So you want to model language acquisition

What does it mean to model something?
So you want to model language acquisition

What does it mean to model something?

It's a scientific technique, like running an experiment. So saying "I want to model $thing" is just like saying "I want to run an experiment about $thing." Basically, it's a fine plan, but the important question is why you're doing it. That is, what question are you trying to answer?

Once you know what question you're trying to answer, you can design the right test of it — whether that's an experiment or a model or something else entirely.

So you want to model language acquisition

"Computational modeling can be used to examine a variety of questions about the language acquisition process, because a model is meant to be a simulation of the relevant parts of a child's acquisition mechanism. In a model, we can precisely manipulate some part of the mechanism and see the results on acquisition....Importantly, some manipulations we can do within a model are difficult to do with children...modeling data are thus particularly useful because of the difficulty of getting those same data through experimental means." - Pearl 2010

Model-y questions

Okay, so what kind of questions do we use models to answer?

I typically see models used in language acquisition to answer the question of how. How exactly does the acquisition process work for a particular thing (like syntactic categorization, word learning, syntactic islands, etc.)?

Some specific questions of how:
• How can children learn certain representations? What representations are easy to learn vs. hard to learn?
• How much impact do different types of input data have on the eventual representation learned? What about different expectations about what’s salient or relevant in the data?

Model-y questions

Okay, so what kind of questions do we use models to answer?

A model that answers these kinds of "how" questions is likely to be an informative model — it tells us something we didn’t know before and didn’t necessarily have another way to find out.
Informative models

How do we make sure our model is informative?

An informative model tells us something about how humans do language acquisition. So, we better have some concrete ideas about the different pieces of the language acquisition process. That way, we can make sure our model captures these important pieces in a realistic way.

One idea about how acquisition works

An informative model captures these important pieces in a realistic way. In particular, it tries to empirically ground these pieces by drawing on available data from formal, experimental, and computational research.

Characterizing the acquisition task

Initial state: What does the child start with? What knowledge, abilities, and learning biases does the child already have?

This allows us, as computational modelers, to define the acquisition task precisely enough to come up with ways children might solve it. The learning strategies we come up with can also be characterized in terms of these acquisition task pieces.
### Initial state: What does the child start with? What knowledge, abilities, and learning biases does the child already have?

**Initial knowledge**
- ex: syntactic categories exist and can be identified
- ex: phrase structure exists and can be identified
- ex: participant roles can be identified

**Initial abilities & biases**
- ex: frequency information can be tracked
- ex: distributional information can be leveraged

---

### Data intake: How does the modeled child perceive the input (=perceptual intake)?

**What part of the perceived data is used for acquisition (=acquisitional intake)?** This is the relevant data for acquisition.

---

Ex: A strategy that depends on the frequency of certain syntactic structures would need the child to know about that syntactic structure via the developing grammar and/or Universal Grammar, recognize it in the input via the developing language processing abilities, and be able to track the frequency of that structure.
Characterizing the acquisition task

**Initial state**: What does the child start with?

**Data intake**: How does the modeled child perceive the input (=perceptual intake)? What part of the perceived data is used for acquisition (=acquisitional intake)? This is the relevant data for acquisition.

- ex: all wh-utterances for learning about wh-dependencies
- ex: all pronoun data when learning about anaphoric one
- ex: syntactic and conceptual data for learning syntactic knowledge that links with conceptual knowledge

[defined by knowledge & biases/capabilities in the initial state]

Characterizing the acquisition task

**Initial state**: What does the child start with?

**Data intake**: The relevant data for acquisition.

**Learning period**: How long does the child have to learn?

- how long children have to reach the target knowledge state
  - ex: 3 years, ~1,000,000 data points
  - ex: 4 months, ~36,500 data points

This is when inference happens, i.e., when updates are made to the developing grammar.
Characterizing the acquisition task

**Initial state:** What does the child start with?

**Data intake:** The relevant data for acquisition.

**Learning period:** How long does the child have to learn?

This is also when iteration happens, i.e., when the developing grammar affects subsequent data encoding.

Lidz & Gagliardi 2015

---

Characterizing the acquisition task

**Initial state:** What does the child start with?

**Data intake:** The relevant data for acquisition.

**Learning period:** How long does the child have to learn?

**Target state:** What does successful acquisition look like? What knowledge is the child trying to attain (often assessed in terms of observable behavior)?

Knowledge

ex: *Where did Jack think the necklace from __ was too expensive? 🙄*
ex: *Where did Jack buy a necklace from __ for Lily for her birthday? 😊*

Lidz & Gagliardi 2015

---

Characterizing the acquisition task

**Initial state:** What does the child start with?

**Data intake:** The relevant data for acquisition.

**Learning period:** How long does the child have to learn?

**Target state:** What does successful acquisition look like? What knowledge is the child trying to attain (often assessed in terms of observable behavior)?

Behavior

looking time preferences  2-score rating

Lidz & Gagliardi 2015

---
Characterizing the acquisition task

**Initial state:** What does the child start with?

**Data intake:** The relevant data for acquisition.

**Learning period:** How long does the child have to learn?

**Target state:** What does successful acquisition look like?

Once we have all these pieces specified, we should be able to implement an informative model of the learning process.

---

Back to the process of modeling

So if your question is a question about how the language acquisition process works, a computational model might be the right tool to use.

Let’s say you have a learning strategy you want to test out. There’s still another important decision to make.

**What level of model do you want to build?**

---

Levels of representation (Marr 1982)

- "Rough winds do shake the darling boughs of tree..."
- "The muscles are engaged in propulsion..."
- "You know, I was thinking about how..."
- "Forecast calls for highs in the 70s..."
"But the important point is that if the notion of different types of understanding is taken very seriously, it allows the study of the information-processing basis of perception to be made rigorous. It becomes possible, by separating explanations into different levels, to make explicit statements about what is being computed and why..."

Our goal: Substitute “language acquisition” for “perception”

The three levels:

Computational
What is the goal of the computation?

Algorithmic
What is the representation for the input and output, and what is the algorithm for the transformation?

Implementational
How can the representation and algorithm be realized physically?

An example with the cash register

Computational
What does this device do?
Arithmetic (ex: addition).
Addition: Mapping a pair of numbers to another number.

Algorithmic

\[(3,4) \rightarrow 7\] [often written \((3+4=7)\)]

Properties:
\[(3+4) = (4+3)\] [commutative]
\[(3+4)+5 = 3+(4+5)\] [associative]
\[(3+0) = 3\] [identity element]
\[(3+\cdot3) = 0\] [inverse element]

True no matter how numbers are represented: this is what is being computed
The three levels: An example with the cash register

Computational
What does this device do?
Arithmetic (ex: addition).
Addition: Mapping a pair of numbers to another number.

Algorithmic
What is the input, output, and method of transformation?
Input: arabic numerals (0,1,2,3,4...)  
Output: arabic numerals (0,1,2,3,4...)  
Method of transformation: rules of addition, where least significant digits are added first and sums over 9 have their next digit carried over to the next column

\[
\begin{array}{c}
99 \\
+ 5 \\
\hline
14
\end{array}
\]
The three levels:  
An example with the cash register

Computational
What does this device do?
Arithmetic (ex: addition).
Addition: Mapping a pair of numbers to another number.

