Computational model building for language acquisition: An introduction

Lisa Pearl
University of California, Irvine

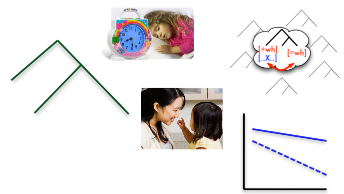
Computation of Language Laboratory
UC Irvine

April 17, 2016:
Great Lakes Expo for Experimental and Formal Undergraduate Linguistics
Michigan State University

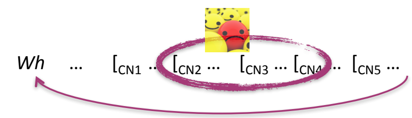
Today's Plan

I. So you want to model language acquisition

II. Modeling case study: Defining the pieces



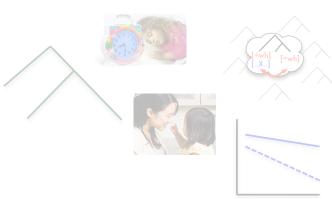
III. Modeling case study: Implementation & Interpretation



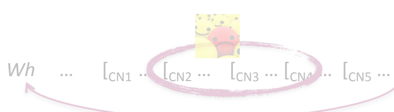
Today's Plan

I. So you want to model language acquisition

II. Modeling case study: Defining the pieces

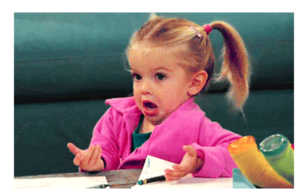


III. Modeling case study: Implementation & Interpretation



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?

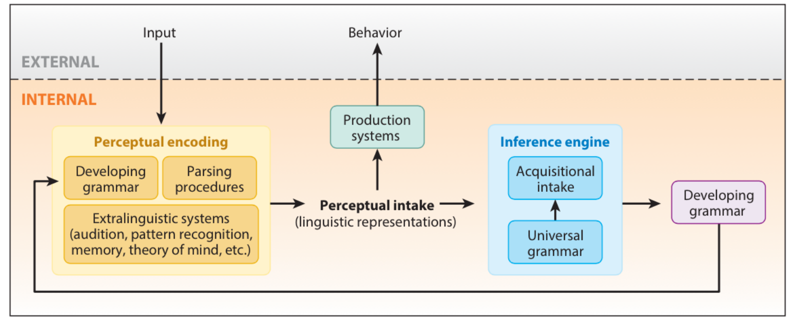
A model that answers these kinds of "how" questions is likely to be an informative model — it tells us something we didn't know before and didn't necessarily have another way to find out.

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.

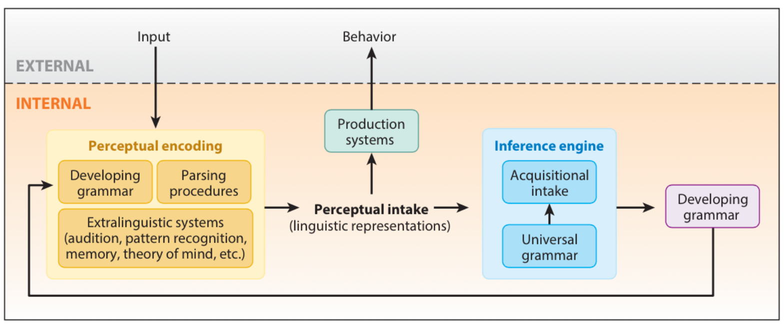
One idea about how acquisition works



Lidz & Gagliardi 2015

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.

One idea about how acquisition works



Lidz & Gagliardi 2015

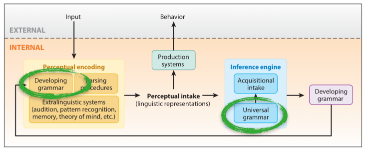
This allows us, as computational modelers, to define the acquisition task precisely enough to come up with ways children might solve it. The learning strategies we come up with can also be characterized in terms of these acquisition task pieces.

Characterizing the acquisition task

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

Characterizing the acquisition task

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



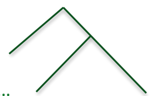
Lidz & Gagliardi 2015

Initial knowledge

ex: syntactic categories exist and can be identified

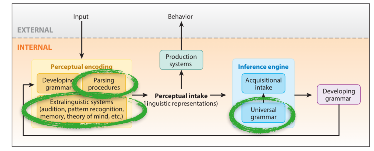
ex: phrase structure exists and can be identified

ex: participant roles can be identified



Characterizing the acquisition task

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

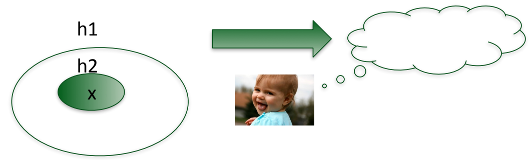


Lidz & Gagliardi 2015

Initial abilities & biases

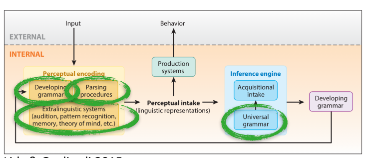
ex: frequency information can be tracked

ex: distributional information can be leveraged



Characterizing the acquisition task

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



Lidz & Gagliardi 2015

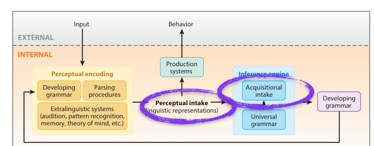
This is typically where a major part of a learning strategy would be concretely realized with the model.

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.

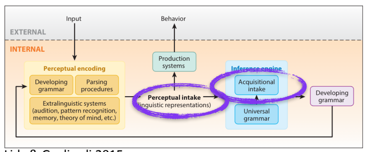


Lidz & Gagliardi 2015

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.



Lidz & Gagliardi 2015

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

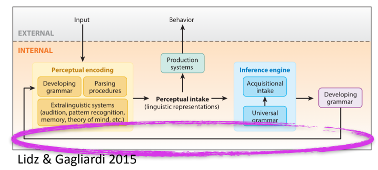
[defined by knowledge & biases/capabilities in the initial state]

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?



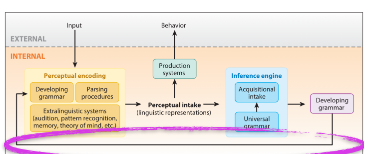
Lidz & Gagliardi 2015

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?



Lidz & Gagliardi 2015

how long children have to reach the target knowledge state

ex: 3 years, ~1,000,000 data points

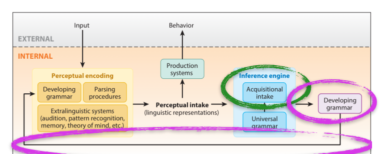
ex: 4 months, ~36,500 data points

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?



Lidz & Gagliardi 2015

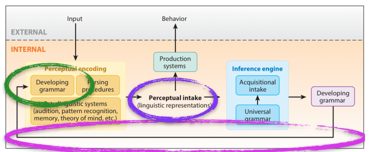
This is when inference happens, i.e., when updates are made to the developing grammar.

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?



Lidz & Gagliardi 2015

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

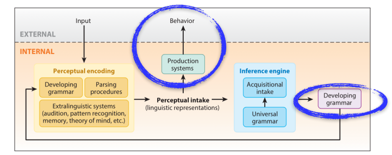
Characterizing the acquisition task

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)?



Lidz & Gagliardi 2015

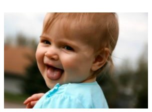
Characterizing the acquisition task

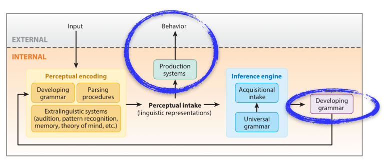
Initial state: What does the child start with?

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)?





Lidz & Gagliardi 2015

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? 

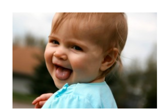
Characterizing the acquisition task

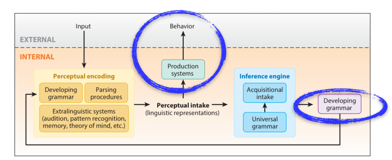
Initial state: What does the child start with?

Data intake: The relevant data for acquisition.

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

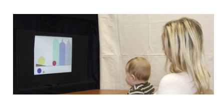
Target state: What does successful acquisition look like? What knowledge is the child trying to attain (often assessed in terms of observable behavior)?

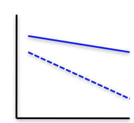




Lidz & Gagliardi 2015

Behavior


looking time preferences


z-score rating

Characterizing the acquisition task

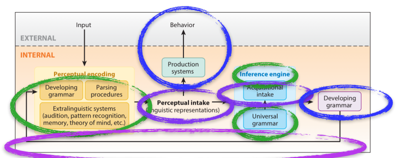
Initial state: What does the child start with?

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.



Lidz & Gagliardi 2015

Characterizing the acquisition task

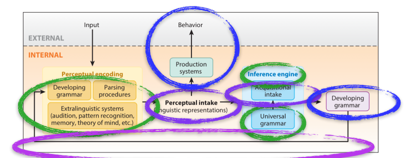
Initial state: What does the child start with?

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?

When we identify a successful learning strategy via modeling, this is an existence proof that children could solve that acquisition problem using the learning biases, knowledge, and capabilities comprising that strategy.



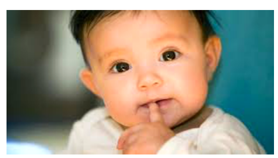
Lidz & Gagliardi 2015

This is something we didn't know before!
Therefore, it's informative.

Back to the process of modeling

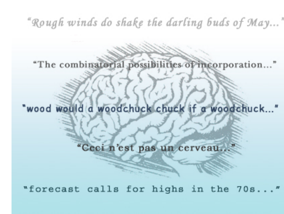
So if your question is a question about how the language acquisition process works, a computational model might be the right tool to use.

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)



On explaining (Marr 1982)

"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…"

On explaining (Marr 1982)

"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?

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.

(3,4) ⟶ 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+ -3) = 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

  99
+ 5

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

  99
+ 5
14

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

 1
  99
+ 5
 4

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

 1
  99
+ 5
104

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.

Properties:

- slices of bread containing both peanutbutter and jelly
- number of bread slices: 2
- sandwich is sliced in half
- crusts are left on
- jelly type: grape
- peanutbutter type: crunchy

etc.

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.

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: PBJ-making steps.

Implementational

How can the representation and algorithm be realized physically?

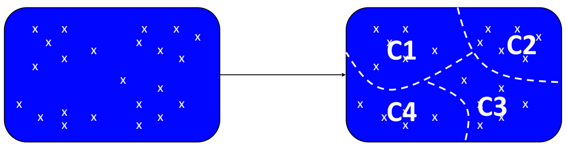
Directing your younger sibling to follow the steps above to make you a sandwich.

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

húwzəfɹéjdəvðəbígbǽdwʌ́lf

↓

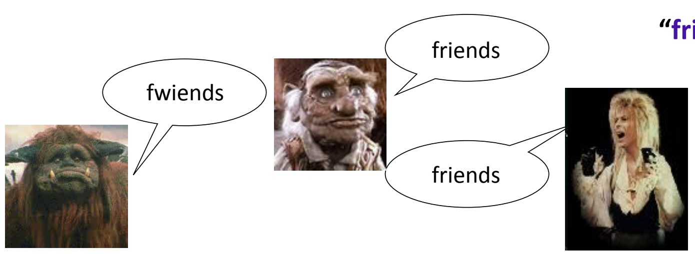
húwz əfɹéjd əv ðə bíg bǽd wʌ́lf
who's afraid of the big bad wolf

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



"friends"

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, …?

