Jihyun Kim & Gabor Csapo
Jk4704 & gc1569
March 3 2017

Homework 4: HMM and POS tagging

System description

Our design consists of two python files. The probability.py read through the training corpus and calculates the transition and emission probabilities and stores them in dictionaries. The transducer.py file uses the previous result and implements the Viterbi algorithm to calculate the most probable path.

How to run:

- Make sure everything is in the same folder
- Running transducer.py creates output file named ‘output.txt’.

Out-of-vocabulary words

I looked at what the most common words are that only appear once in the development corpus. They are mostly words ending in -s, -ly, -er, -ing, numbers and hyphen-containing words. Let’s call these OOV categories.
After adding every word to the emissions and transitions probabilities, I loop through the emissions collection to see the words that have only 1 instance. Then I calculate in each ending category, what the probability of each POS tag is.
If a word that only has one instance and doesn’t match the previous category, gets put in the ‘unknown’ category. I calculate the probability of tags in the unknown category as well.

During parsing, if the probability of an unknown word is requested, I first check if it matches any of the OOV categories. If yes, I use the probabilities obtained from training corpus for the category it falls into to see how likely is a word to have a certain POS tag. If not I’ll use the generic unknown word frequency.

Ways to Improve

It seems really hard to debug and improve the program, because it isn’t always clear how we could change the probability calculation without messing up other results. One interesting example is ‘a’. The character ‘a’ appear 10 times as a SYM in the training corpus and the total number of SYM tagged words is 58. That gives a large percentage. It also appear 20000 times as DT and the total number of DTs is 80000. This gives about the same percentage as SYM, but obviously a is more likely to be a DT.
The emissions probability is the probability of a tag being a word. It seems more intuitive to calculate instead of the emissions probability, the probability of a word having a tag. However, we trusted the slides and did it that way, and haven’t tried the seemingly more intuitive method.

We found that the most commonly appearing words, mostly determiners such as ‘the’, ‘any’, ‘a’, or possessive s, were significantly affecting the accuracy of the program, depending on how the training corpus determined the likelihood of the word. In the cases of these high frequency words, we decided to set up custom rules, that automatically determine the POS tag of a list of predetermined words. This significantly improved our results.