Algorithmic
What is the input, output, and method of transformation?
Input: arabic numerals (0,1,2,3,4...)
Output: arabic numerals (0,1,2,3,4...)
Method of transformation: rules of addition

Implementational
How can the representation and algorithm be realized physically?
A series of electrical and mechanical components inside the cash register.

The three levels:  
An example with a sandwich

Computational
What is the goal?
Make a peanutbutter and jelly sandwich.

Algorithmic
What is the input, output, and method of transformation?
Input: ingredients (peanutbutter, jelly, bread slices), tools (knife, spoon)
Output: completed, edible sandwich with the required properties
Method: Use the spoon to put jelly on one slice & spread it with the knife. Use the spoon to put peanutbutter on the other slice & spread it with the knife. Put the two slices of bread together, with the spread sides facing each other. Cut the joined slices in half with the knife.

Implementational
How can the representation and algorithm be realized physically?
Directing your younger sibling to follow the steps above to make you a sandwich.
Mapping the framework

Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

Computational Problem: Divide sounds into contrastive categories

Mapping the framework

Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

Computational Problem: Divide spoken speech into words

Mapping the framework

Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

Computational Problem: Map word forms to speaker-invariant forms

Mapping the framework

Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

Computational Problem: Identify the concept a word is associated with (Word-meaning mapping)

“I love my daxes.”

Dax = that specific toy, teddy bear, stuffed animal, toy, object, …?
Computational Problem: Identify what a speaker means by using a specific expression.

“I love some of my daxes.”

Does the speaker not love all of them?

Computational Problem: Identify syntactic categories

“This is a DAX.”

DAX = noun

Mapping the framework

Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

Computational Problem: Identify the rules of word order for sentences. (Syntax)

Mapping the framework

Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

A very basic question for an acquisition model: Is it possible for the child with a specific initial state to use the acquisitional intake to achieve the target knowledge/behavior?

Is this the right conceptualization of the acquisition task?
Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

A very basic question for an acquisition model: Is it possible for the child with a specific initial state to use the acquisitional intake to achieve the target knowledge/behavior?

Is this the right conceptualization of the acquisition task?


This kind of analysis is very helpful for determining if this implementation of the acquisition task is the right one. In particular, if children are sensitive to this information in the perceptual intake, is that enough to yield the target knowledge/behavior? Are these useful learning assumptions for children to have to create the acquisitional intake? Are these useful representations?

This is typically implemented as an ideal learner model, which isn’t concerned with the cognitive limitations and incremental learning restrictions children have.

(That is, useful for children is different from useable by children in real life.)

Practical note: Doing a computational analysis is often a really good idea to make sure we’ve got the right conceptualization of the acquisition task (see Pearl 2011 for the trouble you can get into when you don’t do this first).
Goal: Understanding the “how” of language acquisition

First, we need a computational-level description of the learning problem.

Why do none of these learning strategies work?

Because they’re solving the wrong acquisition task...oops.

Input/Intake = sounds, syllables, words, phrases, ...
Output = sound categories, word forms, words with meanings, words with affixes, syntactic categories, phrases, sentences, interpretations...
Method = strategies based on the information in the initial state ...

Algorithmic Level:
Input/intake: Theoretical linguistics, experimental studies, and corpus analysis can tell us what the input is likely to be for a given task and what the intake is likely to be for children at that stage of development.

Example problem: speech segmentation

húwzəfrējdəvədib baggageəwlf

who’s afraid of the big bad wolf

Output: Theoretical linguistics and experimental studies can tell us what the output should look like by observing adult and child knowledge of various linguistic phenomena, as indicated by their behavior.
Algorithmic Level:

**Method:** Learning theories and experimental studies can tell us what are the components in psychologically plausible learning strategies.

**Example problem: speech segmentation**

![Image of a wolf]

húwzafiedođóbígbaédlolf

who’s afraid of the big bad wolf

**Method**

húwz afiéj dá big bád wálf

output

looking time preferences

---

**Mapping the framework**

**Goal:** Understanding the “how” of language acquisition (algorithmic-level)

Another basic question for an acquisition model: Is it possible for the child to use the acquisitional intake to achieve the target knowledge/behavior in the amount of time children typically get to do it, given the incremental nature of learning and children’s cognitive constraints? What algorithm will work in practice?

Is it possible for children to use this strategy? That is, once we know it’s useful for children, it’s important to make sure it’s also usable by children.
So if your question is a question about how the language acquisition process works, a computational model might be the right tool to use.

Let’s say you have a learning strategy you want to test out. There’s still another important decision to make.

What level of model do you want to build?

Back to the process of modeling

Algorithmic: Is it possible for the child with a specific initial state to use the acquisitional intake to achieve the target knowledge/behavior in the amount of time children typically get to do it, given the incremental nature of learning and children's cognitive constraints?

Computational: Is it possible for the child with a specific initial state to use the acquisitional intake to achieve the target knowledge/behavior?
So let’s say you’ve figured out what level of model is appropriate to build in order to test the learning strategy you have in mind.

Now what?

Time to actually build it!

General modeling process

(1) Decide what kind of learner the model represents
(ex: normally developing 6- to 8-month-old child learning first language)

(2) Decide what data the child learns from (ex: Pearl-Brent corpus from CHILDES)

Empirically grounding the input

The CHILDES database has a wealth of child-directed speech transcripts and videos from a number of different languages. This can help us figure out what children’s input looks like.

Video/audio recordings of spontaneous speech samples, along with transcriptions and some structural annotation. Extremely valuable resource to the language acquisition community.

http://childes.psy.cmu.edu
Empirically grounding the input

http://childes.psy.cmu.edu

“In terms of its impact on the field of language development, CHILDES is a game-changer. It allows researchers with limited resources to test hypotheses using an extremely rich data set. It allows for comparison across many different languages, which makes it possible to look for universal cross-linguistic patterns in language development….because the transcripts also include language by the adults that the children are interacting with, it also allows researchers to test detailed quantitative predictions about the relationships between a child’s input and her language production.” — Sedivy 2014, p.224

General modeling process

(1) Decide what kind of learner the model represents
(ex: normally developing 6- to 8-month-old child learning first language)

(2) Decide what data the child learns from (ex: Pearl-Brent corpus from CHILDES) and how the child perceives that data (ex: divide speech stream into syllables)

Lidz & Gagliardi 2015

General modeling process

(1) Decide what kind of learner the model represents
(ex: normally developing 6- to 8-month-old child learning first language)

(2) Decide what data the child learns from (ex: Pearl-Brent corpus from CHILDES) and how the child perceives that data (ex: divide speech stream into syllables)

(3) Decide what hypotheses the child has (ex: what the words are) and what information is being tracked in the input (ex: transitional probability between syllables, stress on syllables)
General modeling process

1. Decide what kind of learner the model represents (ex: normally developing 6- to 8-month-old child learning first language)

2. Decide what data the child learns from (ex: Pearl-Brent corpus from CHILDES) and how the child perceives that data (ex: divide speech stream into syllables)

3. Decide what hypotheses the child has (ex: what the words are) and what information is being tracked in the input (ex: transitional probability between syllables, stress on syllables)

4. Decide how belief in different hypotheses is updated (ex: based on transitional probability between syllables)

5. Decide what the measure of success is (ex: developing knowledge, proto-lexicon of word forms, behavior indicating developed knowledge, recognizing useful units (such as words) in a fluent speech stream, as indicated by looking time behavior)

Cognitively plausible perception & inference

Many models will try to make cognitively plausible assumptions about how the child is representing and processing input data:
- Processing data points as they are encountered
- Assuming children have memory limitations (ex: memory of data points may decay over time)

This makes the model match what we know about children better — therefore, the model is more likely to tell us something real about children.
General modeling process

(6) Implement the model in a programming language of choice

This includes figuring out how the input needs to be represented for the code, how the steps will be implemented, what data structures will be used, what kind of output will be generated, and what kind of format that output will be in.