Mapping the framework

Goal: Understanding the "how" of language acquisition

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

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?

Mapping the framework

Goal: Understanding the "how" of language acquisition

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

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)

Jareth juggles crystals

Subject Verb Object

Kannada

Subject t_{Object} Verb Object

German

Subject Verb $t_{Subject}$ Object t_{Verb}

English

Subject Verb Object

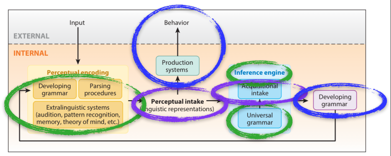
Mapping the framework

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?



Lidz & Gagliardi 2015

Mapping the framework

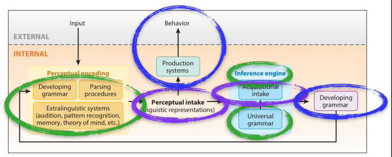
Goal: Understanding the "how" of language acquisition

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?

This is the goal of learnability approaches (often posed at the computational-level of analysis): Frank et al. 2009, Goldwater et al. 2009, Pearl et al. 2010, Pearl 2011, Legate & Yang 2012, Dillon et al. 2013, Doyle & Levy 2013, Feldman et al. 2013, Orita et al. 2013, Pearl & Sprouse in progress



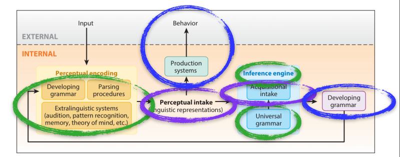
Lidz & Gagliardi 2015

Mapping the framework

Goal: Understanding the "how" of language acquisition

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

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?



Lidz & Gagliardi 2015

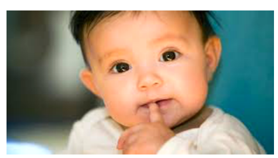
Mapping the framework

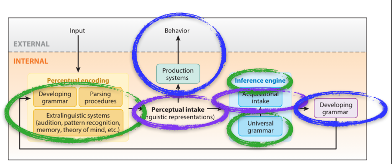
Goal: Understanding the "how" of language acquisition

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

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.)





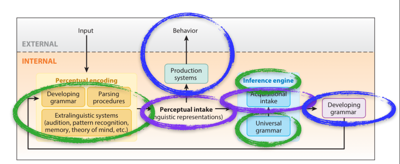
Lidz & Gagliardi 2015

Mapping the framework

Goal: Understanding the "how" of language acquisition

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

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).



Lidz & Gagliardi 2015

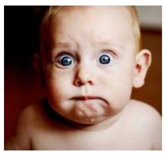
Mapping the framework

Goal: Understanding the "how" of language acquisition

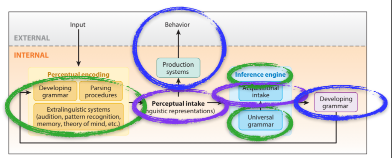
First, we need a computational-level description of the learning problem.

(What happened in a nutshell)

Why do none of these learning strategies work?



Because they're solving the wrong acquisition task…oops.



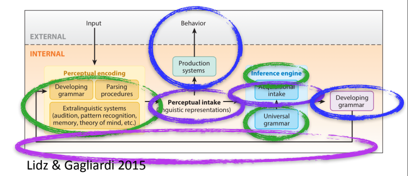
Lidz & Gagliardi 2015

Mapping the framework

Goal: Understanding the "how" of language acquisition

Second, we need to identify the algorithmic-level description:

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 …



Lidz & Gagliardi 2015

Mapping the framework

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

húwzəfɹéjdəvðəbígbǽdwʌ́lf — intake

↓

húwz əfɹéjd əv ðə bíg bǽd wʌ́lf
who's afraid of the big bad wolf

Mapping the framework

Algorithmic Level:

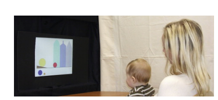
Output: Theoretical linguistics and experimental studies can tell us what the output should look like by observing adult and child knowledge of various linguistic phenomena, as indicated by their behavior.

Example problem: speech segmentation

húwzəfɹéjdəvðəbígbǽdwʌ́lf — intake

↓

húwz əfɹéjd əv ðə bíg bǽd wʌ́lf — output
who's afraid of the big bad wolf



looking time preferences

Mapping the framework

Algorithmic Level:

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

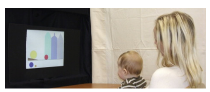
Example problem: speech segmentation

húwzəfɹéjdəvðəbígbǽdwʌ́lf — intake

↓ Method

húwz əfɹéjd əv ðə bíg bǽd wʌ́lf — output

who's afraid of the big bad wolf

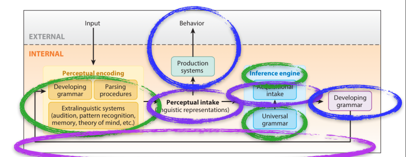


looking time preferences

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?

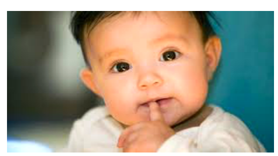


Lidz & Gagliardi 2015

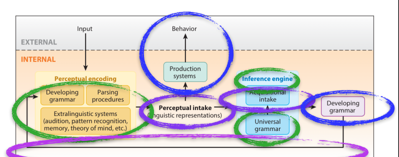
Mapping the framework

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 useable by children.



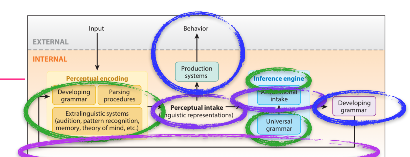
Lidz & Gagliardi 2015

Mapping the framework

Goal: Understanding the "how" of language acquisition (implementation-level)

Another important (not so basic) question for an acquisition model: If we have an algorithm that seems useable by children to usefully solve an acquisition task, how is it implemented in the brain?

(Not something we can easily model yet)



Lidz & Gagliardi 2015

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?

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.

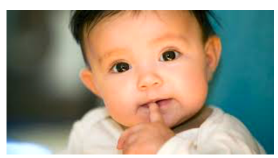
What level of model do you want to build?

Computational: Is it possible for the child with a specific initial state to use the acquisitional intake to achieve the target knowledge/behavior?

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?

Computational: Is it possible for the child with a specific initial state to use the acquisitional intake to achieve the target knowledge/behavior?

Algorithmic: Is it possible for the child 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?

Back to the process of modeling

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?

Back to the process of modeling

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)

General modeling process

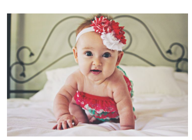
(1) Decide what kind of learner the model represents

(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)

...

UCI_Brent_Syl Corpus: In order to train an automatic segmentation program, Lisa Pearl and Lawrence Phillips at UC Irvine have created a corpus derived from the CDS of the CHILDES Brent corpus. The corpus comes with the scripts and dictionary used to produce it.



Empirically grounding the input

http://childes.psy.cmu.edu

CHILDES Child Language Data Exchange System

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.


```
@Loc:   Eng-NA-MOR/Rollins/al12.cha
@PID:   11312/c-00017262-1
@Begin
@Languages:     eng
@Participants:  CHI Target_Child , MOT Mother
@ID:    eng|rollins|CHI||||Target_Child|||
@ID:    eng|rollins|MOT||||Mother|||
@Media: al12, video
@Activities:    Free Play
*MOT:   you haven't seen this .
%mor:   pro|you aux|have~neg|not part|see&PASTP pro:dem|this .
%gra:   1|4|SUBJ 2|4|AUX 3|2|NEG 4|0|ROOT 5|4|OBJ 6|4|PUNCT
*MOT:   that looks pretty cool .
%mor:   det|that n|look-PL adv:int|pretty adj|cool .
%gra:   1|2|DET 2|0|INCROOT 3|4|JCT 4|2|XMOD 5|2|PUNCT
*MOT:   do you know how to work that .
%mor:   mod|do pro|you v|know adv:wh|how inf|to v|work pro:dem|that .
%gra:   1|3|AUX 2|3|SUBJ 3|0|ROOT 4|3|OBJ 5|6|INF 6|4|XCOMP 7|6|OBJ 8|3|PUNCT
*MOT:   yes you do .
%mor:   co|yes pro|you v|do .
%gra:   1|3|COM 2|3|SUBJ 3|0|ROOT 4|3|PUNCT
```

Empirically grounding the input

http://childes.psy.cmu.edu

CHILDES Child Language Data Exchange System

"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)

húwzəfɹéjdəvðəbígbǽdwʌ́lf

↓

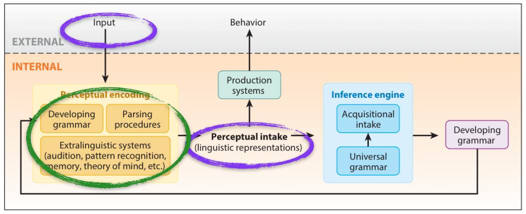
húw zə fɹéj dəv ðə bíg bǽd wʌ́lf

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)

húw zə fɹéj dəv ðə bíg bǽd wʌ́lf

bíg bǽd wʌ́lf

bígbǽd wʌ́lf   bígbǽdwʌ́lf

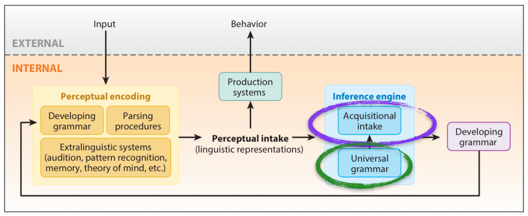
General modeling process

(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)



Lidz & Gagliardi 2015

General modeling process

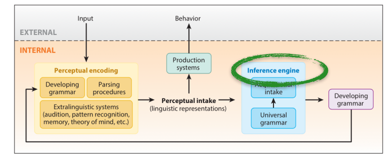
(1) Decide what kind of learner the model represents

(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)

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



Lidz & Gagliardi 2015

bíg bǽd wʌ́lf

bígbǽd wʌ́lf

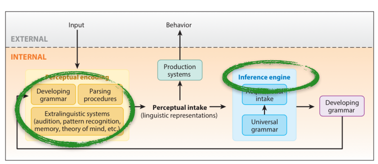
bígbǽdwʌ́lf

Cognitively plausible perception & inference

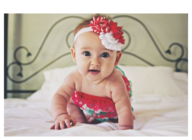
Many models will try to make cognitively plausible assumptions about how the child is representing and processing input data:

- 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.



Lidz & Gagliardi 2015



General modeling process

(5) Decide what the measure of success is

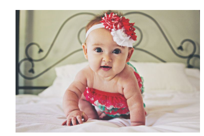
ex: developing knowledge

- Proto-lexicon of word forms

bǽd

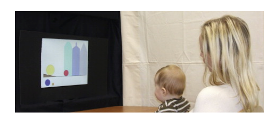
bíg

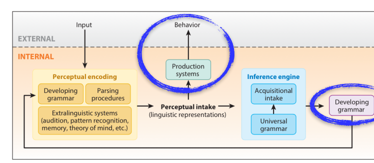
wʌ́lf



ex: behavior indicating developed knowledge

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



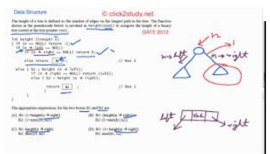


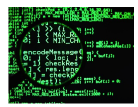
Lidz & Gagliardi 2015

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, wr.t. the measure of success

ex: developing knowledge

- Proto-lexicon of word forms

??? bǽd bíg wʌlf

ex: behavior indicating developed knowledge ???

