Empirically investigating the Universal Grammar hypothesis

Lisa Pearl
University of California, Irvine

Nov 16, 2012: Department of Linguistics Colloquium
New York University
Motivating Universal Grammar

One explicit motivation: The argument from acquisition

Motivating Universal Grammar

Specifically, Universal Grammar consists of the necessary learning biases that are both innate and domain-specific (Chomsky 1965, Chomsky 1975).
Motivating Universal Grammar

What’s so hard about acquiring language?
There seem to be induction problems, given the available data.
(Poverty of the Stimulus, Logical Problem of Language Acquisition, Plato’s Problem)
Motivating the contents of UG

Proposals have traditionally come from characterizing a specific induction problem pertaining to a particular linguistic phenomenon, and describing the (UG) solution to that specific characterization.
Motivating the contents of UG

Proposals have traditionally come from characterizing a specific induction problem pertaining to a particular linguistic phenomenon, and describing the (UG) solution to that specific characterization.

- Structure-dependent rules (Chomsky 1980)

  Pirates who can dance can often fight well.

  Can pirates who can dance ___ often fight well?
Motivating the contents of UG

Proposals have traditionally come from characterizing a specific induction problem pertaining to a particular linguistic phenomenon, and describing the (UG) solution to that specific characterization.

• Constraints on long-distance dependencies (Chomsky 1973)
  Where did Jack think Lily bought the necklace from __?
  *Where did Jack think the necklace from ___ was too expensive?
Motivating the contents of UG

Proposals have traditionally come from characterizing a specific induction problem pertaining to a particular linguistic phenomenon, and describing the (UG) solution to that specific characterization.

- English anaphoric one representation (Baker 1978)
  Look – a red bottle! Do you see another one? one = ?
Motivating the contents of UG

Benefits of a specific characterization of an induction problem:

- Precisely describe a potential solution
- Explicitly test that solution & compare it to other potential solutions
Motivating the contents of UG

Benefits of a specific characterization of an induction problem:
- Precisely describe a potential solution
- Explicitly test that solution & compare it to other potential solutions

When we find a potential solution, we can examine the nature of the learning biases it involves.
Motivating the contents of UG

Benefits of a specific characterization of an induction problem:

- Precisely describe a potential solution
- Explicitly test that solution & compare it to other potential solutions

Benefits for investigating UG:

- If all the solutions involve UG biases:
  - supports the existence of UG
  - provides specific proposals for its contents
Motivating the contents of UG

Benefits of a specific characterization of an induction problem:

- Precisely describe a potential solution
- Explicitly test that solution & compare it to other potential solutions

Benefits for investigating UG:

- If \textit{all} the solutions involve UG biases:
  - supports the existence of UG
  - provides specific proposals for its contents

- If \textit{some solutions do not} involve UG biases
  - takes away the support for UG that comes from that characterization of the induction problem
Characterizing induction problems

Initial state:
Characterizing induction problems

Initial state:
- initial knowledge state
  - ex: grammatical categories exist and can be identified
  - ex: phrase structure exists and can be identified

\[ N^0, N', NP, DP, \ldots \]
Characterizing induction problems

Initial state:

- **Initial knowledge state**
  ex: grammatical categories exist and can be identified
  ex: phrase structure exists and can be identified

- **Learning biases & capabilities**
  ex: frequency information can be tracked
  \[ N^0 = N^0 + 1 \]
  ex: distributional information can be leveraged

\[ N^0, N', NP, DP, ... \]
Characterizing induction problems

**Initial state:** initial knowledge state + learning biases & capabilities

**Data intake:**
Characterizing induction problems

**Initial state**: initial knowledge state + learning biases & capabilities

**Data intake:**
- data perceived as relevant for learning (Fodor 1998)
  - ex: all *wh*-utterances for learning about *wh*-dependencies
  - ex: syntactic data for learning syntactic knowledge
[defined by knowledge & biases/capabilities in the initial state]