General modeling process

(7) See how well the model did, wrt. the measure of success

ex: developing knowledge
- Proto-lexicon of word forms

ex: behavior indicating developed knowledge
- Recognizing useful units (such as words) in a fluent speech stream, as indicated by looking time behavior

From this, we can determine how well the model did — and more importantly, how well the strategy implemented concretely in the model did.

General modeling process

(8) Interpret the results for other people who aren’t you

“The modeled child has the same developing knowledge as we think 8-month-olds do. This strategy can work!”

“The modeled child can reproduce the behavior we see in 8-month-olds. This strategy could be what they’re using to generate that behavior!”

This is incredibly important — otherwise, no one knows what to make of your results (and whether or not they should care).

Let’s take a break for a few minutes
Today's Plan

I. So you want to model language acquisition

II. Modeling case study: Defining the pieces

III. Modeling case study: Implementation & Interpretation

Syntactic islands: Dependencies that aren't okay

A property of language: Long-distance dependencies

Dependencies can exist between two non-adjacent items, and these do not appear to be constrained by length (Chomsky 1965, Ross 1967).

What does Jack think ___?
What does Jack think that Lily said ___?
What does Jack think that Lily said that Sarah heard ___?
What does Jack think that Lily said that Sarah heard that Jareth stole ___?

Syntactic islands: Dependencies that aren't okay

If the gap position appears inside certain structures (called “syntactic islands” by Ross (1967)), the dependency seems to be ungrammatical.

*What did you make [the claim that Jack bought ___]?
*What do you think [the joke about ___] offended Jack?
*What do you wonder [whether Jack bought ___]?
*What do you worry [if Jack buys ___]?
*What did you meet [the scientist who invented ___]?
*What did [that Jack wrote ___] offend the editor?
*What did Jack buy [a book and ___]?
*Which did Jack borrow [___ book]?
De Villiers 1995: comprehension task with 3- to 6-year-olds

“Once there was a boy who loved climbing trees in the forest. One afternoon he slipped and fell to the ground. He picked himself up and went home. That night when he had a bath, he saw a big bruise on his arm. He said to his Dad, ‘I must have hurt myself when I fell this afternoon.’”

Children allow both these structures (and their interpretations), too.
Children’s knowledge of wh-island constraints
De Villiers 1995: comprehension task with 3- to 6-year-olds
“Once there was a boy who loved climbing trees in the forest. One afternoon he slipped and fell to the ground. He picked himself up and went home. That night when he had a bath, he saw a big bruise on his arm. He said to his Dad, ‘I must have hurt myself when I fell this afternoon.’”

When did the boy say how he fell?

---

How could children learn this and other syntactic islands?
*What did you make [the claim that Jack bought __]?
*What do you think [the joke about __] offended Jack?
*What do you wonder [whether Jack bought __]?
*What do you worry [if Jack buys __]?
*What did you meet [the scientist who invented __]?
*What did [that Jack wrote __] offend the editor?
*What did Jack buy [a book and __]?
*Which did Jack borrow [__ book]?

---

Children allow only the top structure (and its interpretation), too.
To model?

We want to understand how the acquisition of constraints on dependencies could work. This concerns the mechanism of acquisition, which will involve a particular strategy.

The strategy is something that can be modeled.

Let’s build an algorithmic model, which will model the process unfolding in time.

Okay, so what empirical data are there?

Remember: We want to empirically ground our modeled child as much as possible, so it’ll end up (hopefully) being informative about how real children learn syntactic islands.

One point: Children’s input doesn’t look so helpful

Pearl & Sprouse 2013: Analysis of child-directed speech (Brown-Adam, Brown-Eve, Suppes, Valian) from CHILDES:

<table>
<thead>
<tr>
<th>Items Encountered</th>
<th>Items in English</th>
<th>Items not in English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wh-questions in input</td>
<td>What did you see?</td>
<td>What happened?</td>
</tr>
</tbody>
</table>

Most of it is fairly simple dependencies — and importantly, dependencies that are grammatical. How could children form the appropriate generalizations about what isn’t allowed?

Syntactic islands: How to generalize?

<table>
<thead>
<tr>
<th>Items Encountered</th>
<th>Items in English</th>
<th>Items not in English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wh-questions in input</td>
<td>What did you see?</td>
<td>What happened?</td>
</tr>
</tbody>
</table>
Syntactic islands: How to generalize?

Grammatical wh-questions
What did you see?
What happened?
Who did Jack think that Lily saw?
What did Jack think happened to Lily in the park?

Syntactic islands: How to generalize?

Ungrammatical wh-questions: Syntactic islands
*What did you make (the claim that Jack bought ___)?
*What do you think (the joke about ___) offended Jack?
*What do you wonder (whether Jack bought ___)?
*What do you worry (if Jack buys ___)?

Another point: Adult behavior
Empirical data: Adult knowledge as measured by acceptability judgment behavior for some islands from Sprouse et al. (2012). This is the eventual target of acquisition.

Adult behavior: acquisition target
Adult knowledge as measured by acceptability judgment behavior
Sprouse et al. (2012) collected magnitude estimation judgments for four different islands, using a factorial definition that controlled for two salient properties of island-crossing dependencies:
- length of dependency (matrix vs. embedded)
- presence of an island structure (non-island vs. island)
Adult knowledge as measured by acceptability judgment behavior

Sprouse et al. (2012) collected magnitude estimation judgments for four different islands, using a factorial definition that controlled for two salient properties of island-crossing dependencies:
- length of dependency
  (matrix vs. embedded)
- presence of an island structure
  (non-island vs. island)

**Complex NP islands**

<table>
<thead>
<tr>
<th>Question</th>
<th>Type</th>
<th>Island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who ___ claimed that Lily forgot the necklace?</td>
<td>matrix</td>
<td>non-island</td>
</tr>
<tr>
<td>What did the teacher claim that Lily forgot ___?</td>
<td>embedded</td>
<td>non-island</td>
</tr>
<tr>
<td>Who ___ made the claim that Lily forgot the necklace?</td>
<td>matrix</td>
<td>island</td>
</tr>
<tr>
<td>*What did the teacher make the claim that Lily forgot ___?</td>
<td>embedded</td>
<td>island</td>
</tr>
</tbody>
</table>

