CS 4650/7650
Dependency parsing[1]

Jacob Eisenstein

October 2, 2014

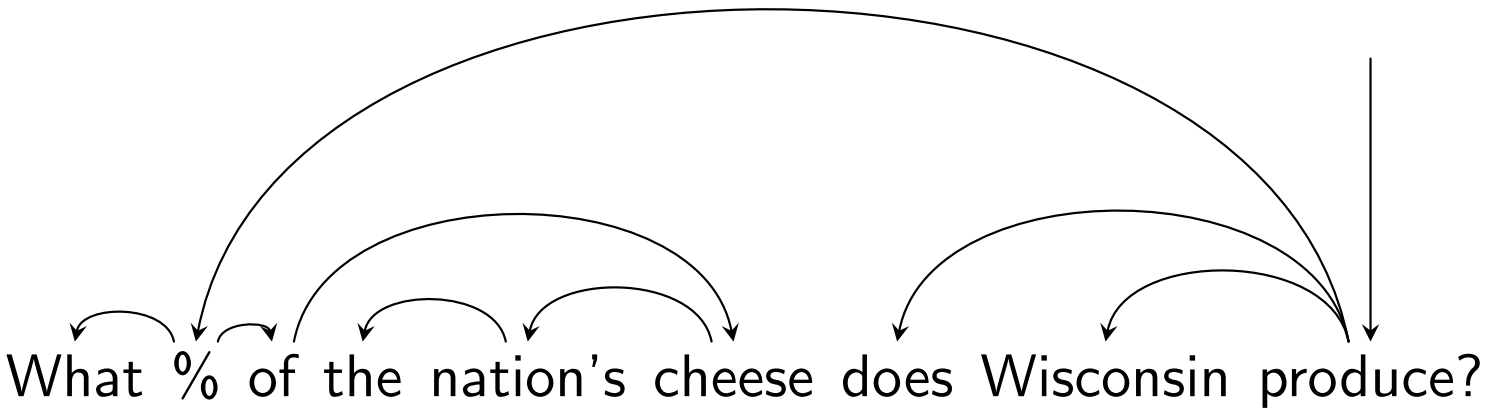
[1] With slides borrowed from Joakim Nivre and Ryan McDonald's tutorial on Dependency Parsing.

Dependency parsing in action

Dependency parsing is used in many real-world applications, like question answering (Cui et al, 2005):

