Goals. This assignment's purpose is to provide a hands-on introduction to the latent structure underlying how words are arranged in language—that is, syntax. This assignment focuses on constituencies, dependencies, parsing, and parts of speech.

Part 1: Sequence Labeling

Take the following transition probabilities and emission probabilities:

```
Transition probabilities:
P(D | START) = 0.5   P(N | START) = 0.4   P(V | START) = 0.1
P(N | D) = 0.9       P(V | D) = 0.1
P(V | N) = 0.6       P(N | N) = 0.3       P(D | N) = 0.1
P(N | V) = 0.4       P(D | V) = 0.3       P(V | V) = 0.3

Emission probabilities:
P("the" | D) = 1.0       P("dog" | N) = 0.3
P("fish" | V) = 0.4      P("can" | V) = 0.4
P("fish" | N) = 0.2      P("can" | N) = 0.2
```

Q1 (10 points). Use the Viterbi algorithm to find the most likely tag sequence for the sentence "the fish can fish". Do this by filling in each valid cell of the Viterbi chart (8 points). If a word cannot be associated with a given POS tag, you do not have to fill in that cell. After drawing the chart, provide the most likely tag sequence (2 points).

	the	**fish**	**can**	**fish**
D	$1.0 \cdot 0.5 =$ 0.5			
N		$0.5 \cdot 0.2 \cdot 0.9 =$ 0.09	$0.3 \cdot 0.09 \cdot 0.2 =$ 0.0054	$0.4 \cdot 0.0216 \cdot 0.2 =$ 0.001728
V		$0.5 \cdot 0.4 \cdot 0.1 =$ 0.02	$0.6 \cdot 0.09 \cdot 0.4$ 0.0216	$0.3 \cdot 0.0216 \cdot 0.4 =$ 0.002592

The most likely tag sequence is D N V V.

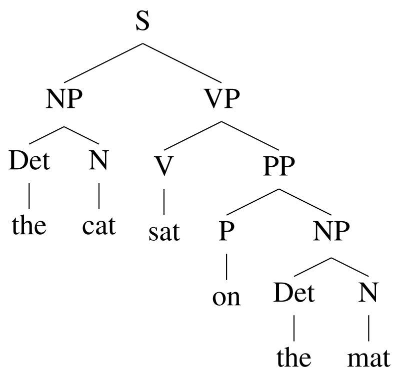
Q2 (3 points). In 2–3 sentences, why might an HMM tagger trained on news articles perform poorly on text from social media?

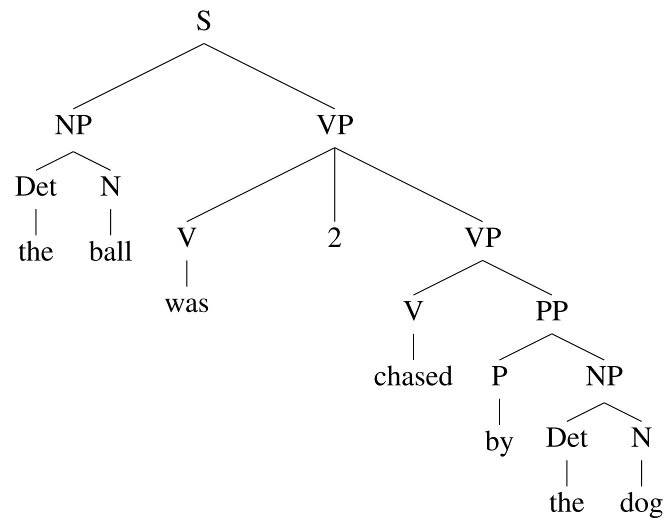
Social media uses very different words and different syntactic structures than news articles. This means that the transition probabilities *and* emission probabilities would not transfer well.

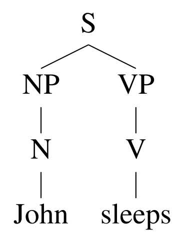
Part 2: Constituencies and Context-free Grammars

As we've discussed in class, words can often work together to form higher-order structures called **constituencies**. Constituencies can recurse into ever-higher-order structures, giving language a sort of infinite potential complexity. While natural language is technically not context-free,[1] context-free grammars (CFGs) still provide a nice theoretical playground for us to work through how to think about these latent structures.