**Subject islands**

<table>
<thead>
<tr>
<th>Question</th>
<th>Type</th>
<th>Island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who ___ thinks the necklace is expensive?</td>
<td>matrix</td>
<td>non-island</td>
</tr>
<tr>
<td>What does Jack think ___ is expensive?</td>
<td>embedded</td>
<td>non-island</td>
</tr>
<tr>
<td>Who ___ thinks the necklace for Lily is expensive?</td>
<td>matrix</td>
<td>island</td>
</tr>
<tr>
<td>*Who does Jack think the necklace for ___ is expensive?</td>
<td>embedded</td>
<td>island</td>
</tr>
</tbody>
</table>

**Whether islands**

<table>
<thead>
<tr>
<th>Question</th>
<th>Type</th>
<th>Island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who ___ thinks that Jack stole the necklace?</td>
<td>matrix</td>
<td>non-island</td>
</tr>
<tr>
<td>What does the teacher think that Jack stole ___?</td>
<td>embedded</td>
<td>non-island</td>
</tr>
<tr>
<td>Who ___ wonders whether Jack stole the necklace?</td>
<td>matrix</td>
<td>island</td>
</tr>
<tr>
<td>*What does the teacher wonder whether Jack stole ___?</td>
<td>embedded</td>
<td>island</td>
</tr>
</tbody>
</table>

**Adjunct islands**

<table>
<thead>
<tr>
<th>Question</th>
<th>Type</th>
<th>Island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who ___ thinks that Lily forgot the necklace?</td>
<td>matrix</td>
<td>non-island</td>
</tr>
<tr>
<td>What does the teacher think that Lily forgot ___?</td>
<td>embedded</td>
<td>non-island</td>
</tr>
<tr>
<td>Who ___ worries if Lily forgot the necklace?</td>
<td>matrix</td>
<td>island</td>
</tr>
<tr>
<td>*What does the teacher worry if Lily forgot ___?</td>
<td>embedded</td>
<td>island</td>
</tr>
</tbody>
</table>
Adult behavior: acquisition target

Adult knowledge as measured by acceptability judgment behavior

Syntactic island = superadditive interaction of the two factors (additional unacceptability that arises when the two factors are combined, above and beyond the independent contribution of each factor).

Superadditivity present for all islands tested = Knowledge that dependencies cannot cross these island structures is part of adult knowledge about syntactic islands.

Importance for acquisition: This is one kind of target behavior that we’d like a modeled child to produce.

Children’s input really doesn’t look so helpful

Data from five corpora of child-directed speech (Brown-Adam, Brown-Eve, Brown-Sarah, Suppes, Valian) from CHILDES (MacWhinney 2000): speech to 25 children between the ages of one and five years old.

Total words: 813,036
Utterances containing a wh-dependency: 31,247

Sprouse et al. (2012) stimuli types:

<table>
<thead>
<tr>
<th></th>
<th>MATRIX + NON-ISLAND</th>
<th>EMBEDDED + NON-ISLAND</th>
<th>MATRIX + ISLAND</th>
<th>EMBEDDED + ISLAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex NP</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Whether</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adjunct</td>
<td>7</td>
<td>295</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

Pearl & Sprouse submitted
These kinds of utterances are fairly rare in general - the most frequent appears about 0.9% of the time (295 of 31,247.)

Sprouse et al. (2012) stimuli types (out of 31,247):

<table>
<thead>
<tr>
<th>Type</th>
<th>Matrix + non-island</th>
<th>Embedded + non-island</th>
<th>Matrix + island</th>
<th>Embedded + island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex NP</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Whether</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adjunct</td>
<td>7</td>
<td>295</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

Being grammatical doesn’t necessarily mean an utterance will appear in the input at all.

Sprouse et al. (2012) stimuli types (out of 31,247):

<table>
<thead>
<tr>
<th>Type</th>
<th>Matrix + non-island</th>
<th>Embedded + non-island</th>
<th>Matrix + island</th>
<th>Embedded + island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex NP</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Whether</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adjunct</td>
<td>7</td>
<td>295</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

Unless the child is sensitive to very small frequencies, it’s difficult to tell the difference between grammatical and ungrammatical dependencies sometimes...

Sprouse et al. (2012) stimuli types (out of 31,247):

<table>
<thead>
<tr>
<th>Type</th>
<th>Matrix + non-island</th>
<th>Embedded + non-island</th>
<th>Matrix + island</th>
<th>Embedded + island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex NP</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Whether</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adjunct</td>
<td>7</td>
<td>295</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

...and impossible to tell no matter what the rest of the time. This looks like an induction problem for the language learner if we’re looking for direct evidence in the input.

Sprouse et al. (2012) stimuli types (out of 31,247):

<table>
<thead>
<tr>
<th>Type</th>
<th>Matrix + non-island</th>
<th>Embedded + non-island</th>
<th>Matrix + island</th>
<th>Embedded + island</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex NP</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Whether</td>
<td>7</td>
<td>295</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adjunct</td>
<td>7</td>
<td>295</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>
Syntactic islands: Specific islands we’ll focus on

- What did you make [the claim that Jack bought ___]?  
- What do you think [the joke about __] offended Jack?  
- What do you wonder [whether Jack bought ___]?  
- What do you worry [if Jack buys ___]?  
- What did you meet [the scientist who invented ___]?  
- What did [that Jack wrote ___] offend the editor?  
- What did Jack buy [a book and ___]?  
- Which did Jack borrow [__ book]?

Great — this seems to be a hard (and therefore interesting) problem. So what kind of learning strategy should we try? Are there existing theories of linguistic representations and learning strategies based on those representations?

Syntactic islands: Representations


1. A dependency cannot cross two or more bounding nodes.

   \( \text{Wh} \rightarrow [\text{BN}_1 \rightarrow [\text{BN}_2 \rightarrow \_]] \)

   Bounding nodes are language-specific
   (CP, IP, and/or NP – must learn which ones are relevant for language)

2. A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

   \( \text{Wh} \rightarrow [\text{CN}_1 \rightarrow [\text{CN}_2 \rightarrow [\text{CN}_3 \rightarrow [\text{CN}_4 \rightarrow [\text{CN}_5 \rightarrow \_]]]]] \)

   Container node: phrase structure node that contains dependency
   \( [\_ \text{Wh} \_ \text{do} \_ \text{you} \_ \text{like} \_ \text{you} \_ \text{in \ this \ picture?]}] \)

Syntactic islands: Representations

Subjacency-ish (Pearl & Sprouse 2013a, 2013b, 2015)

1. A dependency cannot cross two or more bounding nodes.

   \( \text{Wh} \rightarrow [\text{BN}_1 \rightarrow [\text{BN}_2 \rightarrow \_]] \)

2. A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

   \( \text{Wh} \rightarrow [\text{CN}_1 \rightarrow [\text{CN}_2 \rightarrow [\text{CN}_3 \rightarrow [\text{CN}_4 \rightarrow [\text{CN}_5 \rightarrow \_]]]]] \)

How to describe this dependency: What phrases is the gap inside but the wh-word isn’t inside?

Container nodes sequences

- How to describe this dependency: What phrases is the gap inside but the wh-word isn’t inside?
How to describe this dependency:
What phrases is the gap inside but the wh-word isn’t inside?