*Pearl & Mis submitted*
Characterizing induction problems

Initial state: initial knowledge state + learning biases & capabilities

Data intake: data perceived as relevant for learning

Learning period:
Characterizing induction problems

**Initial state:** initial knowledge state + learning biases & capabilities

**Data intake:** data perceived as relevant for learning

**Learning period:**
- 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

*Pearl & Mis submitted*
Characterizing induction problems

**Initial state:** initial knowledge state + learning biases & capabilities

**Data intake:** data perceived as relevant for learning

**Learning period:** how long children have to learn

**Target state:**
Characterizing induction problems

Initial state: initial knowledge state + learning biases & capabilities

Data intake: data perceived as relevant for learning

Learning period: how long children have to learn

Target state:
- the knowledge children are trying to attain
  ex: *Where did Jack think the necklace from ___ was too expensive?*
  ex: *one* is category N’ when it is not NP

Pearl & Mis submitted
Characterizing induction problems

**Initial state:** initial knowledge state + learning biases & capabilities

**Data intake:** data perceived as relevant for learning

**Learning period:** how long children have to learn

**Target state:** the knowledge children must attain

*Pearl & Mis submitted*
Characterizing induction problems

**Initial state:** initial knowledge state + learning biases & capabilities

**Data intake:** data perceived as relevant for learning

**Learning period:** how long children have to learn

**Target state:** the knowledge children must attain

**Induction problem:**
Given a specific initial state, data intake, and learning period, the target state is not the only knowledge state that could be reached.

*Pearl & Mis submitted*
To characterize potential induction problems, we need to draw on a variety of research methods.
To characterize potential induction problems, we need to draw on a variety of research methods.

**Theoretical methods:**

*What* knowledge of language is (and what children have to learn)

[initial state, target state]

SEE the Kitty

si ḏe kiri

---

**Theoretical methods:**

*What* knowledge of language is (and what children have to learn)

[initial state, target state]

SEE the Kitty

see’(the kitty)(x_{listener})
To characterize potential induction problems, we need to draw on a variety of research methods.

Experimental methods:

**When** knowledge is acquired, what the **input** looks like, & plausible capabilities underlying **how** acquisition works

[initial state, data intake, learning period]

\[
p(ki \mid tty) \propto p(H1 \mid \text{tty}) p(H1)
\]

\[
p(ki) \propto p(H1 \mid \text{tty}) p(H1)
\]
To characterize potential induction problems, we need to draw on a variety of research methods.

**Computational methods:**
Strategies that are both useful and useable for **how** children acquire knowledge + quantitative analysis of input [initial state, data intake]
I. Potential induction problem:
   Learning constraints on long-distance dependencies

II. Potential induction problem:
   Learning English anaphoric one
Road Map

I. Potential induction problem:
Learning constraints on long-distance dependencies

II. Potential induction problem:
Learning English anaphoric *one*
Syntactic islands

• **Why**? Central to UG-based syntactic theories.

• **What**? Dependencies can exist between two non-adjacent items. They do not appear to be constrained by length (Chomsky 1965, Ross 1967), but rather by whether the dependency crosses certain structures (called “syntactic islands”).

*Pearl & Sprouse forthcoming*
Syntactic islands

- **Why?** Central to UG-based syntactic theories.

- **What?** Dependencies can exist between two non-adjacent items. They do not appear to be constrained by length (Chomsky 1965, Ross 1967), but rather by whether the dependency crosses certain structures (called “syntactic islands”).

What does Jack think ___?

What does Jack think that Lily said that Sarah heard that Jareth believed ___?
Syntactic islands

• **Why?** Central to UG-based syntactic theories.

• **What?** Dependencies can exist between two non-adjacent items. They do not appear to be constrained by length (Chomsky 1965, Ross 1967), but rather by whether the dependency crosses certain structures (called “syntactic islands”).