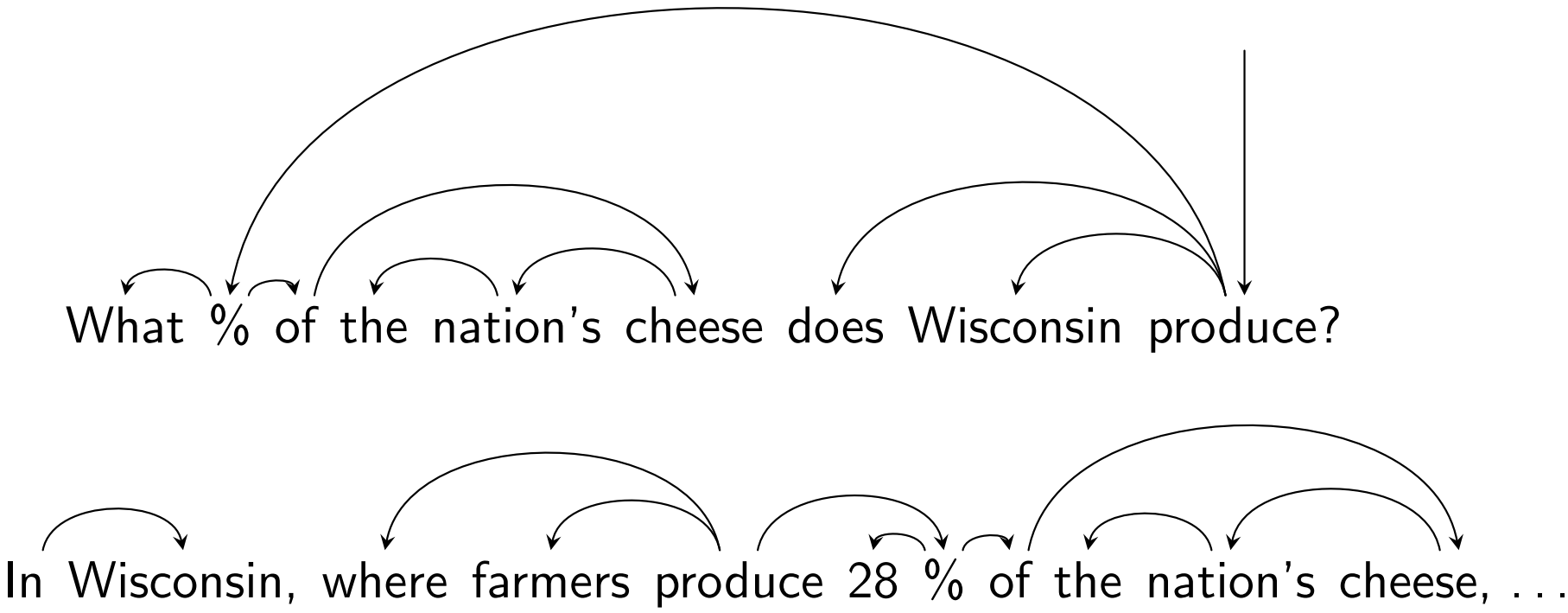
Dependency parsing in action

Dependency parsing is used in many real-world applications, like question answering (Cui et al, 2005):



Dependency parsing in action

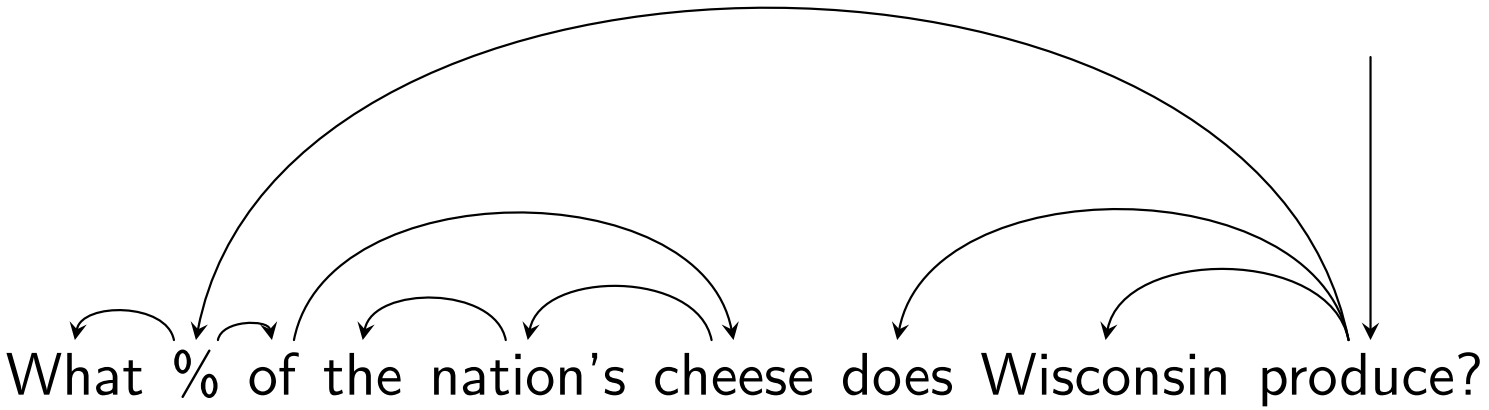
Dependency parsing is used in many real-world applications, like question answering (Cui et al, 2005):



Dependency parsing in action

Question answering works by searching for statements which match well against the query.

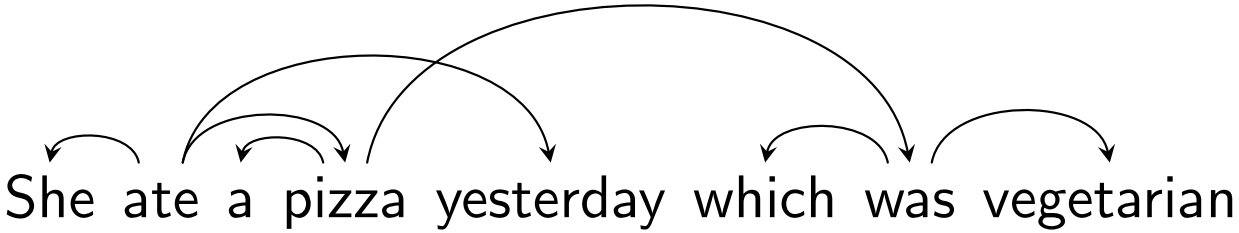
- In the surface form of the question, *produce* and % are six words apart.
- But in the dependency parse, they're adjacent.



Projectivity

In **projective** dependency parsing, there are no crossing edges.

- Crossing edges are rare in English:

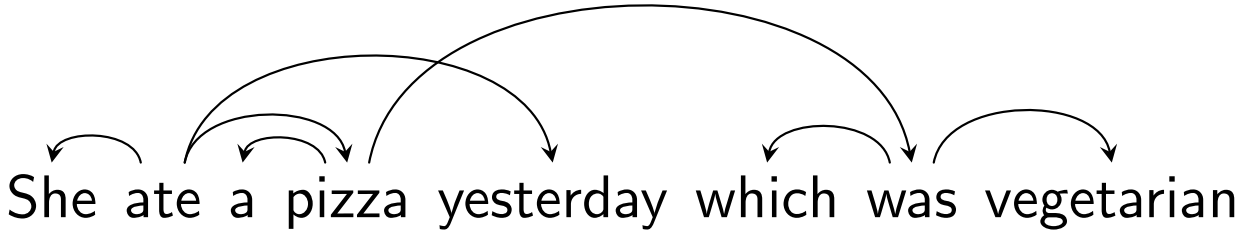


[2] figure from (Nivre 2007)

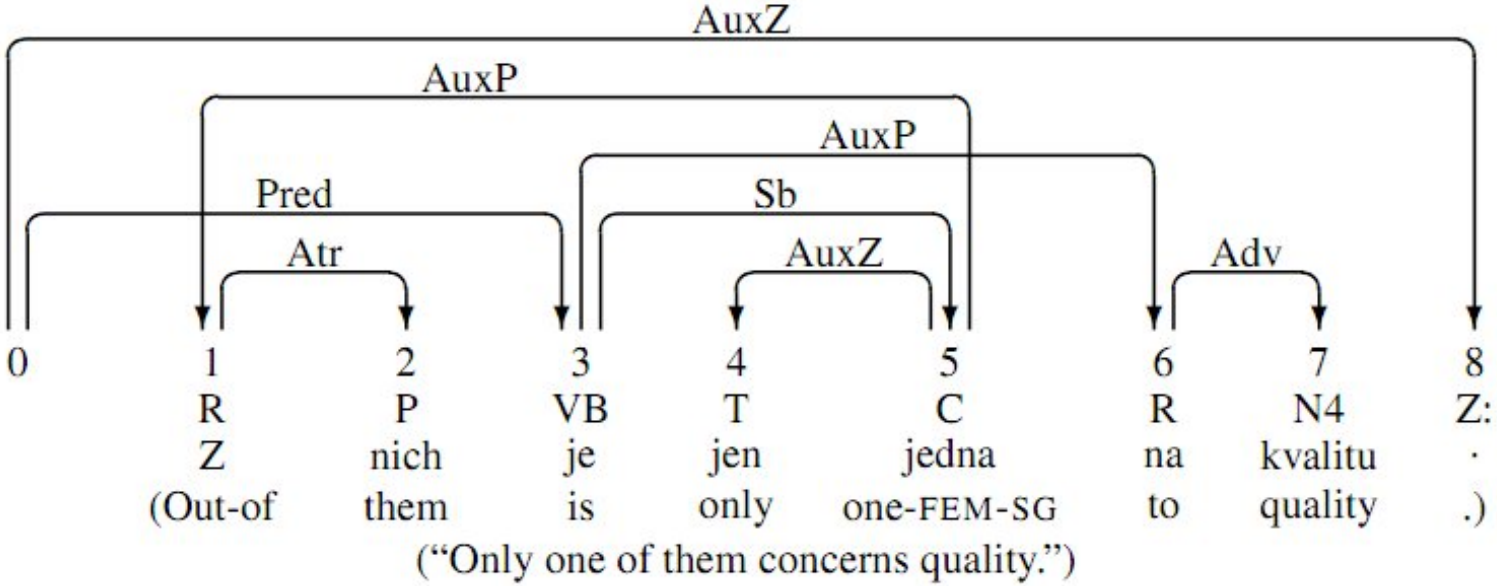
Projectivity

In **projective** dependency parsing, there are no crossing edges.