What did you see __?
= What did [ IP you [ VP see __ ]]?
= IP-VP

What __ happened?
= What [ IP __ happened ]?
= IP

What did she want to do __?
= What did [ IP she [ VP want [ to [ VP do __ ] ] ] ]?
= IP-VP-IP-VP

What did she read from __?
= What did [ IP she [ VP read [ from [ VP do __ ] ] ] ]?
= IP-VP-PP
Syntactic islands: Representations

(1) A dependency cannot cross two or more bounding nodes.

Subjacency-ish (Pearl & Sprouse 2013a, 2013b, 2015)
(2) A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

Low probability regions are language-specific (defined by sequences of container nodes that must be learned)

Different: Amount of language-specific knowledge built in just for islands

In common: Both rely on local structure anomalies (at some level)

Pearl & Sprouse: Focused on evaluating this one

Pearl & Sprouse 2013a, 2013b, 2015
One strategy for some of the islands

Pearl & Sprouse (2013) strategy using the Subjacency-ish representation: Learn what you can from the dependencies you do actually observe in the data (= container node sequence probabilities). Apply that knowledge to make a judgment about the dependencies you haven’t seen before, like these syntactic islands.

Subjacency-ish strategy in more detail

Strategy:
(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.

IP-VP = begin-IP-VP
IP-VP-end
IP-VP-IP-VP
IP-VP-PP
begin-IP-VP
begin-IP-VP
begin-IP-VP
IP-VP-VP
IP-VP-PP
IP-VP-PP-end
IP-VP-end

Subjacency-ish strategy in more detail

Strategy:
(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.

IP-VP = begin-IP-VP
IP-VP-end
IP-VP-IP-VP
IP-VP-PP
begin-IP-VP
begin-IP-VP
begin-IP-VP
IP-VP-VP
IP-VP-PP
IP-VP-PP-end
IP-VP-end

Note that some of these trigrams already appear in multiple dependencies that commonly occur in children’s input.
Subjacency-ish strategy in more detail

Strategy:
(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
(3) Use trigram frequency to calculate the probability of that trigram occurring in a dependency.

When you see a new dependency, break it down into its trigrams and then calculate its probability, based on the trigram probabilities.

```
What does Jack want to do that for__?
= What does [return to] want [return to to] do that [return to]
= IP-VP/IP-VP-PP
= begin-IP-VP
= 0.38*0.03*0.03*0.01*0.01 = 0.00000034
```

Subjacency-ish strategy in more detail

Strategy:
(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
(3) Use trigram frequency to calculate the probability of that trigram occurring in a dependency.
(4) When you see a new dependency, break it down into its trigrams and then calculate its probability, based on the trigram probabilities.

```
What do you think that the joke about__ offended Jack?
= What do [return to] think [return to that [return to the joke [return to about]]] offended Jack?
= IP-VP/IP-VP-CP-NP-PP
= begin-IP-VP
= 0.86*0.01*0.001*0.00*0.02 = 0.00
```

Subject island
**Subjacency-ish strategy in more detail**

**Strategy:**
1. Pay attention to the structure of dependencies.
2. Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
3. Use trigram frequency to calculate the probability of that trigram occurring in a dependency.
4. When you see a new dependency, break it down into its trigrams and then calculate its probability, based on the trigram probabilities.
5. Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

\[
\begin{align*}
p(\text{IP-VP}) &= 0.14 \\
p(\text{IP-VP-VP-PP}) &= 0.000000034 \\
p(\text{IP-VP-CP-IP-NP-PP}) &= 0.00
\end{align*}
\]

**Let’s model!**

Let’s try to pin down all the pieces we need for this strategy based on the Subjacency-ish representation: initial state, data intake, learning period, and target state.

**The Subjacency-ish strategy: Initial state**

Subjacency-ish representation:
A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

Initial state: 
(i) Dependencies defined over container node structure 
(ii) Container nodes recognized 
(iii) Track probability of short container node sequences (trigrams)

**The Subjacency-ish strategy: Initial state**

Perceive wh-dependencies as sequences of container nodes, identifying container node trigrams.

Who did she think that the gift was from?
Perceive wh-dependencies as sequences of container nodes, identifying container node trigrams.

Who did she think that the gift was from?

\[ \text{IP} \ 	ext{VP} \ 	ext{CP} \ 	ext{IP} \ 	ext{VP} \ 	ext{PP} \]

\[ \text{begin-IP-VP CP IP VP PP-end} = \text{begin-IP-VP} \]

\[ \text{IP-VP CP VP-CP CP-IP-VP VP-PP-end} \]

\[ \text{begin-IP-VP CP IP VP PP-end} = \text{begin-IP-VP} \]

\[ \text{IP-VP CP VP-CP CP-IP-VP VP-PP-end} \]

Subjacency-ish strategy: Initial state

The Subjacency-ish strategy: Initial state

A child learns about the frequency of container node trigrams...

Who did she think that the gift was from __?

\[ \text{IP} \ 	ext{VP} \ 	ext{CP} \ 	ext{IP} \ 	ext{VP} \ 	ext{PP} \]

\[ \text{begin-IP-VP CP IP VP PP-end} = \text{begin-IP-VP} \]

\[ \text{IP-VP CP VP-CP CP-IP-VP VP-PP-end} \]

...and at the end of the learning period has a sense of the probability of any given container node trigram, based on its relative frequency.

Subjacency-ish: Developing knowledge

Any wh-dependency can then have a probability, based on the product of the smoothed probabilities of its trigrams.

\[ \text{Probability(begin-IP-VP CP IP VP PP-end)} = \text{p(begin-IP-VP)} \]

\[ \text{p(IP-VP CP)} \]

\[ \text{p(VP-CP IP)} \]

\[ \text{p(IP-VP PP)} \]

\[ \text{p(VP PP-end)} \]

\[ \tau \text{p(trigram)} \]
This allows the modeled learner to generate judgments about the grammaticality of any dependency.

Higher probability dependencies are more grammatical while lower probability dependencies are less grammatical.
The Subjacency-ish strategy: Data intake

A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

```
Wh — (CN1 — (CN2 — (CN3 — (CN4 — (CN5 — __))))
```

Data intake: defined by initial state = all wh-dependencies in child-directed speech, as characterized by container nodes

The CHILDES Treebank can be very helpful, since it annotates phrase structure and dependencies.

[Link to CHILDES Treebank]

CHILDES Treebank

[Visual representation of CHILDES Treebank]

From valian.parsed

```
(ROOT (SBAR) (QP (NP what)))
(SQ (VP (AUX 's))
(NP (PRP (t)))
(VBN got)
(NP (NONE-ABAR-MH- *T*-1)))
(PP (IN on))
(NP (PRP (t)))))))
```

“What’s it got __ on it?”


“What’s it got __ on it?”
The Subjacency-ish strategy: Learning period

Subjacency-ish input & intake:
A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

Learning period:
Hart and Risley (1995) determined that American children in their samples were exposed to approximately one million utterances between birth and three years old.

Let's assume, based on available experimental studies, that intuitions about syntactic islands are acquired in a three year period, such as between the ages of two and five.

So, our modeled learner will get 1,000,000 utterances distributed similarly to the dependencies in the child-directed speech input samples from CHILDES.

