB


Precision per Tag
Precision
0.0
0.2
0.4
0.6
0.8
1.0
Tag
''
-LRB-
-RRB-
:
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
LS
MD
NN
NNP
NNPS
NNS
PDT
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB



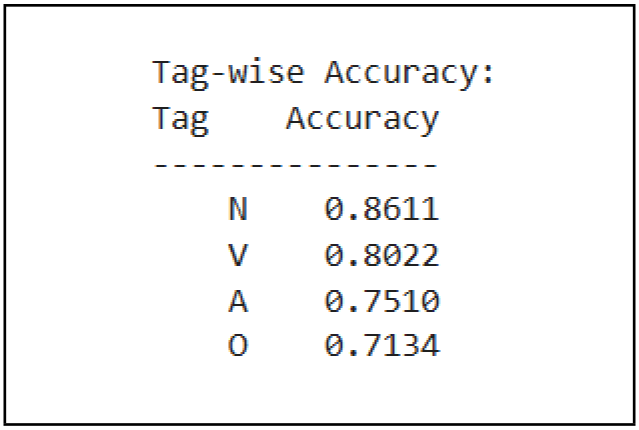
(B)4 Tag Evaluation

First-Order HMM (4-Tag Set) - Performance Analysis

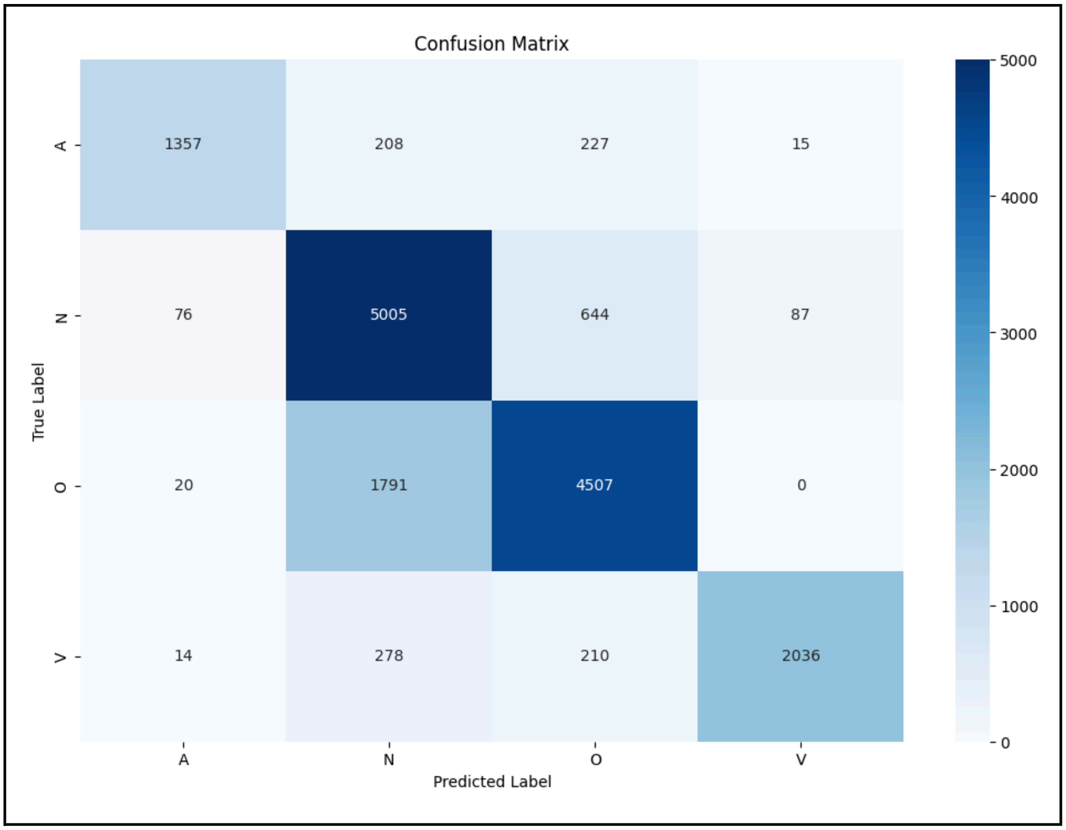
The First-Order Hidden Markov Model (HMM) with a 4-tag set achieves an overall accuracy of 78.33%, demonstrating a solid ability to classify words into broad Part-of-Speech (PoS) categories. The model attains a precision of 84.28%, indicating a high level of confidence in its predictions. Additionally, its recall of 78.19% and F1-score of 80.53% suggest a good balance between identifying correct tags and minimizing false positives.

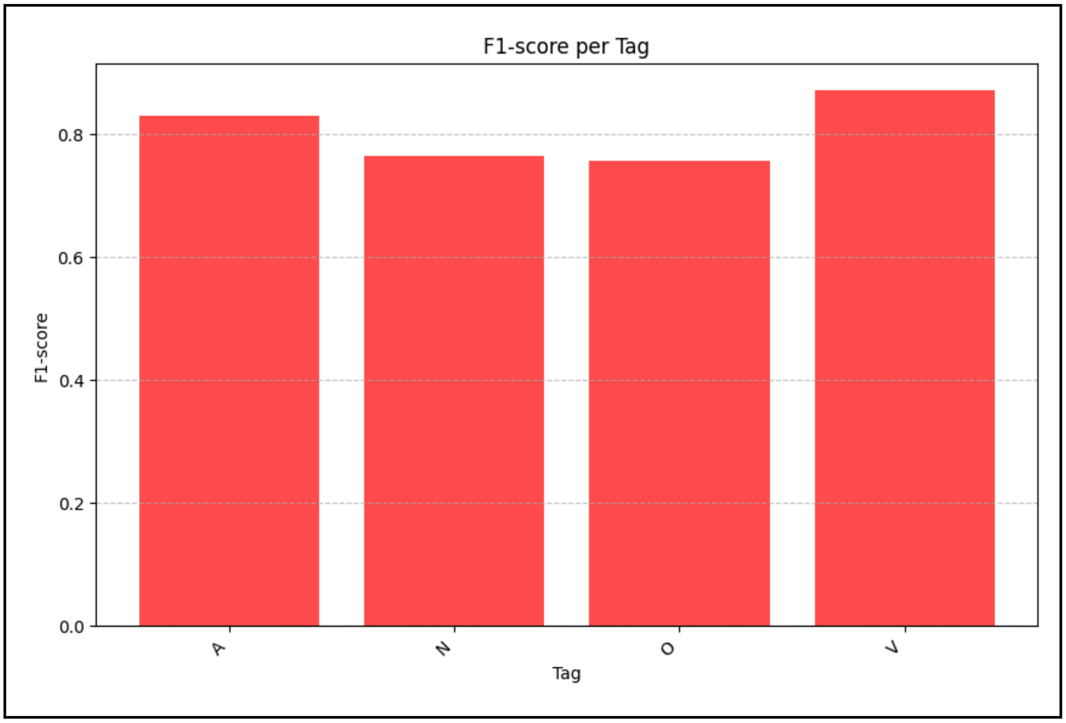
Breaking down performance by category, the noun (N) tag achieves the highest accuracy at 86.11%, indicating that nouns are relatively easy to classify, likely due to their distinct syntactic patterns. Verbs (V) follow with an accuracy of 80.22%, showing that verb recognition is strong but still prone to occasional misclassification, especially for ambiguous forms. Adjectives (A) have an *accuracy of 75.10%, suggesting moderate difficulty in distinguishing descriptive words, particularly when they overlap with other word classes. The other (O) category, which includes function words and less common PoS tags, has the lowest accuracy at 71.34%, reflecting the challenge of correctly classifying auxiliary words, conjunctions, and other grammatical elements.*

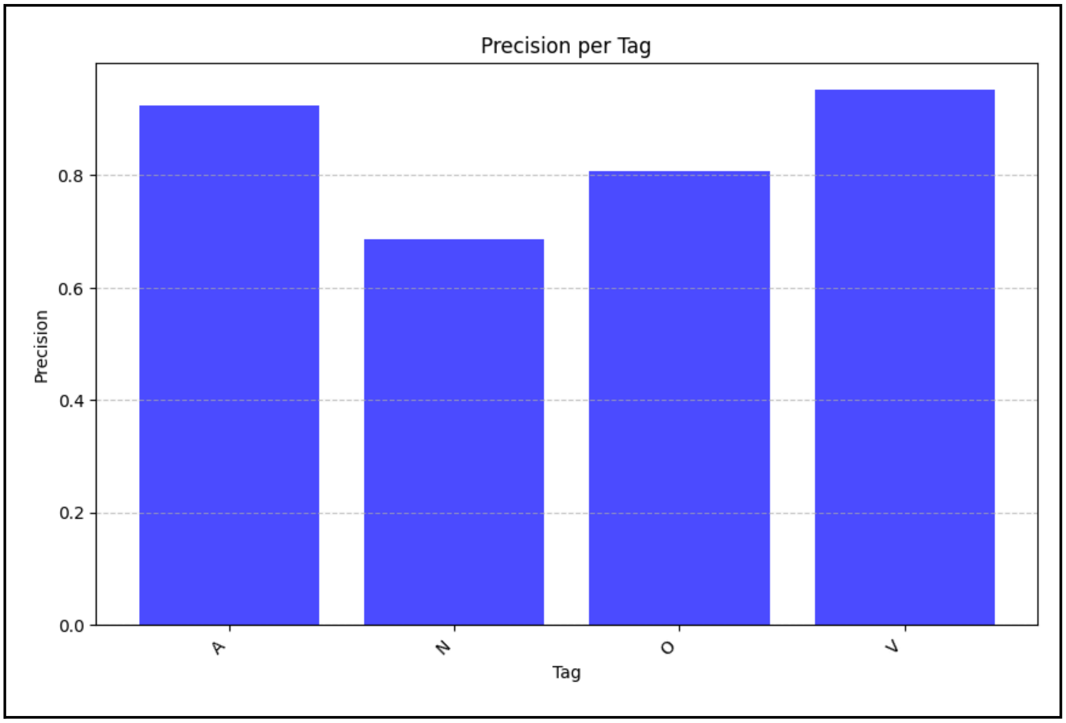
While this First-Order HMM performs well, certain limitations remain, particularly in capturing long-range dependencies and handling ambiguous cases. Future improvements could involve leveraging second-order HMMs or incorporating additional linguistic features to enhance classification accuracy across all categories.