- Crossing edges are rare in English:



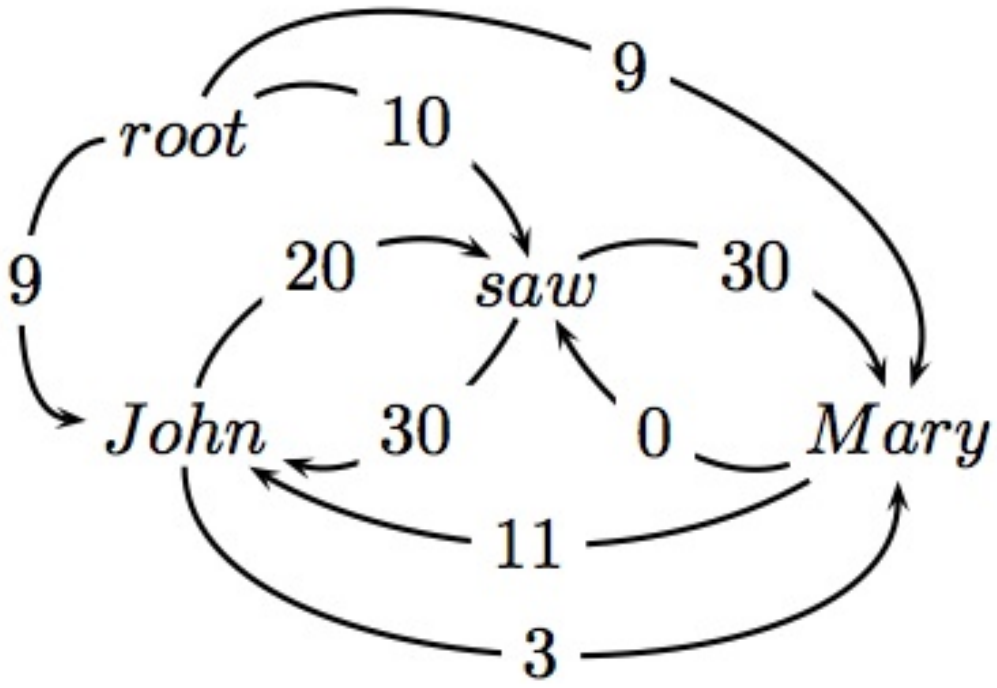
- They are more common in other languages, like Czech:[2]



[2] figure from (Nivre 2007)

