Maico Timmerman
10542590
maico.timmerman@gmail.com

Homework Assignment 2
Natural Language Processing 1, 17/18

2017-11-21

Questions 1

The Chomsky Normal Form of the grammer is written as:

$$\begin{array}{ccc} S & \rightarrow & NP\ VP \\ S & \rightarrow & I\ X \\ X & \rightarrow & VP\ PP \\ NP & \rightarrow & Det\ N \\ VP & \rightarrow & V\ NP \\ PP & \rightarrow & Pre\ NP \\ I & \rightarrow & \textit{I} \\ VP & \rightarrow & \textit{ate} \\ V & \rightarrow & \textit{ate} \\ Det & \rightarrow & \textit{the} \mid \textit{a} \\ N & \rightarrow & \textit{fork} \mid \textit{salad} \\ Pre & \rightarrow & \textit{with} \end{array}$$

Question 2

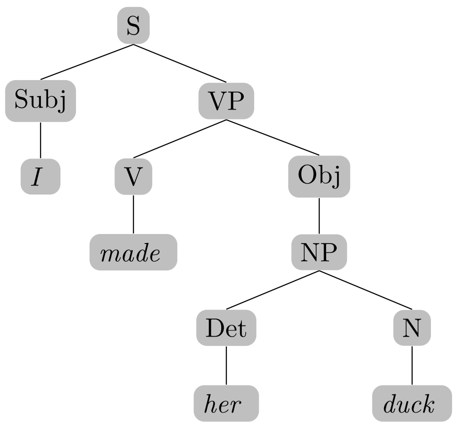
(a) We can write the rules in cell **A**:

$$S \quad \rightarrow \quad Subj\ VP \quad (0.018)$$

The rules in cell **B**:

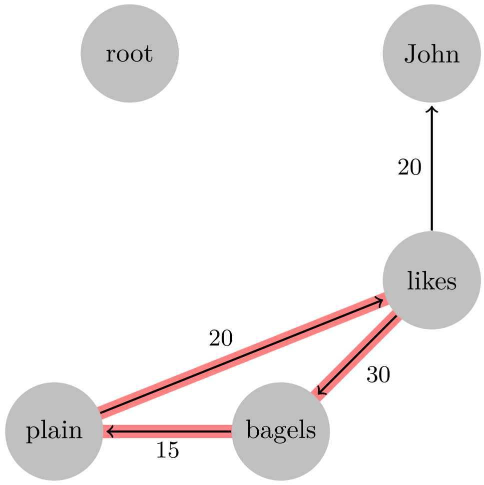
$$\begin{array}{clll} S & \rightarrow & V\ Small & (0.0096 \approx 0.010) \\ S & \rightarrow & V\ Obj\ Obj & (0.0072 \approx 0.007) \\ S & \rightarrow & V\ Obj & (0.06) \end{array}$$

Resulting in a final tree:

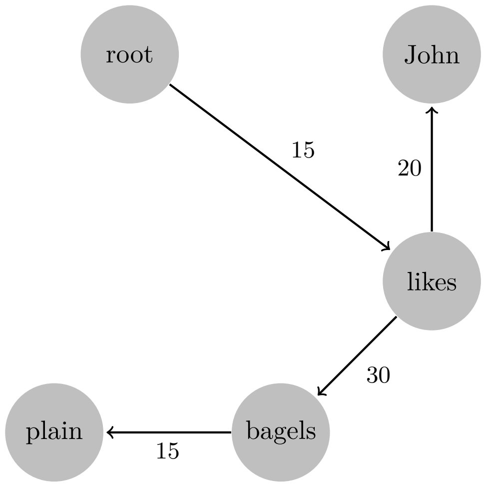


Question 3

(a) For every non-root we keep the highest incoming edge, from where we can see a cycle between the nodes of ‘plain’, ‘bagels’ and ‘likes’:



(b) After completing the algorithm we result in:



Question 4

(a) We define a set A as the set as created arcs, with initial value $A = \emptyset$. From this we start parsing according to Nivre's algorithm[1].

Transition	Stack	Buffer	Arcs (A)
		A koala eats leafs and barks	A
SHIFT	A	koala eats leafs and barks	A
LEFT-ARC(det)		eats leafs and barks	$A = A \cup \text{det(koala, A)}$
SHIFT	koala	eats leafs and barks	A
LEFT-ARC(nsubj)	eats	leafs and barks	$A = A \cup \text{nsubj(eats, koala)}$
RIGHT-ARC(dobj)	eats leafs	and barks	A
RIGHT-ARC(cc)	eats leafs and	barks	$A = A \cup \text{cc(leafs, and)}$
REDUCE	eats leafs	barks	A
RIGHT-ARC(conj)	eats leafs barks		$A = A \cup \text{conj(leafs, barks)}$
REDUCE	eats leafs		A
REDUCE	eats		A

As our buffer is empty and there is only one word left on the stack, we see that the root of the sentence is "eats". We then add the root transition to A: $A = A \cup \text{root(root, eats)}$. This results in $A = \{\text{det(koala, A)}, \text{nsubj(eats, koala)}, \text{dobj(eats, leafs)}, \text{cc(leafs, and)}, \text{conj(leafs, barks)}, \text{root(root, eats)}\}$.

[1]An efficient algorithm for projective dependency parsing, Joakim Nivre, 2003