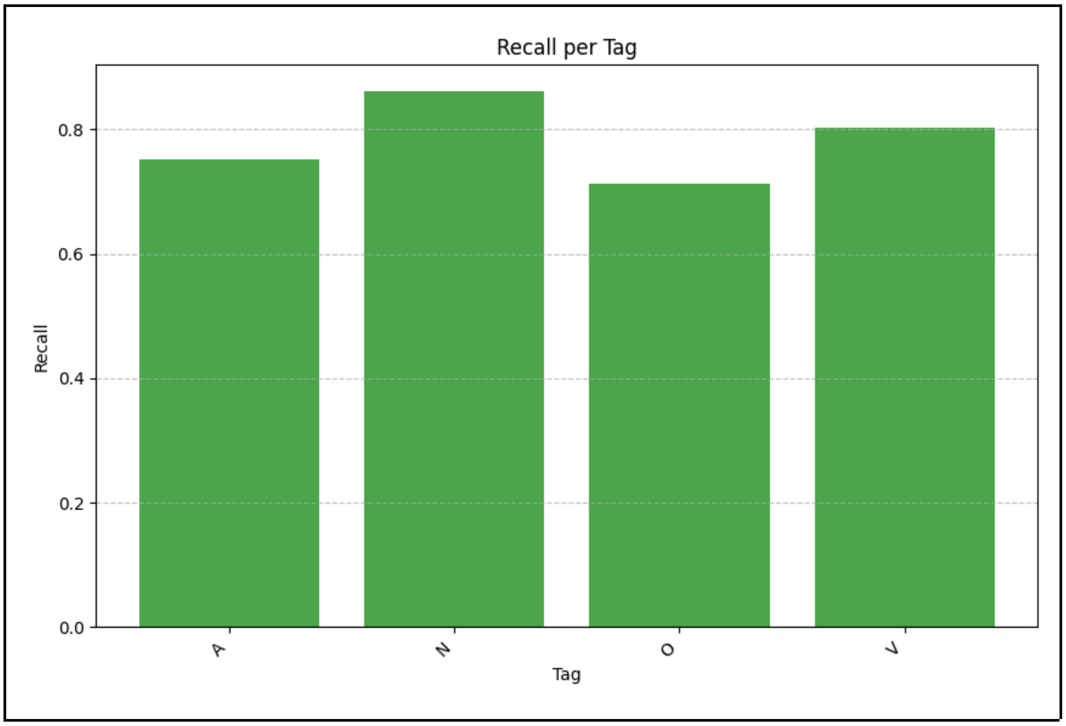


```
Tag-wise Accuracy:
Tag    Accuracy
---------------
    N    0.8611
    V    0.8022
    A    0.7510
    O    0.7134
```


Confusion Matrix
True Label
Predicted Label
A
N
O
V
1357
208
227
15
76
5005
644
87
20
1791
4507
0
14
278
210
2036
A
N
O
V
5000
4000
3000
2000
1000
0


