Course Assignment # 2

May 27, 2021

Andrius Buinovskij
nethz Username: andriusb
Student ID: 18-940-270

By submitting this work, I verify that it is my own. That is, I have written my own solutions to each problem for which I am submitting an answer. I have listed above all others with whom I have discussed these answers.

Q1 a)

$$S \rightarrow NP, VP \tag{1}$$
$$NP \rightarrow Det, N \mid NP, PP \mid "I" \mid "glasses" \tag{2}$$
$$VP \rightarrow VP, PP \mid V, NP \tag{3}$$
$$Det \rightarrow "a" \mid "an" \tag{4}$$
$$N \rightarrow "man" \mid "pencil" \mid "ball" \mid "umbrella" \tag{5}$$
$$PP \rightarrow P, NP \tag{6}$$
$$V \rightarrow "draw" \mid "hit" \tag{7}$$
$$P \rightarrow "with" \tag{8}$$

Q1 b)
Warning: advanced calculations ahead (instead of probabilities We just do counts first):

$$S \rightarrow NP, VP(4) \tag{9}$$
$$NP(14) \rightarrow Det, N(7) \mid NP, PP(2) \mid "I"(4) \mid "glasses"(1) \tag{10}$$
$$VP(6) \rightarrow VP, PP(2) \mid V, NP(4) \tag{11}$$
$$Det(7) \rightarrow "a"(6) \mid "an"(1) \tag{12}$$
$$N(7) \rightarrow "man"(4) \mid "pencil"(1) \mid "ball"(1) \mid "umbrella"(1) \tag{13}$$
$$PP(4) \rightarrow P, NP(4) \tag{14}$$
$$V(4) \rightarrow "draw"(2) \mid "hit"(2) \tag{15}$$
$$P(4) \rightarrow "with"(4) \tag{16}$$

Convert to probabilities (rounded, for precise figures just divide counts on the right by total counts on the left):

$$S \rightarrow NP, VP(1.0) \tag{17}$$
$$NP \rightarrow Det, N(0.5) \mid NP, PP(0.14) \mid "I"(0.29) \mid "glasses"(0.7) \tag{18}$$
$$VP(\rightarrow VP, PP(2 = 0.33) \mid V, NP(0.66) \tag{19}$$
$$Det \rightarrow "a"(0.86(\mid "an"(0.14) \tag{20}$$
$$N \rightarrow "man"(0.57) \mid "pencil"(0.14) \mid "ball"(0.14) \mid "umbrella"(0.14) \tag{21}$$
$$PP \rightarrow P, NP(1) \tag{22}$$
$$V \rightarrow "draw"(0.5) \mid "hit"(0.5) \tag{23}$$
$$P \rightarrow "with"(1) \tag{24}$$

Q1 c)
Our goal is to have different likelihood of expansions on noun phrase depending on whether the expansion is going to be an object or a subject.

I suppose a logical way of going about it would be to replace the noun phrase non-terminal with two other non terminals - object phrase (OP) and subject phrase (SP).

$$S \to NP, VP(1.0) \tag{25}$$
$$NP \to SP \mid OP \tag{26}$$
$$SP \to Det, N(0.5) \mid NP, PP(0.14) \mid "I"(0.29) \mid "glasses"(0.7) \tag{27}$$
$$OP \to Det, N(0.5) \mid NP, PP(0.14) \mid "I"(0.29) \mid "glasses"(0.7) \tag{28}$$
$$VP(\to VP, PP(2 = 0.33) \mid V, NP(0.66) \tag{29}$$
$$Det \to "a"(0.86(\mid "an"(0.14) \tag{30}$$
$$N \to "man"(0.57) \mid "pencil"(0.14) \mid "ball"(0.14) \mid "umbrella"(0.14) \tag{31}$$
$$PP \to P, NP(1) \tag{32}$$
$$V \to "draw"(0.5) \mid "hit"(0.5) \tag{33}$$
$$P \to "with"(1) \tag{34}$$