Therefore, the modeled learner heard 200,000 wh-dependencies distributed this way, encoded as container node sequences:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Example Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>76.7%</td>
<td>IP-VP</td>
</tr>
<tr>
<td>12.8%</td>
<td>IP</td>
</tr>
<tr>
<td>5.6%</td>
<td>IP-VP-PP</td>
</tr>
<tr>
<td>2.5%</td>
<td>IP-VP-CP-IP-VP</td>
</tr>
<tr>
<td>1.1%</td>
<td>IP-VP-CP-IP-VP</td>
</tr>
</tbody>
</table>

20% of the utterances in the child-directed speech sample were wh-dependencies, distributed this way (26 types total):

- 76.7%  What did you see __?
- 12.8%  What ___ happened?
- 5.6%   What did she want to do __?
- 2.5%   What did she read from ___?
- 1.1%   What did she think he said ___?

Pearl & Sprouse 2013a, 2013b, 2015


Note that while this can be helpful for extracting container node sequences, the labels still may not be exactly right. Some post-processing is necessary here. But it sure helps as a basis, instead of having to search text alone and annotate container node sequences by hand.

---

**CHILDES Treebank**

http://www.socsci.uci.edu/~lpearl/CoLaLab/CHILDESTreebank/childestreebank.html


“What’s it got __ on it?”

Pearl & Sprouse 2013a, 2013b, 2015
The Subjacency-ish strategy: Target state

Subjacency-ish input & intake:
A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

Non-parallel lines indicate superadditivity, which indicates knowledge of islands.

But how do we get acceptability judgment equivalents?

Target state: Behavioral evidence of syntactic islands knowledge

For each set of island stimuli from Sprouse et al. (2012), we generate grammaticality preferences for the modeled learner based on the dependency’s perceived probability and use this as a stand-in for acceptability.

Let’s take a break for a few minutes

The Subjacency-ish strategy: Target state

Subjacency-ish input & intake:
A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).

Target state: Behavioral evidence of syntactic islands knowledge

non-island

Who ___ claimed that Lily forgot the necklace?

What did the teacher claim that Lily forgot ___?

island

Who ___ made the claim that Lily forgot the necklace?

What did the teacher make the claim that Lily forgot ___?

Pearl & Sprouse 2013a, 2013b, 2015

Lutz & Gagliardi 2015
Today’s Plan

I. So you want to model language acquisition

II. Modeling case study: Defining the pieces

III. Modeling case study: Implementation & Interpretation

Details: What counts as a container node and why

Encoding a dependency as a sequence of container nodes.

Who did she think that the gift was from?

\[ \text{CP} \quad \text{IP} \quad \text{VP} \quad \text{CP} \quad \text{IP} \quad \text{VP} \quad \text{PP} \]

What phrase structure nodes should children pay attention to?

This is only one option.

Details: What counts as a container node and why

Sanity check: What happens when we look at the dependencies the modeled child will have to make judgments about at the end of learning if we use this version of container nodes?

Important: Can the grammatical dependencies be distinguished from the ungrammatical dependencies with this representation?
### Details: What counts as a container node and why

**Can the grammatical dependencies be distinguished from the ungrammatical ones?**

Sprouse et al. (2012) stimuli:

<table>
<thead>
<tr>
<th>Complex NP islands</th>
<th>Subject islands</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>begin-IP-end</code></td>
<td>matrix</td>
</tr>
<tr>
<td><code>begin-IP-VP-CP-IP-VP-end</code></td>
<td>embedded</td>
</tr>
<tr>
<td><code>begin-IP-end</code></td>
<td>matrix</td>
</tr>
<tr>
<td><em><code>begin-IP-VP-NP-CP-IP-VP-end</code></em></td>
<td>embedded</td>
</tr>
</tbody>
</table>

All the ungrammatical dependencies are distinct from all the grammatical dependencies for these syntactic islands.

---

Uh oh! The ungrammatical dependencies look identical to some of the grammatical dependencies for these syntactic islands.

### Details: What counts as a container node and why

**Can the grammatical dependencies be distinguished from the ungrammatical ones?**

Sprouse et al. (2012) stimuli:

<table>
<thead>
<tr>
<th>Whether islands</th>
<th>Adjunct islands</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>begin-IP-end</code></td>
<td>matrix</td>
</tr>
<tr>
<td><code>begin-IP-VP-CP-IP-VP-end</code></td>
<td>embedded</td>
</tr>
<tr>
<td><code>begin-IP-end</code></td>
<td>matrix</td>
</tr>
<tr>
<td><em><code>begin-IP-VP-CP-IP-VP-end</code></em></td>
<td>embedded</td>
</tr>
</tbody>
</table>

One solution:

- Have CP container nodes be more specified for the learner: 
  - Use the lexical head to subcategorize the CP container node.

  \[ CP_{null}, CP_{that}, CP_{whether}, CP_{if}, \text{etc.} \]

The learner can then distinguish between these structures:

- \[ IP-VP-CP_{null/that} IP-VP \]
- \[ IP-VP-CP_{whether/if} IP-VP \]
Can the grammatical dependencies be distinguished from the ungrammatical ones?

Sprouse et al. (2012) stimuli:

**Complex NP islands**
- `begin-IP-end`
- `begin-IP-VP-CP_mat-IP-VP-end`
- `begin-IP-end`

**Subject islands**
- `matrix`
- `non-island`
- `begin-IP-end`
- `begin-IP-VP-CP_mat-IP-VP-end`
- `embedded`
- `island`

* `begin-IP-VP-CP_mat-IP-VP-end`
  `begin-IP-VP-CP_mat-IP-VP-end`

All the ungrammatical dependencies are still distinct from all the grammatical dependencies for these syntactic islands.

Details: What counts as a container node and why

Details: Subjacency-ish strategy in more detail

Encoding a dependency as a sequence of container nodes.

Who did she think that the gift was from?

![Diagram of dependency structure]

Pearl & Sprouse update: Maybe we should start with “basic” phrase structure nodes for everything except CP which we have lexical detail about.

Details: What counts as a container node and why

Details: What counts as a container node and why

Can the grammatical dependencies be distinguished from the ungrammatical ones?

Sprouse et al. (2012) stimuli:

**Whether islands**
- `begin-IP-end`
- `begin-IP-VP-CP_mat-IP-VP-end`
- `begin-IP-end`

**Adjunct islands**
- `matrix`
- `non-island`
- `begin-IP-end`
- `begin-IP-VP-CP_mat-IP-VP-end`
- `embedded`
- `island`
- `begin-IP-VP-CP_mat-IP-VP-end`

* `begin-IP-VP-CP_mat-IP-VP-end`
  `begin-IP-VP-CP_mat-IP-VP-end`

Now the ungrammatical dependencies are distinct from all the grammatical dependencies for these syntactic islands, too.

Details: What counts as a container node and why

Strategy:
1. Pay attention to the structure of dependencies.
2. Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
3. Use trigram frequency to calculate the probability of that trigram occurring in a dependency.

