CS 534 : Machine Learning
Homework 3
Liang Huang

Eric Klinkhammer

November 23, 2017

1 Structured Perceptron

1.1 Normal Structured Perceptron

After training for five epochs, the best dev error percentage (with a learning rate of 0.1) was 4.97% on the fourth epoch. The best overall dev error rate was 4.68% after 13 epochs.

1.2 Averaged Structured Perceptron

After training for five epochs, the best dev error percentage (with a learning rate of 0.5) was 5.26% on the fifth epoch (ie, it did not improve). The best overall dev error rate was 4.39% after 15 epochs.

1.3 Time Comparison between Unaveraged and Averaged Perceptron

The averaged perceptron was substantially slower to test, but not substantially slower to run. When testing, a copy of all the weights had to be generated with the averaged weights (scaled by the current value of c) incorporated appropriately. When just running (and not outputting any intermediate scores), the run time was nearly identical.

1.4 Plot of error rates

The averaged perceptron performs better than the unaveraged perceptron on the dev set.

1.5 Unks

All single-count words in the training data are replaced with <unk>, and all zero-count words in the test data (ie, words that appear in the test data but do not appear in the training data) are parsed as <unk>.

The first reason is to avoid overfitting your single sample. If a word only appears once, there will be a one-to-one mapping of that word to a tag, regardless of surrounding words or tags. This is - for many English words - not the correct thing to learn.

Unknowns also provide a way to handle unseen words in the test data, and bias the estimate of the tag toward what is common among unknown words. In other words, if one did not have unknowns, the alternative (assuming one did not ignore it) would be to make the guess strictly based on the tags, effectively looking at what the average word in that position would be. However, unknown words are not uniformly distributed words, and may in fact come from a subset of parts

Type of Time	Unaveraged	Averaged
Real	4.147	0.947
User	4.1	0.911
Sys	0.029	0.022

Table 1: Comparison of unix command 'time' when run with averaged and unaveraged perceptron.

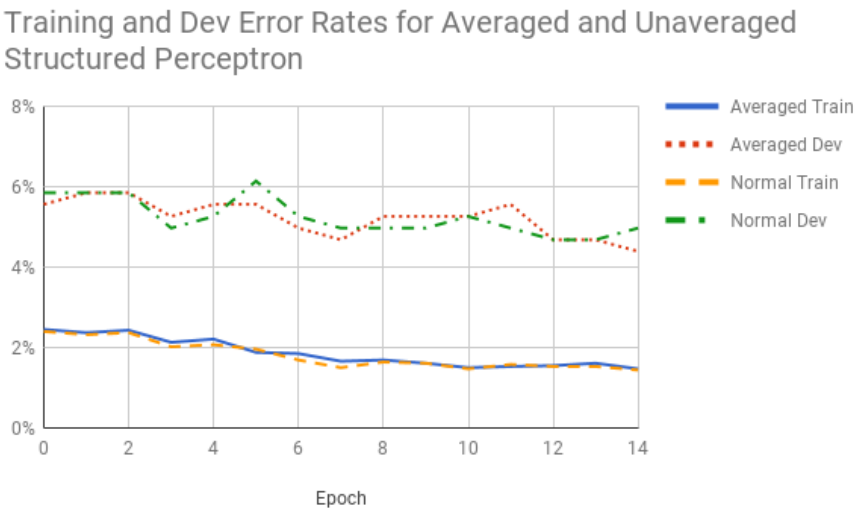

Figure 1:

of speech. Biasing the estimate with only the unknown words instead of the overall dictionary improves performance. For example, consider the sentence "_ house is blue." The correct answer is an adjective (Klinkhammer's). The most likely part of speech would be a determiner so going with only the most likely part of speech would be incorrect. However, while Klinkhammer did not appear in the data set, it is likely other names did. <Unk> represents proper names better than the average word. Moreover, because determiners are so common, it is less likely that you did not encounter it in training. Making a prediction based on the overall bias of the data (as opposed to just the bias of the unknowns) does not make sense, because it is reasonably likely that the training data contains an exhaustive representation of the determiners in the test data. Ironically, it is precisely because determiner is such a likely choice that we should disregard it as being a likely candidate for a zero-count word.

Basically, <unk> is useful at test time because it is better to consider only the set of other one-count words than to consider all words.

2 Feature Engineering

I used a tag trigram (t0_t-1_t-2) and was able (with an averaged perceptron) to get an error rate of 4.97% on the dev set.

Please find the labeled dev and test data attached.

3 Debrief

This section is duplicated in included file debrief.txt and in latex document.

3.1 Hours Spent on Assignment

Including the writeup and catching up on the material, this assignment took about 4 hours.

3.2 Difficulty

I would rate this assignment as easy. With regards to the extent of provided code, I think it is appropriate when the provided code removes boiler plate or non-ML tasks. The Viterbi solution, however, seems a crucial part of POS tagging. With that code provided, this assignment was basically just another rehash of perceptron.

3.3 Lecture Speed

When teaching, just right. However, the bulk of lectures seems to be review of some kind.

3.4 Other Comments

No comment.