LING 571 Project 1: Analysis

Stefan Behr Kathryn Nichols

February 1, 2013

The process of converting the original grammar to Chomsky Normal Form involved essentially two types of transformations: the replacement of nodes in long and hybrid productions, and the deletion of nodes in unit productions. The former operation would be simple to undo, since dummy nodes could just be removed from the trees. The latter, however, would involve storing a history of the unit productions and those they combined with from which we could reconstruct the deleted nodes. From these records we would know which rules should be extended, and what nodes should intervene, some of which could have been eliminated from the grammar entirely during the conversion process.

It's almost impossible to tell with certainty from this data as is whether length or structure causes longer running time since there aren't enough examples of unambiguous, long sentences to compare against. The sentence with the most parses was "An average adult male ..." with 144 due to two prepositional phrases compounded by a coordinating conjunction. It took our parser 13.8 seconds to parse this sentence 100 times. Although this is also the longest sentence, the number of parses is undoubtedly the source of the long run time. The sentences "Do you have pain ..." and "Few people privy ..." are each 13 tokens long, but the first resulted in 10 parses which took 0.78 seconds to generate 100 times, while the second only had two parses, which took 0.597 seconds to generate 100 times. The time difference is not substantial, but it supports the (correct) intuition that it is the number of structural ambiguities in a sentence (and the correspondingly increased size of the search space), rather than the length of the sentence, that most contributes to slowed run times.

To explore this further, we passed two sentences, each with 5 or 10 prepositional phrases ("Do you have pain[in the muscles]*{5|10}?"), to our parser. The first sentence with 5 prepositional phrases was 20 tokens long and had 84 parses for a run time of 0.045s (one iteration). With 10 prepositional phrases, the number of tokens increased to only 35, yet the number of parses shot to 33,592 and the run time was 300 times longer at 13.92s (still just one iteration).

The main area of improvement we see for the parser is in the storing of production rules of the grammar loaded into the parser. Our current implementation uses a dictionary with left hand sides of rules as keys, mapped to lists of corresponding right hand sides. This implementation detail forces the parser, in the worst case, to look through the entirety of each list value in the dictionary while attempting to construct the list of all non-terminal symbols which appear as the left hand sides of productions with a given right hand side.