HMM Tagger

For the HMM Tagger task, we used the `nltk.tag.hmm.HiddenMarkovModelTrainer` to train an HMM tagger. Initially, we used the default `MLE` estimator and observed inferior accuracy (`24.31` on the in-domain test set). This is because it didn't account for OOV tokens and would almost always incorrectly tag these as it was hard-coded to the training corpus. The MLE estimator resulted in overflow errors since it uses raw frequency counts from the training data without smoothing, leading to zero probabilities for OOV tokens. We then adjusted the estimator to use `Laplace Smoothing`, which had better accuracy (`0.7719` on the in-domain test set) than the MLE estimator but still had a worse accuracy than the Lidstone estimator. We suspect that this may be the result of relatively high emission probabilities for all words in tags with very few occurrences. So if `PRP$` only appeared once in the corpus, it would be assigned very high emission probabilities for every word. Compared to the Lidstone estimator, the Laplace estimator with add-one smoothing can skew the probabilities for frequent tokens. Since we had to apply smoothing to account for OOD words, we chose to apply `add-k` smoothing with `k=0.0001`. This was done using the `LidstoneProbDist` class. We achieved a final accuracy of `82.85` on the in-domain test set and `77.85` on the out-of-domain test set. The Lidstone estimator resulted in the best accuracy since the small constant (gamma) helps avoid zero probabilities while maintaining the relative frequencies, allowing the model to generalize better.

Brill Tagger

For the Brill Tagger task, we utilized the `nltk.tag.brill_trainer` to train a Brill Tagger. Initially, we implemented a `DefaultTagger` that assigned the `NN` tag to every token. This approach resulted in low accuracy, so we added more template rules to the trainer. After doing so, we obtained the best test accuracy of around 65% on the test set and 55% on the out-of-domain (OOD) test set. After reviewing the discourse on the eClass forum and the procedures to create a more well-tuned initial tagger, we developed an initial tagger class called `InitialBrillTagger`. This tagger would tag every word in the test corpus with the most frequent tag for that word if found in the training corpus, and NN if not found in the corpus. This change significantly increased accuracy for both test sets, resulting in a final accuracy of 83.41% on the in-domain test set and 80.77% on the out-of-domain test set.

We experimented with multiple templates but ultimately arrived at a simple set. It includes transformations that use the context of the positions and words of only the most immediate neighbouring words. We found that increasing the template list with templates that covered a larger context window did not improve accuracy. The most effective rules considered either the word immediately before or after the current word. For instance, when the tagger transformation improved accuracy, whenever the tagger came across `n't`, the previous word would be transformed from `NN` to `VB`. Apart from this, most rules were standard,

found in an article on the Brill Tagger.

Apart from this, we added a few more rules that took in more context. Only a few additional rules were learnt, including changing `TO` to `IN` if a `FW` was seen previously. These changes did affect the test accuracy. In total, 142 transformations were learned, most of them coming from only immediate contexts. Changing Verb forms, from `VBP` to `VB` based on the previous word, `VBD` to `VBN` based on the previous word, and so on were the most common rules found. We believe these were found in both test corpora as they were likely not well represented in the training corpus.

Accuracy

Tagger	Test %	Test OOD %
HMM	82.85	77.85
Brill	83.41	80.77

Misclassifications

Brill Tagger Misclassifications

For the Brill tagger, many ambiguous words with multiple tags depending on context were misclassified. We suspect this is because the templates provided were unable to capture further rules that would convert these tags. Furthermore, these words were also provided with only one tag to the initial tagger. So a word like `cooking` which can assume both `NN` and `VBG` was always assigned the `NN` tag as it was the most popular in the training corpus. The most common misclassification error was observed to be OOV words. The default tagger that we created either assigns a token to the most common tag found in the training corpus, or the tag `NN` if not found in the training corpus. The tagger also struggles with the plural, almost always using `NN` instead of `NNS`, and we suspect that this is because if the plural word is OOV, the transformation for converting to plural would never be learned. Words like `shoulders` present in the training corpus were correctly labelled in both test corpora. The majority of the errors observed were found to be due to a word being OOV and so the brill tagger struggles. Out of 540 misclassifications, only 90 did not include the `NN` tag.

Some of the templates that increased accuracy, did negatively affect the tagger. For example, many rules would transform the word `to` from `TO` to `IN`, which did have a negative effect on the accuracy of the OOD test corpus. Misclassifications for Verb form also led to further misclassifications in neighbouring words. For incorrectly labelling a `VBZ` as `VBP` would often misclassify the adverb following it.

HMM Tagger Misclassifications

For the HMM tagger, the most commonly occurring misclassification error also occurs with the OOV words. This can be seen as some words will be tagged as non-word tags such as `''`. We assume this is because of the explosion in relative emission probabilities for tags that occur infrequently within the training corpus. This can be seen as tags such as `PRP$` and `WRB` which occur less often in the training corpus, are predicted too often within both test corpora. Another problem the HMM Tagger faces is words that can assume multiple tags. For example, the word `forms` only occurs as `NNS` within the training data, but had a ground truth label of `VBZ` in the test corpus.

In contrast to the Brill Tagger, the HMM tag would often incorrectly tag singular words as plural. Specific nouns have the same form for both singular and plural words, which our HMM tagger is unable to distinguish between. For example, 'smog' was tagged as `NNS` when it should have been `NN`. The HMM tagger also struggled with words that can be used as an adjective or a noun. For example, the word 'downtown' was tagged as `NN` when it was labelled as a `JJ`. We suspect this is because the emission for ambiguous words is similar and minor variations in the training data lead to the HMM tagger predicting the wrong tag.

Comparison of Taggers

Both the Brill Tagger and HMM Tagger struggle with words taking on multiple tags. The HMM tagger overcomes this if the emission probability from tag to word is non-zero and non-trivial (from smoothing) for all tags that the word can assume. The Brill Tagger overcomes this if it is able to learn a rule from the provided templates which allows a transformation to the correct tag. However, rule conflicts dependency on the order of the rules in the template list can lead to misclassifications.

Additionally, both tags misclassify OOV words, however, the Brill Tagger achieves higher accuracy on the OOV words since it defaults to `NN`, which is quite common. The HMM tagger, on the other hand, suffers from explosive emission probabilities and thus often misclassifies those words. Our HMM tagger applies smoothing to handle OOV words, whereas the Brill tagger relies on the initial tagger to handle OOV words.

Comparing the two taggers, we come to the conclusion that when HMM misclassifies a word, it appears a lot more obscure. We found that because of the explosive emission probabilities when it got a word wrong, it would usually be with a very rare tag. This is different from the Brill Tagger, where the misclassification is usually a reasonable guess. We believe this is entirely due to the implementation difference between the two taggers. One significant difference is that the HMM tagger is a generative tagger and the Brill tagger is a transformational tagger. The HMM tagger relies a lot on transition probabilities and emission probabilities, so we found that it does not generalize as well since it may be overfitting to the training corpus probability distribution. But since the

Brill Tagger is capable of learning transformations, it could generalize better even on unseen words. This is because it could learn transformations independent of exact vocabulary and only dependent on position. And since it defaults to `NN` a lot, these transformations had a high probability of occurring. From test accuracy, we can see the gap in performance is larger on the out-of-domain test set compared to the in-domain test set. The HMM tagger is more sensitive to the size of the training data.