F1-score per Tag
F1-score
0.8
0.6
0.4
0.2
0.0
A
N
O
V
Tag



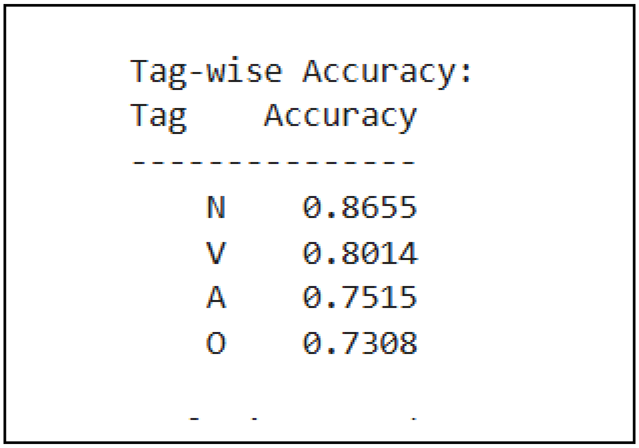


Second-Order HMM (4-Tag Set) - Performance Analysis

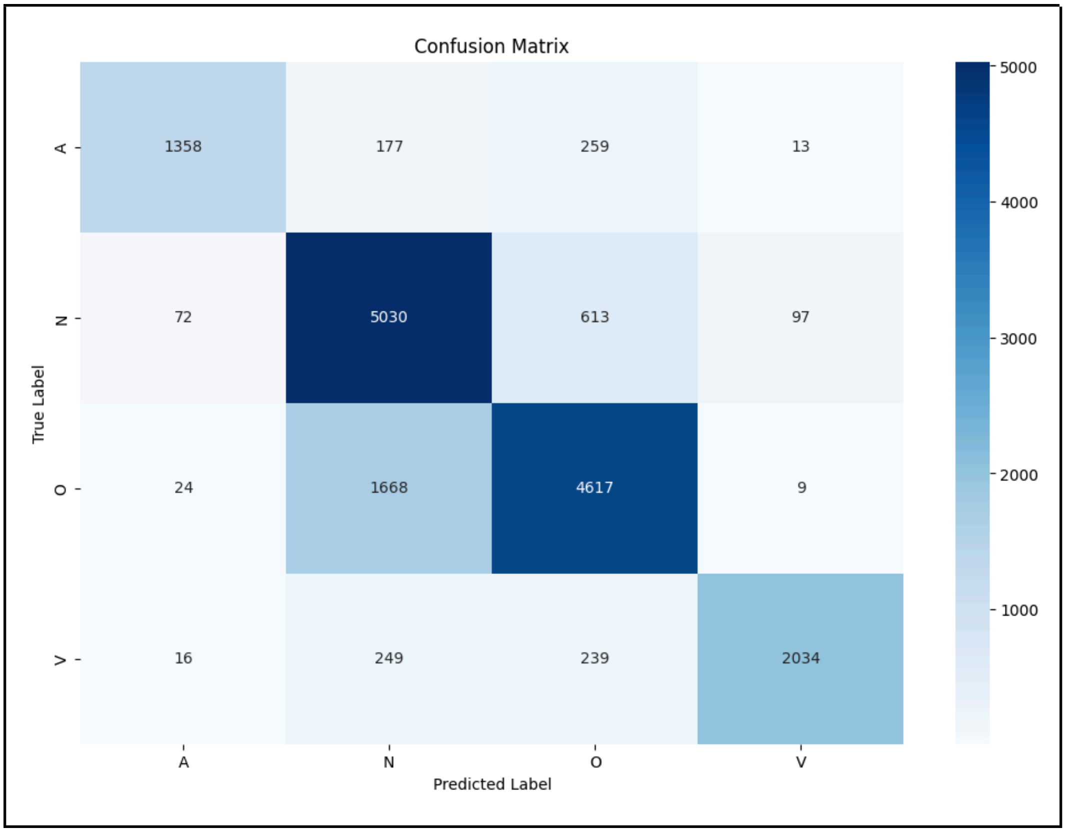
The Second-Order Hidden Markov Model (HMM) with a 4-tag set achieves an overall accuracy of 79.14%, showing a slight improvement over the First-Order HMM. This model, which takes into account dependencies between two preceding tags, refines predictions by incorporating a broader context. It attains a precision of 84.52%, demonstrating a high level of confidence in the assigned PoS tags. The recall of 78.73% suggests that it effectively captures relevant instances of each tag, and the F1-score of 81.01% confirms a well-balanced trade-off between precision and recall.

Examining tag-wise performance, nouns (N) achieve the highest accuracy at 86.55%, reinforcing the idea that nouns are relatively easier to classify due to their syntactic stability. Verbs (V) maintain an accuracy of 80.14%, suggesting that while contextual dependencies help, challenges remain in distinguishing between verb forms and auxiliary verbs. Adjectives (A) show an accuracy of 75.15%, indicating that while the second-order model slightly refines classification, adjectives still pose a moderate challenge. The other (O) category improves to 73.08% accuracy, showing that considering an additional previous tag helps reduce misclassification in function words and miscellaneous categories.

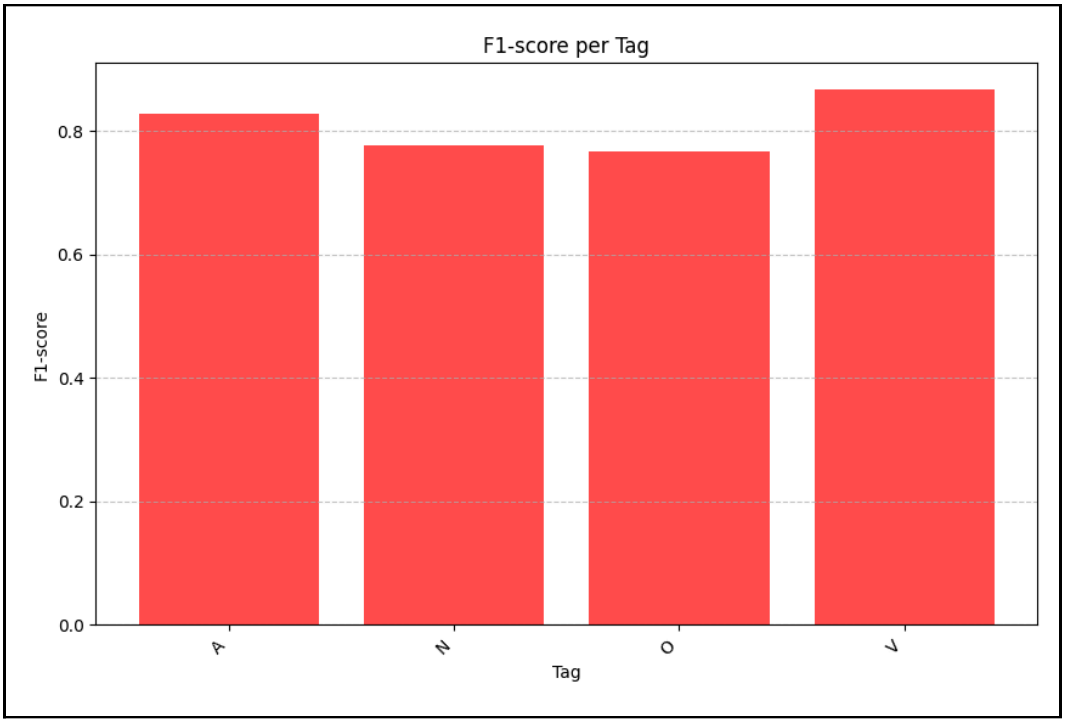
The Second-Order HMM benefits from better sequence modeling, leading to improved accuracy and more refined tag predictions, especially in ambiguous cases. However, while it outperforms the First-Order HMM, further enhancements—such as integrating lexical probabilities or neural network-based language models—could further boost classification performance, particularly for challenging word categories.

