LX 454 / 654
Acquisition of Syntax

Learnability

Logical problem of language acquisition

- The grammar that people end up with is very complicated, and underdetermined by the data.

- The main argument for this ("poverty of the stimulus") is that there are many generalizations that a kid *could* make on the basis of the input data that would be wrong, that would not result in a language that conforms to the principles that we've discovered seem to hold true of all adult languages.

That-t

- John said Mary will meet Bill tomorrow.
- John said that Mary will meet Bill tomorrow.
- Who did John say Mary will meet tomorrow?
- Who did John say that Mary will meet tomorrow?
- Who did John say will meet Bill tomorrow?
- *Who did John say that will meet Bill tomorrow?

Pronouns

- While Mary sat on the T, she read the Metro.
- While she sat on the T, Mary read the Metro.
- Mary said she read the Metro today.
- She said Mary read the Metro today.

CSC

- John drinks Coke with ice.
- What does John drink Coke with?
- What does John drink with ice?
- John drinks rum and Coke.
- *What does John drink rum and?
- *What does John drink and Coke?

The Subset problem

- Suppose the kid made the wrong choice in each case, and generalized.
  - You can either have or not have *that* to introduce an embedded clause.
  - Pronouns can refer to any NP.
  - To form a *who*-question, just move *who* to the front and drop the thing it stands for.
- As far as the kid's concerned, all of the sentences that we just saw are good sentences. (But when the kid grows up, s/he'll know otherwise)

The Subset problem

- Of course, the kid will never hear
  - *Who did John say that will meet Bill tomorrow?
  - She$_{(Mary)}$ said Mary read the Metro today.
  - What does John drink and Coke?
- But the kid will probably also never hear
  - This year's season of *Law & Order* will be the last.
  - ABC just announced a fourth season of *Sports Night*.
- Yet the kid won't have trouble seeing that these are grammatical.

The Subset problem

- So, the trick is: How can the kid get the knowledge (that adults *do* have, invariably) about what sentences are *un*grammatical, given that simply not hearing the sentence before is not evidence.
- The answer: The constraints responsible for ruling out the bad sentences are part of the presuppositions made before acquisition begins—this is UG.

So, some language knowledge is already there

- So, kids come at the task with some kind of framework into which to fit the things they learn from the input.
- Languages do differ, so kids need to learn what language they're actually hearing.
- The basic idea of the Principles & Parameters view is that the Principles are part of the human language faculty we come with, and the points of variation (the parameters) can differ from language to language.

Points of variation

- In the GB/P&P type view, kids need to determine the settings for the individual parameters:
  - Does the V move to T?
  - Can the subject be null?
  - Which of the possible binding domains does the language use?
  - What are the bounding nodes for *wh*-movement?
  - Do any *wh*-words move overtly?
  - Do all *wh*-words move overtly?

Points of variation

- In an OT view of grammar, the (inherently conflicting) constraints themselves are what UG provides, and kids must determine which ones take priority over which others?
  - Is it more important to have a subject or to minimize structure?
  - Is it more important to mark the scope of a question with a *wh*-word or to avoid the effort of movement?
  - …

Navigating grammar spaces

- Regardless of the approach, the idea is that in the space of possible grammars, there is a restricted set that correspond to possible *human* grammars.

- Kids must in some sense navigate that space until they reach the grammar that they' re hearing in the input data.

Questions

- So how do they do it?
- Where do they start?
- What kind of evidence do they need?
- How much evidence do they need?

- Research on learnability in language acquisition has concentrated on these issues.

Are we there yet?

- There are a lot of grammars to choose from, even if UG limits them to some finite number.
- Kids have to try out many different grammars to see how well they fit what they’ re hearing.
- We don’ t want to require that kids remember everything they’ ve ever heard, and sit there and test their current grammar against the whole corpus of utterances—that’ a lot to remember.

Are we there yet?

- We also want the kid, when they get to the right grammar, to stay there.

- Error-driven learning
  - Most theories of learnability rely on a kind of error-detection.
  - The kid hears something, it's not generable by their grammar, so they have to switch their hypothesis, move to a new grammar.

Plasticity

- Yet, particularly as the navigation progresses, we want them to be zeroing in on the right grammar.
- Finding an error doesn’t mean that you (as a kid) should jump to some random other grammar in the space.
- Generally, you want to move to a nearby grammar that improves your ability to generate the utterance you heard—move in baby steps.