Here are the counts as before:

$$S \to NP, VP(1.0) \tag{35}$$
$$NP(14) \to SP(4) \mid OP(10) \tag{36}$$
$$SP(4) \to Det, N(0) \mid NP, PP(0) \mid "I"(4) \mid "glasses"(0) \tag{37}$$
$$OP(10) \to Det, N(7) \mid NP, PP(2) \mid "I"(0) \mid "glasses"(1) \tag{38}$$
$$VP(\to VP, PP(2 = 0.33) \mid V, NP(0.66) \tag{39}$$
$$Det \to "a"(0.86(\mid "an"(0.14) \tag{40}$$
$$N \to "man"(0.57) \mid "pencil"(0.14) \mid "ball"(0.14) \mid "umbrella"(0.14) \tag{41}$$
$$PP \to P, NP(1) \tag{42}$$
$$V \to "draw"(0.5) \mid "hit"(0.5) \tag{43}$$
$$P \to "with"(1) \tag{44}$$

Finally converting to probabilities:

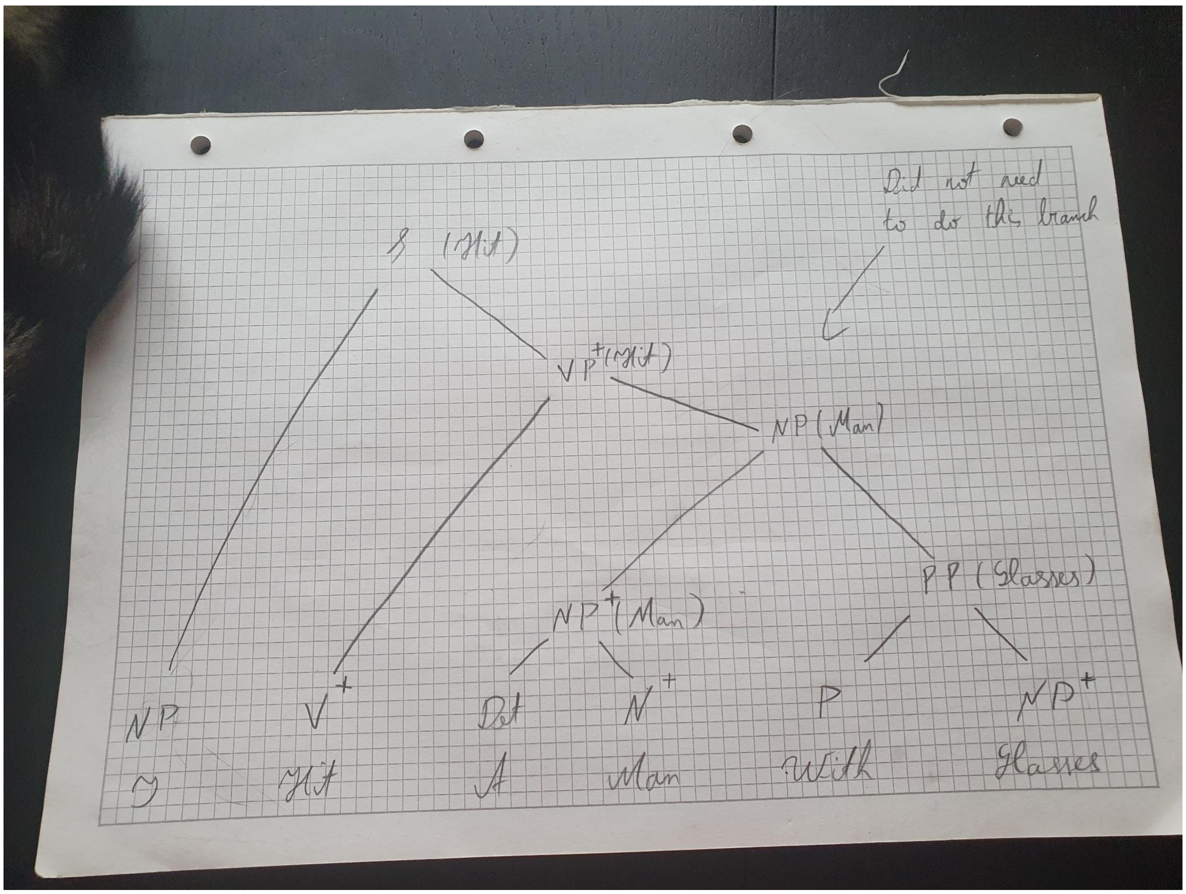
$$S \to NP, VP(1.0) \tag{45}$$
$$NP \to SP(0.29) \mid OP(0.71) \tag{46}$$
$$SP(4) \to Det, N(0) \mid NP, PP(0) \mid "I"(1) \mid "glasses"(0) \tag{47}$$
$$OP(10) \to Det, N(0.7) \mid NP, PP(0.2) \mid "I"(0) \mid "glasses"(0.1) \tag{48}$$
$$VP(\to VP, PP(2 = 0.33) \mid V, NP(0.66) \tag{49}$$
$$Det \to "a"(0.86(\mid "an"(0.14) \tag{50}$$
$$N \to "man"(0.57) \mid "pencil"(0.14) \mid "ball"(0.14) \mid "umbrella"(0.14) \tag{51}$$
$$PP \to P, NP(1) \tag{52}$$
$$V \to "draw"(0.5) \mid "hit"(0.5) \tag{53}$$
$$P \to "with"(1) \tag{54}$$