Some example islands

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 forthcoming*
Syntactic islands

• **Predominant theory in generative syntax:** syntactic islands require *innate, domain-specific* learning biases

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

- **Predominant theory in generative syntax:**
syntactic islands require **innate, domain-specific** learning biases


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

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

\[ Wh \ldots [BN_2 \ldots [BN_1 \ldots \_]] \]

\{CP, IP, NP\}?
Syntactic islands

- Predominant theory in generative syntax:
  syntactic islands require innate, domain-specific learning biases...in addition to whatever else they might require.

```
?                     domain-specific
```

```
derived               |
```

```
domain-general        |
```

```
innate                |
```

Not 2+ bounding nodes (BNs)
BN = \{CP, IP, NP\}

Pearl & Sprouse forthcoming
Syntactic islands

• How do we test this?

(1) Explicitly define the target knowledge state, using adult acceptability judgments.

(2) Identify the data available in the input, using realistic samples. (Is there an induction problem, given what we think children’s data intake is?)

(3) Implement a probabilistic learner that can learn about syntactic islands and see what kind of learning biases it requires. This requires making the initial state and learning period explicit.
The target state: 
Adult knowledge of syntactic islands

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)
The target state: Adult knowledge of syntactic islands

Sprouse et al. (2012) collected magnitude estimation judgments for four different islands, using a factorial definition that controlled for two salient properties of island-crossing dependencies:

- length of dependency (matrix vs. embedded)
- presence of an island structure (non-island vs. island)

Complex NP islands

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 forthcoming
The target state: 
Adult knowledge of syntactic islands

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)

Subject islands

Who ___ thinks the necklace is expensive?  
What does Jack think ___ is expensive?  
Who ___ thinks the necklace for Lily is expensive?  
*Who does Jack think the necklace for ___ is expensive?  

matrix | non-island  
embedded | non-island  
matrix | island  
embedded | island

Pearl & Sprouse forthcoming
The target state: Adult knowledge of syntactic islands

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)

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 forthcoming*
The target state:
Adult knowledge of syntactic islands

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)

Adjunct islands

Who ___ thinks that Lily forgot the necklace?  
*What does the teacher think that Lily forgot ___?*  
Who ___ worries if Lily forgot the necklace?  
*What does the teacher worry if Lily forgot ___?*
The target state:
Adult knowledge of syntactic islands

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).

Pearl & Sprouse forthcoming
The target state:
Adult knowledge of syntactic islands

Sprouse et al. (2012)’s data on the four island types (173 subjects)

Superadditivity present for all islands tested

= Knowledge that dependencies cannot cross these island structures is part of the adult knowledge state

Pearl & Sprouse forthcoming
Characterizing the induction problem: 
Syntactic islands

target state: knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

Pearl & Sprouse forthcoming
The data in the input

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

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

Sprouse et al. (2012) stimuli types:

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

*wh*-dependency rarity
These kinds of utterances are fairly rare in general - the most frequent appears about 0.9% of the time (295 of 31,247).

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

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

*Pearl & Sprouse forthcoming*
The data in the input

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

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

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

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

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

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

Pearl & Sprouse forthcoming
The data in the input

...and impossible to tell no matter what the rest of the time.

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

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

If children are relying only on direct evidence and keying grammaticality directly to frequency, this looks like an induction problem.

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

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

Pearl & Sprouse forthcoming
Characterizing the induction problem: Syntactic islands

initial state:
   Bias: Learn only from direct evidence.

data intake: examples of specific \textit{wh}-dependencies in the input

target state: knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

\textit{Pearl \& Sprouse forthcoming}
Building a computational learner

Idea: Use indirect positive evidence, too.

Similar in spirit to linguistic parameters: Data are deemed informative, even if they are not data about the specific phenomenon of interest.

Here: Dependencies other than the ones of interest (the Sprouse et al. 2012 stimuli) are useful to learn from.
Characterizing the induction problem: Syntactic islands

initial state:

-Bias: Learn only from direct evidence.