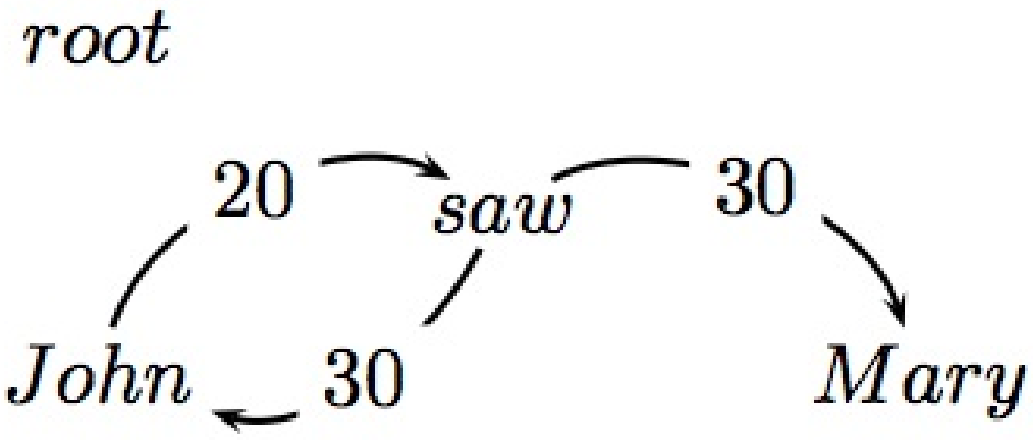
Chu-Liu-Edmonds

- x = root John saw Mary



Chu-Liu-Edmonds

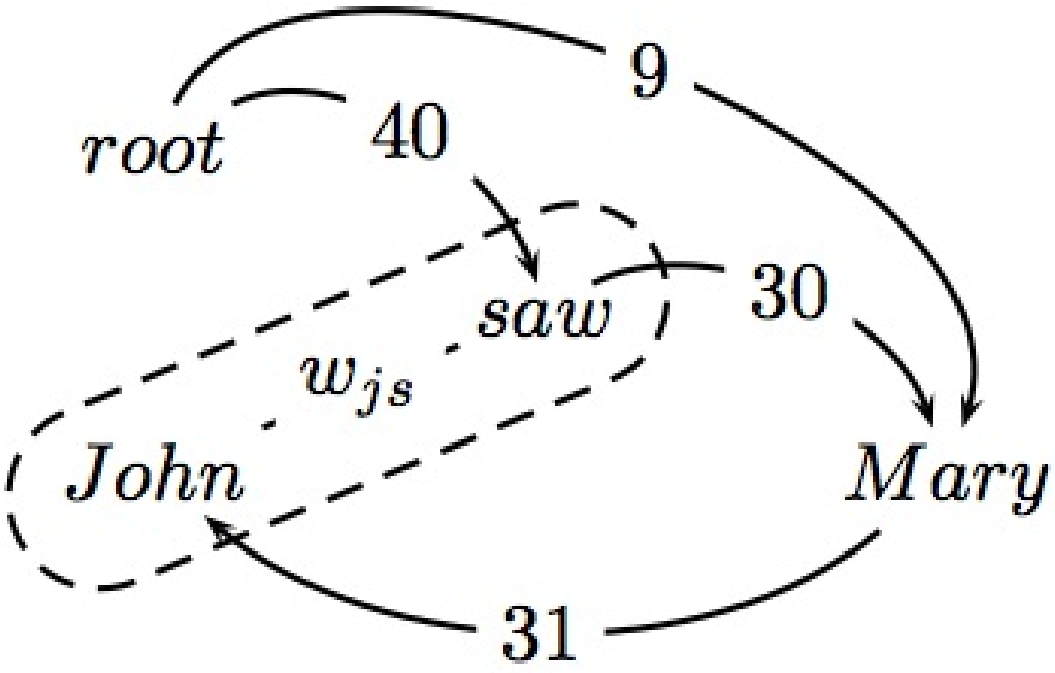
- Find highest scoring incoming arc for each vertex



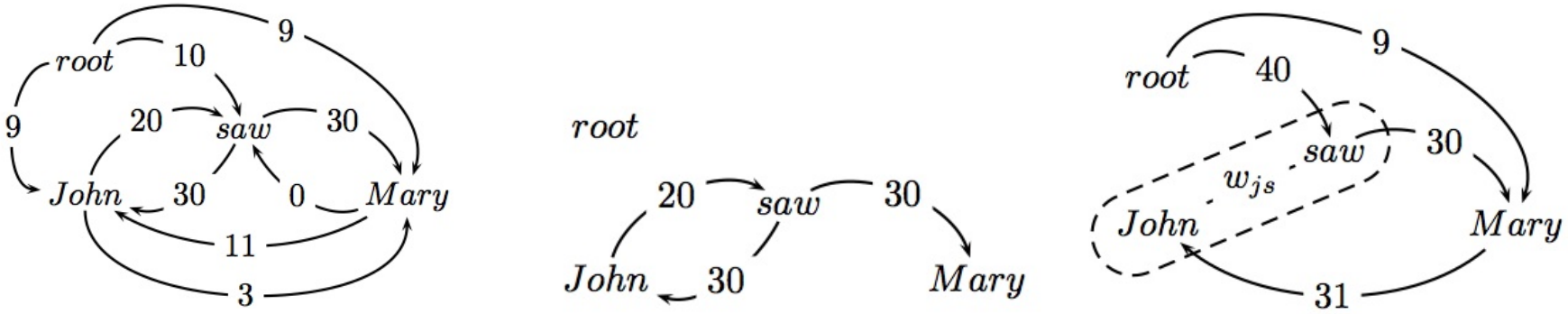
- If this is a tree, then we have found MST!!