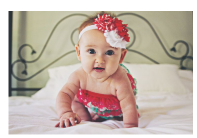
- Recognizing useful units (such as words) in a fluent speech stream, as indicated by looking time behavior

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

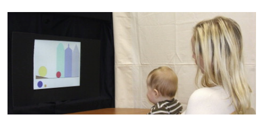
General modeling process

(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!"



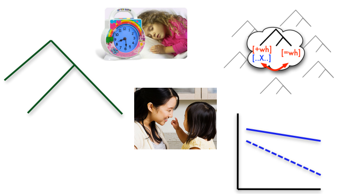
This is incredibly important — otherwise, no one knows what to make of your results (and whether or not they should care).

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

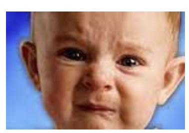
A property of language: Long-distance dependencies

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

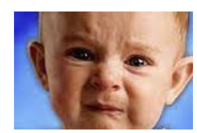
If the gap position appears inside certain structures (called "syntactic islands" by Ross (1967)), the dependency seems to be ungrammatical.



*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]?

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 __]? ← **wh syntactic island**
*What did [that Jack wrote __] offend the editor?
*What did Jack buy [a book and __]?
*Which did Jack borrow [__ book]?

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.'"

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 he fell?

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 he fell? **No island: Two interpretations possible**

—> When did the boy say __ he fell? **When did the saying happen?**

—> When did the boy say he fell __? **When did the falling happen?**

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 he fell? **No island: Two interpretations possible**

—> When did the boy say __ he fell? **That night**

—> When did the boy say he 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?

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]? **wh-island: Only one interpretation**

—> When did the boy say __ [how he fell]? **When did the saying happen?**

X *When did the boy say [how he fell __]?* *When did the falling happen?*

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]? **wh-island: Only one interpretation**

—> When did the boy say __ [how he fell]? **At night**

X *When did the boy say [how he fell __]?* *In the afternoon*

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

How could children learn this and other syntactic islands?

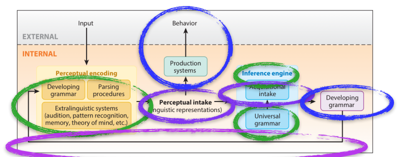
*What did you make [the claim that Jack bought __]?
*What do you think [the joke about __] offended Jack?
*What do you wonder [whether Jack bought __]?
*What do you worry [if Jack buys __]?
*What did you meet [the scientist who invented __]?
*What did [that Jack wrote __] offend the editor?
*What did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

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.

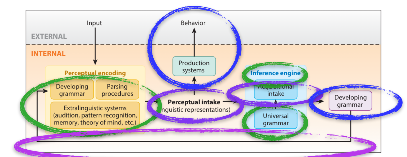
Let's build an algorithmic model, which will model the process unfolding in time.



Lidz & Gagliardi 2015

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.



Lidz & Gagliardi 2015

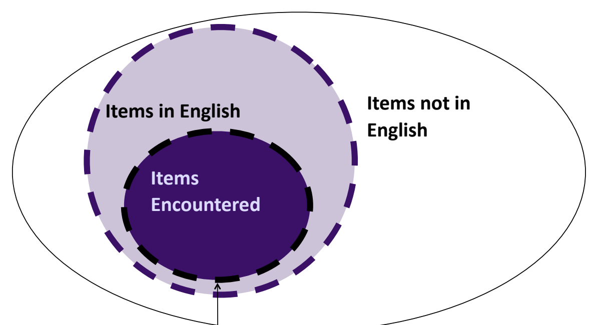
One point: Children's input doesn't look so helpful

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

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 __?*
…	

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?



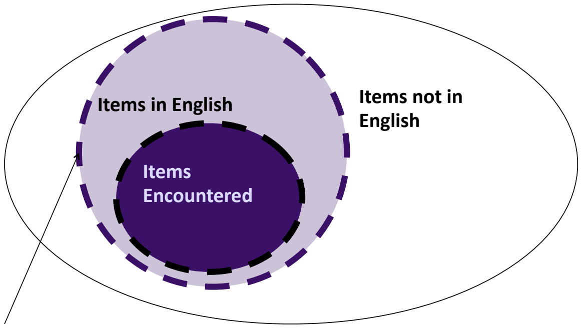
Wh-questions in input

What did you see?

What happened?

…

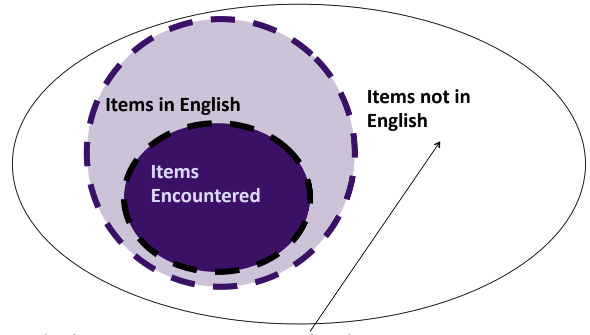
Syntactic islands: How to generalize?



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?

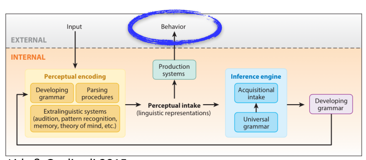


Ungrammatical wh-questions: Syntactic islands

*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.



Lidz & Gagliardi 2015

What does Jack think __?

What does Jack think that Lily said that Sarah heard that Jareth believed __?

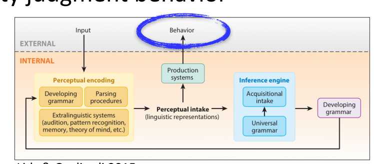
Complex NP island:
*What did you make [the claim that Jack bought __]?
Subject island:
*What do you think [the joke about __] offended Jack?
Whether island:
*What do you wonder [whether Jack bought __]?
Adjunct island:
*What do you worry [if Jack buys __]?

Pearl & Sprouse 2013a, 2013b, 2015

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)



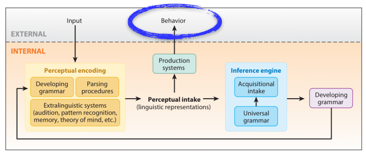
Lidz & Gagliardi 2015

Pearl & Sprouse 2013a, 2013b, 2015

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)



Lidz & Gagliardi 2015

Complex NP islands

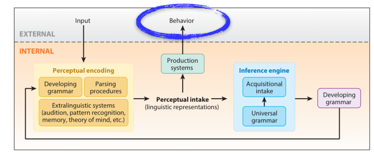
Who __ claimed that Lily forgot the necklace?	matrix | non-island
What did the teacher claim that Lily forgot __?	embedded | non-island
Who __ made the claim that Lily forgot the necklace?	matrix | island
*What did the teacher make the claim that Lily forgot __?	embedded | island

Pearl & Sprouse 2013a, 2013b, 2015

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)


Lidz & Gagliardi 2015

Subject islands

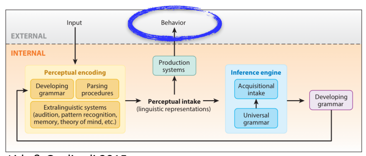
Who __ thinks the necklace is expensive?	matrix | non-island
What does Jack think __ is expensive?	embedded | non-island
Who __ thinks the necklace for Lily is expensive?	matrix | island
*Who does Jack think the necklace for __ is expensive?	embedded | island

Pearl & Sprouse 2013a, 2013b, 2015

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)


Lidz & Gagliardi 2015

Whether islands

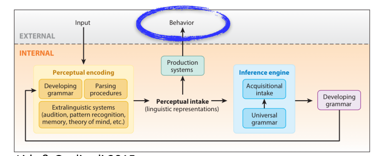
Who __ thinks that Jack stole the necklace?	matrix | non-island
What does the teacher think that Jack stole __ ?	embedded | non-island
Who __ wonders whether Jack stole the necklace?	matrix | island
*What does the teacher wonder whether Jack stole __ ?	embedded | island

Pearl & Sprouse 2013a, 2013b, 2015

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)


Lidz & Gagliardi 2015

Adjunct islands

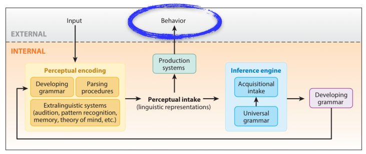
Who __ thinks that Lily forgot the necklace?	matrix | non-island
What does the teacher think that Lily forgot __ ?	embedded | non-island
Who __ worries if Lily forgot the necklace?	matrix | island
*What does the teacher worry if Lily forgot __ ?	embedded | island

Pearl & Sprouse 2013a, 2013b, 2015

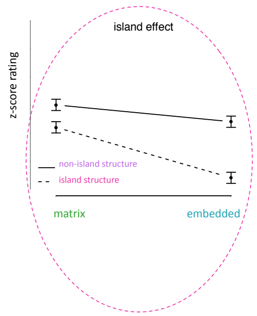
Adult behavior: acquisition target

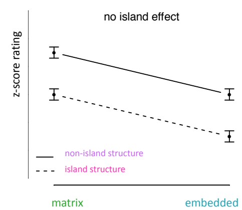
Adult knowledge as measured by acceptability judgment behavior

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).



Lidz & Gagliardi 2015

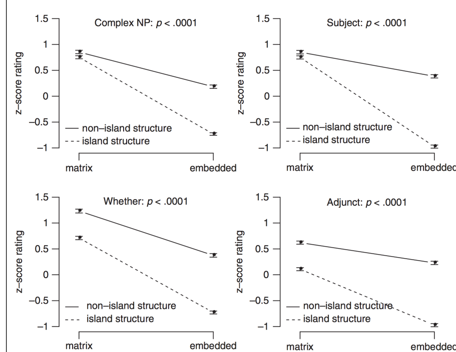


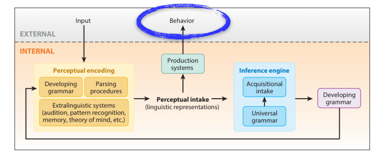


Pearl & Sprouse 2013a, 2013b, 2015

Adult behavior: acquisition target

Sprouse et al. (2012): acceptability judgments from 173 adult subjects





Lidz & Gagliardi 2015

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

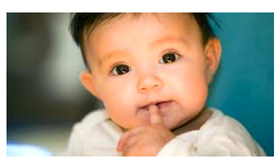
Importance for acquisition: This is one kind of target behavior that we'd like a modeled child to produce.

Pearl & Sprouse 2013a, 2013b, 2015

Syntactic islands: Specific islands we'll focus on

Complex NP	*What did you make [the claim that Jack bought __]?
Subject	*What do you think [the joke about __] offended Jack?
Whether	*What do you wonder [whether Jack bought __]?
Adjunct	*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]?

Question: If we're focusing on these wh-dependencies and that specific target state, what does children's input look like?