+Bias: Learn from both direct and indirect evidence coming from \textit{wh}-dependencies.

data intake: all \textit{wh}-dependencies in the input

target state: knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

Pearl & Sprouse forthcoming
Building a computational learner

Learning Bias: Children track the occurrence of structures that can be derived from phrase structure trees during parsing - container nodes.

\[
[{_{CP} \textbf{Who} \text{ did} \ [_{IP} \text{ she} \ [_{VP} \text{ like} \ \_\_\_\_]]}]?
\]

\[
_{IP} \quad _{VP}
\]

Container node sequence: IP-VP

\[
[{_{CP} \textbf{Who} \text{ did} \ [_{IP} \text{ she} \ [_{VP} \text{ think} \ [_{CP} \ [_{IP} \text{ the gift} \ [_{VP} \text{ was} \ [_{PP} \text{ from} \ \_\_\_\_]]]]]]]]
\]

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

Container node sequence: IP-VP-CP-IP-VP-PP

*Pearl & Sprouse forthcoming*
Building a computational learner

Children’s hypotheses are about what container node sequences are grammatical for dependencies in the language.

Pearl & Sprouse forthcoming
Characterizing the induction problem: Syntactic islands

initial state:
  Bias: Learn from both direct and indirect evidence coming from $wh$-dependencies.
  +Capability: Be able to parse data in the input into phrase structure trees.
  +Bias: Characterize dependencies as sequences of container nodes.

data intake: all $wh$-dependencies in the input

target state: knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

Pearl & Sprouse forthcoming
What does the target knowledge look like?

Sprouse et al. (2012) stimuli:

<table>
<thead>
<tr>
<th>Complex NP islands</th>
<th>Subject islands</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>matrix</td>
</tr>
<tr>
<td>IP-VP-CP-IP-VP</td>
<td>embedded</td>
</tr>
<tr>
<td>IP</td>
<td>matrix</td>
</tr>
<tr>
<td>*IP-VP-NP-CP-IP-VP</td>
<td>embedded</td>
</tr>
</tbody>
</table>

All the ungrammatical dependencies are distinct from all the grammatical dependencies for these syntactic islands.
What does the target knowledge look like?

Sprouse et al. (2012) stimuli:

<table>
<thead>
<tr>
<th>Whether islands</th>
<th>Adjunct islands</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>IP-VP-CP-IP-VP</td>
<td>IP-VP-CP-IP-VP</td>
</tr>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>*IP-VP-CP-IP-VP</td>
<td>*IP-VP-CP-IP-VP</td>
</tr>
</tbody>
</table>

Pearl & Sprouse forthcoming
What does the target knowledge look like?

Sprouse et al. (2012) stimuli:

<table>
<thead>
<tr>
<th>Whether islands</th>
<th>Adjunct islands</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>IP-VP-CP-IP-VP</td>
<td>IP-VP-CP-IP-VP</td>
</tr>
<tr>
<td>*IP-VP-CP-IP-VP</td>
<td>*IP-VP-CP-IP-VP</td>
</tr>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>matrix</td>
<td>matrix</td>
</tr>
<tr>
<td>embedded</td>
<td>embedded</td>
</tr>
<tr>
<td>non-island</td>
<td>non-island</td>
</tr>
<tr>
<td>island</td>
<td>island</td>
</tr>
</tbody>
</table>

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

Pearl & Sprouse forthcoming
Building a computational learner

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

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

The learner can then distinguish between these structures:

\[ \text{IP-VP-CP}_{\text{null/that}}-\text{IP-VP} \]
\[ \text{IP-VP-CP}_{\text{whether/if}}-\text{IP-VP} \]

Pearl & Sprouse forthcoming
What does the target knowledge look like?

