

Python Course: Lecture 18

February 14, 2006

1 Bracket Notation

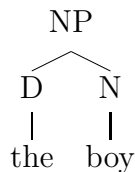
- If we want to write a tree parse as a single text string we use *bracket notation*.
(S (NP He) (VP (V put) (NP (D the) (NP book)) (PP (P on) (NP (D the) (NP table))))))
- This is okay for short sentences, though it quickly becomes hard for a human to read.
- There is a standard format for syntactic information in trees called the Penn Treebank which uses brackets in ASCII files.
- Parsing bracketed sentences is a simple case of the general parsing problem.

2 Shift-Reduce Bracket Parsing Algorithm

Say we want to parse the following string

(NP (D (the)) (N (boy)))

into the tree structure



Assume we know how to tokenize the string into brackets and whitespace-delimited text. So we have the following tokenized input.

Input	(NP	(D	(the))	(N	(boy)))
--------------	---	----	---	---	---	-----	---	---	---	---	---	-----	---	---	---

Step through the input tokens one at a time, removing them from the input and pushing them onto a stack until we reach the first close parenthesis. At this point our stack would look like this.

Input)	(N	(boy)))
Stack	(NP	(D	(the))

We have ensured that the rightmost “)” is the only “)” on the stack. So we can look back in the stack until we find the first matching “(” and use the information between them to create a new node. In this case we would create a “the” node without a parent. We then replace the tokens in the stack used to create the node with the node object itself.

Input)	(N	(boy)))
Stack	(NP	(D		the		

We then repeat the above process, pulling tokens off the input list and pushing them onto the end of the stack. When we reach the next “)”, we convert everything between it in the first matching “(” into a node and add that node onto the end of the stack. Note that the “the” node on the stack will be made into a child of the new node.

Input	(N	(boy)))
Stack	(NP	D		the		

You can continue this process, building a parse stack containing subtrees and labels and “(” corresponding to nodes that have yet to be created.

			Input)
Stack	(NP	D	N
			the	boy

If the parentheses were properly matched, the last “(” in the input will leave you with a single tree node on the stack. This is your fully-parsed tree.

	Input
Stack	<pre> NP / \ D N the boy </pre>

This is an example of a *shift-reduce* parsing algorithm where the shift operation is the addition of new input tokens and the reduce operation is the combination of items at the end of the stack into a single subtree.

This is an efficient way to parse bracketed tree strings that makes use of that Python lists can contain heterogeneous sets of objects.