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:

ungrammatical

	MATRIX + NON-ISLAND	EMBEDDED + NON-ISLAND	MATRIX + ISLAND	*EMBEDDED + ISLAND*
Complex NP	7	295	0	0
Subject	7	29	0	0
Whether	7	295	0	0
Adjunct	7	295	15	0

Pearl & Sprouse submitted

Children's input really doesn't look so helpful

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**):

ungrammatical

	MATRIX + NON-ISLAND	EMBEDDED + NON-ISLAND	MATRIX + ISLAND	*EMBEDDED + ISLAND*
Complex NP	7	295	0	0
Subject	7	29	0	0
Whether	7	295	0	0
Adjunct	7	295	15	0

Pearl & Sprouse submitted

Children's input really doesn't look so helpful

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**):

ungrammatical

	MATRIX + NON-ISLAND	EMBEDDED + NON-ISLAND	MATRIX + ISLAND	*EMBEDDED + ISLAND*
Complex NP	7	295	0	0
Subject	7	29	0	0
Whether	7	295	0	0
Adjunct	7	295	15	0

Pearl & Sprouse submitted

Children's input really doesn't look so helpful

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**):

ungrammatical

	MATRIX + NON-ISLAND	EMBEDDED + NON-ISLAND	MATRIX + ISLAND	*EMBEDDED + ISLAND*
Complex NP	7	295	0	0
Subject	7	29	0	0
Whether	7	295	0	0
Adjunct	7	295	15	0

Pearl & Sprouse submitted

Children's input really doesn't look so helpful

…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**):

ungrammatical

	MATRIX + NON-ISLAND	EMBEDDED + NON-ISLAND	MATRIX + ISLAND	*EMBEDDED + ISLAND*
Complex NP	7	295	0	0
Subject	7	29	0	0
Whether	7	295	0	0
Adjunct	7	295	15	0

Pearl & Sprouse submitted

Syntactic islands: Specific islands we'll focus on

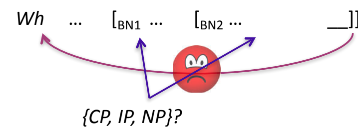
Complex NP	*What did you make [the claim that Jack bought __]?
Subject	*What do you think [the joke about __] offended Jack?
Whether	*What do you wonder [whether Jack bought __]?
Adjunct	*What do you worry [if Jack buys __]?
	*What did you meet [the scientist who invented __]?
	*What did [that Jack wrote __] 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

Subjacency (Chomsky 1973, Huang 1982, Lasnik & Saito 1984)

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



Bounding nodes are language-specific

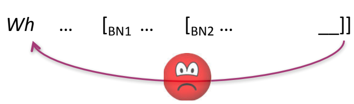
(CP, IP, and/or NP – must learn which ones are relevant for language)

Pearl & Sprouse 2013a, 2013b, 2015

Syntactic islands: Representations

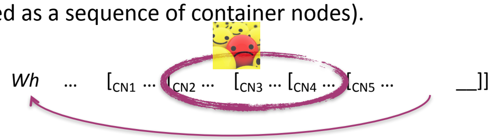
Subjacency (Chomsky 1973, Huang 1982, Lasnik & Saito 1984)

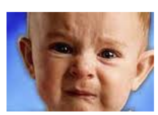
(1) A dependency cannot cross two or more bounding nodes.



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).



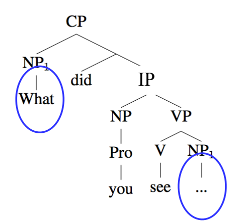


Container node: phrase structure node that contains dependency

[CP *What do* [IP *you* [VP *like __* [PP *in this picture?*]]]]

Pearl & Sprouse 2013a, 2013b, 2015

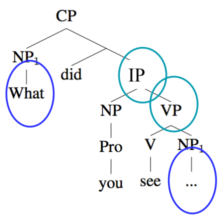
Container nodes sequences



How to describe this dependency:

What phrases is the gap inside but the wh-word isn't inside?

Container nodes sequences



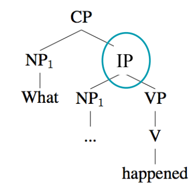
How to describe this dependency:
What phrases is the gap inside but the wh-word isn't inside?

What did you see __?
= What did [$_{IP}$ you [$_{VP}$ see __]]?
= IP-VP

Container nodes sequences

What did you see __?
= What did [$_{IP}$ you [$_{VP}$ see __]]?
= IP-VP

What __ happened?
= What [$_{IP}$ __ happened]?
= IP

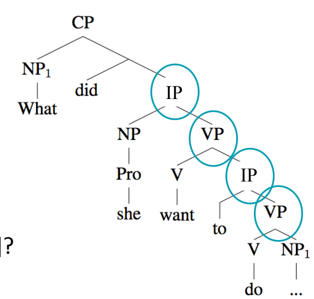


Container nodes sequences

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 [$_{IP}$ to [$_{VP}$ do __]]]]?
= IP-VP-IP-VP



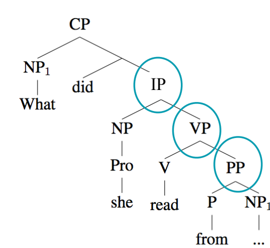
Container nodes sequences

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 [$_{IP}$ to [$_{VP}$ do __]]]]?
= IP-VP-IP-VP

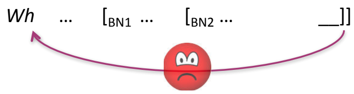
What did she read from __ ?
= What did [$_{IP}$ she [$_{VP}$ read [$_{PP}$ from __]]]]?
= IP-VP-PP



Syntactic islands: Representations

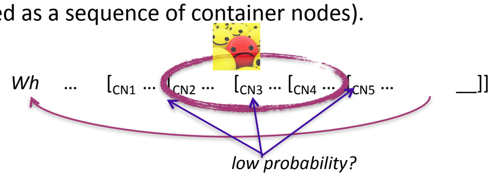
Subjacency (Chomsky 1973, Huang 1982, Lasnik & Saito 1984)

(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).



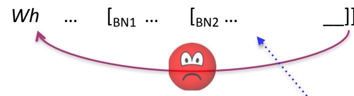
Low probability regions are language-specific
(defined by sequences of container nodes that must be learned)

Pearl & Sprouse 2013a, 2013b, 2015

Syntactic islands: Representations

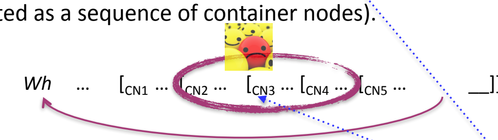
Subjacency (Chomsky 1973, Huang 1982, Lasnik & Saito 1984)

(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).



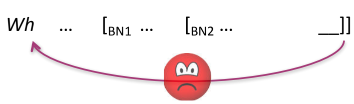
In common: Both rely on local structure anomalies (at some level)

Pearl & Sprouse 2013a, 2013b, 2015

Syntactic islands: Representations

Subjacency (Chomsky 1973, Huang 1982, Lasnik & Saito 1984)

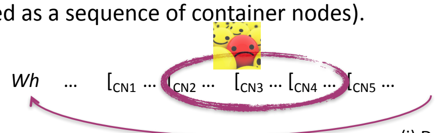
(1) A dependency cannot cross two or more bounding nodes.



(i) Dependencies defined over bounding nodes — track those
(ii) Bounding node = ?
(iii) 2+ 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).



(i) Dependencies defined over container node structure — track that already
(ii) Container node = ?
(iii) low probability =

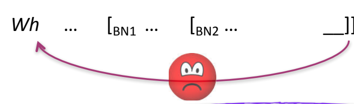
Different: Amount of language-specific knowledge built in just for islands

Pearl & Sprouse 2013a, 2013b, 2015

Syntactic islands: Representations

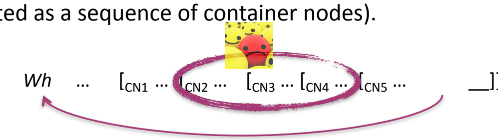
Subjacency (Chomsky 1973, Huang 1982, Lasnik & Saito 1984)

(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).



Pearl & Sprouse: Focused on evaluating this one

Pearl & Sprouse 2013a, 2013b, 2015

One strategy for some of the islands

Complex NP	*What did you make [the claim that Jack bought __]?
Subject	*What do you think [the joke about __] offended Jack?
Whether	*What do you wonder [whether Jack bought __]?
Adjunct	*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]?

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.

What did you see?
What happened?
What did she want to do?
What did she read from?
What did she think he said?

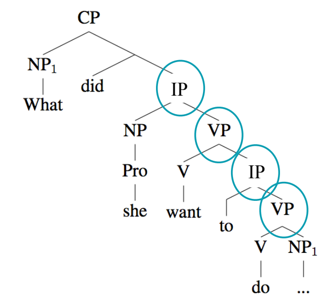
→

*What did you make the claim that Jack bought?
*What do you think the joke about offended Jack?

Subjacency-ish strategy in more detail

Strategy:
(1) Pay attention to the structure of dependencies.

What did she want to do __ ?
= What did [$_{IP}$ she [$_{VP}$ want [$_{IP}$ to [$_{VP}$ do __]]]]?
= IP-VP-IP-VP



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 =
begin-IP-*end*

IP-VP-IP-VP
= *begin*-IP-VP
IP-VP-IP
VP-IP-VP
IP-VP-*end*

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

IP-VP-IP-VP
= *begin*-IP-VP
IP-VP-IP
VP-IP-VP
IP-VP-*end*

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

begin-IP-VP = 86/225
IP-VP-*end* = 83/225
begin-IP-*end* = 13/225
IP-VP-IP = 6/225
VP-IP-VP = 6/225
IP-VP-PP = 3/225
VP-PP-*end* = 3/225
...

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.

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
...	

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 does Jack want __?
= What does [$_{IP}$ Jack [$_{VP}$ want __]]?
= IP-VP
= *begin*-IP-VP
IP-VP-*end*

p(IP-VP) = p(*begin*-IP-VP)*p(IP-VP-*end*)
= 0.38 * 0.37 = 0.14

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 does Jack want to do that for __?
= What does [$_{IP}$ Jack [$_{VP}$ want [$_{IP}$ to [$_{VP}$ do that [$_{PP}$ for __]]?
= IP-VP-IP-VP-PP
= *begin*-IP-VP
IP-VP-IP
VP-IP-VP
IP-VP-PP
VP-PP-*end*

p(IP-VP-IP-VP-PP) = p(*begin*-IP-VP)*p(IP-VP-IP)*p(VP-IP-VP)*p(IP-VP-PP)*p(VP-PP-*end*)
= 0.38*0.03*0.03*0.01*0.01 = 0.000000034

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.

Subject island

What do you think that the joke about __ offended Jack?
= What do [$_{IP}$ you [$_{VP}$ think [$_{CP}$ that [$_{IP}$ [$_{NP}$ the joke [$_{PP}$ about __]]]]]] offended Jack?
= IP-VP-CP-NP-PP
= *begin*-IP-VP
IP-VP-CP
VP-CP-IP
CP-IP-NP
IP-NP-PP
NP-PP-*end*

p(IP-VP-CP-IP-NP-PP) = p(*begin*-IP-VP)*p(IP-VP-CP)*p(VP-CP-S)*p(CP-IP-NP)*p(IP-NP-PP)*p(NP-PP-*end*)
= 0.86*0.01*0.001*0.00*0.00*0.02 = 0.00

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.

