Joshua Shapiro

22 November 2016

Assignment 3

1. Learning Curve Analysis
   1. Learning Curve Table

|  |  |
| --- | --- |
| Training Size (# of Sentences) | LAS Performance on Dev Data |
| 4 | 37.59% |
| 40 | 63.99% |
| 400 | 77.11% |
| 4000 | 84.96% |
| Whole Dataset (about 40,000) | 88.45% |

* 1. Is there a pattern? Predict accuracy of 400,000 sentences and 4,000,000 sentences.

It seems like for every increase in the training size, the error is reduced by half of the last reduction (40-400 is 14, 400-4000 is 7). Therefore, 400,000 would be around 90.5% and 4,000,000 would be around 91.5%.

1. Parsing Algorithms
   1. Performance of dev data for each parsing algorithm

|  |  |
| --- | --- |
| Parsing Algorithm | LAS Performance on Dev Data |
| Nivre Standard | 88.69% |
| Stack Eager | 88.68% |
| Stack Lazy | 88.68% |
| Stack Proj | 88.71% |

* 1. Best performing algorithm

The Stack Projective algorithm has the highest accuracy out of the algorithms tested above.

1. Training POS Tagger
   1. POS tagging accuracy on the development dataset for all explored tagging parameter combinations

|  |  |  |  |
| --- | --- | --- | --- |
| Training | | Tagging |  |
| Order of tag transition probability | Order of emission probability | Number of tags to keep | POS Accuracy |
| 1 | 1 | 1 | 95.69% |
| 2 | 2 | 2 | 96.23% |
| 3 | 3 | 10 | 96.22% |
| 1 | 1 | 2 | 95.79% |
| 2 | 2 | 10 | 96.25% |
| 3 | 3 | 1 | 96.13% |

* 1. Best performing parameters

The best performing parameters was 2 for order of tag transition probability and emission probability, and 10 for number of tags to keep.

* 1. Why are those the best parameters? Are you surprised?

Out of the 6 tests, it is not surprising that these are the best parameters. They are the defaults for hunPOS, so I would hope the default values would perform the best.

1. Evaluating Parser with Predicted Tags
   1. Parsing accuracy using gold POS tags and parsing accuracy using predicted POS tags

|  |  |
| --- | --- |
| POS tags | LAS Performance on Dev Data |
| Gold POS tags (using Stack Proj) | 88.71% |
| Predicted POS tags (using hunPOS defaults & stack Proj) | 85.86% |

* 1. Identify the best performing setting and explain your insights.

Using the .conll file with gold POS tags unsurprisingly yielded better performance. This is because the only source of error for it was running through the maltparser. With the .conll file with predicted POS tags, there were two sources of error. The first was using hunPOS to tag the POS in the first place, and the second was running it through the maltparser. As 96.25% (accuracy of predicted POS) of 88.71(LAS performance on gold POS) is 85.38, the LAS performance of 85.86% is a little better than expected.

1. Parsing Error Analysis
   1. Analysis of different kinds of errors

The lowest accuracy for POSTags was for punctuation, followed by verbs. While it makes sense that verbs may be incorrectly tagged, it seems more improbable that punctuation would be tagged incorrectly, especially with a 29% error. When looking at words where most of the errors occur, there seemed to be some confusion with prepositions “to” and “by”, but it also had problems with the punctuation ` `. I don’t know why this is happening, but it confirms the result above.

* 1. Ideas to solve errors and improve accuracy
     1. I would first incorporate rule based approaches in addition to statistical approaches. Since there seemed to be issues with punctuation, it wouldn’t be too challenging to create rules based on punctuation for tagging instead of depending on statistics. These rules would also be helpful to handle obscure words that get tagged wrong such as names or some of the incorrectly tagged verbs.
     2. Second I would attempt to rerun the parser using more POS tags and less POS tags. Perhaps changing the specificity of the tags would yield better results on classifying obscure words. Using too many tags may result in overfitting though, so it would be important to test multiple sets of tags to see where the greatest accuracy comes from.

1. Parsing Competition
   1. Detailed description of system’s settings

I used the best settings determined in the previous exercises. For my case, that meant using hunPOS with default settings (train with 2 &2 and tag with 10) and maltParser with the stackProj parsing algorithm. To improve my accuracy, I combined the training and dev data together into one training set when running hunPOS and maltParser. Typically this is not a good idea as you no longer can test accuracy on something other than the test set (since I got rid of my dev set). However, since the rules only stated I couldn’t get more gold data from outside the assignment I figured it wasn’t cheating to use my dev data in addition to my training data to train my parser and pos tagger. I hope that the increase in training data will yield more accurate results on my test data.

* 1. Parsed test data

It is located in the Question6Submission folder called test.out.conll.