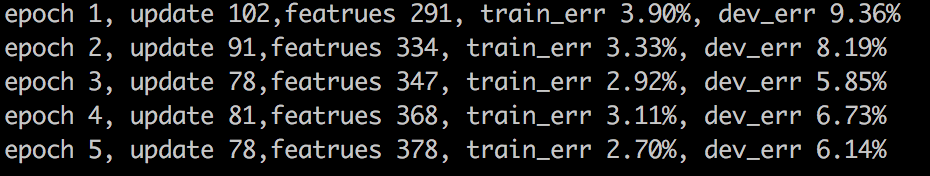
Machine Learning, Individual HW 3: Structured Prediction

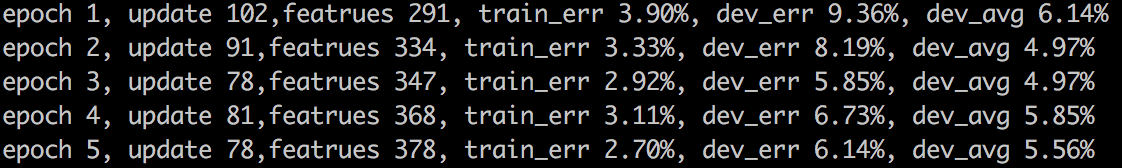
Xiao Tan tanx2@oregonstate.edu

1. Structured Perceptron
   1. First just use two feature templates: ⟨t, t′⟩ and ⟨t, w⟩. Training unaveraged perceptron for 5 epochs. Which epoch gives the best error rates on dev?



Training unaveraged perceptron for 5 epochs, I got the results as the picture shows upon. The best error rates will be 5.85%, while epoch 3.

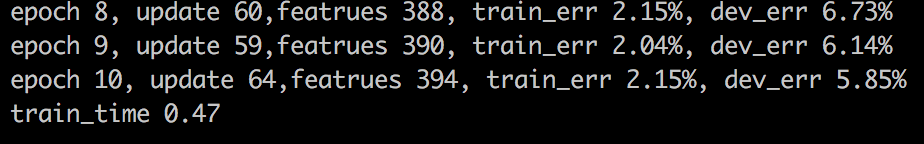
* 1. Now implement the averaged perceptron. What is the new best error rate on dev?

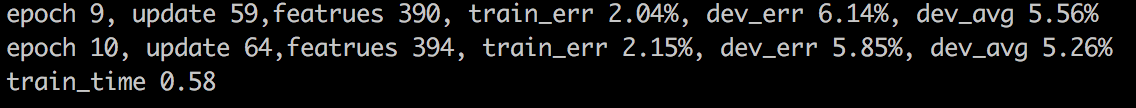


Here are the results of averaged perceptron for 5 epochs, I got the best error rate as 4.97%

in epoch 2 and epoch 3.

* 1. How much slower is averaged perceptron? Compare the time between unaveraged and averaged version.





Picture 1 and 2 show the time used by unaveraged and averaged perception for 10 epoch. Averaged perceptron will cost more time than unaveraged, but not too much.

* 1. Plot a single plot that includes four curves: {unaveraged, averaged} x {train err, dev err}. What did you find from this plot?

Because unaveraged and averaged train err will be the same, so we can only see 3 curves on the plot. The result is showed in following graph.



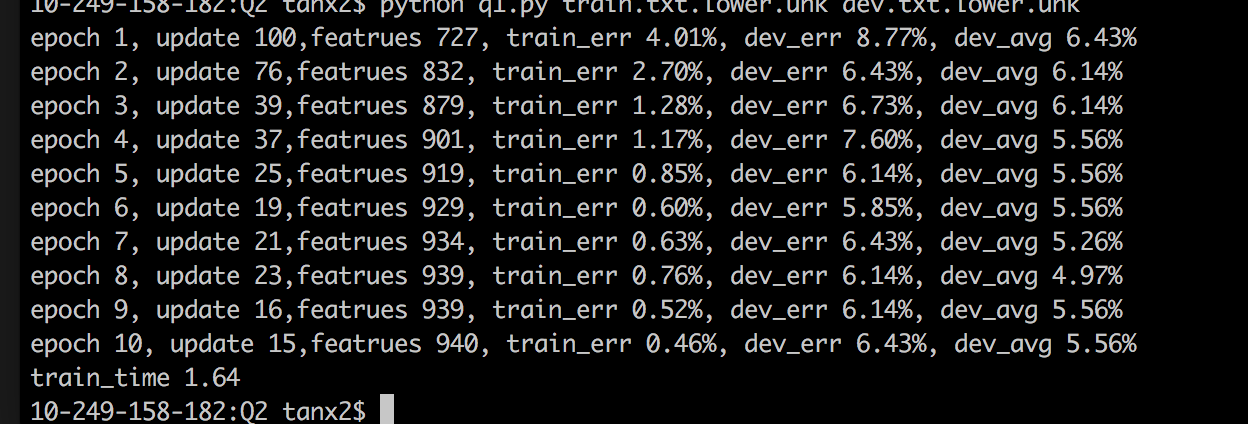
* 1. Explain why we replaced all one-count and zero-count words by <unk>.

Because training data won’t have every word we may test in the future, replaced all one-count and zero-count words by <unk> will make this model can handle unknown words, no matter what words it is, just treat those low frequency words or unknown words as <unk>.

1. Feature Engineering (based on averaged perceptron)

I tried t0\_(t-1\_t-2) and t0\_w0 as template to solve this problem. Use t-1\_t-2 instead of t-1 only to achieve trigrams.

Then, I got the following result. The best dev\_avg is 4.97% in epoch 8.



Every template should include t0, I assume because t0 is the tag position the sentence makes mistake, so that we can know which features we need to reward and which features we need to punish.

Moreover, with the best accuracy, I’ve generated dev.lower.unk.best and test.lower.unk.best in folder.