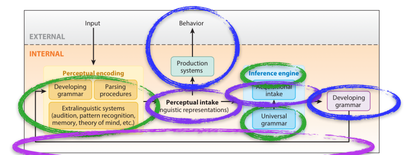
(5) Use calculated dependency probabilities as the basis for grammaticality judgments. Lower probability dependencies are dispreferred, compared to higher probability dependencies.

p(IP-VP) = 0.14
p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00

Let's model!

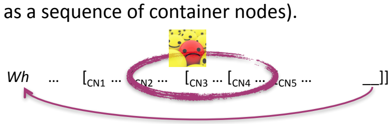
Let's try to pin down all the pieces we need for this strategy based on the Subjacency-ish representation: initial state, data intake, learning period, and target state.

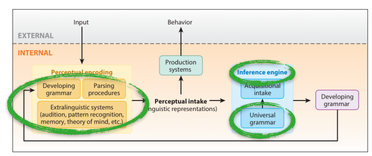


Lidz & Gagliardi 2015

The Subjacency-ish strategy: Initial state

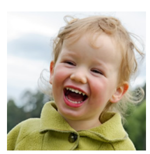
Subjacency-ish representation:
A dependency cannot cross a very low probability region of structure (represented as a sequence of container nodes).





Lidz & Gagliardi 2015

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



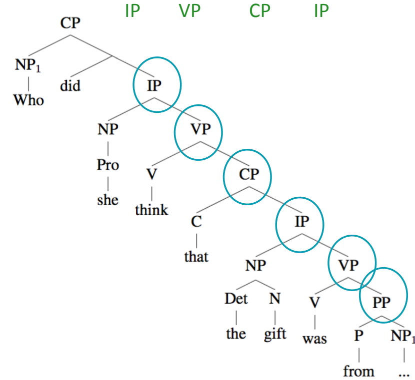
Pearl & Sprouse 2013a, 2013b, 2015

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?

$[_{CP}$ Who did $[_{IP}$ she $[_{VP}$ think $[_{CP}$ that $[_{IP}$ $[_{NP}$ the gift] $[_{VP}$ was $[_{PP}$ from __]]]]]]]?
IP VP CP IP VP PP



Pearl & Sprouse 2013a, 2013b, 2015

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?

[$_{CP}$ Who did [$_{IP}$ she [$_{VP}$ think [$_{CP}$ that [$_{IP}$ [$_{NP}$ the gift] [$_{VP}$ was [$_{PP}$ from __]]]]]]]]?
IP VP CP IP VP PP

begin-IP-VP-CP-IP-VP-PP-*end* =

begin-IP-VP
IP-VP-CP
VP-CP-IP
CP-IP-VP
IP-VP-PP
VP-PP-*end*

Pearl & Sprouse 2013a, 2013b, 2015

Subjacency-ish: Developing knowledge

A child learns about the frequency of container node trigrams…

[$_{CP}$ Who did [$_{IP}$ she [$_{VP}$ think [$_{CP}$ that [$_{IP}$ [$_{NP}$ the gift] [$_{VP}$ was [$_{PP}$ from __]]]]]]]]?
IP VP CP IP VP PP

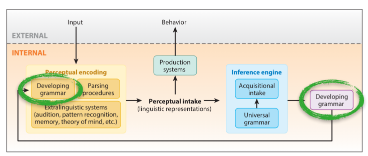
begin-IP-VP-CP-IP-VP-PP-*end* =

begin-IP-VP
IP-VP-CP
VP-CP-IP
CP-IP-VP
IP-VP-PP
VP-PP-*end*

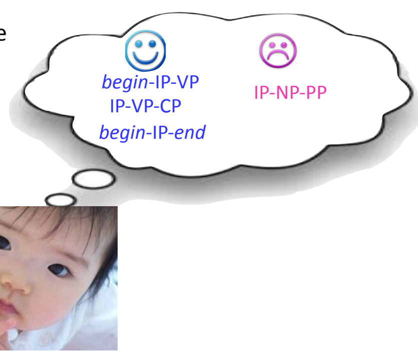
Pearl & Sprouse 2013a, 2013b, 2015

Subjacency-ish: Developing knowledge

…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.



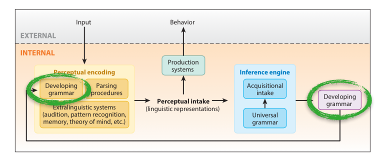
Lidz & Gagliardi 2015



Pearl & Sprouse 2013a, 2013b, 2015

Subjacency-ish: Developing knowledge

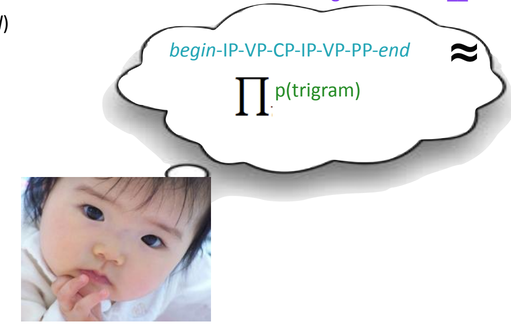
Any *wh*-dependency can then have a probability, based on the product of the smoothed probabilities of its trigrams.



Lidz & Gagliardi 2015

Who did she think the gift was from __?

Probability(*begin*-IP-VP-CP-IP-VP-PP-*end*)
= p(*begin*-IP-VP)
p(IP-VP-CP)
p(VP-CP-IP)
p(CP-IP-VP)
p(IP-VP-PP)
p(VP-PP-*end*)

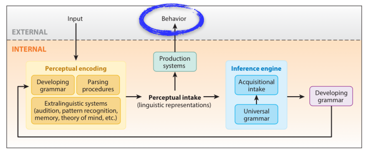


Pearl & Sprouse 2013a, 2013b, 2015

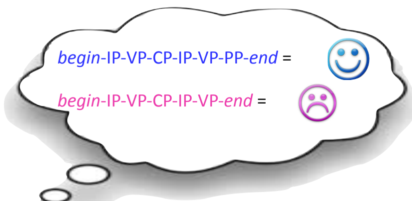
Subjacency-ish: Developing knowledge

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.



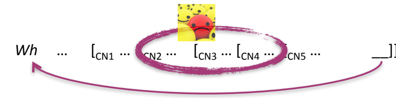
Lidz & Gagliardi 2015

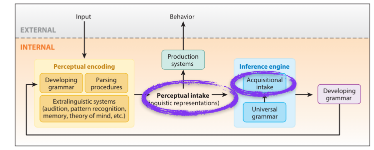


Pearl & Sprouse 2013a, 2013b, 2015

The Subjacency-ish strategy: Data intake

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





Lidz & Gagliardi 2015

Data intake: defined by initial state =

wh-dependencies in child-directed speech, as characterized by container nodes

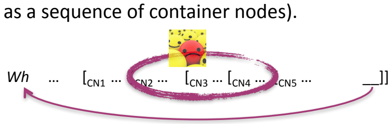
But which *wh*-dependencies? Just the ones being evaluated in the target state?

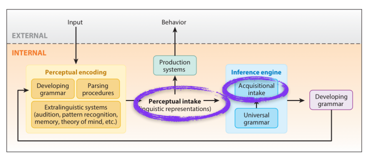
Who __ claimed that Lily forgot the necklace?	matrix | non-island
What did the teacher claim that Lily forgot __?	embedded | non-island
Who __ made the claim that Lily forgot the necklace?	matrix | island
*What did the teacher make the claim that Lily forgot __?	embedded | island

Pearl & Sprouse 2013a, 2013b, 2015

The Subjacency-ish strategy: Data intake

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





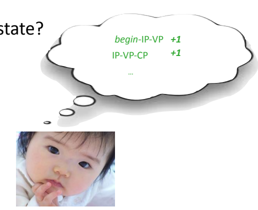
Lidz & Gagliardi 2015

Data intake: defined by initial state =

wh-dependencies in child-directed speech, as characterized by container nodes

But which *wh*-dependencies? Just the ones being evaluated in the target state?

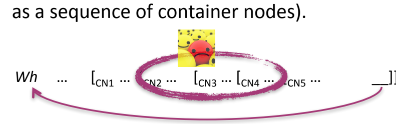
No! Any *wh*-dependency has relevant information about container node trigrams used to determine the grammaticality of *wh*-dependencies in general.

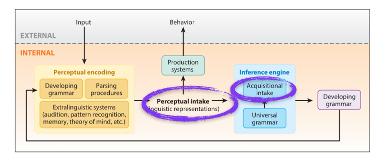


Pearl & Sprouse 2013a, 2013b, 2015

The Subjacency-ish strategy: Data intake

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





Lidz & Gagliardi 2015

Data intake: defined by initial state =

all *wh*-dependencies in child-directed speech, as characterized by container nodes

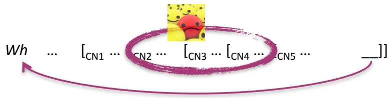
(Brown-Adam, Brown-Eve, Suppes, Valian) from CHILDES:
101,838 utterances containing 20,923 *wh*-dependencies

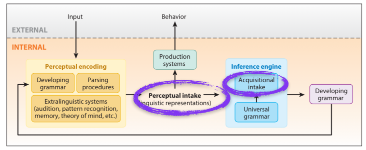
76.7%	*What did you see __?*
12.8%	*What __ happened?*
5.6%	*What did she want to do __?*
2.5%	*What did she read from __?*
1.1%	*What did she think he said __?*
...	

Pearl & Sprouse 2013a, 2013b, 2015

The Subjacency-ish strategy: Data intake

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





Lidz & Gagliardi 2015

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.

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

Pearl & Sprouse 2013a, 2013b, 2015

CHILDES Treebank

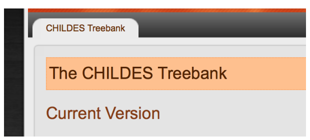
http://www.socsci.uci.edu/~lpearl/CoLaLab/CHILDESTreebank/childestreebank.html

The corpora currently included in the CHILDES Treebank are as follows:

All child-directed speech utterances

- Brown/Adam (**Includes trace-annotation)
- Brown/Eve (**Includes trace-annotation, ***also has version with additional animacy annotation and version with both animacy and thematic role annotation)
- Brown/Sarah
- Soderstrom
- Suppes
- Valian (**Includes trace-annotation)

Pearl & Sprouse 2013a, 2013b, 2015



CHILDES Treebank

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

From valian.parsed

```
(ROOT
  (SBARQ
    (WHNP-1
          (WP what))
    (SQ
          (VP
            (AUX 's)
        (NP
                  (PRP it))
        (VP
                  (VBN got)
                  (NP
                    (-NONE-ABAR-WH- *T*-1))
                  (PP
                    (IN on)
                        (NP
                          (PRP it))))))
    (. ?)))
```

"What's it got __ on it?"

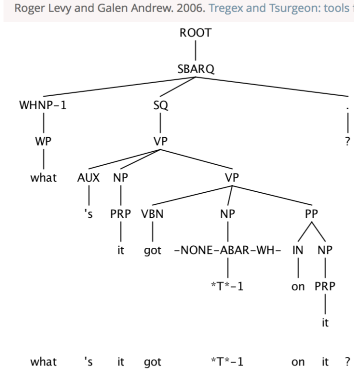
Pearl & Sprouse 2013a, 2013b, 2015

CHILDES Treebank

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

Using the Tregex visualization & query tool available at http://nlp.stanford.edu/software/tregex.shtml

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. .



