Syntactic Parsing: Assignment 3

Johannes Gontrum

Department of Linguistics and Philology
Uppsala University

March 26, 2018

1 Testing on a real treebank

Comparing the derived trees of the `oracle.py` script with the actual treebank, I notice that the oracle often has a tendency to suggest the *root* label, even though it is wrong.

In fact, as far as I could see, all differences in labels can be attributed to this mistake, though I did not see a consistency in the part-of-speech tag of the misclassified token. A similarity I noticed, however, is that the error often occurs in the middle or the end of the sentence. Additionally, these errors seem to appear in groups: If a sentence contains one mistake, there is a high probability that there will be others in the same sentence.

This leads me to the hypothesis that it must have something to do with the *shift-reduce* ambiguity in the arc-eager oracle: Whenever the oracle encounters a situation that allows both transitions, it always chooses *shift*.

If, however, an item is shifted to early from the buffer to the stack, *left arc* transitions that might be needed later in the sentence, cannot be performed. Instead, when the buffer is empty, many right arcs are performed that link the token to the root.

I believe that this mistake could be counteracted by using backtracking: Whenever the oracle predicts a second root transition, we should go back to the latest ambiguous decision, change it and try again. If this does not lead to the success, the next ambiguity is selected and so on. I can imagine that this approach might lead to problems when there are more one incorrect root arcs. Another possibility would be to try out all different combinations of all the *shift-reduce* decisions in a sentence until a correct sentence is found. This brute-force approach might, however, get too complex for long sentences.