Triggers

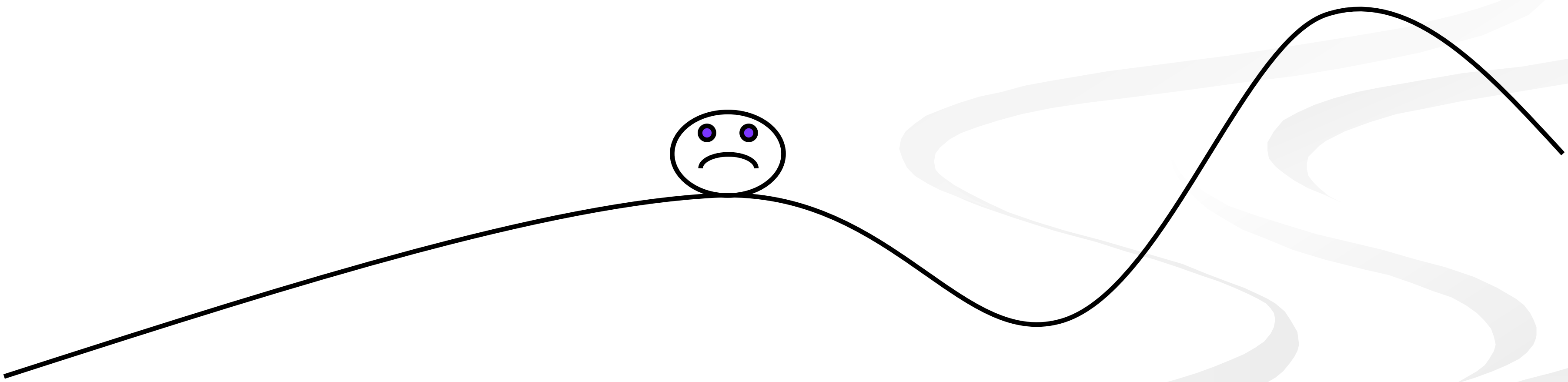
- Gibson & Wexler (1994) looked at learning word order in terms of three parameters (head, spec, V2).
- Their triggering learning algorithm says if you hear something you can't produce, try switching one parameter and see if it helps. If so, that's your new grammar. Otherwise, stick with the old grammar and hope you'll get a better example.

Local maxima

- A problem they encountered is that there are certain places in the grammar space where you end up more than one switch away from a grammar that will produce what you hear.
- This is locally as good as it gets—nothing next to it in the grammar space is better—yet if you consider the whole grammar space, there is a better fit somewhere else, you just can't get there with baby steps.

Local maxima

- This is a point where any move you make is worse, so a conservative algorithm will never get you to the best place.



Children vs. OT

- Optimality Theory is a theory of ranked constraints, some of which are in direct contradiction.
- Standard simple example from phonology:
  - Onset: syllables start with a consonant
  - NoCoda: syllables don't end with a consonant
  - Max: say what you mean (say everything in the input)
  - Dep: say only what you mean (don't add anything to the input).

ba

- If you want to say *ba* (if the word you're trying to say looks like /ba/ in the lexicon), you can say *ba* and satisfy all the constraints.
  - *Ba* starts with a consonant (√Onset)
  - *Ba* ends in a vowel (√NoCoda)
  - *Ba* has all of the input sounds (√Max)
  - *Ba* has no new sounds (√Dep)

bat

- But if the word you want to say is /bat/, there's a problem.
  - Say *bat* and you satisfy Max, Dep, and Onset, but you violate NoCoda (it ends in a consonant).
  - Say *ba* and you satisfy Dep, Onset, and NoCoda, but you violate Max (you left out the /t/).
- Languages make different choices about which wins, so kids have to decide: Is NoCoda more important than Max or vice-versa?

at

- Similarly, /at/ results in these options:
  - Say *at*, satisfying Max and Dep, at the expense of Onset and NoCoda.
  - Say *a*, satisfying Dep and NoCoda, at the expense of Onset and Max.
  - Say *ta*, satisfying Onset and NoCoda, at the expense of Max and Dep.
  - Say *tat*, satisfying Max and Onset, at the expense of Dep and NoCoda.
- Which constraint is more important in the language determines the output.