Q3 (5 points). Here are three constituency trees that were each generated by the same CFG:







Write the minimal CFG that could generate each of these sentences.

```
S -> NP VP          N -> John | cat | mat | ball | dog
NP -> Det N         V -> sat | was | chased | sleeps
NP -> N             P -> on | by
VP -> V             Det -> the
VP -> V PP
VP -> V VP
PP -> P NP
```

Q4 (6 points). The following is a probabilistic context-free grammar (PCFG). The brackets to the left of each rule correspond to the probability that the given nonterminal will generate the associated right-hand side; the probabilities *per non-terminal* sum to 1.

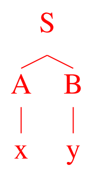
```
[0.6] S -> A B
[0.4] S -> A C
```

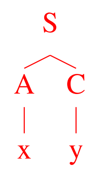
[1] `https://www.eecs.harvard.edu/shieber/Biblio/Papers/shieber85.pdf`

```
[1.0] A -> x
[0.5] B -> y
[0.5] B -> z
[0.5] C -> y
[0.5] C -> z
```

How many parses are there for the string "x y"? Which parse is the most probable under the provided grammar? Provide each parse alongside its probability; feel free to either draw constituency trees or use bracket notation.

There are two parses:





Or equivalently: [S [A x] [B y]], and [S [A x] [C y]].

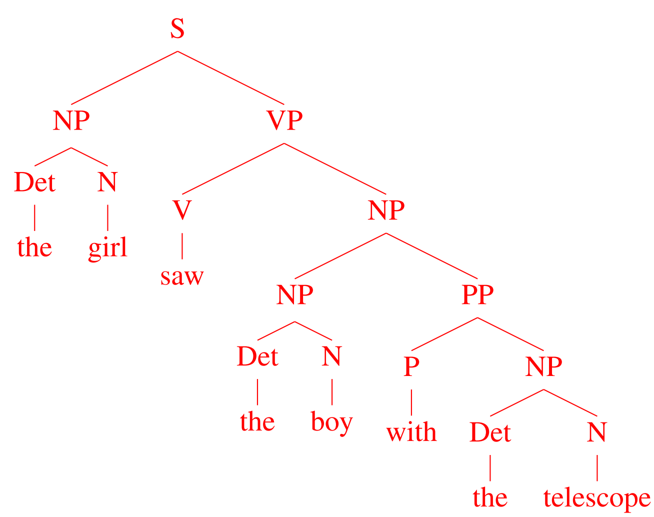
The probability of the AB parse is: $p(S \to A\ B) \times p(A \to x) \times p(B \to y) = 0.6 \times 1.0 \times 0.5 = 0.3$.

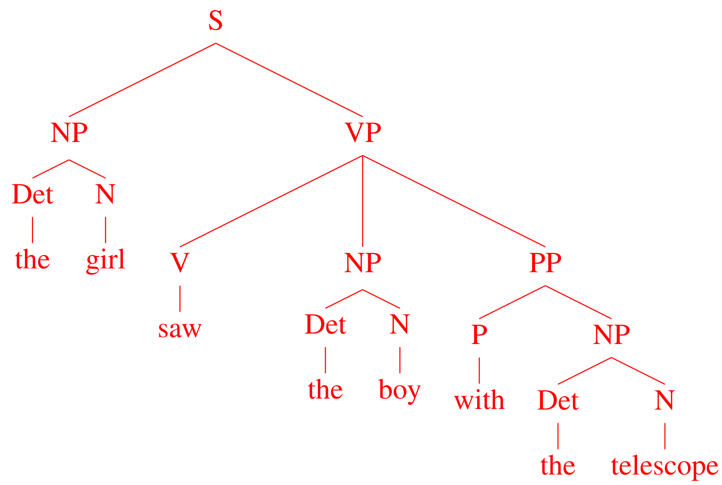
The probability of the AC parse is: $p(S \to A\ C) \times p(A \to x) \times p(C \to y) = 0.4 \times 1.0 \times 0.5 = 0.2$.

The more probable parse is the lefthand one of the above trees, or that beginning with $S \to A\ B$.

Q5. The sentence "The girl saw the boy with a telescope" is ambiguous. Here, you will show precisely how it is ambiguous.