Chu-Liu-Edmonds

- If not a tree, identify cycle and contract
- Recalculate arc weights into and out-of cycle

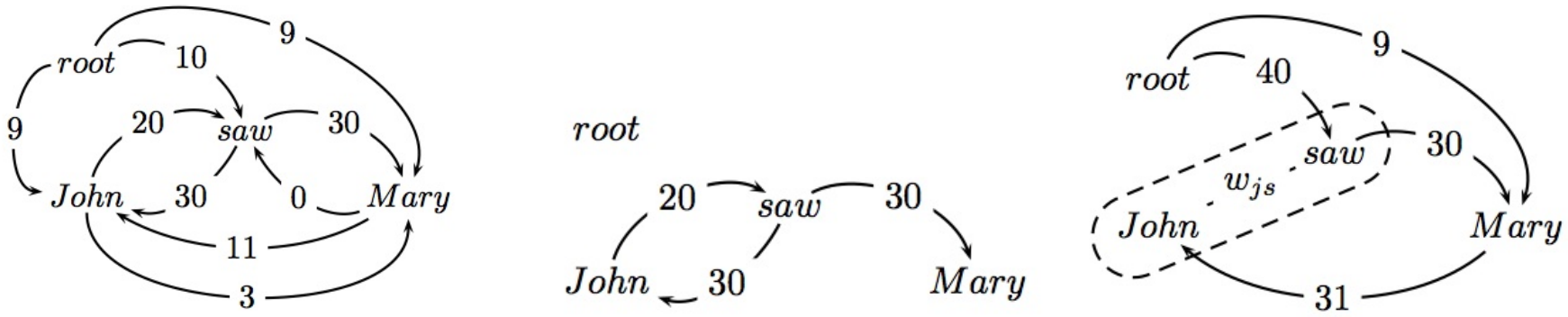


Chu-Liu-Edmonds



- Outgoing arc weights
  - Equal to the max of outgoing arc over all vertexes in cycle
  - e.g., John $\rightarrow$ Mary is 3 and saw $\rightarrow$ Mary is 30

Chu-Liu-Edmonds

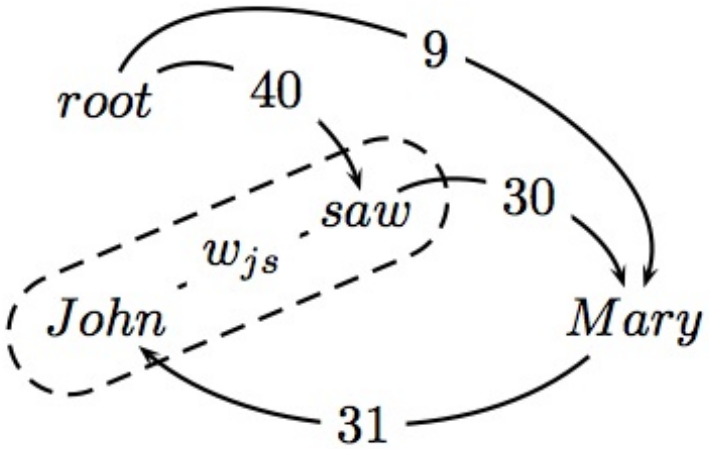


- Incoming arc weights
  - Equal to the weight of best spanning tree that includes head of incoming arc, and all nodes in cycle
  - root $\rightarrow$ saw $\rightarrow$ John is 40 (**)
  - root $\rightarrow$ John $\rightarrow$ saw is 29

