CMPUT 461 Intro to NLP - Assignment 4

1. Introduction

Two POS taggers, the Hidden Markov Model (HMM) tagger and the Brill tagger, are evaluated based on their performance in both in-domain and out-of-domain datasets. The task involves training, tuning, and testing these taggers on (in-domain) and (out-of-domain) texts. This report discusses the tuning efforts, accuracy results, misclassifications, and a comparative analysis of the two taggers.

2. Tuning Efforts

2.1 HMM Tagger Tuning

The HMM tagger was primarily tuned using training data size and smoothing techniques. Below are the key parameters tuned and their impact:

- **Training Data Size**: The amount of training data had a significant impact on the model's accuracy. Performance improved with larger datasets, but the improvement slowed as the dataset size increased beyond a certain threshold. Optimal performance was observed with a large dataset, though further increases in data size provided diminishing returns.
- **Smoothing Techniques**:
  - **Laplace Smoothing**: This technique worked well for generalization, especially for unseen word pairs, and helped mitigate zero probabilities in the model.
  - **Lidstone Smoothing**: After experimenting with different values for the gamma parameter. The optimal gamma value was 0.1, which resulted in a noticeable improvement in accuracy, particularly with unseen events.

2.2 Brill Tagger Tuning

The Brill tagger was tuned by adjusting the maximum number of rules (max_rules) and applying different templates. The following parameters were considered:

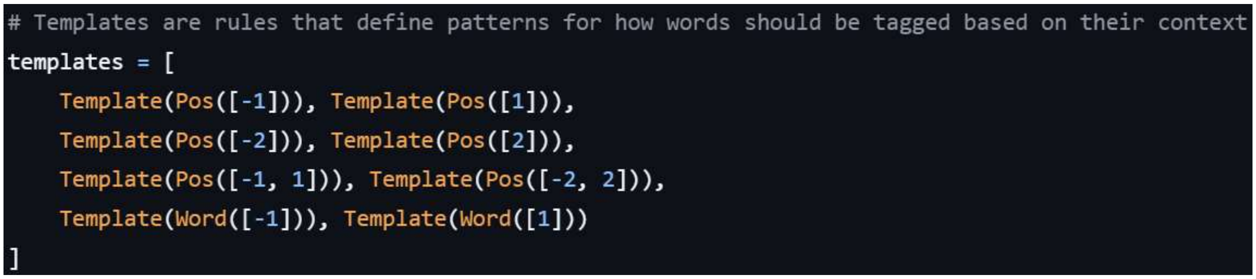
Max_rules:

- Tested values 50, 100, 150, and 200 for the max_rules parameter. The best performance was achieved with **max_rules=200**.
- The max_rules parameter limits how many transformation rules can be learned and applied to improve the accuracy of the tagging. A higher max_rules value allows the Brill tagger to learn and apply more rules, potentially leading to better accuracy, but also increasing the complexity of the model.
- *Overfitting vs. Generalization:*

    - If *max_rules* is too small, the tagger may not have enough rules to correct all the errors in the baseline tagger's predictions, leading to lower accuracy.
    - If *max_rules* is too large, the tagger may overfit to the training data. This means it could learn rules that are specific to the training set but fail to generalize well to new, unseen data (e.g., out-of-domain data).

Template:

A template defines the context of a word by looking at surrounding features, such as the part-of-speech (POS) tag or the word itself, at specific positions relative to the target word.



```
# Templates are rules that define patterns for how words should be tagged based on their context
templates = [
    Template(Pos([-1])), Template(Pos([1])),
    Template(Pos([-2])), Template(Pos([2])),
    Template(Pos([-1, 1])), Template(Pos([-2, 2])),
    Template(Word([-1])), Template(Word([1]))
]
```

- The above template is the one used in src/main.py python code, which represent:
    - **Pos([-1])** and **Pos([1])**: Focus on the immediate context, looking at the POS tag of the surrounding words.
    - **Pos([-2])** and **Pos([2])**: Look further into the past and future of the sentence, helping to capture long-range dependencies.
    - **Pos([-1, 1])** and **Pos([-2, 2])**: Combine information from both directions, which helps in cases where the word's meaning is influenced by both neighboring words.
    - **Word([-1])** and **Word([1])**: Incorporate specific word-level features, useful for distinguishing between different possible tags based on actual word forms.

