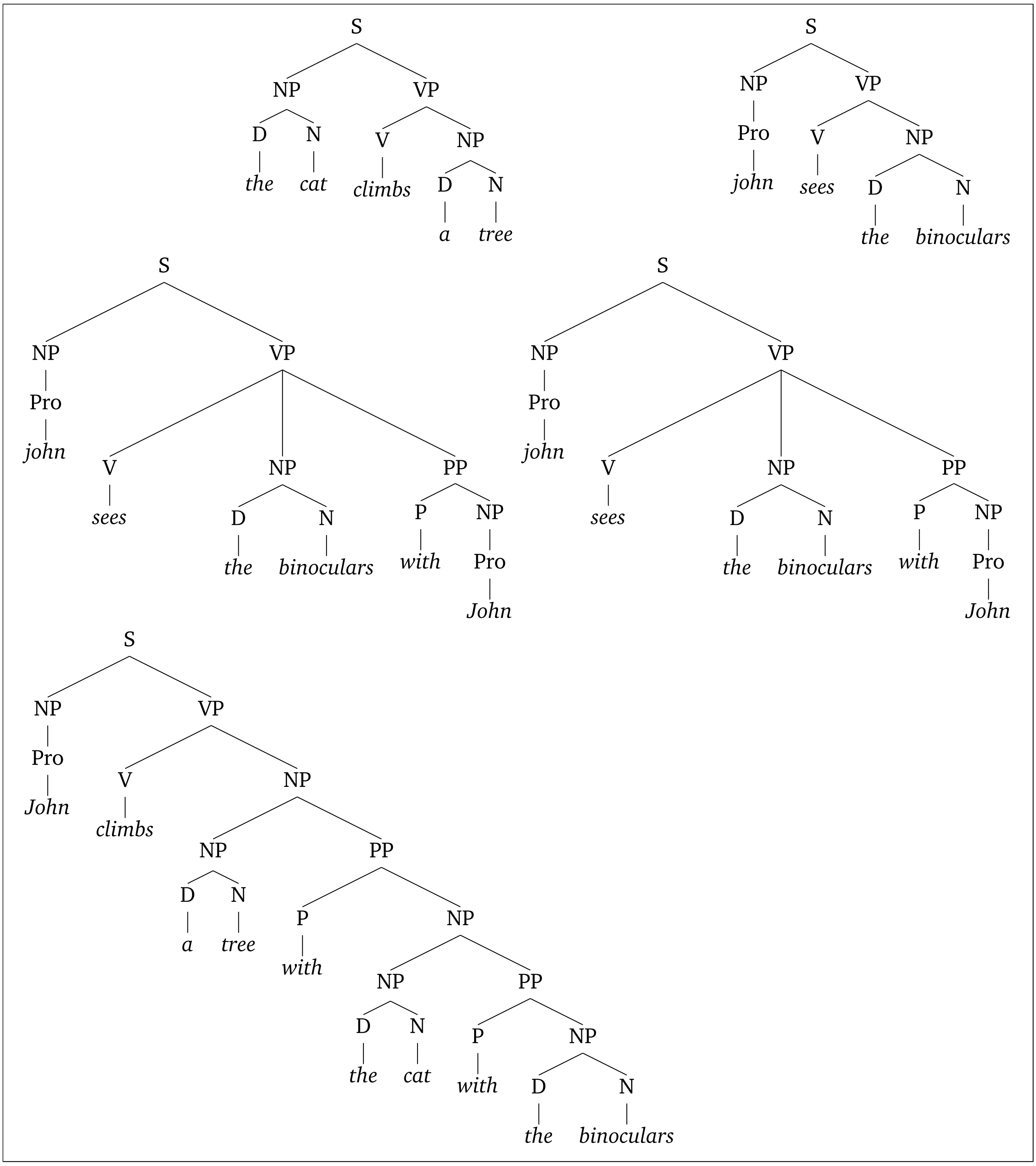
KU Leuven - NLP 2024-2025
12 November 2024

Exercise Session 2 - Solutions
Syntactic Parsing and Semantic Role Labeling with a GNN

For questions contact:
Toledo Forum for Questions (under Discussions) or nlp@ls.kuleuven.be

1 Parsing Context-Free Grammars

1.1 Question 1.A



1.2 Question 1.B

There are two issues.
Issue 1 is with (A → B):

