Lecture 2, 09/03/2019
Motivation for Tree Adjoining Grammars: introduction to sentential structure

Why should we have formal explicit models of language structure , especially in an age of deep learning and learned representations?

Intuition suggests that such structure exists.
We may want to recover this structure to pass down downstream applications.
Inductive bias (?): Limit the search space.
You should now what your options are, even if you choose *not* to use such models.

What are some language characteristics we should try to capture?

("Linguistics amateur hour")
(Explain the *, bracket notation, ignore capitalization and punctuation, XP regularities.)

What we want to (and you should) think about here are *design choices.*

1. The president put $40 billion into department A's budget but only $40 into department B's (some sort of summarization or information extraction system)

2. cashiers put baskets in boxes (a simplification of 1) It turns out to be difficult to say "baskets in boxes" several times in a row.
3. cashiers put boxes in baskets
4. cashiers put boxes in baskets that had lovely bows and were practical -> [*baskets that had lovely bows and were practical*] is like *baskets*
5. * cashiers put boxes in put -> only certain types of things can be put in certain positions . Discussion ensues about whether "put" really has two obligatory

arguments … or whether there are different “put” verbs that take different nubers of obligatory arguments.

6. * cashiers put baskets
7. * cashiers put in boxes

8. cashiers put baskets in boxes ->

[[cashiers]$_{\text{noun phrase (NP), or subject}}$
[[put]$_{\text{V, or main verb}}$
[baskets]$_{\text{NP, or direct object}}$
[[in]$_{\text{preposition}}$ [boxes]$_{\text{NP}}$]$_{\text{prepositional phrase (PP), or location}}$
]$_{\text{verb phrase (VP), or predicate}}$
]$_{\text{S, or sentence; “main word” is the predicate’s verb}}$

Can we reuse a pre-existing, well-known, efficient formalism?

Introduction to context free grammars

What components should a formalism have?
There are rules about categories.
Categories are "invisible"; words aren't.
Finite sets for infinite generative capacity.

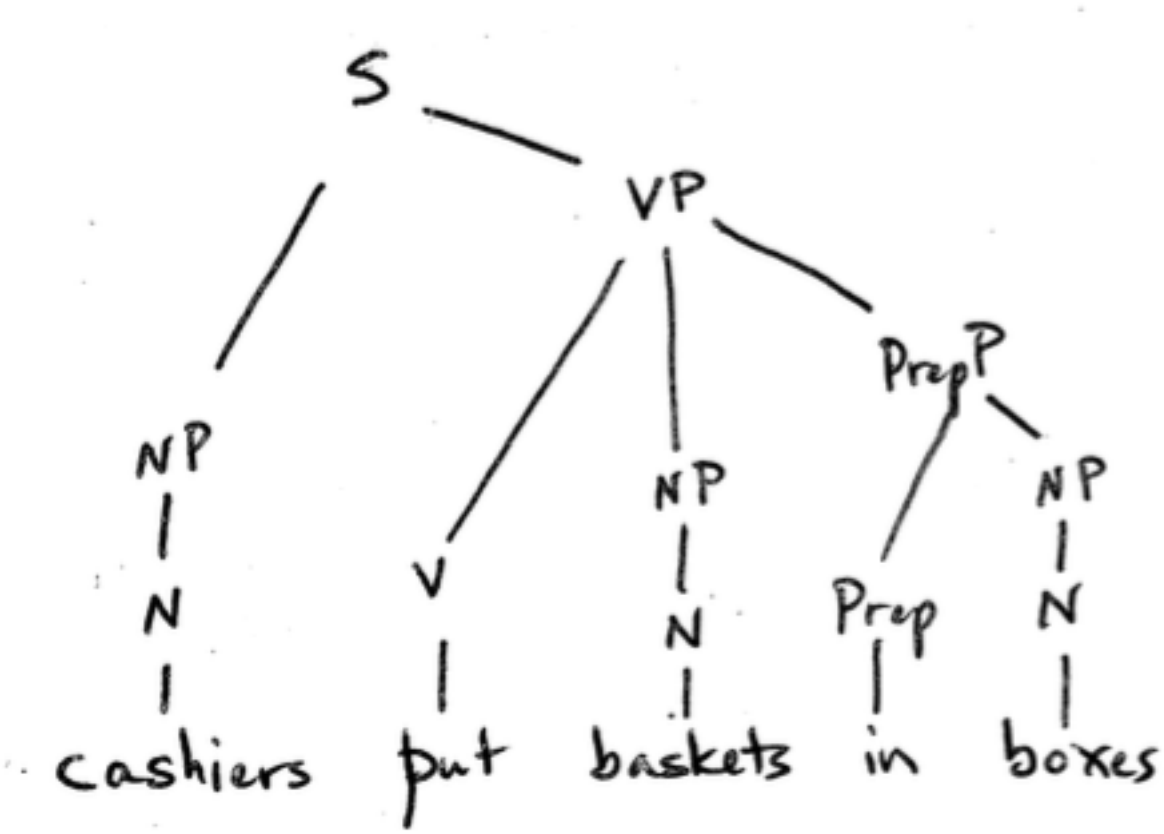
<see handout for formal definition (in English)>

9. VP → V NP PP (decomposable categories are uppercase by convention)
10. V → put (terminals are in lowercase by convention)
11. V → destroy

From a formal point of view, the symbols are arbitrary; from a grammar-writing point of view, “V” and “VP” are meant to be related, as are “NP” and “N”, and so on. “S” means “start symbol” but can also be thought of as “sentence”.

Parse trees are *induced, or the parse trees themselves induce the sentence*

<Very useful to have drawing of full parse tree on board for rest of lecture>



What does efficiency mean, in rough terms?

Do we want infinite generative capacity?
Does the grammar count as part of the input?
If we want all parse trees for a given input sentence, how much space does it take to even store all the trees? Exponential, in principle?

Handling local restrictions (let's be clever engineers)

Lexical information (characteristics of individual words, or *lexical items)* is important.

12. **she** puts boxes in baskets versus * **cashiers** puts boxes in baskets
    we must have VP ---→ "puts boxes in baskets" to create the first sentence but then how do we stop the second sentence?
13. **they** put boxes in baskets versus * cashiers put **they** in baskets -> *case* mismatch ("subject" v. "direct object")

we must have NP---→ "they" to create the first sentence but then how do we prevent the second>

14. ?? cashiers put **sleep** in baskets What if "sleep" is that crud in your eyes when you wake up?
    We should have NP ---→ "sleep" but then how do we stop this sentence?

We want to be able to create/analyze the "good" sentences w/out allowing the bad ones.

15. Lexical entry for "put" includes: subcategorization is 1: "puttable" NP in direct object form; 2: PPLoc; subject is animate NP

Look back at our parse tree; to "communicate" all these constraints to prevent the stars in 12-14, you need decompositions like

$VP_{subject:animateNP,1:puttableNP,2:locationPP} \rightarrow$
$V_{subject:animateNP,1:puttableNP,2:locationPP}$ $NP_{puttable,direct\ object\ form}$ $PP_{location}$

$S \rightarrow NP_{animate}\ VP_{subject:animateNP}$