(a; 6 points) Provide constituency trees for both possible parses, and state which of the parses you believe is more probable. Use only the following non-terminals: S, NP, VP, PP, N, V, P, Det.





As for which is more probable, either choice is defensible. The grammar in part (b) just needs to reflect the choice you made.

(b; 4 points) Based on your answer, write a minimal probabilistic CFG that yields both parses, and assigns a higher probability to the more probable parse.

If the second option was chosen, the grammar could look like this:

```
[1.0] S -> NP VP        [0.33] N -> girl
[0.8] NP -> Det N       [0.33] N -> boy
[0.2] NP -> NP PP       [0.33] N -> telescope
                        [1.0] V -> saw
[0.2] VP -> V NP        [1.0] P -> with
[0.8] VP -> V NP PP     [0.5] Det -> the
[1.0] PP -> P NP        [0.5] Det -> a
```

If the first option was chosen, then `VP -> V NP PP` needs to have a lower probability, and `NP -> NP PP` needs to have a higher probability.

Q6. In class, we've walked through the CKY algorithm using an upper-triangle matrix, like the following:

NP 0.3		S $0.12 \cdot 0.3 =$ 0.036		S $0.3 \cdot 0.0072 =$ 0.00216 S $0.3 \cdot 0.0096 =$ 0.00288
	V 1.0	VP $0.6 \cdot 0.2 =$ 0.12		VP $0.6 \cdot 0.012 =$ 0.0072 VP $0.12 \cdot 0.4 \cdot 0.2 =$ 0.0096
		NP 0.2		NP $0.2 \cdot 0.2 \cdot 0.3 =$ 0.012
			P 1.0	PP 0.2
				NP 0.2
she	ate	rice	with	chopsticks

Here is a PCFG for this sentence:

```
[1.0] S  -> NP VP              [1.0] V -> ate
[0.3] NP -> she                [1.0] P -> with
[0.2] NP -> rice
[0.2] NP -> chopsticks         [0.6] VP -> V NP
[0.3] NP -> NP PP              [0.4] VP -> VP PP
[1.0] PP -> P NP
```

(a; 10 points). Fill in the CKY matrix using the provided PCFG. Include the fully filled out matrix in your written report. Be sure to include probabilities for each non-terminal in each non-empty cell.

Answer in diagram.

(b; 2 points). How many valid parses are there? Which parse is the most likely according to the given PCFG?

According to the CKY chart, there are two valid parses (in the top-right cell). The parse with higher probability is the one where “with chopsticks” attaches to the verb “eats”, rather than the one where “with chopsticks” attaches to the NP “rice”.

(c; 2 points). In your own words, describe the sentence’s meaning according to each parse. Use only 1–2 sentences per possible parse.

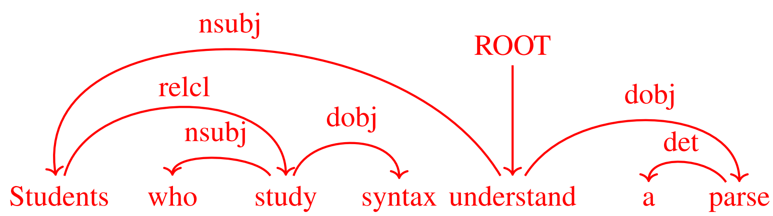
When “with chopsticks” attaches to the verb “eats”, this means that she was using chopsticks to eat the rice.

When “with chopsticks” attaches to the NP “rice”, this means that she was eating the chopsticks as part of the rice.

Part 3: Dependencies

Unlike constituencies, dependencies do not presume that there are latent structures above the word level. Dependencies are asymmetric relationships between *heads* and their *dependents*. For example, a subject is a dependent of its verb head, and an adjective is a dependent of its noun head. These relationships are depicted as arrows from heads (parents) to dependents (children).

Q7. (5 points) Draw the dependency tree for the following sentence: "Students who study syntax understand a parse." Label each arc with its dependency label. You should only need the following dependency labels: nsubj, dobj, relcl, det.

