Viterbi POS Tagger

surya prakash(20CS10038)

1 Introduction

The objective of this assignment is to implement a parts of speech tagger to generate a sequence of parts of speech given a sentence.

2 Implementation

2.1 Training

The first step is to parse the train file and get emission and transition counts. The logic of parsing each line is

- If the line starts with '# sent_id', it marks the start of a sentence. prev_tag is set to 'LINESTART'.
- For other lines, we take the second(token) and fourth(tag) parts of the line, we increment the emission of word from tag and transition of tag from prev_tag counts by 1.

In the second step, we calculate the transition probabilities between tags and store them. The emission probability calculation is done while predicting.

2.2 Viterbi algorithm

For each sentence, we loop through the words in the sentence and calculate the scores for each tag.

For the first word in the sentence, we take 'LINESTART' as the previous tag and computer the score. from the second word, we calculate the score for each possible tag as score(prev_tag, i-1) * transition(tag, prev_tag) * emission(word, tag). We take the tag with the highest score at each word and return the sequence of tags.

2.3 Prediction

This section also requires parsing the file to be predicted as in section 2.1 but whenever a new line's start is seen, we get the predictions for the last line, store them and update the sentence id. For each line, we also write the predicted parts of speech to a tsv file.

3 Results

3.1 Training

The scores for the training dataset are

- Accuracy: 0.83
- Precision: 0.83
- Recall: 0.84
- F1: 0.82
- Number of smoothed words: 0

3.2 Testing

The scores for the testing dataset are

- Accuracy: 0.70
- Precision: 0.70
- Recall: 0.73
- F1: 0.68
- Number of smoothed words: 284