CMPUT 497 Assignment 3 Report

Yonael Bekele
University of Alberta
Edmonton, Canada
yonael@ualberta.ca

Michael Lin
University of Alberta
Edmonton, Canada
michael.lin@ualberta.ca

PART OF SPEECH TAGGING

Stanford POS Tagger

We implemented a Python script to pre-process the data before feeding them into the Stanford POS Tagger. The final output is, each sentence per line, each sentence consists of multiple tokens, each token consists of the word and gold standard (tag) separated by an underscore.

After the pre-processing, we utilized the Stanford POS Tagger CLI to train models with Domain1Train.txt and Domain2Traint.txt. Below is the testing result.

	Domain1Train	Domain2Train	ELLTrain
Domain1Test	94.379%	92.247%	
Domain2Test	91.005%	94.098%	
ELLTest	90.293%	90.794%	95.475%

Table 1: Accuracy of tags between each train and test dataset with Stanford POS Tagger.

Hidden Markov Model Tagger

Our implementation would first deserialize the train and test dataset into a data structure that Hidden Markov Model (HMM) trainer needed.

After the training, the program will evaluate the accuracy of models by testing against each testing data set. Below is the result.

	Domain1Train	Domain2Train	ELLTrain
Domain1Test	84.833%	79.785%	
Domain2Test	80.811%	85.597%	
ELLTest	83.401%	82.817%	92.191%

Table 2: Accuracy of tags between each train and test dataset with HMM Tagger.

Brill Tagger

We used the HMM tagger from the previous section as our baseline for the Brill tagger. Both HMM and Brill are trained with the same training data. We found minimal improvement in comparing the baseline tagger with the final brill tagger.

	Domain1Train	Domain2Train	ELLTrain
Domain1Test	85.124%	80.060%	
Domain2Test	81.333%	85.964%	
ELLTest	83.641%	83.100%	92.681%

Table 3: Accuracy of tags between each train and test dataset with Brill Tagger.

ERROR ANALYSIS

Method

We used two main methods to do error analysis. First, we tracked all the mistakes by the tagger using comparisons with the gold-standard text file. The reason we did this was to establish the faults in the tagger's approach, and to try to find patterns of mistakes by the tagger. The second method was to further find patterns using precision and recall, and the construction of a confusion matrix to visualize the tagger's errors. We ended up using the second method as triangulation for the first method. We also ran analysis to track how taggers dealt with OOV(Out-of-Vocabulary words) by calculating all their mistakes and some random sampling to view it at the sentence level.

Stanford POS Tagger

Results showed that the pattern of mistakes existed due to the trained dataset. When the tagger was trained and tested on their respective datasets, there was a difference between the mistakes found in Domain1 and Domain2. The model trained and tested on Domain1's primary mistakes were tagging JJ as NN, JJ as NNP, NN as JJ, and NN as VB. The model trained and tested on Domain2's primary mistakes were tagging VBD as VBN, JJ as NN, RB as IN, and VB as NN. When tested against the opposite dataset, we found that the mistakes were similar to when the trained tagger had evaluated against its corresponding test domain. There appears to be a bias elicited through training, that causes the tagger to

continuously make similar mistakes no matter what dataset it is tested on. The possibilities of bias exist for the other taggers, but it was more clear to see in the Stanford Tagger because it had so few errors.

When Domain1 and Domain2 tested against the ELLtest set, we found very similar mistakes between the two models. One interesting fact however was that the accuracy was very high at 92%. The tagger mistagged TO as IN, JJ as NN, IN as RB, VB as VBP and vice versa. The errors seem to be similar with the primary mistakes seen by the existing models, except share similarities with mistakes found using both training models, whereas the two did not seem to share similarities in the previous test set.

In terms of OOV handling, our config for Stanford POS tagger will try to "predict the tag of rare or unknown words from the last 1, 2, 3, and 4 characters of the word" (Stanford POS Tagger).

Hidden Markov Model Tagger

Results showed when the tagger was trained and tested on their respective dataset, the HMM tagger most commonly misidentified PRP as NNP, NN as NNP and vice versa, NN as NNS and vice versa, JJ as NN and vice versa. Similar errors were happening when the tagger was tested against the opposite dataset. We found that the HMM tagger had difficulty primarily tagging Pronouns and the singular, plural and proper form of Nouns.

When the HMM tagger trained on Domain1 and Domain2 taggers were tested against the ELL test files, it most commonly misidentified TO as IN, NN as NNS and vice versa, VB as VBP and vice versa. However, the HMM tagger trained on the ELL training data did not make the same frequent error of TO as IN. All HMM tagger models tested on the ELL data struggled with identifying singular and plural Verbs and Nouns. The errors seem to differ greatly when tested against ELL, with only the singular NN and NNS miscategorizations similar to the other test files.

We take a close look at the tagger and its handling of OOV. When focusing on the data of the OOV handling by random sampling, testing against the ELL test set we find that many of the mistakes originate because of misspellings or grammatical errors. For the non-ELL trained and tested models, many of the mistakes were minor in the sense that it was making the correct tag but not the right grammatical number (plural or singular) and whether (in the case of a noun) it is proper or not.

Brill Tagger

Results showed when the tagger was trained and tested on their respective dataset, the Brill tagger most commonly misidentified NN as NNS and vice versa, JJ as NN and vice versa, NN as NNP, and PRP as NNP. Similar errors were found when the tagger was tested against the opposite dataset. We found that the Brill Tagger also had problems classifying Nouns as singular, but would only have trouble tagging singular nouns as plural and not vice versa. It also had much more difficulty classifying adjectives than the previous (HMM) tagger.

When tested against the ELL test files, the Brill tagger's primary mistakes after training on Domain1 and Domain2 was TO as IN, NN as NNS and vice versa, VBN as JJ and NN as NNP. These errors are similar to the misidentification of Nouns whether singular, plural or proper experienced by testing against the prior test files. When ELLtrain was tested against ELLtest it showed very high accuracy. Similar to the other taggers, it also didn't misclassify TO as IN. This might have to do with OOV and the occurrence of these words in the ELLtrain data.

When observing OOV words, the Brill tagger had a slightly better average accuracy than the HMM tagger (12%) when trained on Domain1 and Domain2. Similar to before, the ELLtrain against ELLtest showed less misclassification of OOV words than the Domain1 and Domain2.

Validation

The results outlined for each tagger were consistent with the confusion matrix we made for each tagger. We used this method as validation for our error analysis method of finding patterns through the highest occurrence of mistakes by each tagger. We also used the random sampling method to verify at the sentence depth how well our assumptions hold. We used a sampling of n = 10, but we believe more data would allow our random sampling to be more effective.

Conclusions

All 3 taggers performed remarkably well when tested on the ELLTest data and tested on the ELLTrain data. The Stanford tagger performed the best (95.475%) in this regard, and overall in all categories consistently showed over 90% accuracy and outperformed the other taggers. The OOV words seemed to have an effect on the misclassification of tags, where we found a pattern using random sampling that these words have a higher chance of being mislabelled by

the taggers. Evidence of this can be found by how the models perform better on the parallel dataset (ex. Domain1Train and Domain1Test).

TEAM COLLABORATION

Discussion of data pre-processing, NLTK POS taggers, and accuracy of various models with group Delaney & Daniela and Helen & Flora.

REFERENCES

[1] Stanford POS Tagger documentation