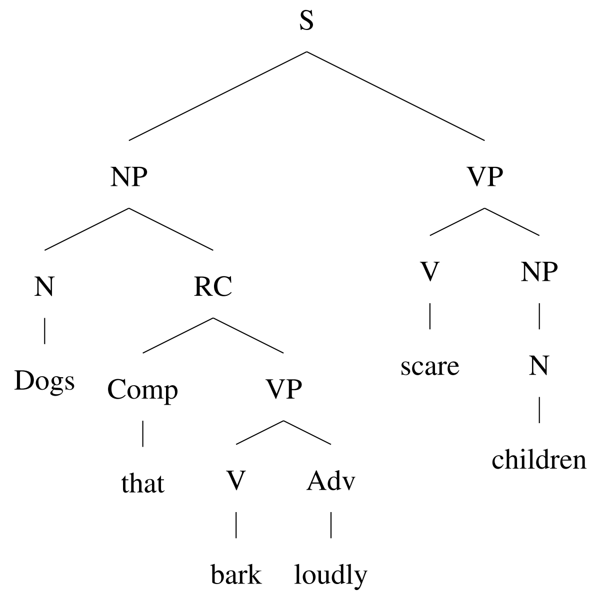


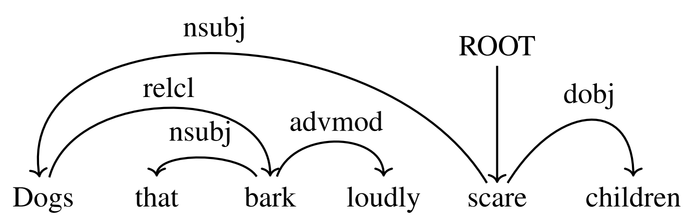
Q8. (10 points) Consider the sentence "The cat chased mice." We'll use a shift-reduce parser to draw the dependency tree. Assume a transition system like the one we've covered in class with three operations: SHIFT, LEFT-ARC, RIGHT-ARC.

Starting with an empty stack and the full sentence in the buffer, show the sequence of operations needed to build the correct dependency tree using a shift-reduce parser. For each step, show the following: the stack contents, the buffer contents, the operation performed for a given step, and any dependency relations created in that step. Use only the following dependencies: det, nsubj, dobj. Note that you may not need all rows of this table.

Step	**Stack**	**Buffer**	**Operation**	**New Relations**
0	[ROOT]	[The, cat, chased, mice]	SHIFT	—
1	[ROOT, The]	[cat, chased, mice]	SHIFT	—
2	[ROOT, The, cat]	[chased, mice]	LEFT-ARC	cat → The (det)
3	[ROOT, cat]	[chased, mice]	SHIFT	—
4	[ROOT, cat, chased]	[mice]	LEFT-ARC	chased → cat (nsubj)
5	[ROOT, chased]	[mice]	SHIFT	—
6	[ROOT, chased, mice]	[]	RIGHT-ARC	chased → mice (dobj)
7	[ROOT, chased]	[]	RIGHT-ARC	ROOT → chased
8	[ROOT]	[]	done	—
9				
10				

Q9. Consider the sentence “Dogs that bark loudly scare children.” Here is a constituency tree (top) and a dependency tree (bottom) for this sentence:





(a; 4 points) In the constituency tree, which node is the relative clause ("that bark loudly") a child of? In 1–2 sentences, what does this grouping imply about the relationship between the relative clause and the noun "Dogs"?

It is a child of the NP corresponding to “dogs”. This implies that the relative clause directly modifies “Dogs”, telling us what the dogs do.

(b; 3 points) In the dependency tree, which word is the head (main word) of the relative clause? What word is this head a dependent of? What is the label of this head–dependent arc?

The main word of the relative clause is “bark”. This word is a dependent of the word “Dogs”; the arc name is “relcl”.

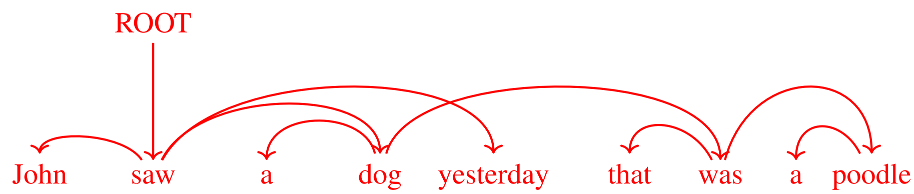
(c; 3 points) Consider the constituent (subtree) defined by the top-level NP node ("Dogs that bark loudly"). Using just “yes” or “no”, does any single node or arc in the dependency tree capture this grouping? In 1–3 sentences, what specific information does this constituency subtree encode that is absent from the dependency tree?

