September 23, 2020

1 Stabilizing the system

Up to now, things have been left kind of free with respect to what we can do with the formalism, but it seems like it might start to be more productive to lay down some assumptions that we can at least presume until we have evidence to the contrary.

1.1 Phrase structure rules

Our description of the grammar of the languages we are investigating will be in terms of phrase structure rules. A phrase structure rule has a single symbol on the left and represent a way that a symbol can be "expanded." These rules are generally context-free, meaning that there is only a single symbol on the left of the arrow. Any context dependence will come from features of lexical items. The derivation from S to the words can be represented more familiarly in a tree structure.

The project is to come up with an adequate/explanatory account of English (or whatever language we are analyzing). So, adhering too close to the motivating data ("'overfitting"') is not scientifically useful. We want the system to predict other things, not just exactly the data that motivated it. If it looks like there is a generalization over most of the observations, try to encode that in the grammar and if there are exceptions, it becomes interesting to understand why they are (or appear to be) exceptions.

A note on parentheses: In a phrase structure rule, we will allow for parentheses as a notational shortcut, a way to write two rules in one line. I won't do this here below, but you could in principle write the first two VP rules in the next block in a single line as "VP $\rightarrow$ V (NP)". That's not making a different proposal from the two rules separately do. However, note that if you write "VP $\rightarrow$ V (NP) (PP)" that *is* a different proposal from what is below, because it also includes the rule "VP $\rightarrow$ V PP"—which might be correct in the end, and probably is, but it is nevertheless a property of this shorthand. If you have two parenthesized elements, you are writing four rules in shorthand.

So, first up. We can start with S, the symbol for a stand-alone main clause sentence. That's fine. We know already that subjects and objects can be much bigger than a single word, and it would seem that anything that can be a subject can be an object, so we will presume that we are generally working with NPs that contain Ns. We also have seen with constituency tests that there is a node that contains the verb and its objects, which we will call VP, and for rule simplicity we will assume that even an intransitive verb "projects" a VP.

$$
\begin{array}{rl}
\vdots & \\
\text{S} \rightarrow & \text{NP VP} \\
\text{VP} \rightarrow & \text{V} \\
\text{VP} \rightarrow & \text{V NP} \\
\text{VP} \rightarrow & \text{V NP PP} \\
\vdots &
\end{array}
$$

Within the NP, we know that we can have a determiner, some number of adjectives, and a main noun. The iterativity of adjectives can be provided by a recusrive rule that rewrites N as N and an adjective.

$$
\begin{array}{rl}
 & \vdots \\
\text{NP} \rightarrow & \text{Det N} \\
\text{N} \rightarrow & \text{Adj N} \\
 & \vdots
\end{array}
$$

The rules above represent a claim that all NPs have a Det in them. This means that we need to in princple allow for silent determiners to cover cases like *Bart* or *spoons*. I have not written Adj as an AdjP for the moment because it's hard to say that we have evidence of a full phrase there. It is possible to modify an adjective with an adverb (like *unbelievably hungry*), but we can handle that with a recursive rule attaching adverbs to adjectives. Later if we find evidence that requires us to expand this to AdjP (and/or AdvP), we can revise our rule.

$$
\begin{array}{rl}
 & \vdots \\
\text{Adj} \rightarrow & \text{Adv Adj} \\
 & \vdots
\end{array}
$$

In fact, it is also possible for an adverb to modify an adverb (*extremely unbelievably hungry*), so we can posit another recursive rule for this, though it feels like it is kind of a confusing one. One adverb is being attached to the other, but the rule doesn't really indicate which is which. At some point, we might want to question that.

$$
\begin{array}{rl}
 & \vdots \\
\text{Adv} \rightarrow & \text{Adv Adv} \\
 & \vdots
\end{array}
$$

Another place we have adverbs is modifying a VP, like in *Bart ate carefully*. These are adjuncts, so we want a recursive rule. It seems like in principle adverbs can attach on either side of a VP (*Bart carefully ate*).

$$
\begin{array}{rl}
 & \vdots \\
\text{VP} \rightarrow & \text{VP Adv} \\
\text{VP} \rightarrow & \text{Adv VP} \\
 & \vdots
\end{array}
$$