Chu-Liu-Edmonds

Theorem

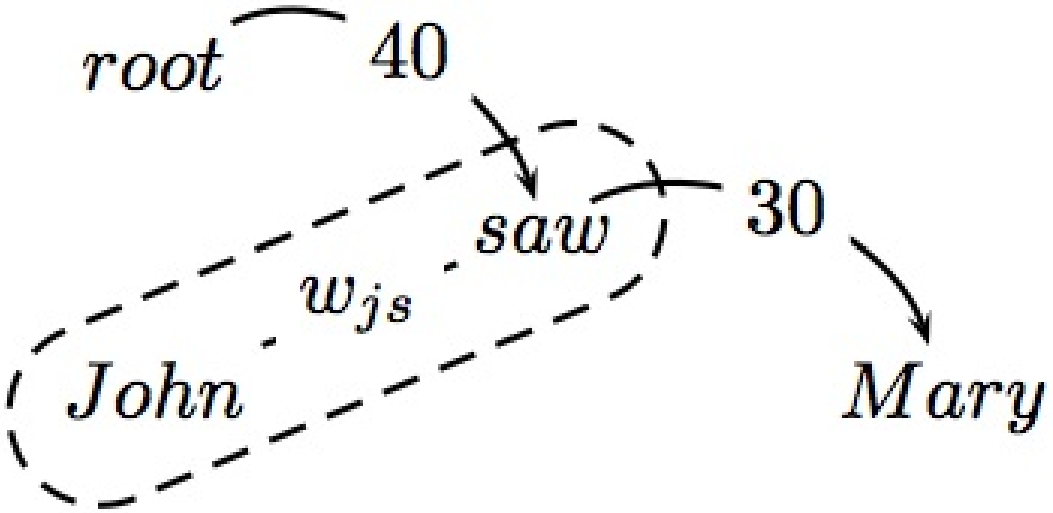
The weight of the MST of this contracted graph is equal to the weight of the MST for the original graph



- Therefore, recursively call algorithm on new graph

Chu-Liu-Edmonds

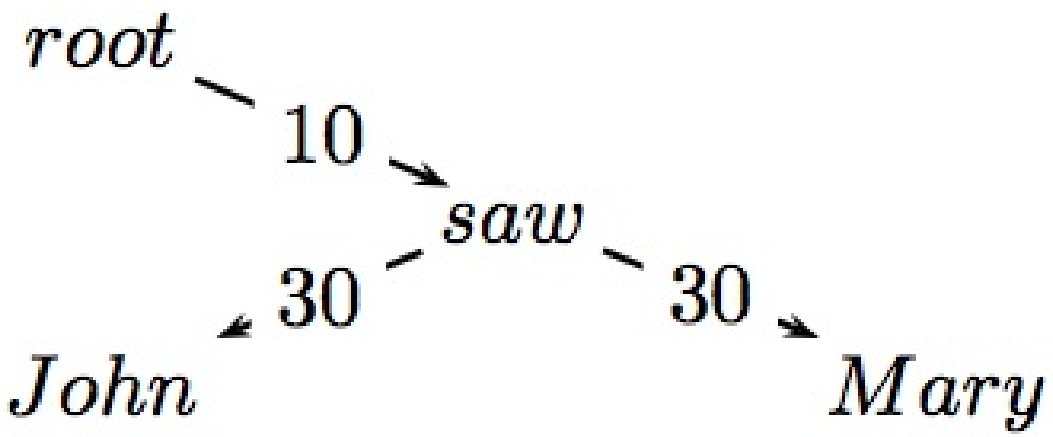
- This is a tree and the MST for the contracted graph!!



- Go back up recursive call and reconstruct final graph

Chu-Liu-Edmonds

- This is the MST!!



Chu-Liu-Edmonds Code

Chu-Liu-Edmonds(G_x, w)

1. Let $M = \{(i^*, j) : j \in V_x, i^* = \arg\max_{i'} w_{ij}\}$
2. Let $G_M = (V_x, M)$
3. If G_M has no cycles, then it is an MST: return G_M
4. Otherwise, find a cycle C in G_M
5. Let $< G_C, c, ma >=$ contract(G, C, w)
6. Let $G =$ Chu-Liu-Edmonds(G_C, w)
7. Find vertex $i \in C$ such that $(i', c) \in G$ and $ma(i', c) = i$
8. Find arc $(i'', i) \in C$
9. Find all arc $(c, i''') \in G$
10. $G = G \cup \{(ma(c, i'''), i''')\}_{\forall (c, i''') \in G} \cup C \cup \{(i', i)\} - \{(i'', i)\}$
11. Remove all vertices and arcs in G containing c
12. return G

