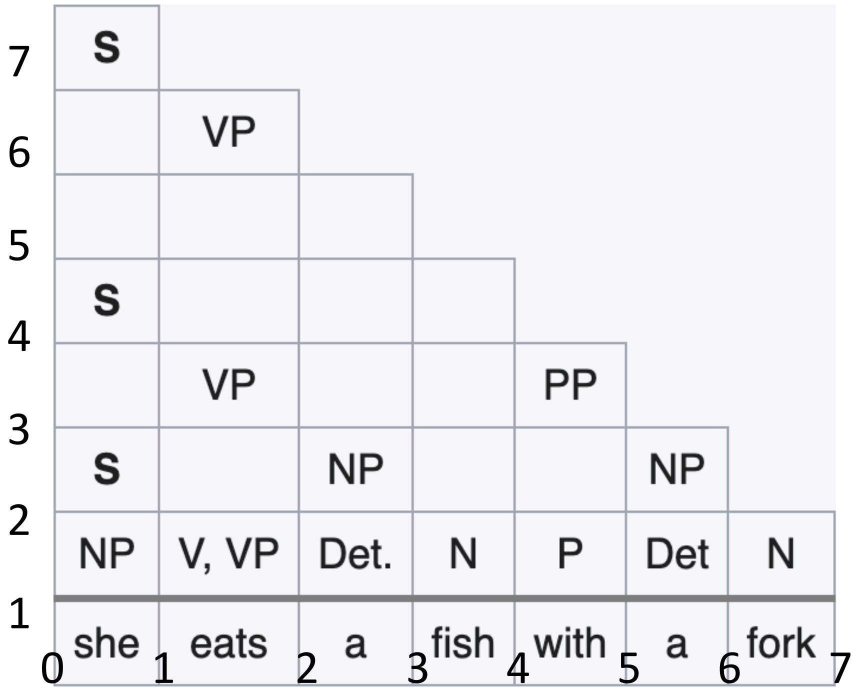
Assignment 3: Parsing problem 1



7	S						
6		VP					
5							
4	S						
3		VP			PP		
2	S		NP			NP	
1	NP	V, VP	Det.	N	P	Det	N
	she	eats	a	fish	with	a	fork

0 she 1 eats 2 a 3 fish 4 with 5 a 6 fork 7

$$
\begin{aligned}
\text{S} &\longrightarrow \text{NP VP} \\
\text{VP} &\longrightarrow \text{VP PP} \\
\text{VP} &\longrightarrow \text{V NP} \\
\text{VP} &\longrightarrow \text{eats} \\
\text{PP} &\longrightarrow \text{P NP} \\
\text{NP} &\longrightarrow \text{Det N} \\
\text{NP} &\longrightarrow \text{she} \\
\text{V} &\longrightarrow \text{eats} \\
\text{P} &\longrightarrow \text{with} \\
\text{N} &\longrightarrow \text{fish} \\
\text{N} &\longrightarrow \text{fork} \\
\text{Det} &\longrightarrow \text{a}
\end{aligned}
$$

a. Consider the above grammar, and the CKY parsing table filled in for the sentence *she eats a fish with a* fork. Suppose you added the rules *N → a* and *NP → N N* to the grammar. These allow you to use "a" as a modifier, as in talking about the "a team" versus "b team".

   i. Suppose you are using CKY as a *recognizer* (SLP Section 13.2.2). What does the table look like with the updated grammar? (Copy the table and show what you need to add.)

   ii. Suppose you are using CKY as a *parser* (SLP Section 13.2.3). Show what the table looks like with the changes you need to make so that it is capable of returning all parses for this sentence. (You can copy the result from the previous question a(i) and show what you need to add.)

b. How many parse trees are there for the whole sentence, with this updated grammar? Enumerate them.

Assignment 3: Parsing problem 2

a. For each of the following sentences, draw a syntactic dependency tree (e.g. see SLP example 14.2) that reasonably captures its structure. Try to use the syntactic dependency relations in SLP Figure 14.3 where possible, but if you have to invent plausible labels, that's ok.

- A lion ate my beagle
- My beagle was eaten by a lion
- The beagle was eager to eat
- The beagle was easy to eat

b. Go to the online spaCy dependency parser demo at https://explosion.ai/demos/displacy. Un-check "merge phrases" and "merge punctuation". Parse each of the above sentences and compare/contrast with your manual analyses. Briefly explain any differences between your analysis and the automatic analysis.

c. Suppose you were consulting for a news agency that had a big database of news reports, and your job was to help a reporter find stories about pets getting eaten by wild animals. Assuming you could (easily) apply the spaCy parser to sentences in the story, what issues would you foresee needing to deal in order to succeed, based on what you saw in part b? How might you imagine (eagerly) dealing with those issues?

Assignment 3: Extra credit (up to 20%)

The repository at https://github.com/psresnik/nlp_assignments/tree/main/assignment3 contains working code that is dense with explanatory comments, including:

assignment3.py (run with −h flag to see arguments)
• Reads in political speeches and reports the most frequent direct objects for a given verb.
cluster_words.py
• Example of how to take a set of words and do K-means clustering using their vector representations.

Some ideas of things you can do for extra credit:

- Run assignment3.py with a selection of commandline arguments to answer questions like the following:
  - When Republican senators use the verb *kill*, what are they talking about killing? Are there any interesting similarities or differences versus Democrats?

  Generate some questions of this kind that one might want to ask about what gets talked about in Congress, e.g. what do different parties speak about reducing? Increasing? Use the program to explore, and report on your results − are you able to find out anything interesting or insightful? Consider whether modifying the program slightly to also use the *nsubj* relation might also yield anything interesting.

- Rather than just looking at frequent direct objects for verbs of interest, extend the code to explore whether those direct objects group together in any interesting ways to form a more general pattern, using clustering.

- As a somewhat ambitious mini-project, explore how language use might tell you interesting things about legislators. For example, you could create your own vector representation for each senator, where each dimension is a verb-object combination (e.g. *reduce-emission*), and then cluster legislators. (Note: if you do this, might want to focus on frequent verbs and/or objects and you might want to use tf-idf weighting to reduce the influence of very frequent but uninteresting verb-combinations like *yield-floor*.)

Those are just a few ideas. Have fun!