CMPUT 501 - Assignment 4

{karimiab, azamani1}@ualberta.ca

In this assignment, we used hmm and brill taggers to tag sentences. Also, to achieve a better performance, we fine-tuned the hmm and brill taggers on the development data and test them on in-domain and out-of-domain text samples to find out the performance of each tagger.

1. Fine-tunning Efforts

For the different hyperparameters, we calculated the accuracy of the taggers, then we chose those hyperparameters that yield the highest accuracy. Then, we test out the taggers with the chosen hyperparameters on the test data.

1.1 HMM Tagger

We used the hmm tagger which is based on 'nltk.hmm' class. The only hyperparameter that is used for hmm tagger is the 'probability estimator', so we initialized the model according to various probability estimators from nltk.probability class, which are listed below:

1. SimpleGoodTuringProbDist
2. LaplaceProbDist
3. MLEProbDist
4. ELEProbDist
5. WittenBellProbDist

One way to do hyperparameter tunning is to use k-fold cross-validation. We utilize sklearn package to initialize a 5 fold object for the hmm tagger, the reason behind selecting 5 as the number of fold is that in machine learning it is common to use number 5 for datasets having less than 10.000 instances. In each iteration, We trained the model on the 4 folds and tested it on the remaining fold. It is worth mentioning that, to avoid overfitting evermore, we shuffled the training data. After that, we took the average of the accuracies that were obtained on each iteration. Hyperparameter tunning ensures that the model is not overfitting the data and can generalize well on the data that has never been seen before. After performing the tunning process, we selected the parameters which resulted in the highest accuracy. That being said, we trained the hmm tagger with the found parameters and tested on the test data. In table 1, we listed accuracies associated with different probability estimators. As you can see, the 'WittenBellProbDist' probability estimator yields the highest accuracy.

Tabel 1: HMM tagger accuracies for various estimator

Estimator	Mean Accuracy (kfold=5)
Simple Good Turing	35.58
Laplace	81.07
Maximum Likelihood	77.22
Witten-Bell	87.39
Expected Likelihood	87.25

The ‘simple good turing’ yields the lowest accuracy, i,e. 35.58%. The next estimator is ‘maximum likelihood’ which achieves an accuracy of 77.22%. The ‘laplace’ estimator performance is 81.07%, while the ‘expected likelihood’ achieves 87.25% accuracy. The ‘written-bell’ estimator yields the highest performance which is 87.39%.

1.2 Brill Tagger

The cross validation was also being done for tuning the Brill Tagger. However, in this time, we changed the number of folds to 2 since the brill tagger is computationally more expensive and the process for this tagger takes a very long time. For tuning the brill tagger, we used the baseline rule templates and also number of rules to fine tune the tagger. For the rules, we used the basic initial rule from nltk documentation along with fntbl37, nltkdemo18, nltkdemo18plus, and brill24. As the name of templates suggests, the fntbl 37 returns 37 templates derived from the POS tagging of the fntbl distribution, brill24 returns 24 templates, and also nltkdemo18 and demo18plus return 18 templates from their distributions. As listed in Table 2, fntbl37 rules obtained the best average accuracy with a rule count of 100.

Tabel 2: Brill tagger accuracies obtained w.r.t the given parameters

Rules	Max Rules	Mean Accuracy
baseline	10	42.94
baseline	50	55.17
baseline	100	59.97
brill24	10	51.56
brill24	50	64.88

brill24	**100**	**69.47**
nltkdemo18	10	39.63
nltkdemo18	50	48.10
nltkdemo18	100	51.84
nltkdemo18plus	10	41.70
nltkdemo18plus	50	51.15
nltkdemo18plus	100	55.34
fntbl37	10	51.56
fntbl37	50	65.48
fntbl37	100	70.12

2. Tagger Performance

In this section, we reported the accuracy of the hmm and brill taggers on the in-domain and out-of-domain test sets. As mentioned in the first section, the best hyperparameter for hmm is ‘written-bell’ brabability estimator, so we used this hyperparameter for testing the hmm tagger. As mentioned in table 3, the accuracy of hmm tagger on the in-domain dataset is 86.40% and on the out-of-domain dataset is 81.78%. According to the section, the best hyperparameters for brill taggers are ‘brill24’ as a rule. Also, the best ‘max rules’ is 100. Hence, we used these hyperparameters to test out the brill tagger on the in-domain and out-of-domain datasets. As you can see in table 3, the accuracy of hmm tagger on the in-domain dataset is 71.18% and on the out-of-domain dataset is 59.35%.

Tabel 3: The HMM and Brill taggers’ performance on the in-domain and out-of-domain datasets

Tagger	**In-domain accuracy**	**Out-of-domain accuracy**
HMM	86.40	81.78
Brill	71.18	59.35

3. Error Analysis:

- In-domain HMM:
  Although the in-domain HMM model performs pretty well, but in several cases, it makes a mistake in identifying Noun vs. Verb. For instance, in the second sentence,
  “Add the quinoa and cook for about 1 minute.” it tagged cook as a Noun while it is obvious that cook is a Verb in this sentence. Another example is “Cover and reduce heat to low”, in which the hmm tagger, tags ‘to’ as ’infinitive’, while the actual tag is ‘prepostion’.
- Out-of-domain HMM:
  Consider “Her broad-brimmed hat is still sailing on the wind.” as an example. HMM tagger tags “Her” as proper noun, while it is obvious that the true tag for it is possessive pronoun. Another example is “The mites missed it.” in which “mites” tags as NN, while we know that the true tag is NNS.
- In-domain BRILL:
  Consider “Drizzle a bit of olive oil in a pan over medium-high heat.” as an example. The brill tagger tags olive, pan, over, and medium-hight as proper noun, adjective, NN, and NN, respectively. While we know that the true tags for them are NN, NN, preposition, and adjective.
- Out-of-domain Brill:
  Consider “Got it?” as an example. We know that “Got” is a verb. But, the hmm tagger tags it as a preposition. Same as HMM tagger for out-of-scope data, the hmm tagger tags “Her” as a proper noun, while it is possessive.

4. Tagger Comparison:

Two taggers can be compared in many aspects. Here are some of them:

- Computational resources:
  The Brill tagger was introduced in 1992 and it is considered a rule-based method for Part of Speech tagging. As experienced in this assignment, training a tagger using rules takes a long time and the performance is also not that good. On the other hand, Hidden Markov Model is a probabilistic model that can predict the POS tag of a word pretty well by using the conditional probabilities and latent space. HMM uses less resource and is the winner of the competition with respecting to computation resource.
- Errors:

Since the Brill tagger is based on rules, it can not generalize well. To be more clear, adding more rules means adding more limitations or decreasing the generalization. This is why the Bril tagger's performance is heavily dependent on the rules that was given to it. On the other hand, HMM, by utilizing the concept of probabilities, can have a big picture of the frequencies and can generilze better.

- In domain and out of domain:
  HMM model's in domain and out of domain's performance were very close while the Brill model's in domain and out of domain's performance were different. This again, strengthen the idea of generalization. In a rule-based method, when a tag or rule is not presented to the model, the model would heavily suffer from this, but this is not true about the probabilities which is the basic idea of what we have in HMMs.