Sprouse et al. (2012) stimuli:

Complex NP islands                                             Subject islands

IP                          matrix | non-island                       IP                          
IP-VP-CP_{that}-IP-VP       embedded | non-island                       IP-VP-CP_{null}-IP
IP                          matrix | island                           IP                          
*IP-VP-NP-CP_{that}-IP-VP   embedded | island                           *IP-VP-CP_{null}-IP-NP-PP

All the ungrammatical dependencies are still distinct from all the grammatical dependencies for these syntactic islands.
What does the target knowledge look like?

Sprouse et al. (2012) stimuli:

<table>
<thead>
<tr>
<th>Whether islands</th>
<th>Adjunct islands</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>IP-VP-CP_{that} - IP-VP</td>
<td>IP-VP-CP_{that} - IP-VP</td>
</tr>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>*IP-VP-CP_{whether} - IP-VP</td>
<td>*IP-VP-CP_{if} - IP-VP</td>
</tr>
<tr>
<td>matrix</td>
<td>non-island</td>
</tr>
<tr>
<td>embedded</td>
<td>non-island</td>
</tr>
</tbody>
</table>

Now the ungrammatical dependencies are distinct from all the grammatical dependencies for these syntactic islands, too.
Characterizing the induction problem: Syntactic islands

initial state:
  Bias: Learn from both direct and indirect evidence coming from wh-dependencies.
  Capability: Be able to parse data in the input into phrase structure trees.
  Bias: Characterize dependencies as sequences of container nodes.
  +Bias: Subcategorize container nodes by CP lexical content.

data intake: all wh-dependencies in the input

target state: knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data
Building a computational learner

Learning Bias: Implicitly assign a probability to a container node sequence by tracking trigrams of container nodes. A sequence’s probability is the smoothed product of its trigrams.
Building a computational learner

Learning Bias: Implicitly assign a probability to a container node sequence by tracking trigrams of container nodes. A sequence’s probability is the smoothed product of its trigrams.

\[
[CP \text{ Who did [IP she [VP think [CP [IP [NP the gift] [VP was [PP from ___]]]]]]]]\]

\[
\text{start-IP-VP-CP\textsubscript{null}-IP-VP-PP-end} = \\
\text{start-IP-VP} \\
\text{IP-VP-CP\textsubscript{null}} \\
\text{VP-CP\textsubscript{null}-IP} \\
\text{CP\textsubscript{null}-IP-VP} \\
\text{IP-VP-PP} \\
\text{VP-PP-end}
\]
Building a computational learner

Learning Bias: Implicitly assign a probability to a container node sequence by tracking trigrams of container nodes. A sequence’s probability is the smoothed product of its trigrams.

\[
\text{Probability}(\text{IP-VP-CP}_{\text{null}}-\text{IP-VP-PP}) = p(\text{start-IP-VP-CP}_{\text{null}}-\text{IP-VP-PP-end}) \\
= p(\text{start-IP-VP}) \times p(\text{IP-VP-CP}_{\text{null}}) \times p(\text{VP-CP}_{\text{null}}-\text{IP}) \times p(\text{CP}_{\text{null}}-\text{IP-VP}) \\
\times p(\text{IP-VP-PP}) \times p(\text{VP-PP-end})
\]

\[
\text{[CP Who did [IP she [VP think [CP [IP [NP the gift] [VP was [PP from ___]]]]]]]
\]

\[
\text{IP VP CP}_{\text{null}}\text{ IP VP PP end} = \\
\text{start-IP-VP} \\
\text{IP-VP-CP}_{\text{null}} \\
\text{VP-CP}_{\text{null}}-\text{IP} \\
\text{CP}_{\text{null}}-\text{IP-VP} \\
\text{IP-VP-PP} \\
\text{VP-PP-end}
\]

Pearl & Sprouse forthcoming
Building a computational learner

Learning Bias: Implicitly assign a probability to a container node sequence by tracking trigrams of container nodes. A sequence’s probability is the smoothed product of its trigrams.

