Part-of-Speech Tagging Using HMMs

COLAB FILE : B21ES001_B21ES003_B21ES017_B21AI010.ipynb

Course: Natural Language Understanding
Instructor: Dr. Asif Ekbal
Group Members:

- Abhijeet Singh Naruka
- Aditya Anand
- Raghav Vijayvergia
- Chauhan Shashank Pravinbhai

1. Abstract

This project uses three types of Hidden Markov Models (HMMs) for Part-of-Speech (PoS) tagging on the Penn Treebank corpus. We apply an 80:20 train/test split and test:

- **First Order HMM** (using **P(w∣t)P(w|t)**)
- **Second Order HMM** (using **P(ti∣ti−2,ti−1)P(t_i|t_{i-2}, t_{i-1})**)
- **First Order HMM with Previous Word Dependency** (using **P(wi∣ti,wi−1)P(w_i|t_i, w_{i-1})**)

We evaluate these on both the original **36-tag dataset** as well as a collapsed version of **4 categories (N, V, A, O)**. The results indicate that the **First Order and Second Order models have high overall accuracy (about 85%)** in both setups, whereas the **model with prior word dependency is much lower (about 52–56%).** We explain why these results are produced and recommend potential avenues for improvement.

2. Introduction

- **Background:** PoS tagging is crucial for a variety of NLP applications. HMMs are a traditional method for tagging because they are simple.

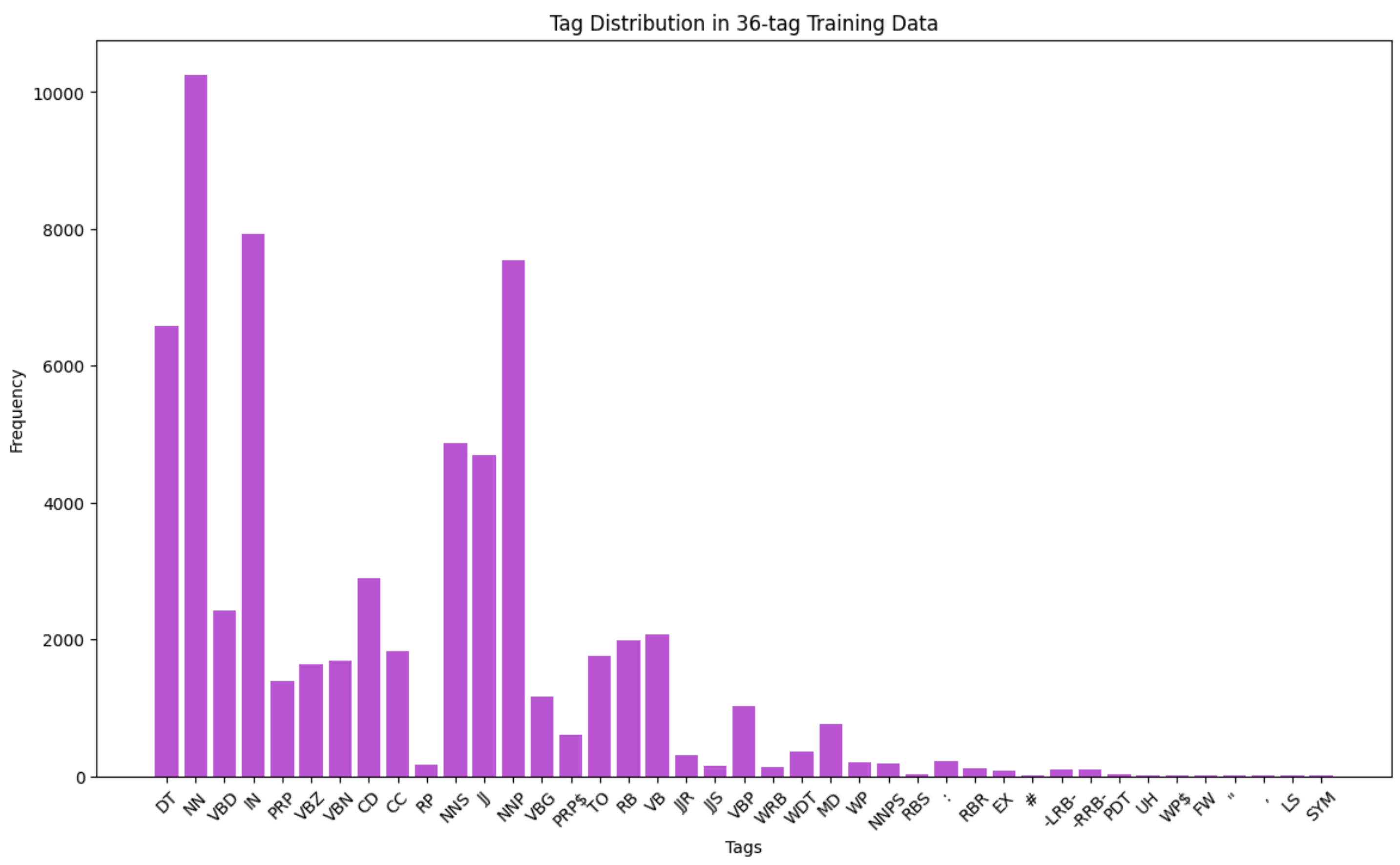
- **Objective:** Implement HMMs from scratch (not using libraries) and contrast varied model configurations and tag granularities.
- **Why Compare 36 vs. 4 Tags? :** Collapsed tags can decrease data sparsity and may enhance parameter estimates. We examine whether this improves performance.