Tableau

/at/	Max	Dep	Onset	NoCoda
[at]			*	*
[a]	*		*	
[ta]	*	*		
[tat]		*		*

Max,Dep>>Ons,NoCoda

/at/	Max	Dep	Onset	NoCoda
☞ [at]			*	*
[a]	*!		*	
[ta]	*!	*		
[tat]		*!		*

Max,Ons>>Dep,NoCoda

/at/	Max	Onset	Dep	NoCoda
[at]		*!		*
[a]	*!	*		
[ta]	*!		*	
☞ [tat]			*	*

NoCoda,Ons>>Dep,Max

/at/	NoCoda	Onset	Dep	Max
[at]	*!	*		
[a]		*!		*
☞ [ta]			*	*
[tat]	*!		*	

NoCoda,Dep>>Ons,Max

/at/	NoCoda	Dep	Onset	Max
[at]	*!		*	
☞[a]			*	*
[ta]		*!		*
[tat]	*!	*		

4 constraints, 24 rankings

- Max, Dep, Ons, and NoCoda have hardly exhausted the systematic knowledge we have about phonology.
- The are *lots* more constraints, but every new constraint we add can in principle be ranked between every two constraints we had before.
  - 4 constraints, 24 = 4x3x2 = 4! rankings
  - 5 constraints, 120 = 5x4x3x2 = 5! Rankings
  - 20 constraints?
  - 30 constraints?

Wide open spaces

- The grammar space that the child has to navigate if OT is the right model of grammar is *vast*.

- Subhierarchies
  - There are some constraints which seem to be fixed in relative ranking to other constraints, cutting down the space a little bit.
  - *[l-onset] >> *[b-onset] >> *[t-onset]
  - But that still leaves a lot of options

Constraint Demotion

- Tesar & Smolensky
- If kid hears [tap] for /tap/ but would have pronounced it [ta], there's a problem—the kid needs to move to a new grammar. The constraints must be reordered so that [tap] comes out.
- Diagnosis: Kid's got NoCoda >> Max, but needs to have Max >> NoCoda

NoCoda>>Ons>>Dep>>Max

/tap/	NoCoda	Onset	Dep	Max
☛ [tap]	*!			
✖ [ta]				*
[a]		*!		**
[ap]	*!	*		*

Constraint demotion

- We demote the constraint that's killing the correct candidate to a point in the ranking below the constraint that would kill the incorrect candidate.
- We re-rank NoCoda to below Max, and solve the problem.

Ons>>Dep>>Max>>NoCoda

/tap/	Onset	Dep	Max	NoCoda
☞ [tap]				*!
[ta]			*!	
[a]	*!		**	
[ap]	*!		*	*

Repeat, repeat, repeat

- Eventually, if you do this long enough, Tesar & Smolensky argue, you'll reach the adult ranking.
  - (or something equivalent)

- Along the way, kids have intermediate grammars (possible human grammars, but not the target).

M vs. F

- Constraints come in two flavors generally, those that say "say exactly what you mean" (Faithfulness—make the output look like the input) and those that say "conform to the admissible shapes for language ouput" (Markedness).
  - Max, Dep = Faithfulness
  - Ons, NoCoda = Markedness

M >> F

- Kids' syllables are generally of the *ba* sort (and not of the *strengths* sort), suggesting that initially the Markedness constraints are outranking the Faithfulness constraints, and then re-ranking brings them more in line with the adults.
  - (The idea is that they may try to say /strengths/, but it comes out like [te] at first)

Wait a minute, this is crazy

- One thing that learnability research of this sort tends to leave mostly undiscussed is how the kid comes to hypothesize /strength/ as the underlying form in the first place. After all, if it sounds like [te], why not suppose it is /te/?

- Be that as it may. A clear place where more work is needed (recent work by Tesar and Smolensky separately make some attempts).
- For now, we assume the kid knows the word.

Optionality

- Another issue arises if the grammar allows true optionality—this seems possible in phonology at least (Arto Anttila and Bill Reynolds have done a lot of work in this domain, with several others since).
- If the adult allows two different surface realizations of the same input, the constraint demotion algorithm is going to eternally flip-flop between rankings.
- The constraint demotion algorithm assumes *strict* ranking, one output per input.