"What's it got __ on it?"

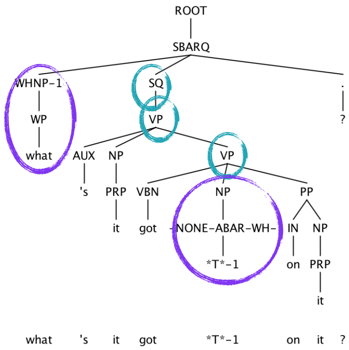
Pearl & Sprouse 2013a, 2013b, 2015

CHILDES Treebank

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

Using the Tregex visualization & query tool available at http://nlp.stanford.edu/software/tregex.shtml

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures.



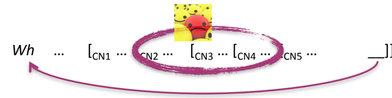
Note that while this can be helpful for extracting container node sequences, the labels still may not be exactly right. Some post-processing is necessary here. But it sure helps as a basis, instead of having to search text alone and annotate container node sequences by hand.

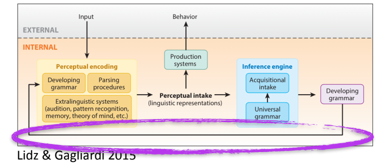
"What's it got __ on it?" = IP VP

Pearl & Sprouse 2013a, 2013b, 2015

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).





Lidz & Gagliardi 2015

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.

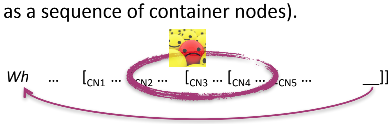
Let's assume, based on available experimental studies, that intuitions about syntactic islands are acquired in a three year period, such as between the ages of two and five.

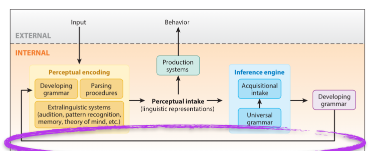
So, our modeled learner will get 1,000,000 utterances distributed similarly to the dependencies in the child-directed speech input samples from CHILDES.

Pearl & Sprouse 2013a, 2013b, 2015

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).





Lidz & Gagliardi 2015

Learning period:

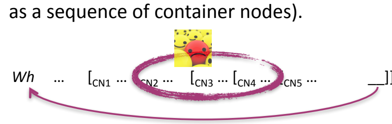
20% of the utterances in the child-directed speech sample were wh-dependencies, distributed this way (26 types total):

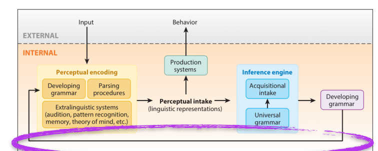
76.7%	*What did you see __?*
12.8%	*What __ happened?*
5.6%	*What did she want to do __?*
2.5%	*What did she read from __?*
1.1%	*What did she think he said __?*
…	

Pearl & Sprouse 2013a, 2013b, 2015

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).





Lidz & Gagliardi 2015

Learning period:

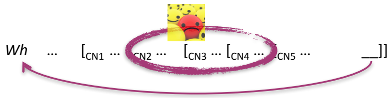
Therefore, the modeled learner heard 200,000 wh-dependencies distributed this way, encoded as container node sequences:

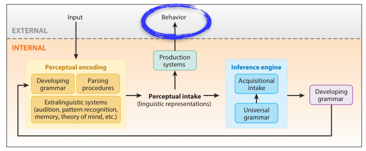
76.7%	IP-VP
12.8%	IP
5.6%	IP-VP-IP-VP
2.5%	IP-VP-PP
1.1%	IP-VP-CP-IP-VP
…	

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).



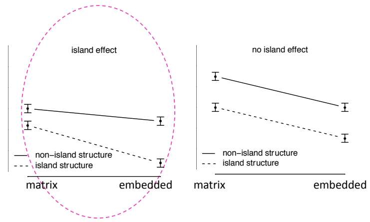


Lidz & Gagliardi 2015

Target state: Behavioral evidence of syntactic islands knowledge

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

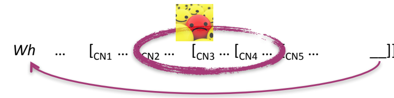
But how do we get acceptability judgment equivalents?

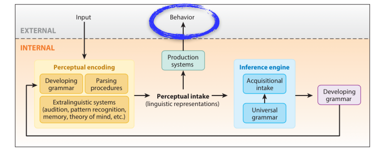


Pearl & Sprouse 2013a, 2013b, 2015

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).

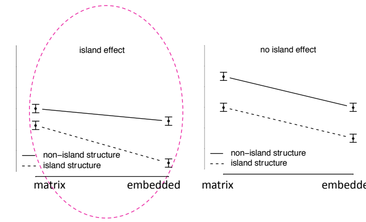




Lidz & Gagliardi 2015

Target state: Behavioral evidence of syntactic islands knowledge

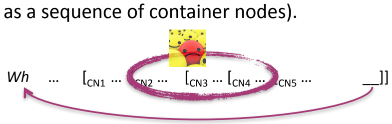
For each set of island stimuli from Sprouse et al. (2012), we generate grammaticality preferences for the modeled learner based on the dependency's perceived probability and use this as a stand-in for acceptability.

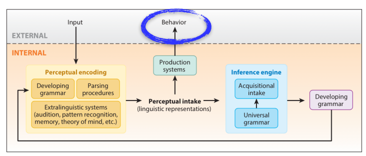


Pearl & Sprouse 2013a, 2013b, 2015

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).

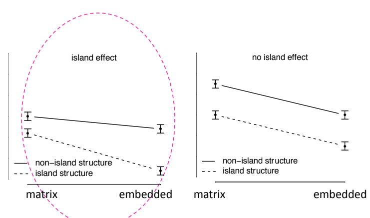




Lidz & Gagliardi 2015

Target state: Behavioral evidence of syntactic islands knowledge

	matrix	embedded
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

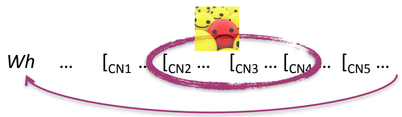
Let's take a break for a few minutes

Today's Plan

I. So you want to model language acquisition

II. Modeling case study: Defining the pieces

III. Modeling case study: Implementation & Interpretation



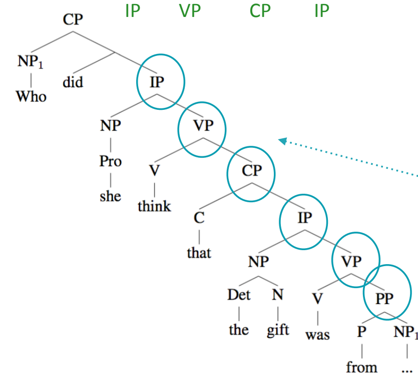
Details: What counts as a container node and why

Encoding a dependency as a sequence of container nodes.

Who did she think that the gift was from?

$[_{CP}$ Who did $[_{IP}$ she $[_{VP}$ think $[_{CP}$ that $[_{IP}$ $[_{NP}$ the gift] $[_{VP}$ was $[_{PP}$ from __]]]]]]]]?

IP VP CP IP VP PP



What phrase structure nodes should children pay attention to?

This is only one option.

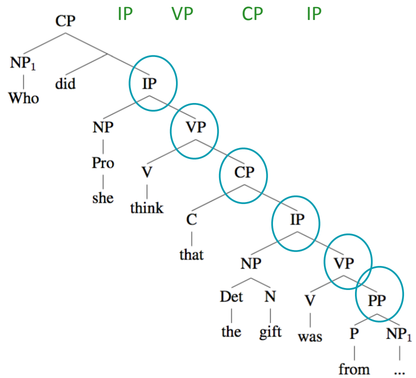
Details: What counts as a container node and why

Encoding a dependency as a sequence of container nodes.

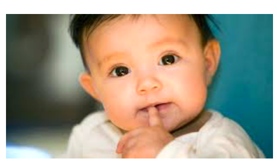
Who did she think that the gift was from?

$[_{CP}$ Who did $[_{IP}$ she $[_{VP}$ think $[_{CP}$ that $[_{IP}$ $[_{NP}$ the gift] $[_{VP}$ was $[_{PP}$ from __]]]]]]]]?

IP VP CP IP VP PP



Pearl & Sprouse: Maybe we should start with "basic" phrase structure nodes (typically associated with a lexical head: CP, IP, VP, AdjP, PP, etc.).



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:

Complex NP islands

begin-IP-*end* — matrix | non-island
begin-IP-VP-CP-IP-VP-*end* — embedded | non-island
begin-IP-*end* — matrix | island
**begin*-IP-VP-NP-CP-IP-VP-*end* — embedded | island

Subject islands

begin-IP-*end*
begin-IP-VP-CP-IP-*end*
begin-IP-*end*
**begin*-IP-VP-CP-IP-NP-PP-*end*

✓ All the ungrammatical dependencies are distinct from all 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:

Whether islands

begin-IP-*end* — matrix | non-island
begin-IP-VP-CP-IP-VP-*end* — embedded | non-island
begin-IP-*end* — matrix | island
**begin*-IP-VP-CP-IP-VP-*end* — embedded | island

Adjunct islands

begin-IP-*end*
begin-IP-VP-CP-IP-VP-*end*
begin-IP-*end*
**begin*-IP-VP-CP-IP-VP-*end*

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:

Whether islands

begin-IP-*end* — matrix | non-island
begin-IP-VP-CP-IP-VP-*end* — embedded | non-island
begin-IP-*end* — matrix | island
**begin*-IP-VP-CP-IP-VP-*end* — embedded | island

Adjunct islands

begin-IP-*end*
begin-IP-VP-CP-IP-VP-*end*
begin-IP-*end*
**begin*-IP-VP-CP-IP-VP-*end*

This means there's no possible way to get these judgments right using this representation. Uh oh!

Details: What counts as a container node and why

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}, etc.

The learner can then distinguish between these structures:

IP-VP-$CP_{null/that}$-IP-VP
IP-VP-$CP_{whether/if}$-IP-VP

Details: What counts as a container node and why

Can the grammatical dependencies be distinguished from the ungrammatical ones?

Sprouse et al. (2012) stimuli:

Complex NP islands | **Subject islands**

Complex NP islands		Subject islands
begin-IP-*end*	matrix | non-island	*begin*-IP-*end*
begin-IP-VP-CP$_{that}$-IP-VP-*end*	embedded | non-island	*begin*-IP-VP-CP$_{null}$-IP-*end*
begin-IP-*end*	matrix | island	*begin*-IP-*end*
**begin*-IP-VP-NP-CP$_{that}$-IP-VP-*end*	embedded | island	**begin*-IP-VP-CP$_{null}$-IP-NP-PP-*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

Can the grammatical dependencies be distinguished from the ungrammatical ones?