What this does:
• longer dependencies are less probable than shorter dependencies, all other things being equal

• individual trigram frequency matters: short dependencies made of infrequent trigrams will be less probable than longer dependencies made of frequent trigrams

Effect: the frequencies observed in the input can temper the detrimental effect of dependency length.
Characterizing the induction problem: 
Syntactic islands

initial state:
  Bias: Learn from both direct and indirect evidence coming from *wh*-dependencies.
  Capability: Be able to parse data in the input into phrase structure trees.
  Bias: Characterize dependencies as sequences of container nodes.
  Bias: Subcategorize container nodes by CP lexical content.
  +Bias: Track trigrams of container nodes in the input.
  +Capability: Generate probability of *wh*-dependency from trigrams of container nodes characterizing it.

data intake: all *wh*-dependencies in the input

target state: knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

*Pearl & Sprouse forthcoming*
Learning process

Hear utterance → Parse utterance, characterizing dependencies as container node sequences → Identify trigrams and update trigram frequencies

XP-YP-ZP...

\[ \text{start-XP-YP + 1} \]

... Repeat until learning period ends

Pearl & Sprouse forthcoming
Generating grammaticality preferences

Parse structure, characterizing dependencies as container node sequences

Identify trigrams

Calculate probability of container node sequence from trigrams

XP-YP-ZP...

start-XP-YP
XP-YP-ZP
...

Probability = $p(\text{start-XP-YP}) \times p(\text{XP-YP-ZP}) \times \cdots$
Building a computational learner: Empirical grounding

Child-directed speech (Brown-Adam, Brown-Eve, Suppes, Valian) from CHILDES:

What kind of dependencies are present?

<table>
<thead>
<tr>
<th>%</th>
<th>Structure</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>76.7%</td>
<td>IP-VP</td>
<td>What did you see __?</td>
</tr>
<tr>
<td>12.8%</td>
<td>IP</td>
<td>What __ happened?</td>
</tr>
<tr>
<td>5.6%</td>
<td>IP-VP-IP-VP</td>
<td>What did she want to do __?</td>
</tr>
<tr>
<td>2.5%</td>
<td>IP-VP-PP</td>
<td>What did she read from __?</td>
</tr>
<tr>
<td>1.1%</td>
<td>IP-VP-CP_{null}-IP-VP</td>
<td>What did she think he said __?</td>
</tr>
</tbody>
</table>

...
Characterizing the induction problem: 
Syntactic islands

**initial state:**
- Bias: Learn from both direct and indirect evidence coming from \(wh\)-dependencies. 
- Capability: Be able to parse data in the input into phrase structure trees. 
- Bias: Characterize dependencies as sequences of container nodes. 
- Bias: Subcategorize container nodes by CP lexical content. 
- Bias: Track trigrams of container nodes in the input. 
- Capability: Generate probability of \(wh\)-dependency from trigrams of container nodes characterizing it.

**data intake: all \(wh\)-dependencies in the input**

**target state:** knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

*Pearl & Sprouse forthcoming*
Building a computational learner: Empirical grounding

Hart & Risley 1995: Children hear approximately one million utterances in their first three years.

Assumption: learning period for modeled learners is 3 years (ex: between 2 and 5 years old for modeling children’s acquisition), so they would hear one million utterances.

Total learning period: 200,000 \(wh\)-dependency data points (\(wh\)-dependencies make up approximately 20% of the input)
Characterizing the induction problem: Syntactic islands

**initial state:**
- Bias: Learn from both direct and indirect evidence coming from *wh*-dependencies.
- Capability: Be able to parse data in the input into phrase structure trees.
- Bias: Characterize dependencies as sequences of container nodes.
- Bias: Subcategorize container nodes by CP lexical content.
- Bias: Track trigrams of container nodes in the input.
- Capability: Generate probability of *wh*-dependency from trigrams of container nodes characterizing it.