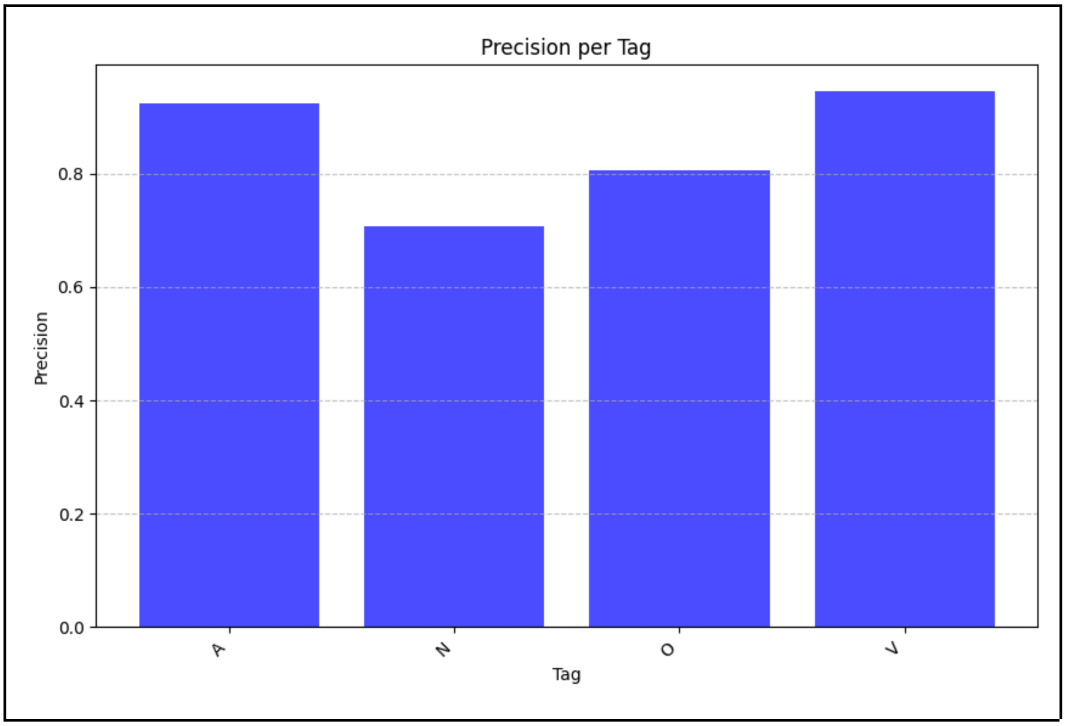


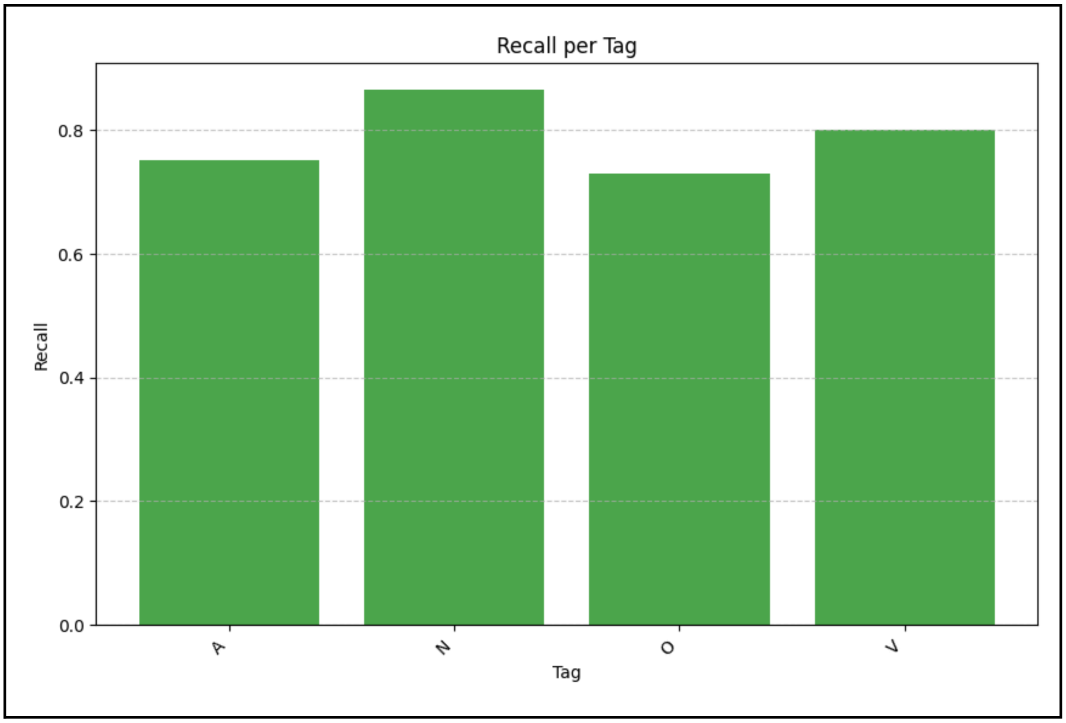
```
Tag-wise Accuracy:
Tag    Accuracy
----------------
    N    0.8655
    V    0.8014
    A    0.7515
    O    0.7308
```