We also have prepositional phrases, which generally seem to be a P and an NP. And these can attach to NPs (or Ns, I suppose, but I'm going to just choose NP) and to VPs (same comment re: V).

$$
\begin{array}{rl}
 & \vdots \\
\text{PP} \rightarrow & \text{P NP} \\
\text{VP} \rightarrow & \text{VP PP} \\
\text{NP} \rightarrow & \text{NP PP} \\
 & \vdots
\end{array}
$$

Since we are shortly going to talk about embedding sentences inside one another (like *Pat thinks that Tracy left*), I will go ahead and provide the rules we need for those.

	⋮
CP →	C S
VP →	V CP
	⋮

We will need to tweak these here and there, but the basic grammatical setup then is:

S →	NP VP
VP →	V
VP →	V NP
VP →	V NP PP
VP →	V CP
VP →	VP Adv
VP →	Adv VP
VP →	VP PP
NP →	Det N
N →	Adj N
NP →	NP PP
Adj →	Adv Adj
Adv →	Adv Adv
PP →	P NP
CP →	C S

1.2 Lexical entries

Because we kept our phrase structure rules pretty general and context-free, whatever context sensitivity we need to add is going to have to be part of the lexical items.

A lexical item has the following structure:

word, category, features

The category is something like V, N, P, Det, Adv, Adj. The features are things like [+pl] for plural, or a subcategorization frame that determines whether a lexical item will "fit" into the structure, which is called a **subcategorization frame**.

We will have classes of lexical items that will be defined by subcategorization frames.

- [+ __] is the subcategorization frame for an instransitive verb.
- [+ __ NP] is the subcategorization frame for a transitive verb.
- [+ __ NP NP] is the subcategorization frame for a ditransitive verb.

Determiners too can have subcategorization frames that put them in classes

- [+ __ $N_{[-pl, -prop]}$] is the subcategorization frame for determiners like *each*.
- [+ __ $N_{[-prop]}$] is the subcategorization frame for determiners like *no* or *the*.
- [+ __ $N_{[-pl, -prop]}$] is the subcategorization frame for determiners like *two* or *both*.

And there can be silent determiners, like the one for bare plurals and the one for proper names.

- SOME (no phonology), Det, [–def], [+ __ $N_{[+pl]}$]
- PROP (no phonology), Det, [+def], [+ __ $N_{[+prop]}$]
- *some*, Det, [–def]
- *a(n)*, Det, [–def], [+ __ $N_{[-pl]}$]

We will assume that generally the subcategorization frames in a head-initial language like English look like [+ __ . . .]. That is, they generally constrain things to their right, at least when we are considering arguments/complements (not adjuncts). Practically, this means Det can constrain Ns, rather than vice-versa. There is something deeper here, but this will do for now.

Lexical item	Category	Features	Subcategorization frame
the	Det	[+def]	[+ _ $N_{[-prop]}$]
a	Det	[–def]	[+ _ $N_{[-prop, -pl]}$]
some	Det	[–def]	[+ _ $N_{[-prop]}$]
SOME(∅)	Det	[–def]	[+ _ $N_{[-prop, +pl]}$]
PROP(∅)	Det	[+def]	[+ _ $N_{[+prop]}$]
Bart	N	[–pl, +prop]	
boy	N	[–pl, –prop]	
boys	N	[+pl, –prop]	
sneeze	V	[+sub]	[+ __]
hit	V	[+sub]	[+ __ NP]
give	V	[+sub]	[+ __ NP PP]

Another note on parentheses: What was mentioned above about parentheses in phrase structure bit rules is basically still true with lexical items. We can write a lexical item with multiple subcategorization frame features, or a single subcategorization frame feature in parentheses, but this is really a notational shorthand for two lexical items. By writing them together, we are kind of acknowledging that they share a lot (pronunciation, basic semantics) and so probably we want the system to derive this someday, somehow. But strictly speaking, here too, parenthesized elements (or, in the case of multiple subcategorization frames, a set of options to choose among) could be expanded to multiple lexical items. It is in principle possible to have no subcategorization frame feature as well (meaning that it would be unconstrained).