- `begin-IP-VP = 86/225`
  `p(begin-IP-VP) = 0.38`
- `IP-VP-end = 83/225`
  `p(IP-VP-end) = 0.37`
- `begin-IP-end = 13/225`
  `p(begin-IP-end) = 0.06`
- `IP-VP-IP = 6/225`
  `p(IP-VP-IP) = 0.03`
- `VP-IP-VP = 6/225`
  `p(VP-IP-VP) = 0.03`
- `IP-VP-PP = 3/225`
  `p(IP-VP-PP) = 0.01`
- `VP-PP-end = 3/225`
  `p(VP-PP-end) = 0.01`
- `IP-NP-PP = 0/225`
  `p(IP-NP-PP) = 0.00`
Strategy:
(1) Pay attention to the structure of dependencies.

(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.

(3) Use trigram frequency to calculate the probability of that trigram occurring in a dependency.

One way to do smoothing: Lidstone’s Law

\[
\frac{\text{total observations of } t + \alpha}{\text{total observations of all } N \text{ trigrams} + Na}
\]

Allowing a very small default probability is known as “smoothing”.

A number less than 1 (e.g., \(\alpha = 0.5\)) is added to all counts. This means all N trigrams have \(\alpha\) added to them (that’s why \(Na\) is in the denominator). This is true no matter how frequently each trigram was observed (so some may have appeared 83 times while others appeared only once).
How does this part work?

<table>
<thead>
<tr>
<th>Dependency Structure</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>begin-IP-VP</code></td>
<td>0.38</td>
</tr>
<tr>
<td><code>IP-VP-end</code></td>
<td>0.37</td>
</tr>
<tr>
<td><code>begin-IP-end</code></td>
<td>0.06</td>
</tr>
<tr>
<td><code>IP-VP-VP</code></td>
<td>0.03</td>
</tr>
<tr>
<td><code>begin-IP-VP</code></td>
<td>0.01</td>
</tr>
<tr>
<td>Total types</td>
<td>0.002</td>
</tr>
</tbody>
</table>

(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
(3) Use trigram frequency to calculate the probability of that trigram occurring in a dependency.

Strategy:

- **Smoothing**: Allows a very small default probability to be assigned to infrequent trigrams. This is known as “smoothing.”
- If we had 100 trigram types, we could use the matrix:
  
<table>
<thead>
<tr>
<th>Trigram Types</th>
<th>p(IP-NP-PP)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IP-NP-PP</code></td>
<td>0.002</td>
</tr>
<tr>
<td><code>VP-PP-end</code></td>
<td>0.01</td>
</tr>
<tr>
<td><code>IP-VP-VP</code></td>
<td>0.03</td>
</tr>
<tr>
<td><code>begin-IP-end</code></td>
<td>0.06</td>
</tr>
<tr>
<td><code>begin-IP-VP</code></td>
<td>0.38</td>
</tr>
<tr>
<td><code>IP-VP-end</code></td>
<td>0.37</td>
</tr>
</tbody>
</table>

(4) When you see a new dependency, break it down into its trigrams and then calculate its probability, based on the trigram probabilities.
(5) Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.
Let’s use log probabilities:

1. They’re easier to compare visually, especially when the probabilities are very small numbers.

5. Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

How does this part work?

Let’s use log probabilities:

(1) They’re easier to compare visually, especially when the probabilities are very small numbers.

\[ \log_{10}(0.14) = -0.85 \]
\[ \log_{10}(0.000000034) = -7.46 \]
\[ \log_{10}(0.0000000001) = -11.0 \]

5. Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

How does this part work?

Let’s use log probabilities:

(2) The integer lets us quickly compare the order of magnitude in difference.

\[ \log_{10}(0.14) = -0.85 \]
\[ \log_{10}(0.000000034) = -7.46 \]
\[ \log_{10}(0.0000000001) = -11.0 \]

This one is 4 times \([10^4]\) smaller than the one above it.

How does this part work?

Let’s use log probabilities:

(3) Multiplication is addition in log space. This is handy when working with lots of trigrams with small probabilities.

\[ \log_{10}(0.0000000001) = \log_{10}(0.00000000034) + \log_{10}(0.0000000001) \]
\[ -7.46 + -11.0 = -18.46 \]

5. Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

How does this part work?
Let's use log probabilities:
Interpretation: Since all log probabilities are negative, what matters is less negative (closer to 0) vs. more negative. This is what we plot on the interaction plots.

Less negative = more probable = more grammatical.

(5) Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

How does this part work?

Implementation: Subjacency-ish strategy

Strategy:
(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
(3) Use trigram frequency to calculate the probability of that trigram occurring in a dependency.
(4) When you see a new dependency, break it down into its trigrams and then calculate its probability, based on the trigram probabilities.
(5) Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

So this is what we want to implement in a computer program. What decisions remain?

Details: Subjacency-ish strategy in more detail

Let's use log probabilities:
Interpretation: Since all log probabilities are negative, what matters is less negative (closer to 0) vs. more negative. This is what we plot on the interaction plots.

Less negative = more probable = more grammatical.

(5) Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

How does this part work?

Implementation: Subjacency-ish strategy

Strategy:
(1) Pay attention to the structure of dependencies.
(2) Break these dependency structures into smaller pieces made up of three units (trigrams) that you can track the frequency of in the input you encounter.
(3) Use trigram frequency to calculate the probability of that trigram occurring in a dependency.
(4) When you see a new dependency, break it down into its trigrams and then calculate its probability, based on the trigram probabilities.
(5) Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

So this is what we want to implement in a computer program. What decisions remain?

Things to consider when implementing a model

What programming language should you use?
This depends a lot on
(a) what things you need to be able to do, and
(b) what’s handy (either because you’re already familiar with it or because you have easy access to it)

Some languages are excellent at quick text processing (ex: perl), some are fantastic for visualization (ex: R, matlab), some are wonderful at fast math operations (ex: numpy libraries of python, matlab, C++), some are great for generative models (ex: Church, WebPPL), some are brilliant for portability (ex: java, python), ...
Things to consider when implementing a model

What programming language should you use?
This depends a lot on
(a) what things you need to be able to do, and
(b) what’s handy (either because you’re already familiar with it or because you have easy access to it)

Some languages are excellent at quick text processing (ex: perl), some are fantastic for visualization (ex: R, matlab), some are wonderful at fast math operations (ex: numpy libraries of python, matlab, C++), some are great for generative models (ex: Church, WebPPL), some are brilliant for portability (ex: java, python), ...

If you know what you need to do (because you’ve mapped out your learning strategy implementation in glorious detail), it’s easier to choose.

Things to consider when implementing a model

Wait, what do I need to be able to do?
This includes things like:
(i) how the input needs to be represented for the code
(ii) what data structures will be used
(iii) what kind of output will be generated & what format that output will be in

Which dependencies to generate probabilities for

Dependency frequency in the input
Things to consider when implementing a model

Wait, what do I need to be able to do?

This includes things like:

(i) how the input needs to be represented for the code
(ii) what data structures will be used
(iii) what kind of output will be generated & what format that output will be in

Information useful for analysis

Probabilities over time

For the Pearl & Sprouse learning strategy for syntactic islands, I used perl for the model implementation and R for the interaction plots because I knew perl already and Jon Sprouse had previously made interaction plot graphs in R.

Useful skill: Being able to adapt someone else’s freely available code to what you need to do. This is why it can be handy to know a little about a variety of programming languages.

Things to consider when implementing a model

Now back to what programming language you should use...

