

LING571 – Hw4

Probabilistic Cocke-Kasami-Younger Parsing

1. Overview

I worked alone to complete this assignment.

2. Challenges

The biggest challenge for this assignment was reading the assignment brief.

In the end, I created the following mapping, which helped me out a lot. (Please feel free to use it for future cohorts.)

treebank_filename	parses.train
output_PCFG_file	hw4_trained.pcfg
test_sentence_filename	sentences.txt
baseline_parse_output_filename	parses_base.out
input_PCFG_file	hw4_trained.pcfg
improved_parse_output_filename	parses_improved.out
baseline_eval	parses_base.eval
improved_eval	parses_improved.eval

3. Learning Outcomes

This was an interesting assignment and a good review of condition probabilities. Having already done Ling 572 and Ling 570, I was pleased that the calculation of conditional probability was intuitive and relatively straightforward.

My base implementation achieved accuracy of 99.04% with a runtime of about 7 seconds. I think that this is pretty good, both in terms of efficiency and accuracy. In part, I believe the high accuracy is because my original CNF implementation relied on a back-trace mechanism that returned *all* possible parses and it was simple to adapt that algorithm to also return the probability of each parse and select the highest *global* parse, as opposed to making a greedy, local assessments along the way.

4. Improvements

Make sure to discuss the improvements you implemented and compare your 'improved' results to your baseline results.

The baseline implementation achieved 99.04% accuracy even without handling out-of-vocabulary (OOV) symbols; so certainly, being able to handle OOV would go a long way to improving the accuracy even further.

I decided to make improvements to my parser and did so in the following ways:

- Using log probabilities rather than simple probabilities.

This really didn't have nearly as much of an effect as I expected. This didn't appear to affect tagging accuracy at all with both versions reporting 99.04%.

However, a marginal improvement was noticed in Bracketing Recall, Bracketing Precision, and Bracketing FMeasure, which all improved equivalently from 87.74 to 88.05.

Average crossing, however, seemed to degrade from 0.53 to 0.49.

- Attempted to handle out-of-vocabulary (OOV) words:

Unfortunately, this was not as successful as I had hoped. The solution was able to produce parses for all-but-two of the input sentences, however these sentences were flagged as errors by the evalb assessment. This was indicated by a decrease in the reported 'Skip sentences' (which decreased from 6 to 4) and a corresponding increase in the reported 'Error sentences' (which increased from 0 to 4).

The mechanism I employed to handle OOV words relied on recognizing an OOV word when building up a parse and populating the table with all possible transitions at that point; however with a reduced probability in comparison to the other productions. (More exactly, a probability of half the lowest probability was assigned to each such production.) In other words, any OOV was assumed to be a valid token for *every* word type. Then, the highest probability sentence was selected from those successful parses.

5. Closing Comments

A very interesting assignment. I would have liked to have been able to spend more time attempting to improve my algorithm, but this was deemed infeasible by management (i.e. my wife).

6. Completeness

I was able to complete the assignment.