Well this at least shows that subject phrases are likely to be "I", and object phrases are unlikely to be "I", so it's capturing some of what We wanted to.

There is a problem here of course - nothing is stopping anyone from using SP instead of OP wherever they like, so the onus is on the user of the grammar to use the subject non-terminal for subjects and the object non-terminal for objects.

Q1 d)

$$S \rightarrow NP, VP^{+} \tag{55}$$
$$NP \rightarrow Det, N^{+} \mid NP^{+}, PP \mid ''I'' \mid ''glasses'' \tag{56}$$
$$VP \rightarrow VP^{+}, PP \mid V^{+}, NP \tag{57}$$
$$Det \rightarrow ''a'' \mid ''an'' \tag{58}$$
$$N \rightarrow ''man'' \mid ''pencil'' \mid ''ball'' \mid ''umbrella'' \tag{59}$$
$$PP \rightarrow P, NP^{+} \tag{60}$$
$$V \rightarrow ''draw'' \mid ''hit'' \tag{61}$$
$$P \rightarrow ''with'' \tag{62}$$



Q2 a)

So, it is my understanding that for the projective dependency tree, to obtain the score We would simply take all dependency pairs and multiply the scores corresponding to those dependencies (taking care to pay attention to the direction of the dependency etc.)

To obtain the score of a lexicalized probabilistic context free grammar tree, one also simply multiplies the score for each "rule" applied, except this rule is also lexicalized so You have very many parameters indeed (page 15 from _ click me_.)

With all that out of the way, I would construct a weighted lexicalized CFG as follows:

1. Add root productions $R \to X_j$, $j \in 1 \ldots M$.
2. Add self-productions $X_j \to j$.
3. For each production $\psi(i \to j)$, add bidirectional productions to the language as follows:

$$X_i \to X_i^+, X_j \tag{63}$$

$$X_i \to X_j, X_i^+ \tag{64}$$

And of course they weight for that transformation corresponds to the value of $\psi(i \to j)$ does not You anything about the ordering of words within a sentence.