S	→ NP	VP						
NP	→ D	N	|	NP	PP	|	Pro	
VP	→ V	NP	|	V	NP	PP		
PP	→ P	NP						
D	→ *the*	|	*a*					
P	→ *with*							
Pro	→ *john*							
N	→ *cat*	|	*tree*	|	*binoculars*			
V	→ *sees*	|	*climbs*					

Issue 2 is with (A → B C D):

S	→ NP	VP						
NP	→ D	N	|	NP	PP	|	Pro	
VP	→ V	NP	|	V	NP	PP		
PP	→ P	NP						
D	→ *the*	|	*a*					
P	→ *with*							
Pro	→ *john*							
N	→ *cat*	|	*tree*	|	*binoculars*			
V	→ *sees*	|	*climbs*					

The solution is:

S	→ NP	VP					
NP	→ D	N	|	NP	PP	|	*john*
VP	→ V	NP	|	X	PP		
X	→ V	NP					
PP	→ P	NP					
D	→ *the*	|	*a*				
P	→ *with*						
N	→ *cat*	|	*tree*	|	*binoculars*		
V	→ *sees*	|	*climbs*				

1.3 Question 1.C

lets make sure we use our updated grammar:

S	→ NP	VP					
NP	→ D	N	|	NP	PP	|	*john*
VP	→ V	NP	|	X	PP		
X	→ V	NP					
PP	→ P	NP					
D	→ *the*	|	*a*				
P	→ *with*						
N	→ *cat*	|	*tree*	|	*binoculars*		
V	→ *sees*	|	*climbs*				

Sentence 1:

$$
\begin{array}{rl}
S & \rightarrow NP \quad VP \\
NP & \rightarrow D \quad N \quad | \quad NP \quad PP \quad | \quad \textit{john} \\
VP & \rightarrow V \quad NP \quad | \quad X \quad PP \\
X & \rightarrow V \quad NP \\
PP & \rightarrow P \quad NP \\
D & \rightarrow \textit{the} \quad | \quad \textit{a} \\
P & \rightarrow \textit{with} \\
N & \rightarrow \textit{cat} \quad | \quad \textit{tree} \quad | \quad \textit{binoculars} \\
V & \rightarrow \textit{sees} \quad | \quad \textit{climbs}
\end{array}
$$

Sentence 2:
used rules for meaning 1:

$$
\begin{array}{rl}
S & \rightarrow NP \quad VP \\
NP & \rightarrow D \quad N \quad | \quad NP \quad PP \quad | \quad \textit{john} \\
VP & \rightarrow V \quad NP \quad | \quad X \quad PP \\
X & \rightarrow V \quad NP \\
PP & \rightarrow P \quad NP \\
D & \rightarrow \textit{the} \quad | \quad \textit{a} \\
P & \rightarrow \textit{with} \\
N & \rightarrow \textit{cat} \quad | \quad \textit{tree} \quad | \quad \textit{binoculars} \\
V & \rightarrow \textit{sees} \quad | \quad \textit{climbs}
\end{array}
$$

used rules for meaning 2:

$$
\begin{array}{rl}
S & \rightarrow NP \quad VP \\
NP & \rightarrow D \quad N \quad | \quad NP \quad PP \quad | \quad \textit{john} \\
VP & \rightarrow V \quad NP \quad | \quad X \quad PP \\
X & \rightarrow V \quad NP \\
PP & \rightarrow P \quad NP \\
D & \rightarrow \textit{the} \quad | \quad \textit{a} \\
P & \rightarrow \textit{with} \\
N & \rightarrow \textit{cat} \quad | \quad \textit{tree} \quad | \quad \textit{binoculars} \\
V & \rightarrow \textit{sees} \quad | \quad \textit{climbs}
\end{array}
$$

Please refer to the slides (relevant slides also added at the end of this document) to see the CKY algorithm in action.

1.4 Question 1.D

The ambiguity arises from the fact that the prepositional phrase (PP) “with the binoculars” can be syntactically attached to “sees” or to “the tree”. This is modelled in our grammar, where the Prepositional phrase can be attached to both a NP and a VP. Both are syntactically correct, but the sentence gets an entirely different (non-sensible) meaning.

2 Unsupervised latent trees

Sentence: “I saw an elephant”. Log-probs:

I	-1
saw	-2
an	-5
elephant	-7

2.1 Inside pass

See figure 1a.