	A	N	O	V
A	1358	177	259	13
N	72	5030	613	97
O	24	1668	4617	9
V	16	249	239	2034


F1-score per Tag
F1-score
0.0
0.2
0.4
0.6
0.8
A
N
O
V
Tag


Precision per Tag
Precision
0.0
0.2
0.4
0.6
0.8
A
N
O
V
Tag

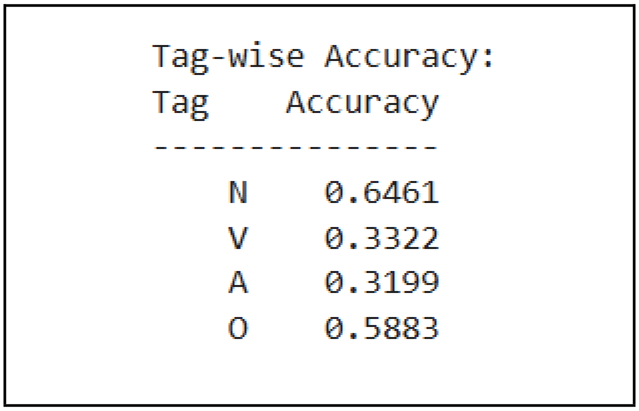


First-Order HMM with Previous Word Dependency (4-Tag Set) - Performance Analysis

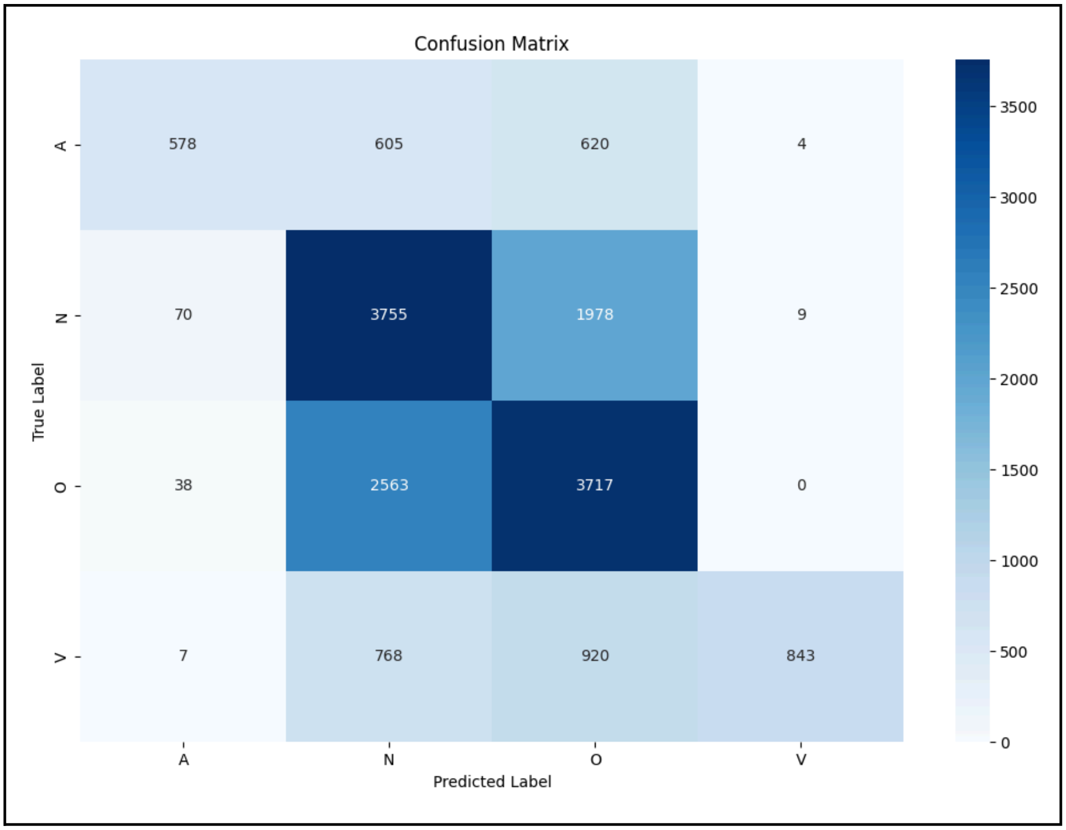
The First-Order Hidden Markov Model (HMM) with previous word dependency achieves an overall accuracy of 53.98%, significantly lower than the standard First-Order and Second-Order HMMs. This suggests that relying on the previous word instead of the previous tag introduces additional ambiguity, making predictions less reliable. Despite this, the model still achieves a precision of 70.52%, indicating that when it assigns a tag, it is fairly confident in its choice. However, the recall of 47.16% highlights its struggle to correctly capture all relevant instances, leading to an F1-score of 51.60%, reflecting an imbalanced trade-off between precision and recall.

