CSCI 544 – Applied Natural Language Processing, Spring 2020

Written Homework 4: Parsing

Out: February 18, 2020

Total: 19 pages.

General instructions

1. This is not a graded assignment. Do not turn it in.
2. The assignment is meant as preparation for the in-class exams. You should aim to answer all the questions on your own, without help.
3. Space is provided as it would be on the exams. Answers should be concise and fit in the space provided; in the exams it will not be possible to add space, and long and rambling answers will be penalized.
4. After solving the problems (or giving them your best try), you are encouraged to discuss and compare solutions with your classmates.
5. You are welcome to discuss the problems with us. We encourage open discussion on Piazza, so that the entire class can benefit from the discussion.
6. Answers to select problems will be distributed at a later time.

Problem 1. Assume the following grammar and lexicon for a context-free grammar.

Grammar		Lexicon
S $\rightarrow$ NP VP	NP $\rightarrow$ N	N $\rightarrow$ squirrels | rabbits
NP $\rightarrow$ AP NP	VP $\rightarrow$ V	A $\rightarrow$ gray
NP $\rightarrow$ NP ConjP	AP $\rightarrow$ A	V $\rightarrow$ ate
ConjP $\rightarrow$ Conj NP		Conj $\rightarrow$ and

In order to use this grammar with the CKY parser, we need to convert it to Chomsky Normal Form, that is a grammar where the right-hand side of each rule must expand either to exactly two non-terminals or to a single terminal (terminal nodes are the words of the lexicon).

a. Identify the rules in the above grammar which are not in Chomsky Normal Form.

Conversion to Chomsky Normal Form proceeds as follows. Rules that expand to a single non-terminal (*unit productions*) are eliminated by rewriting the right-hand side of the original rules with the right-hand side of the rules that they ultimately lead to. Rules that expand to more than one non-terminal are eliminating by introducing new non-terminals such that a rule of the form $A \rightarrow B\,C\,\gamma$ is replaced with two rules, $A \rightarrow X\,\gamma$ and $X \rightarrow B\,C$.

b. Write down the new rules that replace the rules from part a.

The CKY parsing algorithm works as follows (adapted from Jurafsky and Martin, chapter 13: Constituency Parsing; assume an input sentence is an array *words*, indexed starting at 1, and the results are stored in a matrix *table*).

```
# Loop over words from left to right
for $j$ from 1 to LENGTH(words) :
    # Add lexical rules to the table
    for each rule $A \rightarrow \text{words}[j]$ :
        add $A$ to $\text{table}[j-1, j]$
    # Loop over preceding words from right to left
    for $i$ from $j-2$ down to 0 :
        # Loop over places to split the string
        for $k$ from $i+1$ to $j-1$ :
            # Add non-lexical rules to the table
            for each rule $A \rightarrow B\,C$ such that $B \in \text{table}[i,k]$ and $C \in \text{table}[k,j]$ :
                add $A$ to $\text{table}[i,j]$
```

Use the CKY algorithm to parse the sentence *Gray squirrels and rabbits ate*, and answer the questions on the following page.

gray	squirrels	and	rabbits	ate
[0,1]	[0,2]	[0,3]	[0,4]	[0,5]
	[1,2]	[1,3]	[1,4]	[1,5]
		[2,3]	[2,4]	[2,5]
			[3,4]	[3,5]
				[4,5]

c. In which cells does the parser identify at least one S constituent?

d. In which cells does the parser not identify any constituents whatsoever?

e. In which cell does the parser first encounter an ambiguity? List the various interpretations.

f. Now assume the above grammar is probabilistic, that is a (non-lexicalized) PCFG: Each rule carries a probability, and given a rule r of the form $C \to C_1 \ldots C_n$, we calculate the probability of the constituent C as follows.

