# CMPUT 497 Assignment 3: PoS Tagging

**David Hoeppner**
dhoeppne@ualberta.ca

**Abstract**

This assignment had me create 3 models trained on 3 sets each. The sets were two native English documents, and one English Language Learner document. In order to evaluate the documents, the Stanford PoS tagger was used, and the NLTK HMM and NLTK Brill tagger were used. Then, error analysis and comparisons were done between the results and accuracies of the different taggers.

## 1 Credits

The sources consulted for this assignment can be found in the README file.

## 2 Part of Speech Tagging

### 2.1 Stanford Tagger

The Stanford was quite good at tagging when trained on the Domain1 and Domain2 files. It consistently achieved greater than 70% across the tested files when trained on Domain1 and Domain2, sometimes getting above 80%.

| Test File | Tags Right | UNKs Right |
|---|---|---|
| Domain1 | 78.0% | 24.6% |
| Domain2 | 76.4% | 29.3% |
| ELL | 71.9% | 30.8% |

Table 1: Domain1 Stanford Tagger Results.

| Test File | Tags Right | UNKs Words Right |
|---|---|---|
| Domain1 | 74.8% | 26.1% |
| Domain2 | 79.5% | 25.8% |
| ELL | 76.4% | 32.8% |

Table 2: Domain2 Stanford Tagger Results.

The Stanford tagger was trained using the English language rules, rather than generic. This meant that it could quickly train and test the models. In addition to this, the choice was made to run the jar from the command line, in order to make the tagger run with little intervention from me. This led to a little manual data collection, but it was easier than trying to get it to run outside of this fact. The failures of these models seem to be largely in part minor mis-tags. For instance, rather than classifying American as a proper noun, the tagger assigns it the noun tag. Where the Stanford tagger seems to miss quite a bit of tags is when it is trained on the ELL set. Here, it fails to get above 55%, and does not recognize unknown words very well.

| Test File | Tags Right | UNKs Right |
|---|---|---|
| Domain1 | 48.8% | 25.0% |
| Domain2 | 51.4% | 27.3% |
| ELL | 50.2% | 29.6% |

Table 3: ELL Stanford Tagger Results.

I believe this is because of the many misspelled words in the ELL corpus. Because of this, the tagger fails to make heads or tails of what the word could possibly be tagged as, and makes a guess that is usually wrong. As the tagger is trained on English, it fails to recognize words that do not quite fit into the English language; it has a low fault tolerance for misspellings.

### 2.2 HMM Tagger

The HMM tagger, on average, performed better than the Stanford Tagger. It regularly achieves an accuracy of above 80%, and on one occassion achieves a 92%. This tagger also managed the ELL corpus' much better.

As you can see, the HMM tagger performs

| Test File | Accuracy |
|---|---|
| Domain1 | 84.8% |
| Domain2 | 80.7% |
| ELL | 83.4% |

Table 4: Domain1 HMM Tagger Results.

| Test File | Accuracy |
|---|---|
| Domain1 | 79.8% |
| Domain2 | 85.6% |
| ELL | 82.8% |

Table 5: Domain2 HMM Tagger Results.

best when testing on the corpus it was trained on, which was not always the case for the Stanford tagger.

### 2.3 Brill Tagger

The Brill tagger is a PoS tagging tool that takes another model that has been trained and builds rules to hopefully do a better job of tagging than the model it was supplied. Overall for me, this was marginally true at best. The Brill tagger did technically do a better job, but it took longer than the HMM and Stanford taggers, while only delivering results that were slightly improved. This is clearly seen in the Brill Tables below.

As you can see, the Brill tagger is only on average a third of a percentage point higher than its HMM equivalent. This makes the Brill significantly slower than the HMM, as it not only has to train itself, it also relies on a trained HMM model to train off of.

## 3 Learner English

All three taggers generally performed worse on the English Language Learner corpus', with the exception of the NLTK taggers trained on the ELL corpus. As discussed early, this is likely do to the misspellings of different words, and in some cases the incorrect usage of homonyms (for instance, using the wrong their, there, or they're). The Stanford tagger especially fails to grasp the "Non-Standard" English of the corpus. I used the English default for the tagger rather than the generic one, and this has likely led to the abysmal score.

| Test File | Accuracy |
|---|---|
| Domain1 | 76.7% |
| Domain2 | 76.8% |
| ELL | 92.2% |

Table 6: ELL HMM Tagger Results.

| Test File | Accuracy | HMM Accuracy |
|---|---|---|
| Domain1 | 85.1% | 84.8% |
| Domain2 | 81.3% | 80.7% |
| ELL | 83.7% | 83.4% |

Table 7: Domain1 Brill Tagger Results.

## 4 Conclusion

Clearly, the HMM tagger was the most efficient and most accurate tagger. With very little tuning, the HMM tagger was up and running very quickly, and produced the best results with very little interaction on my part to achieve the high results. Technically the Brill performed better, but the added computation cost makes the HMM tagger the superior tool. In addition to this, the ability of this tagger to more accurately tag the ELL corpus makes it a preferable tool to the Stanford tagger. In addition to this, the difficulties with dealing with non-Standard English corpus' is evident in this assignment, as each tagger struggled more so with the ELL corpus' over the Standard English ones.

| Test File | Accuracy | HMM Accuracy |
|---|---|---|
| Domain1 | 80.1% | 79.8% |
| Domain2 | 86.0% | 85.6% |
| ELL | 83.0% | 82.8% |

Table 8: Domain2 Brill Tagger Results.

| Test File | Accuracy | HMM Accuracy |
|---|---|---|
| Domain1 | 77.0% | 76.7% |
| Domain2 | 77.1% | 76.8% |
| ELL | 92.7% | 92.2% |

Table 9: ELL Brill Tagger Results.