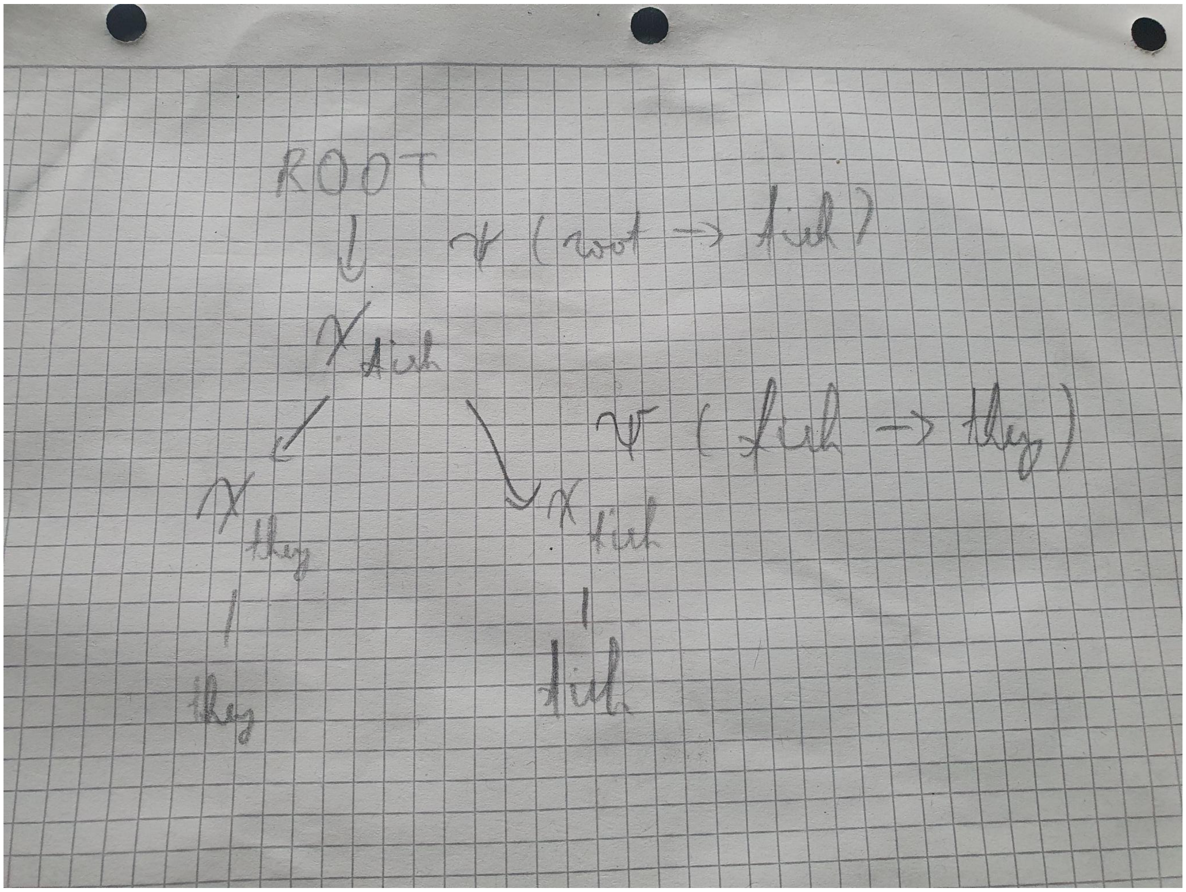
The reason for adding bidirectional rules is that just having access to the set of weights generated by

And that's all She wrote.

Q2 b)

So the dependency tree would simply be $ROOT \to fish$, $fish \to they$, and the score would be $\psi(ROOT, fish) \cdot \psi(fish, they)$.

In our CFG We would end up with $ROOT \to S$, then $S \to NP$,


ROOT
ψ (root → fish)
χ fish
ψ (fish → they)
χ they
χ fish
they
fish

Q4 a)

number	sample strings	accepted	weight
1	educational is this not	NO	
2	is this assignment educational	NO	
3	not educational is not educational	YES	12
4	this assignment is not educational	YES	7
5	is this assignment educational	NO	
6	this assignment course is educational	NO	
7	is this assignment not educational	YES	15
8	this assignment not	YES	8
9	this course assignment is not educational	YES	12
10	this course is not not educational	YES	21
11	not educational is this	YES	10
12	course assignment is not educational	YES	10
13	not this assignment is educational	NO	
14	not not not educational	YES	25
15	is this course assignment not educational	YES	18
16	course assignment is this	YES	8
17	this course is interesting	NO	
18	this course assignment not educational	YES	14

Table 1: Some strings from $\mathcal{Y}_{\geq 2, \leq 6}$

Q4 b)

Cost matrices on left, backtracking matrices on the right.

"inf" means an infinite cost, i.e. no path exists, and similarly "N" in the backtracking matrix means a backtrack is not possible.