Sprouse et al. (2012) stimuli:

Whether islands		Adjunct islands
begin-IP-*end*	matrix | non-island	*begin*-IP-*end*
begin-IP-VP-CP$_{that}$-IP-VP-*end*	embedded | non-island	*begin*-IP-VP-CP$_{that}$-IP-VP-*end*
begin-IP-*end*	matrix | island	*begin*-IP-*end*
**begin*-IP-VP-CP$_{whether}$-IP-VP-*end*	embedded | island	**begin*-IP-VP-CP$_{if}$-IP-VP-*end*

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

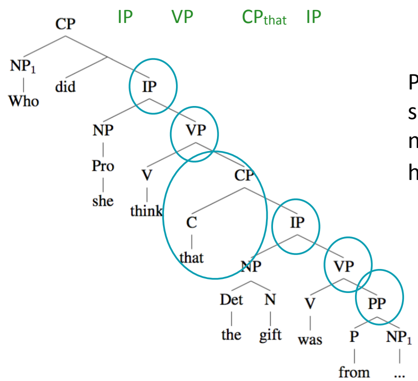
Details: What counts as a container node and why

Encoding a dependency as a sequence of container nodes.

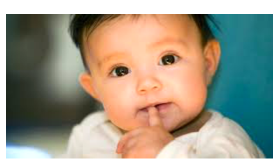
Who did she think that the gift was from?

[$_{CP}$ Who did [$_{IP}$ she [$_{VP}$ think [$_{CP}$ that [$_{IP}$ [$_{NP}$ the gift] [$_{VP}$ was [$_{PP}$ from __]]]]]]]]?

IP VP CP$_{that}$ IP VP PP



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



Details: 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.

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

Details: 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.

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

A small problem: Trigrams we never observe have a frequency of 0. This then yields a probability of 0. In general, we prefer not to assign 0 probabilities — what if whatever it is is simply very rare? It's better to allow a very small probability for things not yet observed.

Details: 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.

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

Allowing a very small default probability is known as "smoothing".

Details: 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.

One way to do smoothing: Lidstone's Law

$$\frac{\text{total observations of } t + \alpha}{\text{total observations of all N trigrams} + N\alpha}$$

Allowing a very small default probability is known as "smoothing".

Details: 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.

One way to do smoothing: Lidstone's Law

$$\frac{\text{total observations of } t + \alpha}{\text{total observations of all N trigrams} + N\alpha}$$

Allowing a very small default probability is known as "smoothing".

A number less than 1 (ex: $\alpha = 0.5$) is added to all counts. This means all N trigrams have α added to them (that's why Nα 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).

Details: 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.

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

Allowing a very small default probability is known as "smoothing".

(ex: $\alpha = 0.5$)

Details: 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.

begin-IP-VP = $(86+\alpha)/(225+100\alpha)$	p(*begin*-IP-VP) = 0.31
IP-VP-*end* = $(83+\alpha)/(225+100\alpha)$	p(IP-VP-*end*) = 0.30
begin-IP-*end* = $(13+\alpha)/(225+100\alpha)$	p(*begin*-IP-*end*) = 0.05
IP-VP-IP = $(6+\alpha)/(225+100\alpha)$	p(IP-VP-IP) = 0.02
VP-IP-VP = $(6+\alpha)/(225+100\alpha)$	p(VP-IP-VP) = 0.02
IP-VP-PP = $(3+\alpha)/(225+100\alpha)$	p(IP-VP-PP) = 0.01
VP-PP-*end* = $(3+\alpha)/(225+100\alpha)$	p(VP-PP-*end*) = 0.01
…	
IP-NP-PP = $(0+\alpha)/(225+100\alpha)$	p(IP-NP-PP) = 0.002

if we had 100 trigram types total

Allowing a very small default probability is known as "smoothing".

(ex: $\alpha = 0.5$)

Details: 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.

How does this part work?

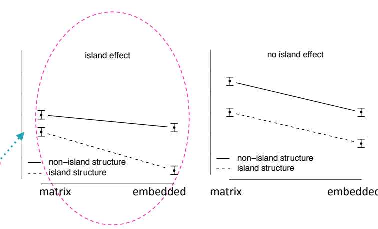
p(IP-VP) = 0.14

p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00000000001

Details: Subjacency-ish strategy in more detail

	matrix	embedded
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 __?*



We want to get numbers we can plot on an interaction plot.

(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?

p(IP-VP) = 0.14

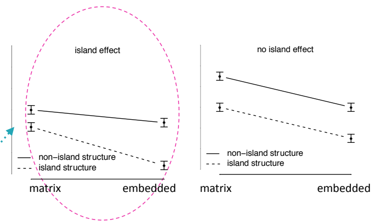
p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00000000001

Details: Subjacency-ish strategy in more detail

Let's use log probabilities:

(1) They're easier to compare visually, especially when the probabilities are very, 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?

p(IP-VP) = 0.14

p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00000000001

Details: Subjacency-ish strategy in more detail

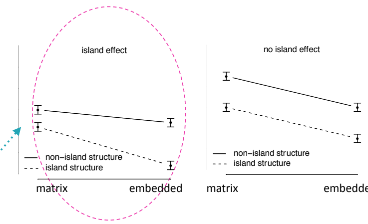
Let's use log probabilities:

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

$\log_{10}(0.14) = -0.85$

$\log_{10}(0.000000034) = -7.46$

$\log_{10}(0.00000000001) = -11.0$



(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?

p(IP-VP) = 0.14

p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00000000001

Details: Subjacency-ish strategy in more detail

Let's use log probabilities:

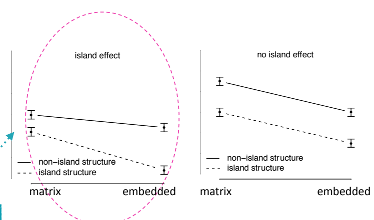
(2) The integer lets us quickly compare the order of magnitude in difference.

$\log_{10}(0.14) = -0.85$

$\log_{10}(0.000000034) = -7.46$

$\log_{10}(0.00000000001) = -11.0$

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



(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?

p(IP-VP) = 0.14

p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00000000001

Details: Subjacency-ish strategy in more detail

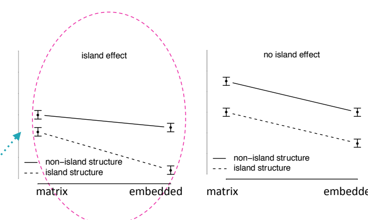
Let's use log probabilities:

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

$\log_{10}(0.000000034 * 0.00000000001) =$

$\log_{10}(0.000000034) + \log_{10}(0.00000000001)$

$-7.46 + -11.0 = -18.46$



(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?

p(IP-VP) = 0.14

p(IP-VP-IP-VP-PP) = 0.000000034

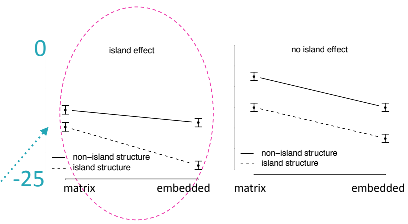
p(IP-VP-CP-IP-NP-PP) = 0.00000000001

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?

p(IP-VP) = 0.14

p(IP-VP-IP-VP-PP) = 0.000000034

p(IP-VP-CP-IP-NP-PP) = 0.00000000001

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)

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

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

```
IP-VP   0.5
IP      0.5
IP-VP-NP-CP_null-IP-VP  0.5
IP-VP-NP-CP_that-IP-VP  0.5
IP-VP-CP_null-IP-VP             0.5
IP-VP-CP_that-IP-VP             0.5
```

Which dependencies to generate probabilities for

Dependency frequency in the input

```
IP      2680
IP-VP   16039
IP-VP-ADJP-IP-VP        0
IP-VP-ADJP-IP-VP-PP     0
IP-VP-ADJP-PP           0
IP-VP-CP_null-IP        24
IP-VP-CP_null-IP-VP     236
```

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

```
#############################
##### main body of program #####
#############################
# (1) read in chain information from $inputfile
#       (a) initialize data structure that holds chains, based on chain nodes extracted
#            (should initialize with smoothing constant, rather than all 0s)
#       (b) print out list of chain nodes extracted from chains
# (2) read in chains to track from $chainfile
#       (a) initialize data structure that contains probabilities of chains to track
my ($ref_arr_learner_input, $ref_hash_node_types, $ref_hash_chains_to_track) = read_chain
print_nodes($ref_hash_node_types);
my ($ref_trigrams, $total_trigrams, $total_tri_observations) = initialize_trigrams($ref_h
my %observed_chains; # used in updating probabilities
my $num_obs_chains = 0; # used in updating probabilities
```

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

```
Options selected:
print_incremental = yes
print_interval = 25000
inputfile = input/child-all.countsCP
chainfile = input/child-all.chainsCP
chain_separator = -
learner_runs = 1
data_points_to_run = 200000
smoothing_constant = 0.5
alpha = 1
beta = 1
Node types learner is using (extracted from input file):
ADJP
CP_after
CP_for
CP_null
CP_that
CP_whether
IP
NP
PP
VP
```

Information useful for analysis

Probabilities over time

```
debug: chain gen probabilities over time
chain:  0       25000   50000   75000   100000  125000  150000  175000  200000
IP      0.000826446280991736    0.0599188056077248      0.0615245754455672      0.0614775792904201
96      0.0610480695614131      0.0609562935013291
IP-VP   6.83013455365071e-07    0.164263256618967       0.16569359342645        0.166288957803426
9       0.166750550672405       0.16685112868796
IP-VP-CP_after-IP-VP    3.85543289429532e-16    1.40183610522217e-16    1.80468152683841e-17    5.392
6.76533446635375e-19    4.26119252504539e-19    2.85677163542943e-19
IP-VP-CP_null-IP        4.66507380209733e-13    1.03698307735959e-08    1.07071261408854e-08    1.136
1.31909055172949e-08    1.32101607826206e-08    1.29112680725489e-08
```

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)

Things to consider when implementing a model

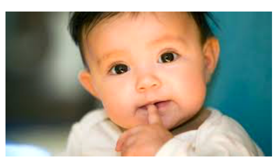
Now back to what programming language you should use…

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.



Things to consider when implementing a model

Now back to what programming language you should use…

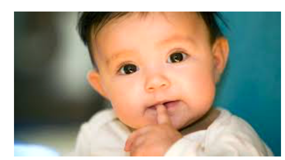
This depends a lot on

(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.



Model results & interpretation

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

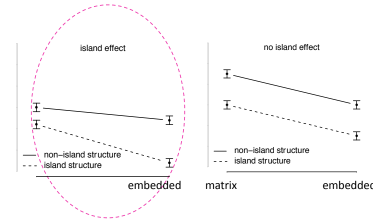
Now what?

Model results & interpretation

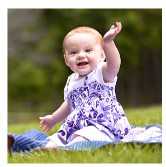
Now we need to link them back to the target state we're interested a little more precisely. This way, other people can understand what our results *mean*.

```
debug: chain gen probabilities over time
chain:  0       25000   50000   75000   100000  125000  150000  175000  200000
IP      0.000826446280991736    0.0599188056077248      0.0615245754455672      0.061477579290420196
96      0.0610480695614131      0.0609562935013291
IP-VP   6.83013455365071e-07    0.164263256618967       0.16569359342645        0.166288957803426
9       0.166750550672405       0.16685112868796
IP-VP-CP_after-IP-VP    3.85543289429532e-16    1.40183610522217e-16    1.80468152683841e-17    5.392
6.76533446635375e-19    4.26119252504539e-19    2.85677163542943e-19
IP-VP-CP_null-IP        4.66507380209733e-13    1.03698307735959e-08    1.07071261408854e-08    1.136
1.31909055172949e-08    1.32101607826206e-08    1.29112680725489e-08
```

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



Model results & interpretation



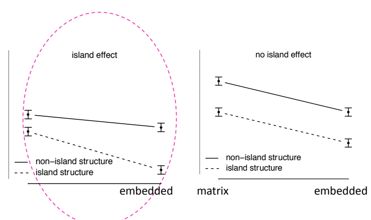
Complex NP islands

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 __?*

matrix | embedded

Each island type had four stimuli dependencies in Sprouse et al. (2012).