$$P(C) = P(r)\prod_{i=1}^{n} P(C_i)$$

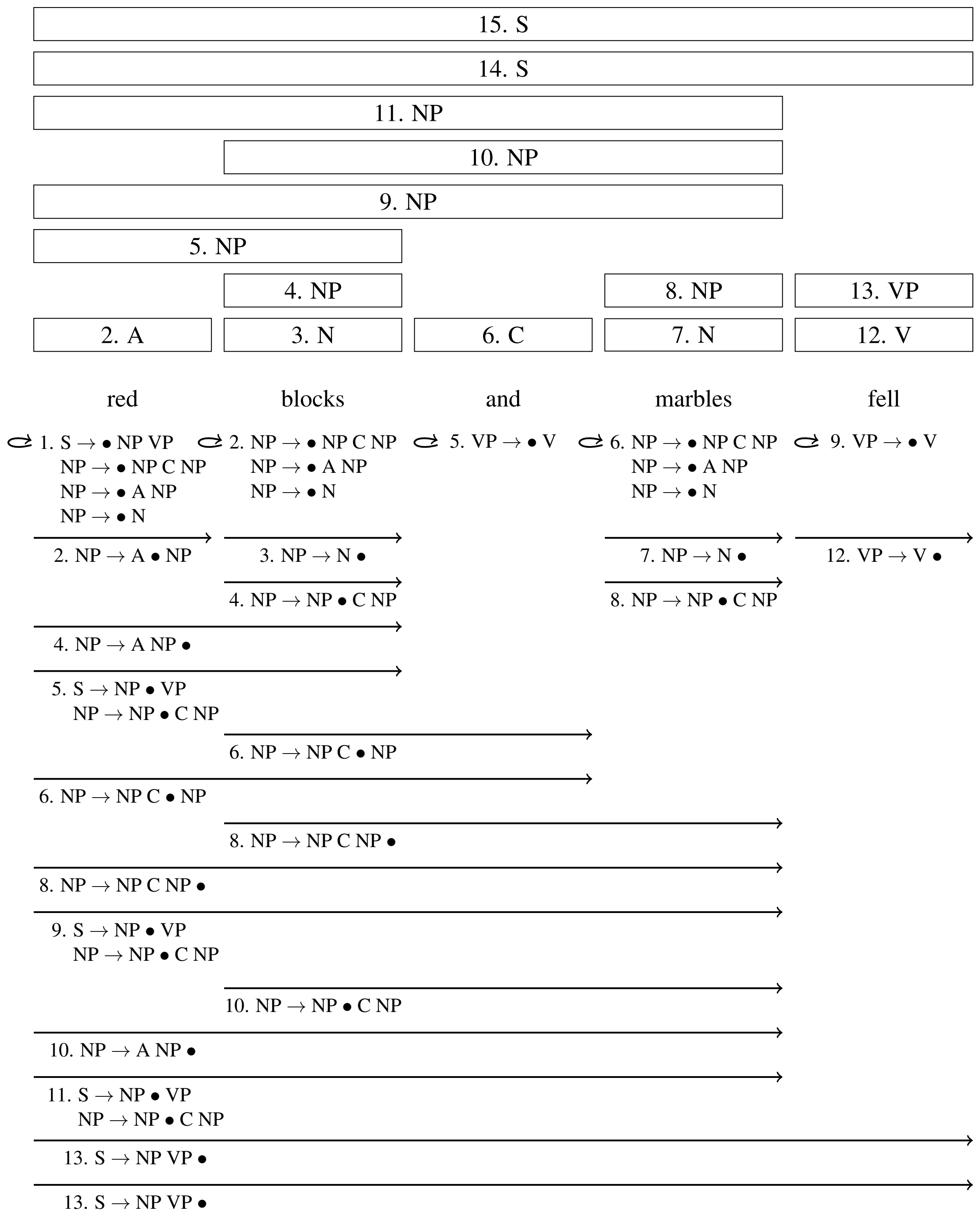
Can such a parser disambiguate between the two structures of *Gray squirrels and rabbits ate*? Why or why not? Which rules are the ones whose probabilities matter for disambiguation?

Top-down chart parsing algorithm (for Problem 2)

```
# Initialize
Agenda $\leftarrow$ list of all possible categories of each input word, in order of input

For all grammar rules S $\rightarrow$ X$_1 \ldots$ X$_N$:
    Add arc S $\rightarrow \bullet$X$_1 \ldots$ X$_N$ from position 0 to position 0
    Apply arc addition (step b below) to the new arc

# Iterate
while Agenda not empty:
    $C(i, j) \leftarrow$ pop Agenda        # C is a completed constituent from position i to position j

    Add $C(i, j)$ to the chart

    # (a) Arc extension
    For all arcs X $\rightarrow$ X$_1 \ldots \bullet$C$\ldots$ X$_N$ from $k$ to $i$:
        Add arcs X $\rightarrow$ X$_1 \ldots$ C$\bullet \ldots$ X$_N$ from $k$ to $j$

    # (b) Arc addition
    For all arcs X $\rightarrow$ X$_1 \ldots \bullet$Y$\ldots$ X$_N$ from $k$ to $j$ added in step (a):
        Add arcs Y $\rightarrow \bullet$Y$_1 \ldots$ Y$_N$ from $j$ to $j$

    # (c) Arc completion
    For all arcs X $\rightarrow$ X$_1 \ldots$ X$_N$ C$\bullet$ from $k$ to $j$ added in step (a):
        push (Agenda , X)
```