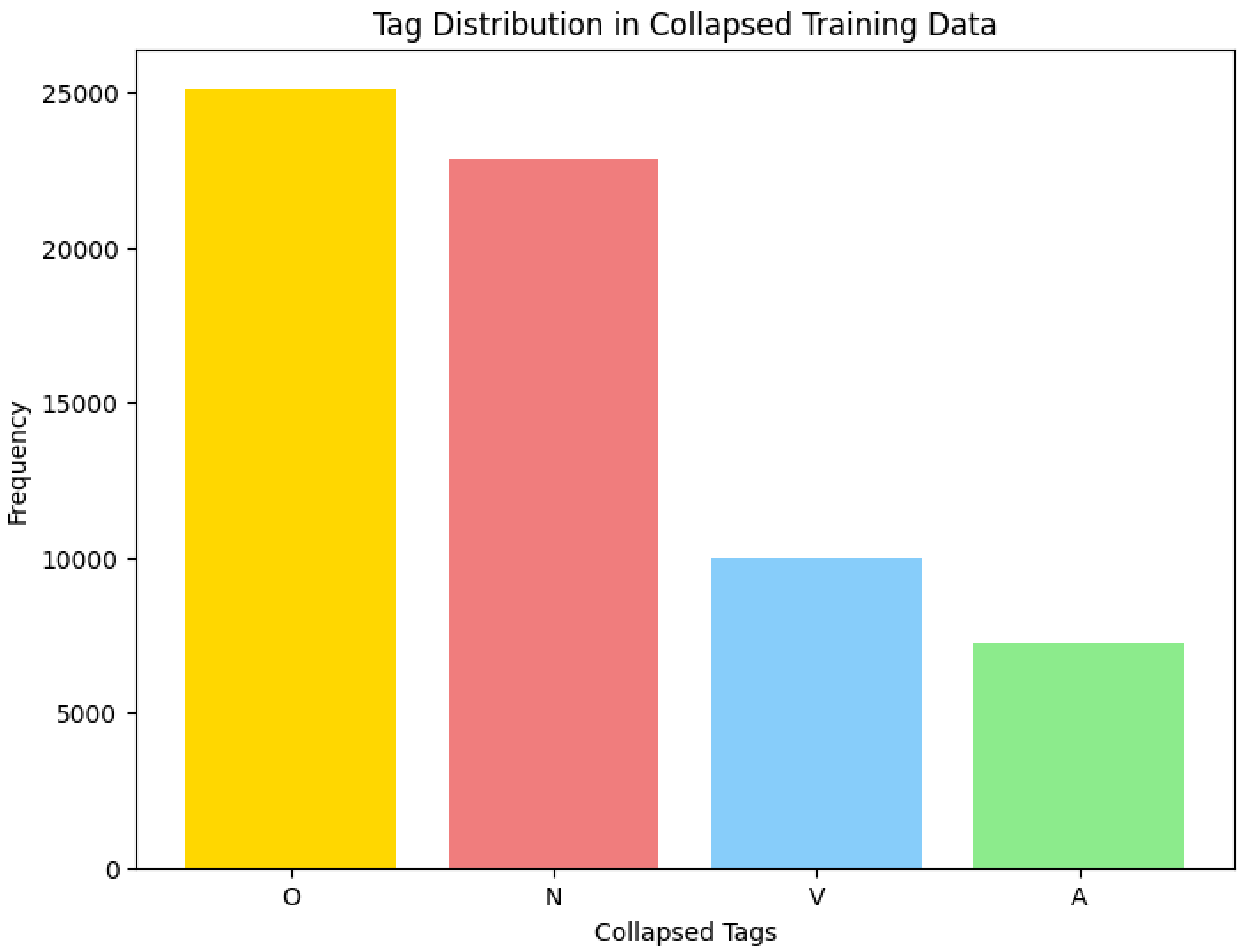
---

3. Dataset and Preprocessing

- **Dataset:**
  Penn Treebank dataset provided in JSON format (each entry contains a sentence and its tag list).
- **Preprocessing Steps:**
  - **Cleaning:** Removed entries where the number of tokens did not match the number of tags.
  - **Tokenization:** If a sentence is given as a string, it is tokenized (using whitespace and NLTK's word_tokenize if necessary).
  - **Train/Test Split:** 80% of sentences are used for training and 20% for testing.
- **Tag Collapsing:**
  The 36 tags are collapsed into 4 groups:
  - **N:** Noun tags (NN, NNS, NNP, NNPS)
  - **V:** Verb tags (VB, VBD, VBG, VBN, VBP, VBZ)
  - **A:** Adjectives and adverbs (JJ, JJR, JJS, RB, RBR, RBS)
  - **O:** All other tags

Figure 1: Bar chart showing tag distribution in original and collapsed training data.


Tag Distribution in 36-tag Training Data
Frequency
10000
8000
6000
4000
2000
0
Tags
DT
NN
VBD
IN
PRP
VBZ
VBN
CD
CC
RP
NNS
JJ
NNP
VBG
PRP$
TO
RB
VB
JJR
JJS
VBP
WRB
WDT
MD
WP
NNPS
RBS
:
RBR
EX
#
-LRB-
-RRB-
PDT
UH
WP$
FW
''
,
LS
SYM



---

4. Methodology

4.1 HMM Model Variants

- **First Order HMM:**
  - **Probabilities:**
    - Initial: P(t1)P(t_1)
    - Transition: P(ti∣ti−1)P(t_i|t_{i-1})
    - Emission: P(wi∣ti)P(w_i|t_i)
  - **Decoding:**
    Viterbi algorithm used to compute the best tag sequence.

- **Second Order HMM:**
    - **Probabilities:**
        - Initial and second tag probabilities are computed.
        - Transition: P(ti∣ti−2,ti−1)P(t_i|t_{i-2},t_{i-1})
    - **Decoding:**
      Modified Viterbi algorithm tracks tag pairs.
- **HMM with Previous Word Dependency:**
    - **Probabilities:**
      Emission probability is conditioned on both the current tag and the previous word: P(wi∣ti,wi−1)P(w_i|t_i, w_{i-1}).
    - **Note:**
      This model has a higher parameter count and suffers from data sparsity.

4.2 Handling Unseen Words

- For unseen words, the model uses the most frequent tag from the training data as the default.