This depends a lot on

(a) what things you need to be able to do, and
(b) what’s handy (either because you’re already familiar with it or because you have easy access to it)

For the Pearl & Sprouse learning strategy for syntactic islands, I used perl for the model implementation and R for the interaction plots because I knew perl already and Jon Sprouse had previously made interaction plot graphs in R.

Useful skill: Being able to adapt someone else’s freely available code to what you need to do. This is why it can be handy to know a little about a variety of programming languages.
Now what?

Non-island dependencies we specified that we cared about. Once we run this model, we get some numbers for a variety of island types.

Each island type had four stimuli embedded.

Now we need to link them back to the target state we're interested a little more precisely. This way, other people can understand what our results mean.

Probabilities at the end of learning are what's useful for generating the numbers needed for the interaction plots.

Each of those is characterized by a sequence of container nodes.
Superadditivity observed for all four islands — the qualitative behavior suggests that this learner has knowledge of these syntactic islands.

The Subjacency-ish representation that relies on container node trigram probabilities can solve this learning problem using this learning strategy.

We can get the log probability of all these dependencies and then plot them on interaction plots.

When we compare this against the desired target behavior...

Note: We’re careful to say “qualitative” behavior fit because there are lots of other factors that impact acceptability judgment behavior, and we’ve only modeled one (presumably) large part of them, which is the grammaticality of the dependency.
But is this all we can say?

No! One useful aspect of models is that we can look inside the modeled child to see why it’s behaving the way that it is. (This is something that’s harder to do with real children — that is, opening up their minds and seeing how they work.)

What’s going on?

Why are the island-spanning dependencies so much worse than the grammatical ones?

Let’s look inside them and see!

It turns out that each island-spanning dependency contains at least one very low probability container node trigram. So these are the relevant “island” representations.

a. Complex NP
   (i) * What did the teacher [the] [say] the claim [the] that [Lily] say forgot [the] [her] [homework]?
   (ii) start-IP-VP-NP-CP-IP-NP / start-IP-VP-CP-IP-NP
   (iii) Low probability:
      VP-NP-CP-IP
      NP-CP-IP

b. Subject
   (i) * Who does [I [make] the claim [the] that [Lily] think [the] [necklace] [worth $50] is [expensive]?
   (ii) start-IP-VP-CP-IP-NP-PP
   (iii) Low probability:
      CP-IP-NP-PP

Model results & interpretation

Table 4. Inferred grammaticality of different v-b-dependencies from Sprouse et al. (2012a), represented w/log proba.

<table>
<thead>
<tr>
<th>v-b-dependencies</th>
<th>Child-directed speech</th>
<th>Adult-directed speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grammatical dependencies</td>
<td>-1.22</td>
<td>-0.93</td>
</tr>
<tr>
<td>Embedded subject</td>
<td>-0.99</td>
<td>-0.67</td>
</tr>
<tr>
<td>Embedded object</td>
<td>-1.84</td>
<td>-1.00</td>
</tr>
<tr>
<td>Island-spanning dependencies</td>
<td>-1.91</td>
<td>-1.69</td>
</tr>
<tr>
<td>Complex NP</td>
<td>-1.91</td>
<td>-1.69</td>
</tr>
<tr>
<td>Subject</td>
<td>-2.07</td>
<td>-1.69</td>
</tr>
<tr>
<td>Whether</td>
<td>-1.84</td>
<td>-1.69</td>
</tr>
<tr>
<td>Adjunct</td>
<td>-1.84</td>
<td>-1.69</td>
</tr>
</tbody>
</table>

Model results & interpretation

Table 4. Inferred grammaticality of different v-b-dependencies from Sprouse et al. (2012a), represented w/log proba.

<table>
<thead>
<tr>
<th>v-b-dependencies</th>
<th>Child-directed speech</th>
<th>Adult-directed speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grammatical dependencies</td>
<td>-1.22</td>
<td>-0.93</td>
</tr>
<tr>
<td>Embedded subject</td>
<td>-0.99</td>
<td>-0.67</td>
</tr>
<tr>
<td>Embedded object</td>
<td>-1.84</td>
<td>-1.00</td>
</tr>
<tr>
<td>Island-spanning dependencies</td>
<td>-1.91</td>
<td>-1.69</td>
</tr>
<tr>
<td>Complex NP</td>
<td>-1.91</td>
<td>-1.69</td>
</tr>
<tr>
<td>Subject</td>
<td>-2.07</td>
<td>-1.69</td>
</tr>
<tr>
<td>Whether</td>
<td>-1.84</td>
<td>-1.69</td>
</tr>
<tr>
<td>Adjunct</td>
<td>-1.84</td>
<td>-1.69</td>
</tr>
</tbody>
</table>
Model results & interpretation

What’s going on?
Why are the island-spanning dependencies so much worse than the grammatical ones?

Let’s look inside them and see!
It turns out that each island-spanning dependency contains at least one very low probability container node trigram. So these are the relevant “island” representations.

c. Whether
   (i) * What does Jack [IP VP wonder [CN1 CP whuber IP VP-end]]?
   (ii) starts-IP-VP-CP_whuber-IP-VP-end
   (iii) Low probability:
         IP-VP-CP_whuber
         VP-CP_whuber-IP
         CP_whuber-IP-VP

Big picture
Representation validation: Rather than needing to know about specific island constraints, humans could simply be sensitive to the local pieces of structure captured by container node trigrams.

Lobs & Gagliardi 2015

What’s going on?
Why are the island-spanning dependencies so much worse than the grammatical ones?

Let’s look inside them and see!
It turns out that each island-spanning dependency contains at least one very low probability container node trigram. So these are the relevant “island” representations.

d. Adjunct
   (i) * What does [IP VP worry [CN1 CP nil CP VP]]?
   (ii) starts-IP-VP-CP-nil-IP-VP-end
   (iii) Low probability:
         IP-VP-nil-IP-VP
         VP-nil-CP-IP
         CP-nil-IP-VP

Model results & interpretation

Model results & interpretation

Model results & interpretation

Model results & interpretation
Big picture debate: What’s in UG?

**Subjacency-ish**

\[Wh \rightarrow [BN1 \rightarrow [BN2 \rightarrow [\ldots] \rightarrow \ldots]]\]

- Attend to container nodes of a particular kind
- Low probability items are dispreferred

**UG = innate + domain-specific**

<table>
<thead>
<tr>
<th></th>
<th>Innate</th>
<th>Derived</th>
<th>Domain-specific</th>
<th>Domain-general</th>
</tr>
</thead>
<tbody>
<tr>
<td>innate</td>
<td>?</td>
<td>?</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>derived</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

- Fewer pieces of knowledge necessarily in UG + empirically-motivated alternative proposal for one component.

Big picture: Syntactic islands

Informing theories of representation & acquisition

1. **Broadening the set of relevant data** in the acquisitional intake to include all wh-dependencies
2. Evaluating output by how useful it is for generating acceptability judgment behavior
3. Not necessarily needing the prior knowledge we thought we did in UG: container nodes rather than bounding nodes, no domain-specific constraint on length
Computational acquisition modeling: 
Big picture

This technique is a useful tool — so let’s use it to inform our theories of representation and acquisition!

Thank you!

Jon Sprouse


This work was supported in part by NSF grants BCS-0843896 and BCS-1347028.