Report

Anonymous ACL submission

In this assignment, we implemented hmm and Brill part-of-speech taggers. Furthermore, we tuned them and compared their performance on in-domain and out-of-domain text samples.

1 Tuning efforts

Different hyperparameters for each POS tagger are used, and then we compared their influence on each tagger's performance.

1.1 HMM Tagger

For hmm, various estimators, including the maximum likelihood, Laplace, Lidstone, expected likelihood, Witten-Bell, and Simple Good Turing estimators are selected as hyperparameters. Furthermore, for the Lidstone estimator, we considered various gamma values and investigated their influence on tagger performance.

We implemented 10 fold cross-validation and trained the model on nine folds, and tested it on the hold-out fold to compare various models. We repeated the training ten times and used the mean accuracy to compare the models. 10 fold cross-validation is usually implemented to statistically ensure that the model performs better and the performance is not due to a chance. The distribution of the mean value of the accuracy will follow the t-distribution, which can be used to compare the models. In this assignment, we calculated the mean value of accuracy and selected the model that provided the highest mean value accuracy.

As reported in the table (1) Witten-Bell estimator provided the best mean accuracy equal to 84%; the next best performing estimator is Lidstone with mean accuracy equal to 80.5%. We created several Lidstone estimators with various gamma values and tested them with each of the parameters. Lidstone performance continuously went down as we changed the Lidstone gammas from between 0.1 and 0.9. Then, we searched the gammas between 0.01 and 0.1. Lidstone did best with 0.06 gamma with an accuracy of 80.5 % but still not as good as Witten-Bell with 84%. Simple good turing did almost as well as Lidstone with 80% accuracy. The Laplace provided 77% accuracy and the expected likelihood procided 72% accuracy. The maximum likelihood estimator provided the lowest performance on POS tagging, which was equal to 24% because it is highly affected by unknown tokens, which are not found in the training corpus.

Table 1: HMM POS-tagging performance for various hyperparamters

Hyper parameter	Mean Accuracy
Witten-Bell	**0.84**
Lidstone, gamma = [0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.805
Maximum likelihood	0.24
Laplace	0.77
Expected likelihood	0.72
Simple Good Turing	0.80

1.2 Brill Tagger

We did the same k-fold splitting on the training data and defined various hyperparameters, and investigated their influence on its performance. However, since Brill tagger was computationally more expensive and took more time we only used two folds to report the accuracy for this section. We considered the baseline rules template and the maximum number of rules for the Brill tagger. We considered the baseline rules template and the maximum number of rules for the Brill tagger. We considered base which was the default template in nltk documentation, fntbl37, nltkdemo18, and brill24 templates. The fntbl37 return 37 templates taken from the pos tagging task of the fntbl distribution, the nltkdemo18 return 18 templates, from the original nltk demo, in multi-feature syntax, the brill24 return 24 templates of the seminal TBL paper Brill (1995). All the mentioned templates are from the nltk package and we directly used them in our POS tagging model. As shown in table (2) the fntbl37 baseline rules with maximum 100 rules proided the best performance with mean accuracy eqaul to 67%.

The parameters for hmm and Brill tagger are analysed systematically using modes *trainhmm* and *trainbrill* mode in - -tagger [tagger mode] in the code. We defined a list of possible hyperparamters and used grid search to evaluate all possible combinations of parameters to find the best model. However, due to computational limitations we selected a limited possible parameters.

Table 2: Brill POS-tagging performance for various hyperparamters

baseline rules	Max rules	Mean Accuracy
base	10	0.429
base	50	0.542
base	100	0.585
fntbl37	10	0.513
fntbl37	50	0.637
fntbl37	**100**	**0.679**
nltkdemo18	10	0.394
nltkdemo18	50	0.488
nltkdemo18	100	0.537
brill24	10	0.513
brill24	50	0.635
brill24	100	0.675

2 Tagger performance

In this section we provided the accuracy results on the in-domain and out-of-domain test sets for both HMM and Brill tagger. For the HMM we used Witten-Bell estimator, and for Brill tagger we used fntbl37, with 100 rules, which provided the best results in tuning process.

Table 3: HMM and Brill POS-tagging performance for in-domain and out-of-domain test sets

Tagger	In-domain accuracy	Out-of-domain accuracy
HMM	0.864	0.817
Brill	0.711	0.593

3 Error analysis

- **In-domain HMM:** Messes up nouns and verbs in ambiguous cases i.e. ”cook” in the second sentence is a verb tagged as a noun. Or time on line 161 is a noun tagged as a verb. fluff on line 177 is tagged as a noun instead of a verb.

- **Out-of-domain HMM:** Would often default to NN or NNP for out of vocabulary words such as broad-brimmed and memory.

- **In-domain BRILL:** Could not map ?, missing a rule, always marked them as NN. Everything is defaulted to NN if not in initial rules Not very good with sentence starts as it didn't have as much to go off of.

- **Out-of-domain BRILL:** Could not map question marks, missing a rule, sometimes marked as NN sometimes as VB or VBP. Both brills did poorly with punctuation. tagged far too much as NN or NNP. didn't get nearly enough RB

4 Tagger comparison

In general Brill made the same mistakes more often, and got the same things right. Hmm would get more right because it would consider whole sentences, however it would be less consistent. For example HMM mapped “her” incorrectly several times, even though the training data did not have “her” as anything other than PRP. In domain brill had a much higher accuracy than out of domain. While in domain hmm and out of domain were much closer.