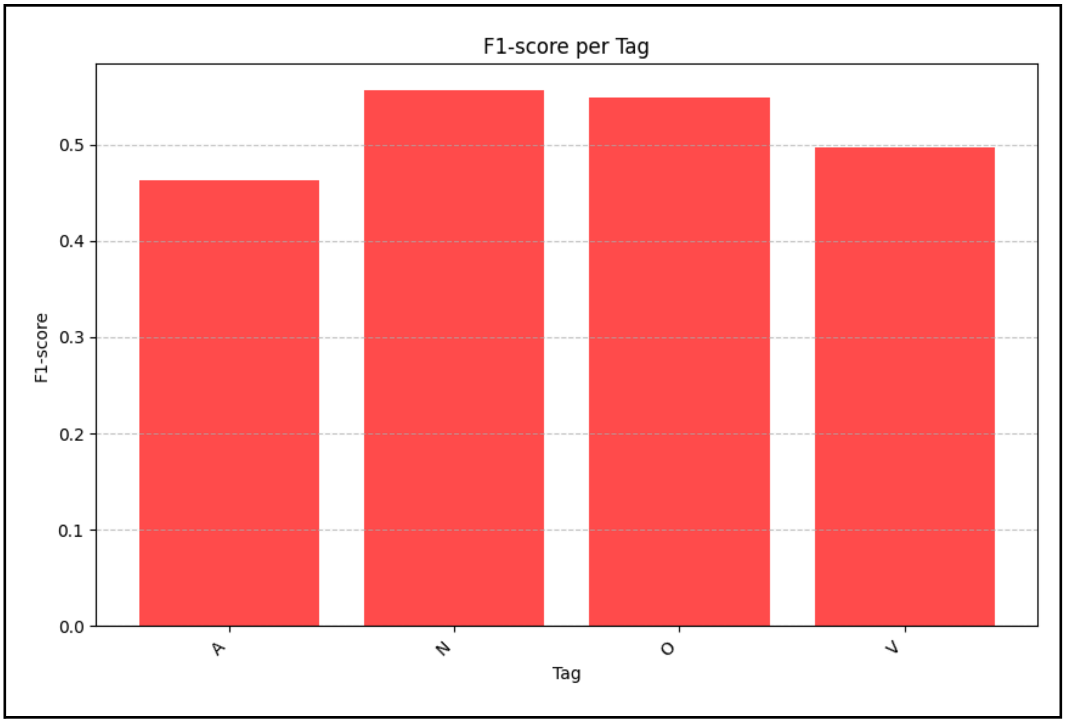
Looking at tag-wise performance, nouns (N) have the highest accuracy at 64.61%, suggesting that previous-word dependency helps in noun identification to some extent. However, verbs (V) perform poorly at just 33.22% accuracy, *likely due to the complex syntactic variations in verb usage that are not well captured by relying solely on the previous word. Adjectives (A) also suffer, with an accuracy of 31.99%, showing that descriptive words are difficult to classify without considering grammatical structure. The other (O) category achieves 58.83% accuracy, indicating moderate success in tagging function words, but still lagging behind models that use tag dependencies instead of word dependencies.*

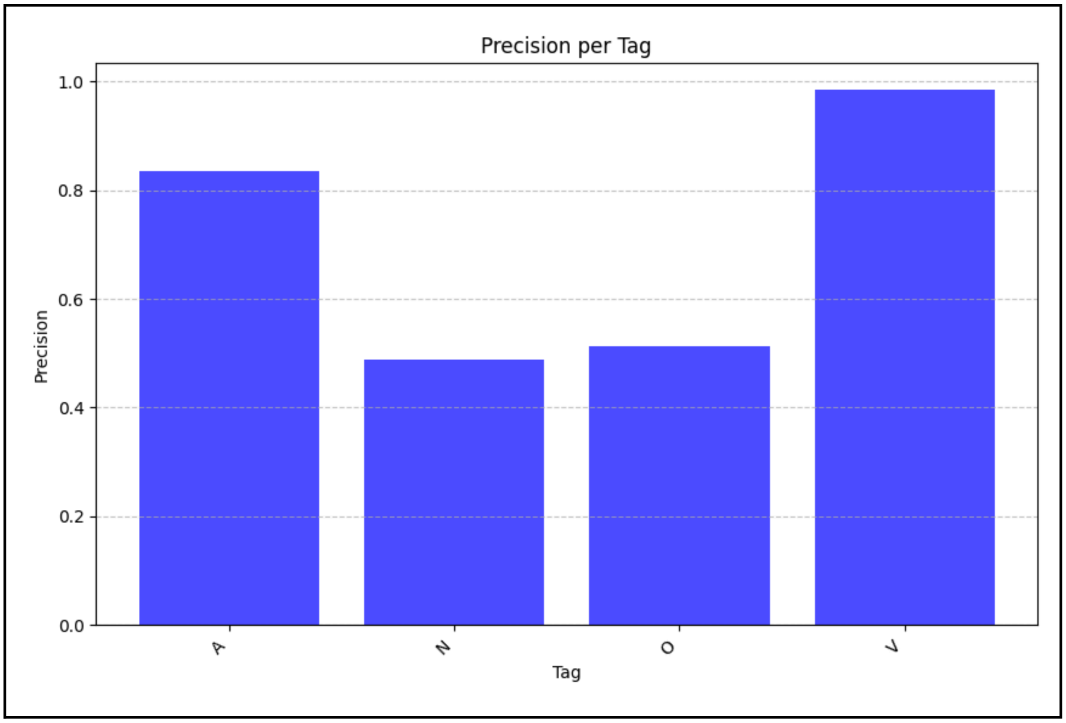
This model highlights the limitations of word-based dependencies in Part-of-Speech (PoS) tagging, as words alone do not provide enough syntactic and grammatical context for accurate classification. While it captures some patterns, tagging based on previous tags proves to be a more effective approach. Future improvements could involve hybrid models combining both word and tag dependencies or integrating neural network-based embeddings to better capture context.