Model results & interpretation

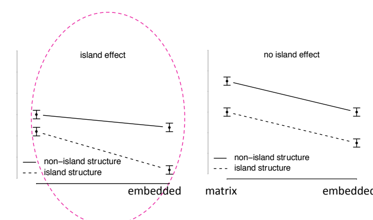
Complex NP islands

non-island  *IP*  *IP-VP-CP$_{that}$-IP-VP*

island  *IP*  **IP-VP-NP-CP$_{that}$-IP-VP*

matrix | embedded

Each of those is characterized by a sequence of container nodes.

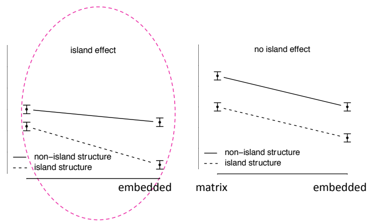


Model results & interpretation

Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

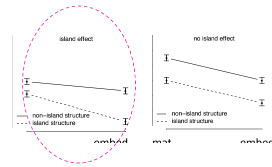
		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

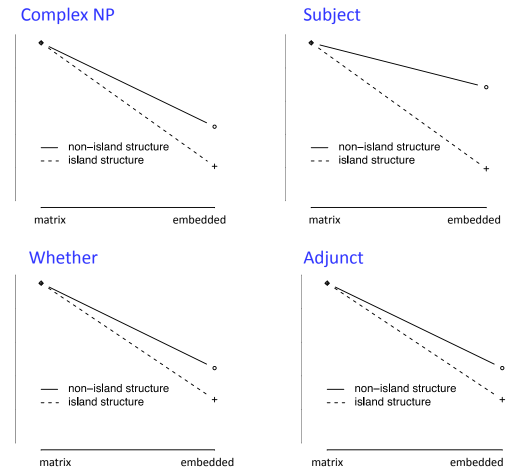
We can get the log probability of all these dependencies and then plot them on interaction plots.



Model results & interpretation

When we compare this against the desired target behavior…



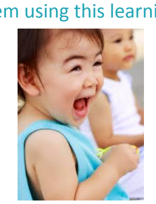


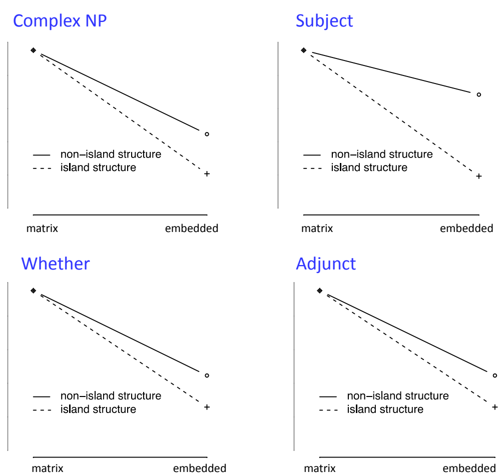
Pearl & Sprouse 2013a, 2013b, 2015

Model results & interpretation

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.

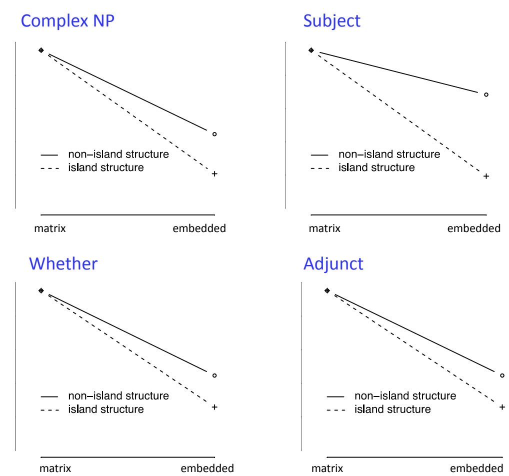




Pearl & Sprouse 2013a, 2013b, 2015

Model results & interpretation

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.

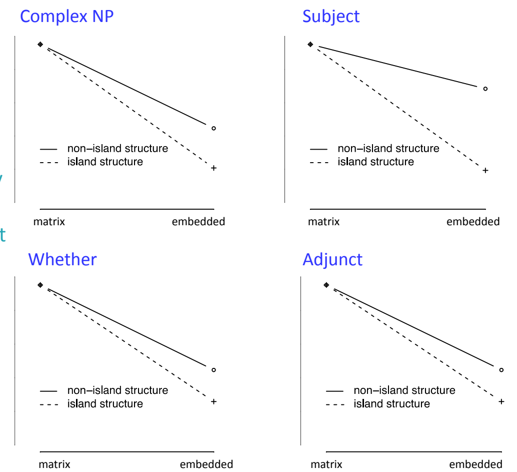


Pearl & Sprouse 2013a, 2013b, 2015

Model results & interpretation

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.)



Pearl & Sprouse 2013a, 2013b, 2015

Model results & interpretation

What's going on?

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

Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

Model results & interpretation

What's going on?

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

Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

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 [$_{IP}$ the teacher [$_{VP}$ make [$_{NP}$ the claim $_{CP_{that}}$ that [$_{IP}$ Lily $_{VP}$ forgot __]]]]]?
(ii) *start*-IP-VP-NP-CP_{that}-IP-VP-*end*
(iii) Low probability:
VP-NP-CP_{that}
NP-CP_{that}-IP

Model results & interpretation

What's going on?

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

Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

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.

b. Subject
(i) * Who does [$_{IP}$ Jack [$_{VP}$ think [$_{CP_{null}}$ [$_{IP}$ [$_{NP}$ the necklace [$_{PP}$ for __]] is expensive]]]]?
(ii) *start*-IP-VP-CP_{null}-IP-NP-PP-*end*
(iii) Low probability:
CP_{null}-IP-NP

Model results & interpretation

What's going on?

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

Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

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 [$_{IP}$ the teacher [$_{VP}$ wonder [$_{CP_{whether}}$ whether [$_{IP}$ Jack [$_{VP}$ stole __]]]]]?
(ii) *start*-IP-VP-$CP_{whether}$-IP-VP-*end*
(iii) Low probability:
IP-VP-$CP_{whether}$
VP-$CP_{whether}$-IP
$CP_{whether}$-IP-VP

Model results & interpretation

What's going on?

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

Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

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 [$_{IP}$ the teacher [$_{VP}$ worry [$_{CP_{if}}$ if [$_{IP}$ Lily [$_{VP}$ forgot __]]]]]?
(ii) *start*-IP-VP-CP_{if}-IP-VP-*end*
(iii) Low probability:
IP-VP-CP_{if}
VP-CP_{if}-IP
CP_{if}-IP-VP

Model results & interpretation

What's going on?

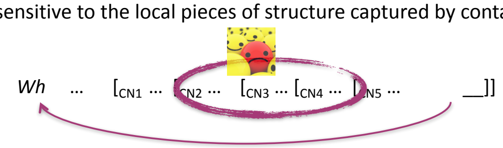
Why are the island-spanning dependencies so much worse than the grammatical ones?

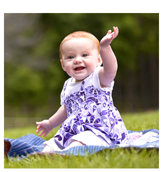
Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

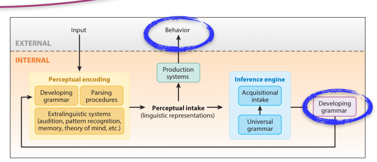
		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

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.







Lidz & Gagliardi 2015

Model results & interpretation

What's going on?

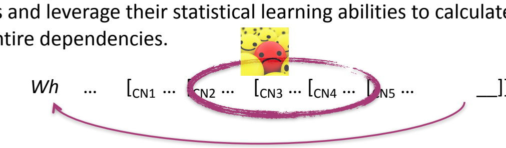
Why are the island-spanning dependencies so much worse than the grammatical ones?

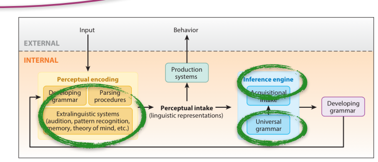
Table 4. Inferred grammaticality of different *wh*-dependencies from Sprouse et al. (2012a), represented with log probability.

		Child-directed speech	Adult-directed speech & text
Grammatical dependencies			
matrix subject	IP	-1.21	-0.93
embedded subject	IP-VP-CP_{null}-IP	-7.89	-7.67
embedded object	IP-VP-CP_{that}-IP-VP	-13.84	-11.00
Island-spanning dependencies			
Complex NP	IP-VP-NP-CP_{that}-IP-VP	-19.81	-18.93
Subject	IP-VP-CP_{null}-IP-NP-PP	-20.17	-20.36
Whether	IP-VP-$CP_{whether}$-IP-VP	-18.54	-18.46
Adjunct	IP-VP-CP_{if}-IP-VP	-18.54	-18.46

Big picture

Acquisition: To learn using this representation, children need to be able to parse utterances into container node trigrams and leverage their statistical learning abilities to calculate probabilities of trigram pieces and entire dependencies.





Big picture debate: What's in UG?

Subjacency-ish

Wh … [$_{CN1}$ … [$_{CN2}$ … [$_{CN3}$ … [$_{CN4}$ … [$_{CN5}$ … __]]

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

UG = innate + domain-specific

	Innate	Derived	Domain-specific	Domain-general
Attend to container nodes of a particular kind	?	?	*	
Low probability items are dispreferred	*			*

Subjacency

Wh … [$_{BN1}$ … [$_{BN2}$ … __]]

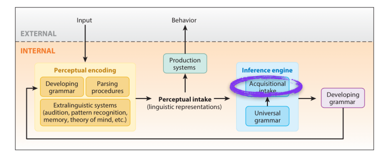
	Innate	Derived	Domain-specific	Domain-general
Attend to bounding nodes (BNs)	*		*	
Dependencies crossing 2+ BNs are not allowed	*		*	

Pearl & Sprouse 2013a, 2013b, 2015

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



Lidz & Gagliardi 2015

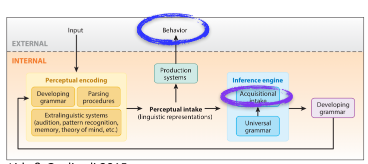
Pearl & Sprouse 2013a, 2013b, 2015

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



Lidz & Gagliardi 2015

Pearl & Sprouse 2013a, 2013b, 2015

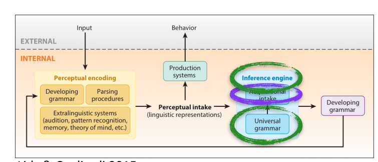
Big picture: Syntactic islands

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



Lidz & Gagliardi 2015

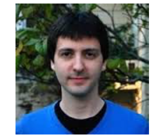
Pearl & Sprouse 2013a, 2013b, 2015

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



Audiences at: Input and Syntactic Acquisition 2012, UMaryland Mayfest 2012, New York University Linguistics colloquium 2012, GALANA 2015, UCLA Linguistics colloquium 2015

Computation of Language Laboratory

UC Irvine

Lisa S. Pearl
Associate Professor
Department of Cognitive Sciences
SBSG 2314
University of California, Irvine
lpearl@uci.edu

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