

# Dependency Parsing exercises: Graph-based parsing I

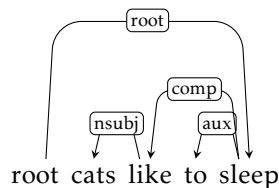
Deadline: 29.06.2021. Please send the homework to [evang@hhu.de](mailto:evang@hhu.de) with subject "dependency homework" and an attachment named "ex8\_lastname1\_lastname2.pdf".

1. We would like to parse the sentence "Cats like to sleep". To do so, we will use the weight function  $w$ , defined as follows:

$w(\text{root}, \text{cats}) = 1$	
$w(\text{root}, \text{like}) = 2$	
$w(\text{root}, \text{to}) = 0$	
$w(\text{root}, \text{sleep}) = 3$	$w(\text{to}, \text{cats}) = 2$
	$w(\text{to}, \text{like}) = 0$
$w(\text{cats}, \text{like}) = 0$	$w(\text{to}, \text{sleep}) = 0$
$w(\text{cats}, \text{to}) = 2$	$w(\text{to}, \text{root}) = 0$
$w(\text{cats}, \text{sleep}) = 0$	
$w(\text{cats}, \text{root}) = 0$	$w(\text{sleep}, \text{cats}) = 3$
	$w(\text{sleep}, \text{like}) = 3$
$w(\text{like}, \text{cats}) = 4$	$w(\text{sleep}, \text{to}) = 3$
$w(\text{like}, \text{to}) = 2$	$w(\text{sleep}, \text{root}) = 0$
$w(\text{like}, \text{sleep}) = 5$	
$w(\text{like}, \text{root}) = 0$	

where  $w(i, j)$  is the weight of the arc from node  $i$  to node  $j$ . Here, we ignore the labels of the arcs (only the weight of the best label is given).

- (a) According to the weights for the arcs starting from root, it would be easy to think that "sleep" should be the head of the sentence. This would lead to this parse:



What is the weight of this parse? Is this the best parse that can be obtained with "sleep" as head of the sentence?

- (b) Using Chu-Liu-Edmonds' algorithm, determine the best parse of the sentence with the given weights. Please show all the intermediate graphs used during the parsing process. Compare the weight of the resulting analysis with the one of the previous question.
- (c) Optional task for extra credit: Consider a word  $w$  that was never found in the corpus which was used to learn the weights. What are the problems encountered by the algorithm when  $w$  occurs in a sentence to be parsed? Think also about the labels of the dependencies (which we can usually ignore). Which solution(s) can you propose to solve these problems?