Example chart parse for the sentence *red blocks and marbles fell*

Grammar:	S → NP VP	NP → NP C NP	NP → A NP	NP → N	VP → V
Lexicon:	red: A	blocks: N	and: C	marbles: N	fell: V



Problem 2. Assume the following grammar and lexicon:

S $\rightarrow$ NP VP	VP $\rightarrow$ V NP	cops: N	chased: V
NP $\rightarrow$ N	VP $\rightarrow$ V NP PP	thieves: N	on: P
NP $\rightarrow$ N PP	PP $\rightarrow$ P NP	skates: N	

a. Show the complete chart (including all the completed constituents and active arcs) for parsing the following sentence: ***cops chased thieves on skates***

b. Draw the parse tree (or trees) for any complete sentence structures found by the algorithm (that is, S constituents from position 0 to 5).

Problem 3. Assume the following grammar and lexicon:

S $\rightarrow$ NP VP	pets: N	bowls: N, V	for: P
NP $\rightarrow$ (Poss) N (PP) (IP)	owners: N	drink: N, V	
VP $\rightarrow$ V (NP) (PP)	food: N	their: Poss	
PP $\rightarrow$ P NP	show: N, V	to: P, Inf	
IP $\rightarrow$ Inf VP	desire: N, V	from: P	

Note that in the above grammar formalism, parentheses in a rule denote an optional constituent. So the VP rule above is shorthand for the following 4 rules:

VP $\rightarrow$ V
VP $\rightarrow$ V NP
VP $\rightarrow$ V PP
VP $\rightarrow$ V NP PP

a. Draw the most plausible parse tree given by the above grammar for the following sentence:

pets show their desire to drink to owners

That is, the tree that corresponds to the meaning where pets desire to drink, and they show this desire to owners.

b. Draw all the additional parse trees for the same sentence:

pets show their desire to drink to owners

c. Choose *two* of the parse trees from part (b), and explain how the grammar can be modified to rule those out. Make sure the modifications do not rule out the following sentences.

pets drink from bowls
pets drink their food from bowls
pets show desire
pets show desire for food

Problem 4. Assume the following grammar and lexicon for a PCFG (probabilistic context-free grammar), where each grammatical rule is associated with a probability (the probabilities for the NP $\rightarrow \ldots$ rules and the VP $\rightarrow \ldots$ rules don't sum up to 1 because there are other rules that are not shown here).

Grammar				Lexicon
S $\rightarrow$ NP VP	1	VP $\rightarrow$ V NP	0.2	N: United, Houston, flights
NP $\rightarrow$ N	0.1	VP $\rightarrow$ V NP PP	0.3	V: diverted
NP $\rightarrow$ N PP	0.2	PP $\rightarrow$ P NP	1	P: to

Since the words are unambiguous (in this lexicon), assume that the probability of each lexical constituent (N, V, or P) is 1. For non-lexical constituents (S, NP, VP, PP), given a rule r of the form $C \rightarrow C_1 \ldots C_n$, we calculate the probability of C as follows.

$$P(C) = P(r) \prod_{i=1}^{n} P(C_i)$$

a. Draw the two parse trees generated by the grammar for the following sentence; use constituent trees in the same format as Jurafsky and Martin Figures 14.2, 14.5, 14.7 etc.

United diverted flights to Houston

(meaning: flights landed in Houston) | *(meaning: flights did not land in Houston)*

b. Label each non-lexical constituent in the above trees with its probability.

Problem 5. Assume the same grammar and lexicon, but with lexicalized probabilities. This problem assumes a simplified version of the **Charniak Parser**, which was mentioned in class but not covered in detail. Lexicalized rules come in two flavors:

1. The probability of a rule *r* is conditioned on its head *h*, so, for example, we will have separate probabilities for the following rules:

$$\text{NP} \to \text{N}|\text{N} = \text{United}$$
$$\text{NP} \to \text{N}|\text{N} = \text{Houston}$$

2. The probability of a head *h* is conditioned on the head of its parent constituent *m*, so we also collect probabilities such as:

$$P(\text{head} = \text{United}|\text{parent} = \text{Houston}).$$

For a non-lexical constituent C, given a rule r of the form $C \to C_1 \ldots C_n$, where h is the head of C and m is head of the parent constituent of C, we calculate the probability of C as follows.

$$P(C) = P(r|h)P(h|m)\prod_{i=1}^{n} P(C_i)$$

a. List all the lexicalized probabilities that are relevant for choosing which of the two structures from Problem 1 is more probable. Do not list any other probabilities.

b. Are there lexical dependencies that are relevant for choosing attachment of PPs that are not captured by the lexicalized model? Support your answer with examples.