---

5. Experiments and Results

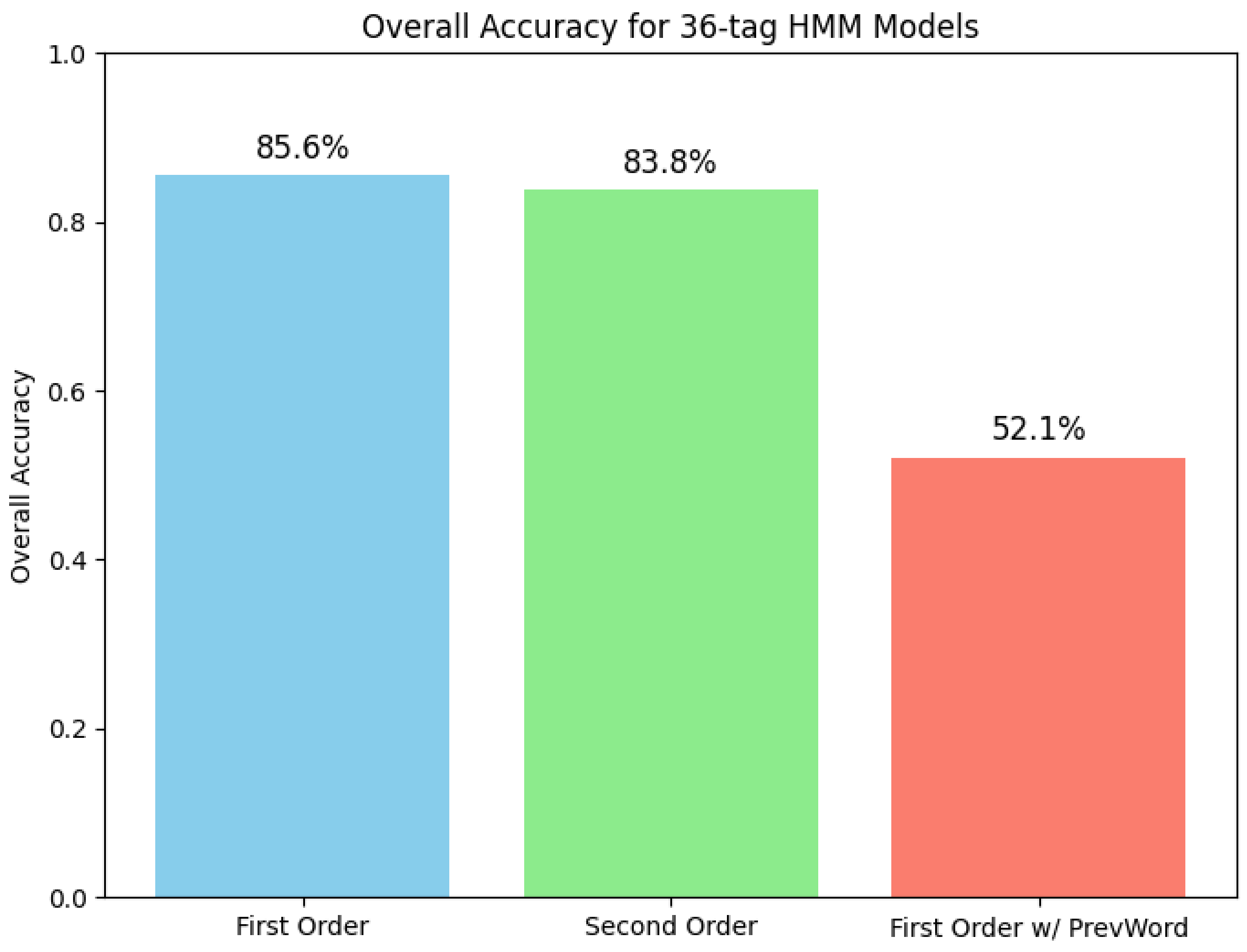
5.1 Experimental Setup

- **Data Split:**
  Training: 3131 sentences; Test: 783 sentences.
- **Evaluation Metrics:**
    - Overall Accuracy: Percentage of tokens correctly tagged.
    - Tag-wise Accuracy: Accuracy computed for each individual tag.