I	saw	an	elephant
$q(1,1) = -1$	$q(1,2) = -3$	$q(1,3) = -8$	$q(1,4) = -15$
	$q(2,2) = -2$	$q(2,3) = -7$	$q(2,4) = -14$
		$q(3,3) = -5$	$q(3,4) = -12$
			$q(4,4) = -7$

(a) Chart of inside scores for exercise in section 2.1

I	saw	an	elephant
$p(1,1) = -54$	$p(1,2) = -24$	$p(1,3) = -7$	$p(1,4) = 0$
	$p(2,2) = -59$	$p(2,3) = -16$	$p(2,4) = -1$
		$p(3,3) = -41$	$p(3,4) = -6$
			$p(4,4) = -27$

(b) Chart of outside scores for exercise in section 2.1

Figure 1

$$
\begin{aligned}
&q(1,1) = -1 \text{ (given)}\\
&q(2,2) = -2 \text{ (given)}\\
&q(1,2) = q(1,1) + q(2,2) = -3\\
&q(3,3) = -5 \text{ (given)}\\
&q(2,3) = q(2,2) + q(3,3) = -7\\
&q(1,3) = \frac{s_{1,3,1}}{\sum_l s_{1,3,l}}(q(1,1) + q(2,3)) + \frac{s_{1,3,2}}{\sum_l s_{1,3,l}}(q(1,2) + q(3,3)) = -\frac{1}{2}\cdot 8 - \frac{1}{2}\cdot 8 = -8\\
&q(4,4) = -7 \text{ (given)}\\
&q(3,4) = q(3,3) + q(4,4) = -12\\
&q(2,4) = \frac{s_{2,4,2}}{\sum_l s_{2,4,l}}(q(2,3) + q(4,4)) + \frac{s_{2,4,3}}{\sum_l s_{2,4,l}}(q(2,2) + q(3,4)) = -\frac{1}{2}\cdot 14 - \frac{1}{2}\cdot 14 = -14\\
&q(1,4) = \frac{s_{1,4,1}}{\sum_l s_{1,4,l}}(q(1,1) + q(2,4)) + \frac{s_{1,4,2}}{\sum_l s_{1,4,l}}(q(1,2) + q(3,4)) + \frac{s_{1,4,3}}{\sum_l s_{1,4,l}}(q(1,3) + q(4,4))\\
&\qquad = -\frac{1}{3}\cdot 15 - \frac{1}{3}\cdot 15 - \frac{1}{3}\cdot 15 = -15
\end{aligned}
$$

2.2 Outside pass

See figure 1b. p are outside scores, q are inside scores.

$$
\begin{aligned}
&p(1,4) = 0 \text{ (given)}\\
&p(2,4) = p(1,4) + q(1,1) = -1\\
&p(3,4) = p(1,4) + q(1,2) + p(2,4) + q(2,2) = -6\\
&p(4,4) = p(1,4) + q(1,3) + p(2,4) + q(2,3) + p(3,4) + q(3,3) = -27\\
&p(1,3) = p(1,4) + q(4,4) = -7\\
&p(2,3) = p(2,4) + q(4,4) + p(1,3) + q(1,1) = -1 - 7 - 7 - 1 = -16\\
&p(3,3) = p(1,3) + q(1,2) + p(2,3) + q(2,2) + p(3,4) + q(4,4) = -7 - 3 - 16 - 2 - 6 - 7 = -41\\
&p(1,2) = p(1,3) + q(3,3) + p(1,4) + q(3,4) = -7 - 5 - 0 - 12 = -24\\
&p(2,2) = p(1,2) + q(1,1) + p(2,3) + q(3,3) + p(2,4) + q(3,4) = -24 - 1 - 16 - 5 - 1 - 12 = -59\\
&p(1,1) = p(1,2) + q(2,2) + p(1,3) + q(2,3) + p(1,4) + q(2,4) = -24 - 2 - 7 - 7 - 0 - 14 = -54
\end{aligned}
$$

2.3 Maximum score tree

All scores get an equal score. In the slides, the term in the formula for the scores that involves the vector representations h makes sure different subtrees get different scores. In the algorithm in Manning and Schütze, it is the fact that there are different rules that have different probabilities for different nonterminal symbols that results in different scores for different subtrees.