  The key idea behind choosing these templates is to make sure that the Brill tagger has access to enough relevant context to correctly identify the POS tags for words.

    - **Immediate Context:** Helps the tagger make quick decisions based on the most obvious clues (previous/next word).
    - **Longer-range Context:** Captures more complex dependencies between words, improving accuracy in sentences where the relationship between words spans more than just one/two positions.
    - **Combined Context:** Allows the tagger to consider the between words in both directions.
    - **Word-Level Features:** Provides the model with direct clues from the actual words, helping resolve ambiguities that might be missed by only considering the POS tags.
- These templates helped the Brill tagger generalize transformations for surrounding words, improving accuracy, particularly on the in-domain dataset.

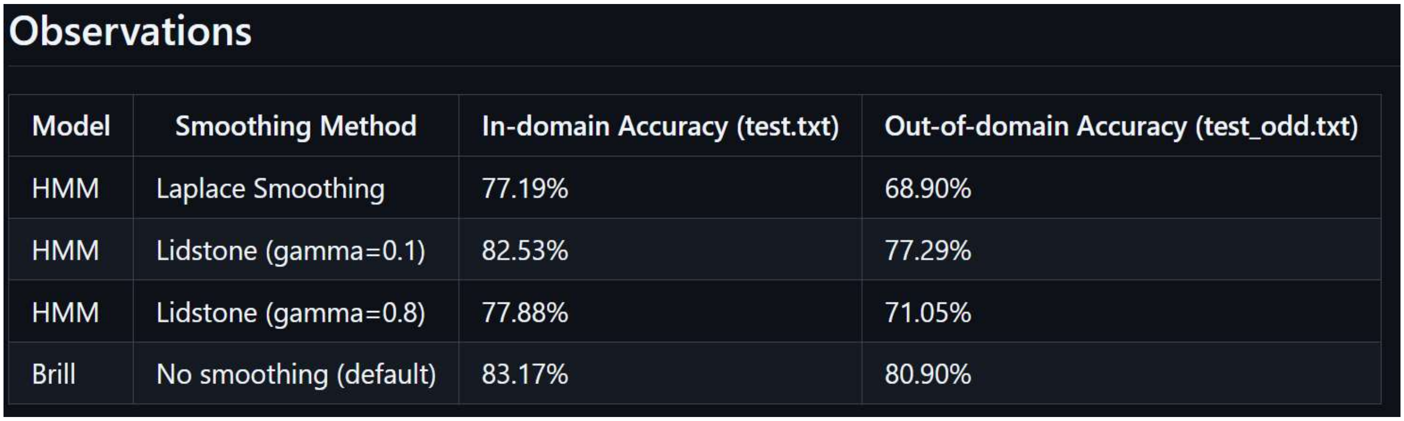
2.3 Parameter Set Influences

- **HMM Tagger**: Larger training datasets significantly improved accuracy, but the effect was more pronounced with richer and more diverse data. After a certain point, adding more data did not provide substantial improvements.
- **Brill Tagger**: Increasing the number of rules beyond 200 did not lead to a noticeable increase in accuracy.