5.2 Results for 36-Tag Models

- **First Order HMM:**
    - Overall Accuracy: **85.60%**
    - Tag-wise Accuracy: (e.g., 'NN': 97.53%, 'DT': 99.24%, etc.)
- **Second Order HMM:**
    - Overall Accuracy: **83.77%**
    - Tag-wise Accuracy: (e.g., 'NN': 97.25%, 'DT': 98.48%, etc.)
- **HMM with Previous Word Dependency:**
    - Overall Accuracy: **52.11%**
    - Tag-wise Accuracy: (e.g., 'NN': 68.58%, 'DT': 81.17%, etc.)

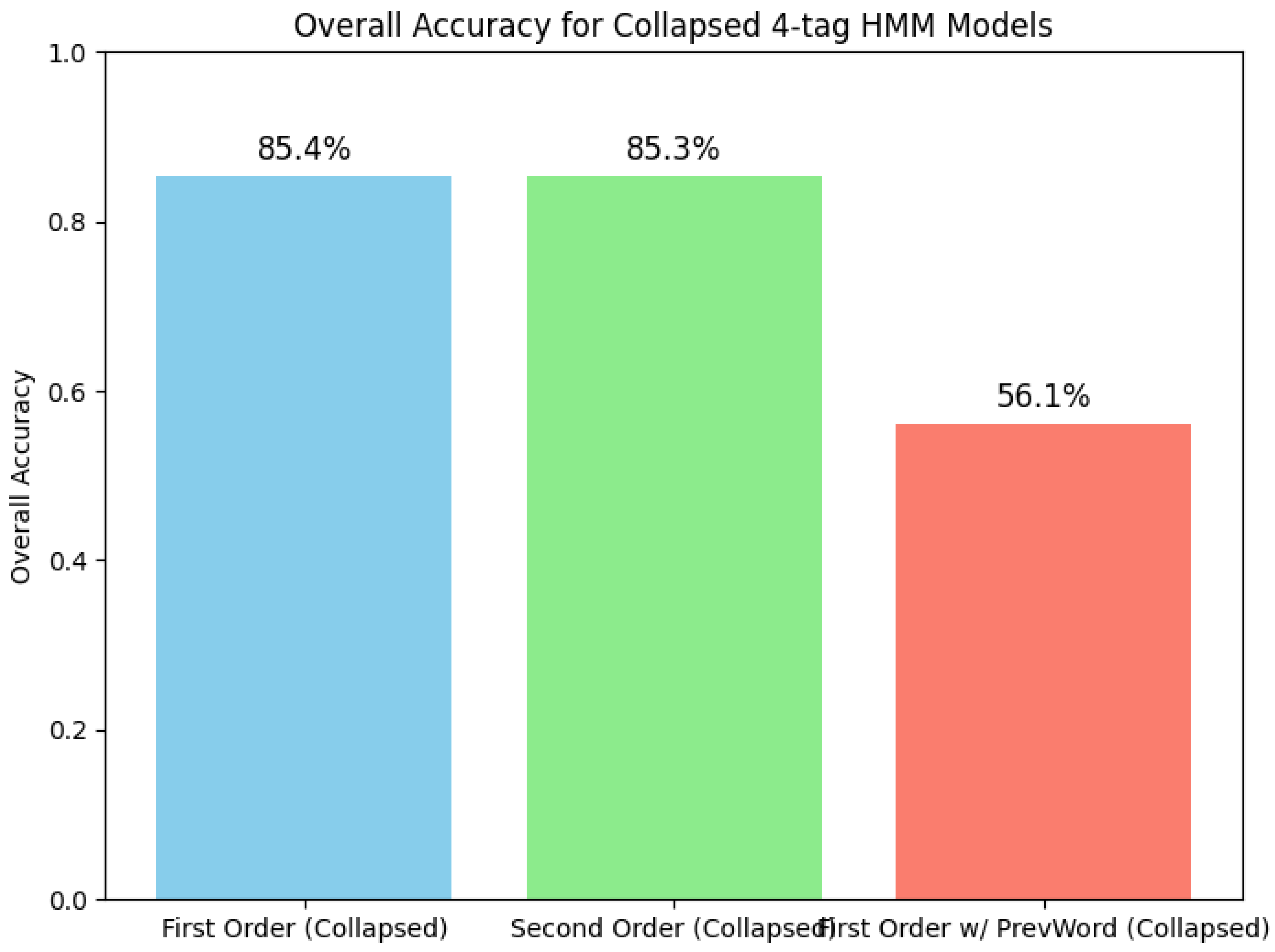
Figure 2: Bar chart comparing overall accuracies for the three variants on the 36-tag dataset.