- Reminder: $w_{ij} = \arg\max_k w_{ij}^k$

Projective dependency parsing: The Eisner Algorithm

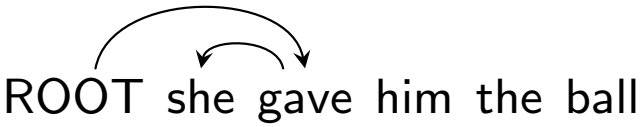
- ROOT She gave him the ball
- ROOT (She $\leftarrow$ gave) (gave $\rightarrow$ him) (the $\leftarrow$ ball)
- [ROOT] [She $\leftarrow$ gave] [gave $\rightarrow$ him] [the $\leftarrow$ ball]

ROOT she gave him the ball

Projective dependency parsing: The Eisner Algorithm

- ROOT She gave him the ball
- ROOT (She $\leftarrow$ gave) (gave $\rightarrow$ him) (the $\leftarrow$ ball)
- [ROOT] [She $\leftarrow$ gave] [gave $\rightarrow$ him] [the $\leftarrow$ ball]

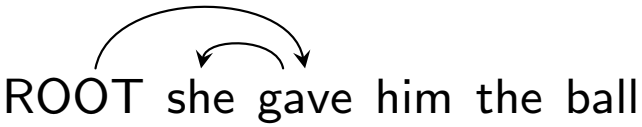
- ([ROOT] $\rightarrow$ [She $\leftarrow$ gave]) ([gave $\rightarrow$ him] $\rightarrow$ [the $\leftarrow$ ball])



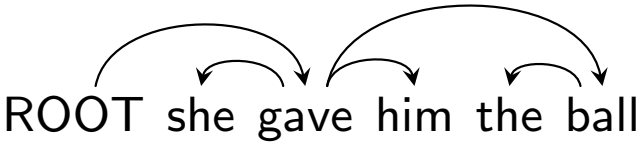
Projective dependency parsing: The Eisner Algorithm

- ROOT She gave him the ball
- ROOT (She $\leftarrow$ gave) (gave $\rightarrow$ him) (the $\leftarrow$ ball)
- [ROOT] [She $\leftarrow$ gave] [gave $\rightarrow$ him] [the $\leftarrow$ ball]

- ([ROOT] $\rightarrow$ [She $\leftarrow$ gave]) ([gave $\rightarrow$ him] $\rightarrow$ [the $\leftarrow$ ball])



- ([ROOT] $\rightarrow$ [She $\leftarrow$ gave]) [[gave $\rightarrow$ him] $\rightarrow$ [the $\leftarrow$ ball]]
- [([ROOT] $\rightarrow$ [She $\leftarrow$ gave]) $\rightarrow$ [[gave $\rightarrow$ him] $\rightarrow$ [the $\leftarrow$ ball]]]



The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

incomplete

$\rightarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

complete

$\leftarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

complete

$\rightarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

complete

$\leftarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

complete

$\rightarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

complete

$\rightarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT			
plastic			
cup			

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$		
plastic		2	
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	6
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1		
plastic		-1	
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	6
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	$-\infty$
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	6
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	$-\infty$
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	7
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	
plastic		2	6
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	$-\infty$
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	7
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	$-\infty$
plastic		2	6
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	
plastic		-1	3
cup			-1

The Eisner Algorithm

Edge weights:

	ROOT	plastic	cup	holders
ROOT		1	1	1
plastic	$-\infty$		-1	-1
cup	$-\infty$	2		-1
holders	$-\infty$	0	4	

incomplete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	$-\infty$
plastic		2	4
cup			4

incomplete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	7
plastic		-1	3
cup			-1

complete

$\leftarrow$	plastic	cup	holders
ROOT	$-\infty$	$-\infty$	$-\infty$
plastic		2	6
cup			4

complete

$\rightarrow$	plastic	cup	holders
ROOT	1	3	7
plastic		-1	3
cup			-1