Problem 6. Assume a dependency grammar with *unlabeled* arcs.

a. Draw dependency structures that correspond to the two structures from Problem 1. Use dependency graphs similar to those in Jurafsky and Martin Figure 15.1, but without the labels.

United diverted flights to Houston

(meaning: flights landed in Houston) | *(meaning: flights did not land in Houston)*

b. Assume a dependency parser is parsing the above string, as in Jurafsky and Martin, section 15.4. At what point in the parser's operation does it reach an ambiguity? Show the parser's configuration at that point.

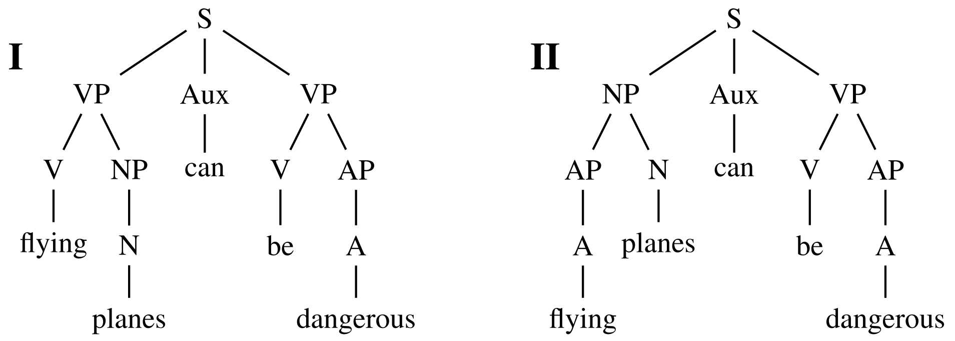
Stack	Word list	Relations

c. At the above point, which parser action corresponds to each of the structures from part (a)?

Meaning:	*flights landed in Houston*	*flights did not land in Houston*
Action:		

d. What is the difference between the above structures that allows the *training algorithm* (Jurafsky and Martin section 15.4.1) to learn the correct action for each structure?

Problem 7. The following structures are generated by a probabilistic context-free grammar (PCFG).



Each rule in the grammar is associated with a probability. For the non-lexical constituents (S, NP, VP, AP), given a rule r of the form $C \rightarrow C_1 \ldots C_n$, we calculate the probability of C as follows.

$$P(C) = P(r)\prod_{i=1}^{n} P(C_i)$$

For the lexical constituents (N, V, A), the probability of the constituent is the probability of the part-of-speech tag, as determined by a POS tagger.

a. Which structure is associated with which meaning?

   danger is in flying:

   danger is in the planes:

b. List all the syntactic rules (for the non-lexical constituents) which are relevant for choosing between the structures. Do not list any other rules.

c. Which of the lexical constituents will affect the choice between the structures? Why?

Problem 8. Assume the same structures as before, but with lexicalized probabilities.

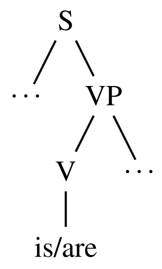
The probability of a rule r is conditioned on its head h, so, for example, we will have separate probabilities for the following rules: AP $\rightarrow$ A|A = flying; AP $\rightarrow$ A|A = dangerous.

The probability of a head h is conditioned on the head of its parent constituent m, so we also collect probabilities such as $P(\text{head} = \text{dangerous}|\text{parent} = \text{flying})$.

For a non-lexical constituent C, given a rule r of the form $C \rightarrow C_1 \ldots C_n$, where h is the head of C and m is head of the parent constituent of C, we calculate the probability of C as follows.

$$P(C) = P(r|h)P(h|m)\prod_{i=1}^{n} P(C_i)$$

a. List all the lexicalized probabilities that are relevant for choosing which of the two structures from Problem 5 is more probable (assume that Aux is the head of S). Do not list any other probabilities.



b. Now assume a modified grammar, where the top of the structures is replaced as in the image to the left (assume that V is head of both VP and S). How can the lexicalized PCFG ensure that the appropriate parses are given for the sentences *flying planes are dangerous* and *flying planes is dangerous*? What are the problems with using lexicalized probabilities for this purpose?

