Parsing: Example for k-best parsing

Laura Kallmeyer

SS 2021, Heinrich-Heine-Universität Düsseldorf

Consider the PCFG G with $N = \{S, A\}$, $T = \{a\}$, start symbol S and productions

$0.1(1) : N \to NN$	$0.2(0.7) : N \to AN$	$0.1(1) : N \to red$	$0.1(1) : N \to green$
$0.1(1) : N \to car$	$0.1(1) : N \to bike$	$0.2(0.7) : N \to camping$	$0.1(1) : N \to house$
$0.3(0.5) : A \to nice$	$0.25(0.6) : A \to ugly$	$0.2(0.7) : A \to red$	$0.25(0.6) : A \to green$

For weights, we use $|log_{10}(p)|$ (given in brackets).

As outside estimates for sentence length 4, we have

$out(A, 0, 4, 0) = \infty$, $out(N, 0, 4, 0) = 0$

$out(A, 0, 3, 1) = 1.4$, $out(A, 1, 3, 0) = \infty$
$out(N, 0, 3, 1) = 1.7$, $out(N, 1, 3, 0) = 1.2$

$out(A, 2, 2, 0) = \infty$, $out(A, 1, 2, 1) = 2.6$, $out(A, 0, 2, 2) = 2.6$
$out(N, 0, 2, 2) = 2.9$, $out(N, 1, 2, 1) = 2.9$, $out(N, 2, 2, 0) = 2.9$

$out(A, 3, 1, 0) = \infty$, $out(A, 2, 1, 1) = 4.3$, $out(A, 1, 1, 2) = 3.8$, $out(A, 0, 1, 3) = 3.8$
$out(N, 3, 1, 0) = 4.1$, $out(N, 2, 1, 1) = 4.1$, $out(N, 1, 1, 2) = 4.1$, $out(N, 0, 1, 3) = 3.8$

We perform a k-best weighted deductive parsing of the input *red ugly camping car* with $k = 2$, following the Pauls & Klein algorithm.

I-items: first component is viterbi inside score, second SX outside estimate.
O-items: first component is viterbi inside score, second viterbi outside score.
tree items: first component is score of the tree, second viterbi outside score, except for the axioms where the second is the SX outside estimate.

We give only the subsequent agenda contents, marking the best item each time. The chart is then understood as containing everything that has been marked in the previous agenda contexts. The sequence of agenda contents is as follows (best item that is processed next is in bold, newly added items are in red):

I items	O items	tree items
(0.6,3.8):I[A,1,2]		**(0.6,3.8):[A(ugly),1,2]** (0.7,3.8):[A(red),0,1]
(0.7,3.8):I[A,0,1]		(0.7,4.1):[N(camping),2,3] (1,3.8):[N(red),0,1] (1,4.1):[N(car),3,4]
(0.7,4.1):I[N,2,3]		
(1,3.8):I[N,0,1]		
(1,4.1):I[N,3,4]		
(0.7,3.8):I[A,0,1]		**(0.7,3.8):[A(red),0,1]** (0.7,4.1):[N(camping),2,3] (1,3.8):[N(red),0,1]
(0.7,4.1):I[N,2,3]		(1,4.1):[N(car),3,4]
(1,3.8):I[N,0,1]		
(1,4.1):I[N,3,4]		
(0.7,4.1):I[N,2,3]		**(0.7,4.1):[N(camping),2,3]** (1,3.8):[N(red),0,1] (1,4.1):[N(car),3,4]
(1,3.8):I[N,0,1]		
(1,4.1):I[N,3,4]		
(1,3.8):I[N,0,1]		**(1,3.8):[N(red),0,1]** (1,4.1):[N(car),3,4]
(2.0,2.9):I[N,1,3]		
(1,4.1):I[N,3,4]		
(2.0,2.9):I[N,1,3]		(1,4.1):[N(car),3,4]
(1,4.1):I[N,3,4]		
(1,4.1):I[N,3,4]		**(1,4.1):[N(car),3,4]**
(3.4,1.7):I[N,0,3]		

(3.4,1.7):I[N,0,3] (4,1.2):I[N,1,4] (2.7,2.9):I[N,2,4]		
(4,1.2):I[N,1,4] (5.4,0):I[N,0,4] (2.7,2.9):I[N,2,4]		
(5.4,0):I[N,0,4] (2.7,2.9):I[N,2,4]		
(2.7,2.9):I[N,2,4]	**(5.4,0):O[N,0,4]**	
(2.7,2.9):I[N,2,4]	**(0.7,4.7):O[A,0,1]** (4.0,1.4):O[N,1,4] (3.4,2):O[N,0,3] (1,4.4):O[N,3,4] (1,5):O[N,0,1]	
(2.7,2.9):I[N,2,4]	**(4.0,1.4):O[N,1,4]** (3.4,2):O[N,0,3] (1,4.4):O[N,3,4] (1,5):O[N,0,1]	
(2.7,2.9):I[N,2,4]	**(3.4,2):O[N,0,3]** (1,4.4):O[N,3,4] (2,3.4):O[N,1,3] (1,5):O[N,0,1]	
(2.7,2.9):I[N,2,4]	**(1,4.4):O[N,3,4]** (2,3.4):O[N,1,3] (1,5):O[N,0,1]	
(2.7,2.9):I[N,2,4]	**(2,3.4):O[N,1,3]** (1,5):O[N,0,1]	
(2.7,2.9):I[N,2,4]	(1,5):O[N,0,1]	**(2,3.4):[N(A(ugly))(N(camping)),1,3]**
(2.7,2.9):I[N,2,4]	(1,5):O[N,0,1]	**(3.4,2):[N(A(red))(N(A(ugly))(N(camping))),0,3]** (4,1.4):[N(N(A(ugly))(N(camping)))(N(car)),1,4] (4,2):[N(N(red))(N(A(ugly))(N(camping))),0,3]
(2.7,2.9):I[N,2,4]	(1,5):O[N,0,1]	**(4,1.4):[N(N(A(ugly))(N(camping)))(N(car)),1,4]** (5.4,0):[N(N(A(red))(N(A(ugly))(N(camping)))(N(car)),0,4] (4,2):[N(N(red))(N(A(ugly))(N(camping))),0,3]
(2.7,2.9):I[N,2,4]	(1,5):O[N,0,1]	**(5.4,0):[N(N(A(red))(N(A(ugly))(N(camping)))(N(car)),0,4]** (5.4,0):[N(A(red))(N(N(A(ugly))(N(camping)))(N(car))),0,4] (4,2):[N(N(red))(N(A(ugly))(N(camping))),0,3]
	first goal item found	
(2.7,2.9):I[N,2,4]	(1,5):O[N,0,1]	**(5.4,0):[N(A(red))(N(N(A(ugly))(N(camping)))(N(car))),0,4]** (4,2):[N(N(red))(N(A(ugly))(N(camping))),0,3]
	second goal item found	

Consequently, the two best parse trees (with equal probability) are: