Elements of Language Processing and Learning

Assignment 1

Group Jaysinh Sagar 10318771, John Zervos 10333630, Liveris Avgerinidis 10318828

For this assignment, we are asked to compute the probability of a given sentence tree. In order to do this, we use rule look up table for parent->child node pairs and then retrieve the learned probability of that pair. For this, we consider 3 types of node relations. First being where each parent has only 1 child where we apply unary parent-child rule. Another type of relation we have is when a parent has 2 children for which we apply the binary parent-children rules. Lastly, we have a third type of relation where the parent has a child which is a leaf node of that tree illustrating a natural language word of the sentence where we apply lexicon rules. Based on the presence of each of these parent-child relations in a given tree, we can easily compute the trees overall probability.

The first step for us was to find a way to traverse through the annotated tree structure in order to get word-token pairs or parent-child pairs. For this, we try an iterative approach as well as a recursive approach. Both of these approached work well in our case where in our submitted implementation we have the recursive method. For traversal over the tree, we implement a method called traversalAnnotatedTree in the baselineCkyParser.java class where we take in an annotated tree and generate a matrix of parent->child pairs along with the type of child with that relations log probability. We illustrate a part of this table below in Table 1. Once we generate this matrix for the tree, we simply add up over all log probabilities we attain from 2 custom methods we define in the grammar.java class, getUnScore and getBiScore. Each of these functions matches pairs of given parent and child labels and returns the logged score of that pair.

Parent	Child	No./type Children	Log Probability
ROOT	S	1	0.1
S	INTJ	2	0.33
INTJ	NP	1	0.2
NP	“No”	“lexicon”	0.002
S	@S->_INTJ	2	0

Table 1. Matix table of tree token information.

From our experiment, we draw results for test sentences shown in table 2. We also notice some -infinity values which represent absence of the probability of a certain rule for a parent-child pair.

```
0:  logscore =  -50.99367882096118
1:  logscore =  -Infinity
2:  logscore =  -39.85517820793024
3:  logscore =  -41.67112250048548
4:  logscore =  -92.5165915926479
5:  logscore =  -Infinity
6:  logscore =  -101.62447403037062
7:  logscore =  -68.99276774663788
8:  logscore =  -55.708379070109174
9:  logscore =  -57.39394243941492
Total log prob: -Infinity
```