No. The constituency tree encodes that the relative clause as a whole is a part of the overall noun phrase headed by “Dogs”. The dependency tree doesn’t directly this information in a single edge, instead just giving pairwise word relations.

Extra Credit

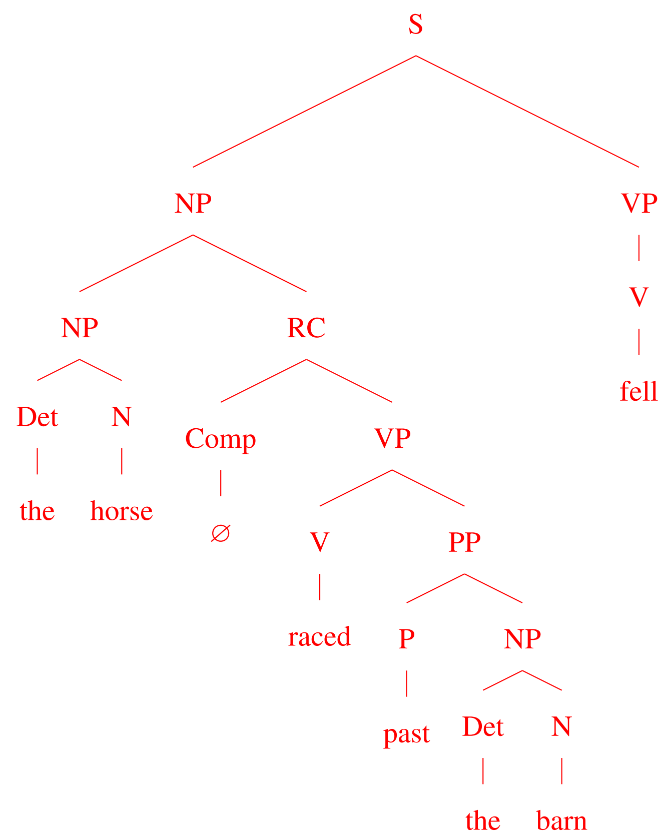
E1. (6 points) As stated in Part 1, natural language is not actually context-free. Come up with an example sentence that shows that natural language is not context-free, and draw its dependency parse (just the arrows; you do not have to label the edges). You may not use the examples from the class slides, nor the example given in the linked paper by Stuart Shieber.

We were looking for a sentence with a non-projective dependency parse. One possibility is "John saw a dog yesterday that was a poodle." Here's its dependency parse:

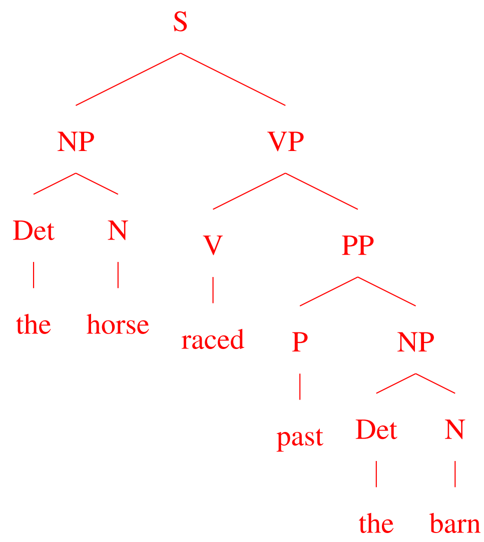


E2. Consider the sentence "The horse raced past the barn fell." This is a grammatical English sentence (believe it or not), but it is difficult for people to parse.

(a; 3 points) Draw the constituency tree for this sentence.



(b; 3 points) Draw the constituency tree for "The horse raced past the barn."



(c; 5 points) In 2–3 sentences, explain why humans find "The horse raced past the barn fell." difficult to process, and what specific structural property of the sentence causes this difficulty.

Humans find this hard to parse because the sentence initially suggests one parse up until "barn". However, when we see "fell", we find out that the correct parse is completely different. This means we have to go back and revise our initial parse significantly.

Submission

Upload your write-up to Gradescope as a PDF to **HW3: Syntax and Parsing**.