3. Tagger Performance

Accuracy on In-domain and Out-of-domain Data

The table below shows the accuracy of both taggers on in-domain and out-of-domain datasets:



Observations

Model	Smoothing Method	In-domain Accuracy (test.txt)	Out-of-domain Accuracy (test_odd.txt)
HMM	Laplace Smoothing	77.19%	68.90%
HMM	Lidstone (gamma=0.1)	82.53%	77.29%
HMM	Lidstone (gamma=0.8)	77.88%	71.05%
Brill	No smoothing (default)	83.17%	80.90%

Observations:

1. The Brill tagger slightly outperformed the HMM tagger in both in-domain and out-of-domain datasets. The Brill tagger's **rule-based approach (template)** allowed it to make more **precise adjustments** based on the surrounding context, whereas the HMM tagger's performance was more dependent on the **training data and smoothing techniques.**
2. Both taggers performed better on the in-domain dataset, which is expected due to the **similarity between training and testing data** in terms of domain-specific vocabulary and sentence structure. This allowed both models to **leverage the contextual patterns** present in the training data more effectively
3. The optimal value of **gamma = 0.1** which provided the best performance for both taggers. For the **HMM tagger**, this value of gamma worked well with Lidstone smoothing, significantly improving accuracy by better handling unseen word pairs. In contrast, a **higher gamma (e.g., gamma = 0.8)** resulted in reduced accuracy for both models, suggesting that a lower value helps the taggers generalize better to unseen data **without overfitting to the training set**. This shows that smoothing is crucial for both taggers, and the **right balance** of smoothing parameters is important for optimal performance.

4. Tagger Errors

4.1 HMM Tagger Errors

The HMM tagger made several types of errors, primarily related to word ambiguity and insufficient training data on specific patterns:

- **Ambiguous Word Errors**:
    - The HMM tagger struggled with words that could be both nouns and verbs, such as "lead" and "project". These words were often misclassified due to context-dependent meanings.
- **Multi-word Expressions**:
    - The HMM tagger misclassified multi-word expressions like "New York" due to a lack of training on proper nouns, resulting in errors such as misclassifying proper nouns as common nouns or verbs.

4.2 Brill Tagger Errors

The Brill tagger, being rule-based, made different kinds of errors based on its transformation rules (from the template used):

- **Plural/Singular Confusion**:
    - The Brill tagger sometimes confused singular and plural nouns due to the rules it applied. (e.g., "dog" tagged as NNS), likely due to the template rules failing to capture the correct singular/plural context.
- **Verb Tense Confusion**:
    - The Brill tagger misclassified verb tenses, such as tagging "run" as a past tense verb instead of present, which suggests that the transformation rules might not have captured the full scope of verb tense variations.

5. Comparison of Tagger Errors

5.1 Misclassification Patterns

Error Type	HMM	Brill
Ambiguous Word Errors	High, especially for homonyms	Low (minimal)
Contextual Errors	Struggled with surrounding words	Low, mostly rule-based errors
Training Data Influence	Significant with large data	Improved with more rules

- **Summary of Differences**:
  - The HMM tagger's errors were largely due to context-based ambiguities and its inability to handle multi-word expressions effectively.
  - The Brill tagger's errors stemmed mainly from misclassifications caused by the transformation rules, such as errors in handling plural/singular distinctions and verb tense.

5.2 Parameter Sensitivity: Effect of Gamma

Gamma	**Accuracy for HMM**	**Accuracy for Brill**
0.1	82.53%	83.17%
0.8	77.88%	80.90%

- **Observations**: A gamma value of 0.1 provided the best results for both the HMM and Brill taggers. This value helped both models better generalize to unseen data, with Lidstone smoothing in HMM and transformation-based learning in the Brill tagger benefitting the most.

6. Conclusion

In conclusion, the **Brill tagger outperformed** the **HMM tagger** on both in-domain and out-of-domain datasets, This shows that rule-based taggers can be **more accurate** when their rules are carefully **chosen and fine-tuned**.Training **data size** had a **significant impact** on the HMM tagger's performance, while the Brill tagger benefited from an increased number of **transformation rules** up to a certain point. Both models showed sensitivity to their respective parameters, with HMM benefiting from a larger and more diverse dataset, while the Brill tagger's performance improved as **more transformation rules were applied**.

Ishaan Meena

1780950