3 SRL with GNN

In the following table we define every word of the sentence, its corresponding index, and then a list of all the incoming and outgoing neighbors:

word	index	incoming	outgoing
the	0	[1, 1]	[1]
cat	1	[0, 2, 2]	[0, 2, 0]
sees	2	[1, 3]	[1, 3, 1, 4]
the	3	[2, 4, 4]	[2, 4]
tree	4	[3, 5, 2]	[3, 5, 3, 7]
with	5	[4, 6, 7]	[4, 6]
the	6	[5, 7, 7]	[5, 7]
binoculars	7	[6, 4]	[6, 5, 6]

First we declare $\boldsymbol{h}^0$ for all words, by concatenating the corresponding embeddings with the predicate boolean:

$$\boldsymbol{h}_0 = \begin{bmatrix} -0.3000, & 0.1000, & 0.0000 \\ 1.4000, & 1.1000, & 0.0000 \\ 0.8000, & -0.5000, & 1.0000 \\ -0.3000, & 0.1000, & 0.0000 \\ 1.3000, & 0.8000, & 0.0000 \\ 0.8000, & -0.8000, & 0.0000 \\ -0.3000, & 0.1000, & 0.0000 \\ -0.5000, & 0.7000, & 0.0000 \end{bmatrix}$$

Now we will do the update for all the words in the sentence separately.

1. the

$$\begin{aligned} v_{in} &= \begin{bmatrix} -4.6000, & 7.1000, & 1.8800 \end{bmatrix}^T \\ v_{out} &= \begin{bmatrix} -1.3300, & -2.6000, & 3.8000 \end{bmatrix}^T \\ h_1^1 &= \begin{bmatrix} -6.2300, & 4.6000, & 5.6800 \end{bmatrix}^T \\ \text{ReLU} h_1^1 &= \begin{bmatrix} 0.0000, & 4.6000, & 5.6800 \end{bmatrix}^T \end{aligned}$$

2. cat

$$\begin{aligned} v_{in} &= \begin{bmatrix} 6.9500, & 7.1500, & -4.7200 \end{bmatrix}^T \\ v_{out} &= \begin{bmatrix} -0.8700, & 3.7000, & -0.6000 \end{bmatrix}^T \\ h_2^1 &= \begin{bmatrix} 7.4800, & 11.9500, & -5.3200 \end{bmatrix}^T \\ \text{ReLU} h_2^1 &= \begin{bmatrix} 7.4800, & 11.9500, & 0.0000 \end{bmatrix}^T \end{aligned}$$

3. sees

$$\begin{aligned} v_{in} &= \begin{bmatrix} -1.7500, & 6.6000, & -0.1400 \end{bmatrix}^T \\ v_{out} &= \begin{bmatrix} -4.5900, & -6.0000, & 10.8000 \end{bmatrix}^T \\ h_3^1 &= \begin{bmatrix} -5.5400, & 0.1000, & 11.6600 \end{bmatrix}^T \\ \text{ReLU} h_3^1 &= \begin{bmatrix} 0.0000, & 0.1000, & 11.6600 \end{bmatrix}^T \end{aligned}$$

4. the

$$v_{in} = \begin{bmatrix}-0.1000, & 8.8500, & -0.6600\end{bmatrix}^T$$
$$v_{out} = \begin{bmatrix}-0.4000, & -0.4000, & 2.3000\end{bmatrix}^T$$
$$h_4^1 = \begin{bmatrix}-0.8000, & 8.5500, & 1.6400\end{bmatrix}^T$$
$$\text{ReLU}h_4^1 = \begin{bmatrix}0.0000, & 8.5500, & 1.6400\end{bmatrix}^T$$

5. tree

$$v_{in} = \begin{bmatrix}5.5500, & 7.7000, & -4.2200\end{bmatrix}^T$$
$$v_{out} = \begin{bmatrix}-2.9000, & 4.1000, & 0.7000\end{bmatrix}^T$$
$$h_5^1 = \begin{bmatrix}3.9500, & 12.6000, & -3.5200\end{bmatrix}^T$$
$$\text{ReLU}h_5^1 = \begin{bmatrix}3.9500, & 12.6000, & 0.0000\end{bmatrix}^T$$

6. with

$$v_{in} = \begin{bmatrix}-1.6500, & 9.8000, & -1.1000\end{bmatrix}^T$$
$$v_{out} = \begin{bmatrix}-1.9300, & -0.8000, & 3.2000\end{bmatrix}^T$$
$$h_6^1 = \begin{bmatrix}-2.7800, & 8.2000, & 2.1000\end{bmatrix}^T$$
$$\text{ReLU}h_6^1 = \begin{bmatrix}0.0000, & 8.2000, & 2.1000\end{bmatrix}^T$$

