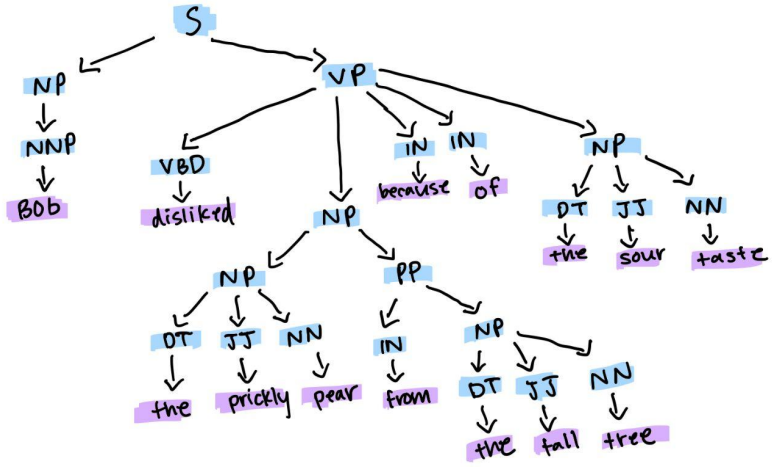
Portfolio 5: Sentence Parsing
Name: Shreya Valaboju
CS 4395.001

This document highlights the concepts related to sentence syntax, the 3 types of sentence parses: PSG, dependency, and SRL, and how to use sentence parsers.

Let's look at a fairly complex sentence:

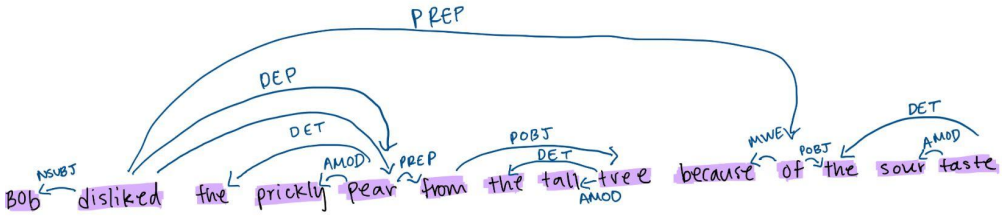
> *Bob disliked the prickly pear from the tall tree because of the sour taste*

Here is the PSG tree, also known as constituent parsing, for the example sentence:



S: simple declarative clause
NP: noun phrase
VP: verb phrase
NNP: singular proper noun
VBD: past tense verb phrase
DT: determiner, modifies/introduces a noun
JJ: adjective
NN: singular noun
IN: preposition or subordinating conjunction

Below is a dependency parse for the sentence with labeled relations:



NSUBJ: nominal subject, the noun phrase for the subject of a clause
PREP: prepositional modifier
DEP: dependent
DET: determiner, modifies/introduces a noun
AMOD: adjectival modifier, adjective phrase that modifies a noun phrase
POBJ: object of a preposition, head of a noun phrase following a preposition
MWE: multi-word expression, behave like one word

Let's take a look at a SRL (Semantic Role Label) parse for the sentence.

Predicate (verb): *'disliked'*

Arguments:
A0: 'Bob'
A1: 'the prickly pear'
A2: 'from the tall tree'

A0 is 'Bob' because it is the agent doing the action, as 'Bob' is the one who disliked 'the prickly pear.' A1 represents the passive actor, 'the prickly pear,' as it is the entity that is acted upon and 'disliked' by 'Bob.' A2 is the instrument, the entity used in action. In our case, it is 'from the tall tree,' as it's used in the action of 'Bob' 'disliking the prickly pear.' Arguments are the actors in a sentence.

Modifiers:
CAU : 'because of the sour taste'

The modifier 'CAU' is the reason for action. The action was that 'Bob disliked the prickly pear.' But why Bob disliked' was because of the sour taste' of that pear. Modifiers generally add more information and are not arguments.

For my sentence, the constituency parser or the PSG tree was fairly straightforward. I was able to identify the various types of phrases and their parts of speech. Because my sentence was not structurally ambiguous, the PSG parser worked fairly well. However, a downside to PSG parsing is when sentences are ambiguous, it can be unclear in determining the correct parse based on intended syntactical meaning. In the dependency parse for the sentence, an advantage was that it was straightforward, showcasing the dependencies and relationships between words. A disadvantage was that it can be difficult to put together and comprehend. It was difficult to understand the types of dependencies and their abbreviations. Also, my sentence had a lot of adjectives and 2 clauses, making it harder to correctly identify the dependencies and which tokens they originate from. Lastly, the SRL parser's advantage for the sentence is that it was easy to parse. For my sentence, identifying the modifiers and arguments were clear and simple. A disadvantage of this parser is that it does not go into detail for each token in the sentence. It looks at arguments, which could be phrases. It does not take into account, as indicatively, the specific part of speech for particular words and the relationship between them as does a dependency parse or a PSG tree.

Sources:

- https://gist.github.com/nlothian/9240750
- https://demo.allennlp.org/machine-comprehension
- https://corenlp.run/
- The Stanford Dependencies Manual