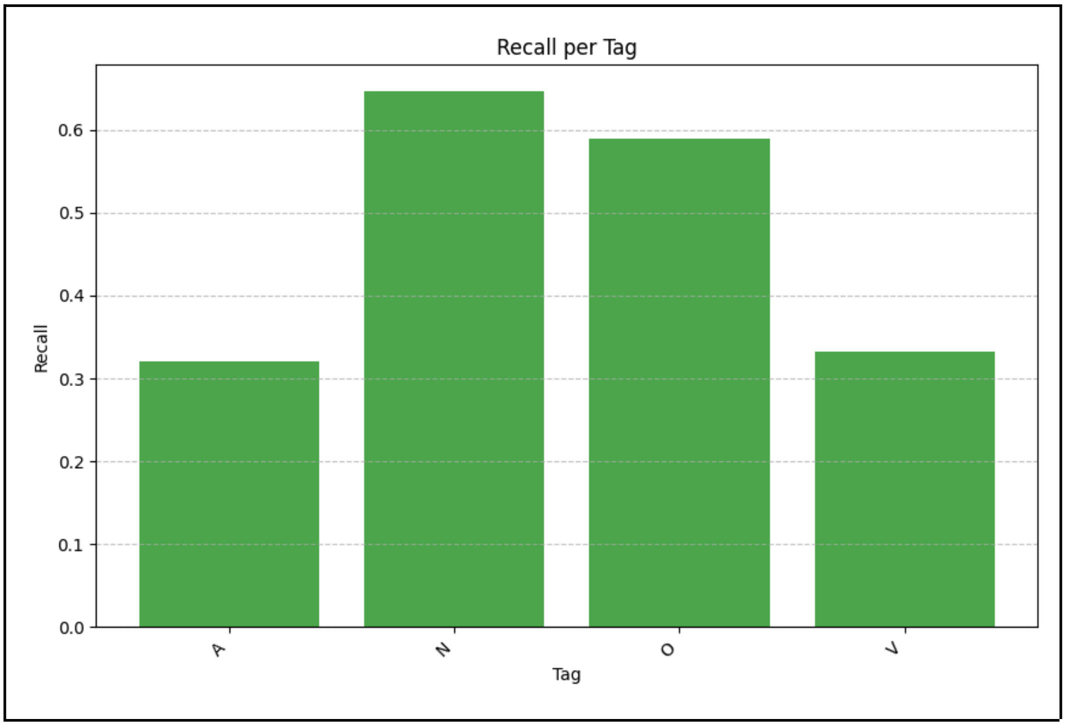


```
Tag-wise Accuracy:
Tag    Accuracy
---------------
    N    0.6461
    V    0.3322
    A    0.3199
    O    0.5883
```


Confusion Matrix
True Label
Predicted Label
A
N
O
V
A
N
O
V
578
605
620
4
70
3755
1978
9
38
2563
3717
0
7
768
920
843
3500
3000
2500
2000
1500
1000
500
0


F1-score per Tag
F1-score
Tag
0.0
0.1
0.2
0.3
0.4
0.5
A
N
O
V


Precision per Tag
Precision
1.0
0.8
0.6
0.4
0.2
0.0
A
N
O
V
Tag


Recall per Tag
Recall
0.6
0.5
0.4
0.3
0.2
0.1
0.0
A
N
O
V
Tag

Conclusion

In this study, we explored the use of the Hidden Markov Model (HMM) for Part-of-Speech (PoS) tagging, with a focus on evaluating different HMM configurations. Our results show that collapsing the 36 PoS tags into 4 broad categories improves performance, highlighting the importance of tag granularity in model efficiency. We also demonstrated the effectiveness of the Viterbi algorithm in determining the most probable sequence of PoS tags.

As observed from the results, the accuracy for the 4-tag HMM models is significantly higher than that of the 36-tag HMM models. The 4-tag model often outperforms the 36-tag model due to several key factors related to data sparsity, probability estimation, and model complexity:

A. *Reduced Data Sparsity*: *Merging similar tags in the 4-tag model increases the amount of data available per tag. This leads to more reliable probability estimates and reduces zero-probability issues that arise in the 36-tag model.*

B. *Smoother Transition Probabilities*: *The 4-tag model has only 16 possible transitions (4×4), leading to smoother probability distributions and better generalization. In contrast, the 36-tag model has 1296 transitions (36×36), making it harder to estimate probabilities accurately and leading to noisier results.*

C. *Simpler Decision Boundaries*: *The 4-tag model makes tagging decisions at a higher level of abstraction, reducing confusion between similar tags (e.g., "NN" vs. "NNP"). On the other hand, the 36-tag model includes finer distinctions, which increases the likelihood of misclassification.*

D. *Improved Robustness to Data Variability*: *With fewer tag categories, the 4-tag model generalizes better across different text domains and corpora. The 36-tag model, being more granular, is more sensitive to domain-specific variations, making it less adaptable to diverse datasets.*

Regarding the interpretation of our findings, we observed that, similar to the first-order model, incorporating previous word dependencies led to reduced accuracy for most tags. This suggests that adding word-level context was not beneficial in this reduced-tag scenario, likely due to overfitting or the model's *inability to effectively capture the complex relationships between words and tags.*

Overall, this work provides valuable insights into the application of probabilistic models for PoS tagging and lays the groundwork for future research aimed at enhancing the accuracy and efficiency of NLP tasks.