Problem 9. Assume a dependency grammar with unlabeled arcs.

a. Consider the ambiguous sentence *flying planes can be dangerous*. Draw unlabeled dependency structures (arc format) that correspond to the two meanings; assume the root of the sentence is the word *be*.

(meaning: danger is in the planes)

$\downarrow$ (over **be**)

flying planes can be dangerous

(meaning: danger is in flying)

$\downarrow$ (over **be**)

flying planes can be dangerous

b. Assume a shift-reduce dependency parser is parsing the above string. At what point in the parser's operation does it reach an ambiguity? Show the parser's configuration at that point.

Stack	Word list	Relations

c. At the above point, which parser action (RIGHTARC, LEFTARC, SHIFT) corresponds to each of the structures from part (a)?

Meaning:	*danger is in the planes*	*danger is in flying*
Action:		

d. Now consider the unambiguous sentences *flying planes are dangerous* and *flying planes is dangerous*. Place each sentence with the appropriate meaning, and draw unlabeled dependency structures.

(meaning: danger is in the planes)

$\downarrow$

flying planes ______ **dangerous**

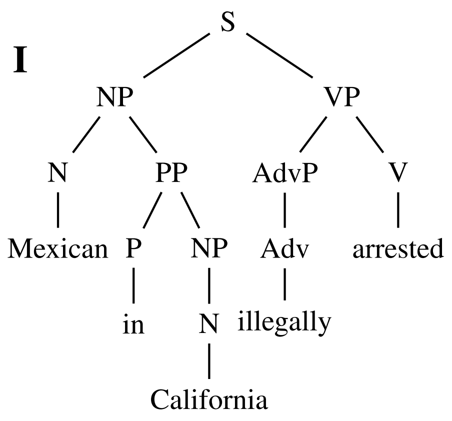
(meaning: danger is in flying)

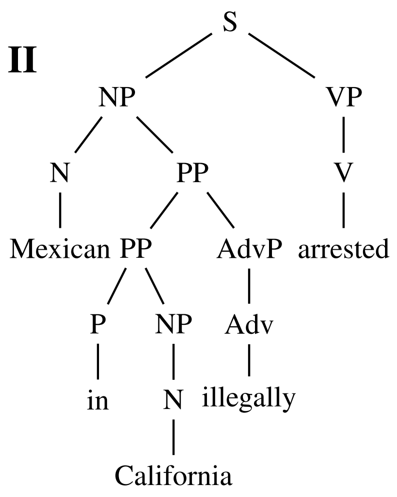
$\downarrow$

flying planes ______ **dangerous**

e. What about the parser's configuration at the point of ambiguity allows it to choose the correct action in each case?

Problem 10. The following structures are generated by a probabilistic context-free grammar (PCFG).





Assume the grammar has lexicalized probabilities: The probability of a rule r is conditioned on its head h, so, for example, we will have separate probabilities for the following rules:

$$\text{NP} \rightarrow \text{N} | \text{N} = \text{Mexican}$$
$$\text{NP} \rightarrow \text{N} | \text{N} = \text{California}$$

The probability of a head h is conditioned on the head of its parent constituent m, so we also collect probabilities such as:

$$P(\text{head} = \text{California} | \text{parent} = \text{in})$$

For a non-lexical constituent C, given a rule r of the form $C \rightarrow C_1 \ldots C_n$, where h is the head of C and m is head of the parent constituent of C, we calculate the probability of C as follows.

$$P(C) = P(r|h)P(h|m)\prod_{i=1}^{n} P(C_i)$$

a. List all the lexicalized probabilities that are relevant for choosing between the structures. Do not list any other probabilities.

Problem 11. Assume a dependency grammar with unlabeled arcs.

a. Consider the ambiguous sentence *Mexican in California illegally arrested.* Draw unlabeled dependency structures (arc format) that correspond to the two meanings; assume the root of the sentence is the word *arrested.*

(meaning: arrest is illegal)	*(meaning: presence is illegal)*
$\downarrow$	$\downarrow$
Mexican in California illegally arrested	**Mexican in California illegally arrested**

b. Assume a shift-reduce dependency parser is parsing the above string. At what point in the parser's operation does it reach an ambiguity? Show the parser's configuration at that point.

Stack	Word list	Relations

c. At the above point, which parser action (RIGHTARC, LEFTARC, SHIFT) corresponds to each of the structures from part (a)?

Meaning:	*arrest is illegal*	*presence is illegal*
Action:		

d. Compared to the lexicalized PCFG from the previous problem, which word relations do both parsers consider in a similar fashion? Which relations do they treat differently (that is, considered by one parser and not the other)? Why?