7. the

$$v_{in} = \begin{bmatrix}0.7000, & 9.3000, & -2.5200\end{bmatrix}^T$$
$$v_{out} = \begin{bmatrix}-1.3000, & 1.9000, & 0.5000\end{bmatrix}^T$$
$$h_7^1 = \begin{bmatrix}-0.9000, & 11.3000, & -2.0200\end{bmatrix}^T$$
$$\text{ReLU}h_7^1 = \begin{bmatrix}0.0000, & 11.3000, & 0.0000\end{bmatrix}^T$$

8. binoculars

$$v_{in} = \begin{bmatrix}-1.1000, & 6.4500, & -0.5000\end{bmatrix}^T$$
$$v_{out} = \begin{bmatrix}-1.6600, & 4.0000, & -0.4000\end{bmatrix}^T$$
$$h_8^1 = \begin{bmatrix}-3.2600, & 11.1500, & -0.9000\end{bmatrix}^T$$
$$\text{ReLU}h_8^1 = \begin{bmatrix}0.0000, & 11.1500, & 0.0000\end{bmatrix}^T$$

For simplicity we concatenate all vectors of $\boldsymbol{h}^1$ into a matrix:

$$\boldsymbol{H}^1 = \begin{bmatrix} 0.0000, & 4.6000, & 5.6800 \\ 7.4800, & 11.9500, & 0.0000 \\ 0.0000, & 0.1000, & 11.6600 \\ 0.0000, & 8.5500, & 1.6400 \\ 3.9500, & 12.6000, & 0.0000 \\ 0.0000, & 8.2000, & 2.1000 \\ 0.0000, & 11.3000, & 0.0000 \\ 0.0000, & 11.1500, & 0.0000 \end{bmatrix}$$

Next we compute the prediction layer and perform the softmax:

$$\boldsymbol{O} = \begin{bmatrix} 5.4360, & -2.6600, & -2.9400 \\ 24.6350, & 8.4400, & -34.8540 \\ 9.1720, & -23.6200, & 22.5200 \\ 3.0030, & 13.3200, & -22.8700 \\ 14.1100, & 16.8000, & -37.5100 \\ 3.2900, & 11.7000, & -20.9000 \\ 2.1300, & 22.1000, & -34.4000 \\ 2.1150, & 21.8000, & -33.9500 \end{bmatrix}$$

$$\text{softmax}(\boldsymbol{O}) = \begin{bmatrix} 9.99465201e-01 & 3.04592709e-04 & 2.30206255e-04 \\ 9.99999907e-01 & 9.25978555e-08 & 1.45967284e-26 \\ 1.59601346e-06 & 9.15486352e-21 & 9.99998404e-01 \\ 3.30650668e-05 & 9.99966935e-01 & 1.91808663e-16 \\ 6.35660057e-02 & 9.36433994e-01 & 2.42629559e-24 \\ 2.22580254e-04 & 9.99777420e-01 & 6.94869181e-15 \\ 2.12392501e-09 & 9.99999998e-01 & 2.89975783e-25 \\ 2.82431480e-09 & 9.99999997e-01 & 6.13878721e-25 \end{bmatrix} = \begin{bmatrix} 1. & 0. & 0. \\ 1. & 0. & 0. \\ 0. & 0. & 1. \\ 0. & 1. & 0. \\ 0.06 & 0.94 & 0. \\ 0. & 1. & 0. \\ 0. & 1. & 0. \\ 0. & 1. & 0. \end{bmatrix}$$

So the resulting predictions are:
`['arg0', 'arg0', 'O', 'arg1', 'arg1', 'arg1', 'arg1', 'arg1']`

4 Dependency Parsing

Please look at the solutions in the slides accompanying the exercise session.

5 Complexity

a. We know that we have M words and R relation types. Let us consider in steps what we need to calculate:

- We have to make a computation for every word M: $O(M)$.
- Every word is connected to all other words, so that results in total to M computations per word. Combined with the previous step we have $O(M^2)$.
- Now there are still R relation types, so there is not just one connection to every other word, but R connections. This results in the overall complexity of:

$$O(M^2 R)$$

b. If there are cycles, we have to do additional recursive computations. Since every cycle needs at least two nodes, we cannot have up to M cycles, but we can have up to a $M - k$, with k some small constant. If M goes towards infinite, we can ignore the small constant k in our big O notation. So for the number of recursive computations we have:

$$O(M)$$

c. Finally we can simply combine the two complexities of the previous two steps. this results in:

$$O(M^3 R)$$