CSCI.GA.2590 Natural Language Processing

Assignment 4

Leslie Huang (LH1036)

February 28, 2018

Results

My minimal implementation of the Viterbi algorithm on the WSJ_24 development corpus achieved accuracy of 94.874136%. After implementing changes to the handling of unknown words, my tagger achieved accuracy of 95.345935% on the WSJ_24 development corpus.

Handling of unknown words

After inspecting the most common unknown words in the test data, I implemented several changes to handle certain categories of words:

- I assume unknown numbers (e.g. "5" or "1,000") are only emitted from the POS state "CD." I handle the word emission probabilities of an unknown number from CD with $Pr(unknownNum|CD) = mean[Pr(word|CD) \forall word \in CDemittedWords]$, and the (unlikely) emissions of an unknown number from any of the other states with $Pr(unknownNum|notCD) = 0$.

- I assume that any word in uppercase is an acronym e.g. "NASA" and that these are only emitted from the POS "NNP." Similar to above, I handle the word emission probability of an unknown acronym or title-cased word from NNP with $Pr(unknownAcronym|NNP) =$

$mean[Pr(word|NNP) \forall word \in NNPemittedWords]$ and $Pr(unknownAcronym|notNNP) = 0$.

- There are also unknown names of people or places not in the known vocabulary. These words are generally in title case (e.g. "Smith") and I assume they are only emitted from "NNP." These are handled in the same manner as acronyms.

- For unknown words that don't meet the above criteria, I assign a small emission probability for all of the possible states.

I trained the tagger on the union of the training and development data (WSJ_02-21 and WSJ_24) before running it on the test corpus WSJ_23 and generating my wsj_23.pos file.

NB:

To run my program, the following arguments should be specified in the command line:

```
python3 tagger.py trainingdata.pos testdata.words -o outputfilename.pos
```

e.g.

```
python3 tagger.py WSJ_POS_CORPUS_FOR_STUDENTS/combined_WSJ_02-21_24.pos WSJ_POS_CORPUS_F
```