One approach: multiple grammars

- One way to look at this is as taking people to have multiple grammars.
- Plausible; registers, for example, or multilingualism, or dialects.
- (Of course, how would the kid decide which grammar to demote constraints in?)

Multiple grammars vs. Darwin

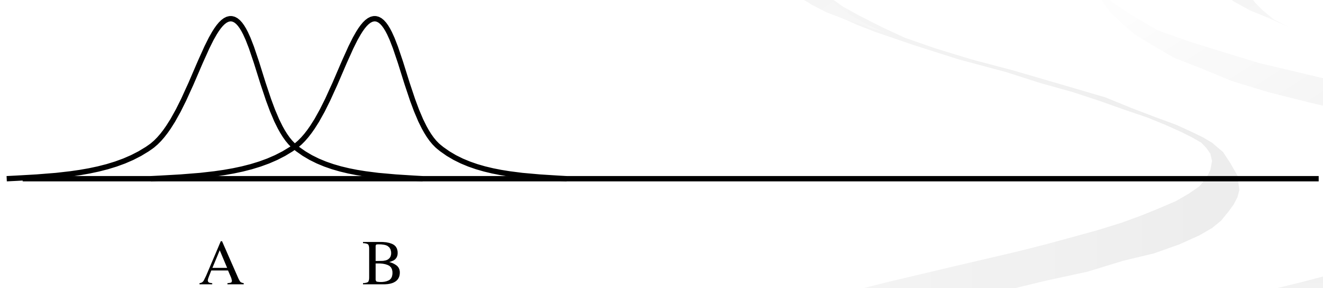
- One approach (e.g., Charles Yang) says that kids maintain several grammars and test them all at once, the ones which do a worse job being demoted to less frequent use.
- You have several grammars you choose between, each with a probability of choosing it, and if a grammar never seems to be predicting the right forms, it will become less and less likely to be used in the future. Acquisition in the end lands on one (or a couple, if there is free variation).

Another approach: Boersma

- Paul Boersma (and more recently Bruce Hayes) have been championing a more statistical approach.
- Constraints have a position in a ranking (associated with a number), but when you do an evaluation, the number it's actually evaluated at is governed by a normal ("bell curve") distribution—there's noise in the system.

Boersma

- This is a grammar where A usually >> B, but sometimes B>>A.
- You can compute exactly how often B>>A.
- This can get adult variation, and provides a way for kids to learn the rankings too.



GLA

- Boersma's gradual learning algorithm is sensitive to the frequencies in the input, and will move the centers of the constraints up or down in order to try to match the frequency of A>>B with respect to B>>A. Works similarly to constraint demotion otherwise.

- Advantages:
  - Gets statistical properties of the input
  - More resilient in the face of ungrammatical noise.
  - Fancy simulation program (Praat).

NoCoda>>Ons>>Dep>>Max

/tap/	NoCoda	Onset	Dep	Max
☛ [tap]	*!			
✖ [ta]				*
[a]		*!		**
[ap]	*!	*		*

Praat

- Praat is Boersma's super-program, that either already does or in the future will do everything a phonologist/phonetician could ever want.
- For the moment, our concern is that it can do learning simulations on datasets to try to reach an adult grammar.

Legendre et al.

- The system we discussed for French last time (*F, *F2, ParseT, ParseA) is a slightly different system again (based on work by Reynolds, related to work by Anttila).
- Under that system, learning proceeds by starting with M>>F, and promoting Faithfulness constraints, but not in an all-or-nothing manner—rather, constraints are promoted such that they span a range of the constraint hierarchy. This also yields percentages. It's a form of multiple grammars, but very *related* grammars.

Praat

- We're going to try out Praat's learning capabilities.
- Praat comes with some built-in grammars, for example the NoCoda grammar.
- Each Praat grammar lists
  - The constraints and their position
  - Any "fixed rankings" (e.g. $*F^2 >> *F$).
  - The possible inputs
  - The candidate outputs
  - The constraints each candidate violates

Seeing what Praat grammars can do

- To learn a grammar (that is, given the constraints in the grammar, but set at some random place in the hierarchy initially), the learner needs data.
- You can write **pair distributions**, which say how frequently /form/ comes out as [form] and as [for], for example.
- You can use these distributions to create a corpus (input strings) which the learner will process to try to set the ranking.

Finnish plurals

- Anttila's data.