5.3 Results for Collapsed 4-Tag Models

- **First Order HMM (Collapsed):**
  - Overall Accuracy: **85.35%**
  - Tag-wise Accuracy: 'N': 75.10%, 'V': 80.40%, 'O': 99.73%, 'A': 75.90%
- **Second Order HMM (Collapsed):**
  - Overall Accuracy: **85.34%**
  - Tag-wise Accuracy: 'N': 75.16%, 'V': 80.52%, 'O': 99.65%, 'A': 75.68%
- **HMM with Previous Word Dependency (Collapsed):**
  - Overall Accuracy: **56.08%**
  - Tag-wise Accuracy: 'N': 62.29%, 'V': 32.80%, 'O': 68.06%, 'A': 28.83%

Figure 3: Bar chart comparing overall accuracies for the three variants on the collapsed dataset.



---

6. Comparative Analysis and Discussion

6.1 36-Tag vs. Collapsed 4-Tag Models

- **Total Accuracy:**
  The First and Second Order HMMs function nearly similarly in both settings (approximately 85%).

- **Intuition:**
  Reducing the number of tags collapses them and increases examples per tag, making it less sparse. But in our scenario, the overall accuracy did not appreciably vary from 36 to 4 tags, showing that the fine-grained distinctions don't impair performance at the token level.

6.2 Model Variant Comparison

- **HMM with Previous Word Dependency**
  Conducted much worse (52–56% total), probably because there are more parameters, which has led to sparse counts and unusable estimates.

- **Why Having Fewer States Could Help**
  With a collapsed tag set (4 tags), transition and emission estimates are made from more data per state, which has the potential to result in smoother and more accurate probability estimates. Yet, overall performance of our First and Second Order models is also the same in both scenarios.

6.3 Handling Unseen Words

- The use of the most frequent tag as a default for unseen words seems effective, as common tags like 'DT' and 'IN' show very high accuracies.

6.4 Future Improvements

Laplace smoothing would assist in dealing with low-frequency counts, particularly for the previous word dependency model. Incorporating features like word prefixes, suffixes, or capitalization could also enhance tagging precision. Blending HMM output with rule-based or CRF-based systems might be more effective.

---

7. Conclusion

- **Summary:**We tested three different HMM versions and compared their performance on the 36-tag and collapsed 4-tag set-ups. Both the First Order and Second Order HMMs worked well (approximately 85% overall accuracy), whereas the model with a dependency on previous word did poorly (approximately 52–56% overall accuracy).
- **Insights:**Collapsing tags is useful for alleviating sparsity, but in our experiments the full 36-tag model still performed with high accuracy. Adding more complexity to conditioning on the prior word appears to degrade performance because of sparsity of the data.
- **Future Work:**Future experiments could involve smoothing methods, more features, and experimenting with hybrid models.

---

8. Group Contributions

- **Abhijeet Singh Naruka:** Data cleaning, tokenization, and implementation of the First Order HMM.
- **Aditya Anand:** Implementation of the Second Order HMM and its Viterbi algorithm.
- **Raghav Vijayvergia:** Implementation of the HMM with Previous Word Dependency variant and preliminary experiments.
- **Chauhan Shashank Pravinbhai:** Evaluation, error analysis, graph generation, and report compilation.

---

9. References

- Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing*.
- Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*.

THANK YOU