Iteration 0

-	A	B	C	D	E	F
A	inf	1	2	inf	inf	inf
B	inf	inf	inf	1	5	inf
C	inf	2	1	2	3	inf
D	4	inf	inf	1	2	4
E	inf	inf	inf	inf	10	2
F	inf	inf	inf	3	inf	inf

-	A	B	C	D	E	F
A	-	B	C	-	-	-
B	-	-	-	D	E	-
C	-	B	C	D	E	-
D	A	-	-	D	E	F
E	-	-	-	-	E	F
F	-	-	-	D	-	-

Iteration 1

-	A	B	C	D	E	F
A	inf	1	2	inf	inf	inf
B	inf	inf	inf	1	5	inf
C	inf	2	1	2	3	inf
D	4	5	6	1	2	4
E	inf	inf	inf	inf	10	2
F	inf	inf	inf	3	inf	inf

-	A	B	C	D	E	F
A	-	B	C	-	-	-
B	-	-	-	D	E	-
C	-	B	C	D	E	-
D	A	A	A	D	E	F
E	-	-	-	-	E	F
F	-	-	-	D	-	-

Iteration 2

-	A	B	C	D	E	F
A	inf	1	2	2	6	inf
B	inf	inf	inf	1	5	inf
C	inf	2	1	2	3	inf
D	4	5	6	1	2	4
E	inf	inf	inf	inf	10	2
F	inf	inf	inf	3	inf	inf

-	A	B	C	D	E	F
A	-	B	C	B	B	-
B	-	-	-	D	E	-
C	-	B	C	D	E	-
D	A	A	A	D	E	F
E	-	-	-	-	E	F
F	-	-	-	D	-	-

Iteration 3

-	A	B	C	D	E	F
A	inf	1	2	2	5	inf
B	inf	inf	inf	1	5	inf
C	inf	2	1	2	3	inf
D	4	5	6	1	2	4
E	inf	inf	inf	inf	10	2
F	inf	inf	inf	3	inf	inf

-	A	B	C	D	E	F
A	-	B	C	B	C	-
B	-	-	-	D	E	-
C	-	B	C	D	E	-
D	A	A	A	D	E	F
E	-	-	-	-	E	F
F	-	-	-	D	-	-

Iteration 4

-	A	B	C	D	E	F
A	6	1	2	2	4	6
B	5	6	7	1	3	5
C	6	2	1	2	3	6
D	4	5	6	1	2	4
E	inf	inf	inf	inf	10	2
F	7	8	9	3	5	7

-	A	B	C	D	E	F
A	B	B	C	B	B	B
B	D	D	D	D	D	D
C	D	B	C	D	E	D
D	A	A	A	D	E	F
E	-	-	-	-	E	F
F	D	D	D	D	D	D

Iteration 5

-	A	B	C	D	E	F
A	6	1	2	2	4	6
B	5	6	7	1	3	5
C	6	2	1	2	3	5
D	4	5	6	1	2	4
E	inf	inf	inf	inf	10	2
F	7	8	9	3	5	7

-	A	B	C	D	E	F
A	B	B	C	B	B	B
B	D	D	D	D	D	D
C	D	B	C	D	E	E
D	A	A	A	D	E	F
E	-	-	-	-	E	F
F	D	D	D	D	D	D

Final 6th iteration

-	A	B	C	D	E	F
A	6	1	2	2	4	6
B	5	6	7	1	3	5
C	6	2	1	2	3	5
D	4	5	6	1	2	4
E	9	10	11	5	7	2
F	7	8	9	3	5	7

-	A	B	C	D	E	F
A	B	B	C	B	B	B
B	D	D	D	D	D	D
C	D	B	C	D	E	E
D	A	A	A	D	E	F
E	F	F	F	F	F	F
F	D	D	D	D	D	D

Q4 c)

The number of iterations is bound by $\mathcal{O}(N^3)$, where N is the number of nodes.

The algorithm terminated after 6 iterations, so after looking at all of the nodes.

Q4 d)

The time complexity is $\mathcal{O}(N^3)$ and for the method I used the space complexity is N^2. Strictly speaking it's more like $4N^2$ since I was keeping new and old versions of matrices around in my implementation, but this can be reduced. Asymptotically, it's simply $\mathcal{O}(N^2)$ for space complexity and $\mathcal{O}(N^3)$ for time complexity.

Backtracking takes $\mathcal{O}(n)$ and is achieved as follows: let starting node be u and ending node be v. Now take the v'th column of the backtracking matrix, and begin at the u'th entry. The value of the u'th entry is the next node in the shortest path. Then simply follow the entries until the desired destination is reached.

This makes sense intuitively - it may be the case that to go from u to v one has to traverse all the nodes, but no node should be traversed more than once since if it were, then there is a cycle in the path that could be cut out.