Aniq Shahid

CS 5832 – Natural Language Processing

Assignment 2

10/1/2018

Report

Transmission and emission probabilities:

I started by developing a base model for gathering transition and emission probabilities without smoothing and a closed tag set. I developed emission matrix using dictionaries data structure by doing a count for each unique word in the corpus and counting the corresponding tags. I developed transition matrix in the similar way by using each POS tag as a key and counting the number of transitions to different tags. I did not include transition from any tag to "." because of the assumption all the sentences in the corpus finished with tag '.', and therefore tag for the period can always be correctly predicted as ".".

Viterbi algorithm

I used pseudo code from the book for developing Viterbi algorithm. I used transitions from tag "." to any other tag to determine start of the sentence.

Training/Dev distribution and evaluation:

I divided the corpus into four different sets of training and development data using 80 to 20 ratio, and evaluating the accuracy using provided script (eval.py)

The accuracy for unsmoothed model with unknown words across different training/validation sets was around 82 percent. The incorrectly predicted words are a combination of unknown words, low frequency words, and words with multiple POS tags.

Smoothing:

I added add-one smoothing to both transition and emission matrices by adding 1 to the counts and dividing by the sum of token and vocabulary count. The results were improved to 86 percent accuracy. The unknown words were still poorly predicted but the total of low frequency words unpredicted slightly reduced (from 3500 words to 2800 words).

Unknown words:

I tried different methods for tagging unknown words. I started by assigning a default tag of 'NN' for all the unknown words with the assumption that as per morphology many unknown words tend to be some kind of noun. This improved the accuracy by 1 percent since some of the previously untagged words were now getting successfully tagged as 'NN'.

I tried to improve the implementation by not considering emission probability for the unknown words and only use transmission probability since that is the only information available for the unknown words. This changed worsened the results by 4 percent across different test sets, suggesting that transmission probability just by itself can be too noisy for accurate prediction of low frequency words.

I implemented one-count smoothing, where each transmission probability is being multiplied by a fraction where fraction = 1/(sum of total count for a given tag and vocabulary). The idea is the high

probability tags appearing in the corpus are more likely to be the tag for unknown word as opposed to the lower probability tags. Therefore, multiplying by the fraction adjusts the likelihood of a given tag getting selected in proportion to its frequency in the corpus. This implementation improves unknown words prediction by 2 percent, lowering mis-predicted unknown words from 940 to 750.