**data intake:** all *wh*-dependencies in the input

**learning period:** ~3 years = ~200,000 *wh*-dependency data points

**target state:** knowledge of grammatical and ungrammatical dependencies, as indicated by Sprouse et al. (2012) judgment data

*Pearl & Sprouse forthcoming*
Success metrics

Compare learned grammaticality preferences to Sprouse et al. (2012) judgment data.

Then, for each island, we plot the predicted grammaticality preferences from the modeled learner on an interaction plot, using log probability of the dependency on the y-axis. Non-parallel lines indicate knowledge of islands.

Pearl & Sprouse forthcoming
Learning results

Superadditivity observed for all four islands:

This learner has knowledge of these syntactic islands!

That means this learner can solve this induction problem.

Now...what did it need to do so?

_Pearl & Sprouse forthcoming_
Proposed learning biases/capabilities

Several learning biases/capabilities are potentially both innate and domain-specific.

<table>
<thead>
<tr>
<th></th>
<th>Innate</th>
<th>Derived</th>
<th>Domain-specific</th>
<th>Domain-general</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn from all <em>wh</em>-dependencies</td>
<td>?</td>
<td>?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Parse data into phrase structure trees</td>
<td>?</td>
<td>?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Attend to container nodes &amp; subcategorize by CP</td>
<td>?</td>
<td>?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Extract &amp; track container node trigrams</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Calculate dependency probability from trigrams</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Pearl & Sprouse forthcoming
Learn from all *wh*-dependencies

<table>
<thead>
<tr>
<th></th>
<th>Innate</th>
<th>Derived</th>
<th>Domain-specific</th>
<th>Domain-general</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>?</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
Learn from all *wh*-dependencies

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

Clearly *domain-specific*, since this is language data.

May seem reasonable to attend to *wh*-dependency data when learning about *wh*-dependencies (and so this would be *derived*)
Learn from all *wh*-dependencies

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

Clearly *domain-specific*, since this is language data.

May seem reasonable to attend to *wh*-dependency data when learning about *wh*-dependencies (and so this would be *derived*)

...but then why not attend to *all* dependencies (ex: relative clause dependencies, binding dependencies) since *wh*-dependencies are a kind of dependency?

Empirical necessity of just using *wh*-dependency data:
There are different island effects for relative clauses (Sprouse et al. submitted) and no island effects for binding dependencies, so the learner needs to know to pay attention just to *wh*-dependencies.

*Pearl & Sprouse forthcoming*
Parse data into phrase structure trees
Parse data into phrase structure trees

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

Clearly **domain-specific**, since the structure is specific to language.

May be possible to bootstrap this information (acquiring syntactic categories: Mintz 2003, 2006; acquisition of hierarchical structure given syntactic categories as input: Klein & Manning 2002). If so, this would be **derived**...

_Pearl & Sprouse forthcoming_
Parse data into phrase structure trees

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

Clearly **domain-specific**, since the structure is specific to language.

May be possible to bootstrap this information (acquiring syntactic categories: Mintz 2003, 2006; acquisition of hierarchical structure given syntactic categories as input: Klein & Manning 2002). If so, this would be **derived**...

...but it’s **currently unclear** if all the necessary phrase structure knowledge can be bootstrapped.

**Important:**
The need for this capability is not specific to learning islands – it’s (presumably) needed for learning any kind of syntactic knowledge.

*Pearl & Sprouse forthcoming*
**Attend to container nodes** & subcategorize by CP

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

*Pearl & Sprouse forthcoming*
Attend to container nodes & subcategorize by CP

Identifying container nodes
- applies to language data: domain-specific
- derived from ability to parse utterances
Attend to container nodes & subcategorize by CP

**Identifying container nodes**
- applies to language data: **domain-specific**
- derived from ability to parse utterances

**Attending to container nodes (among all the other data out there)**
- applies to language data: **domain-specific**
- innate vs. derived?
  - could be specified **innately** (like bounding nodes)
  - could be **derived** from a bias to use representations that are already being used for parsing
Attend to container nodes & **subcategorize by CP**

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

_Pearl & Sprouse forthcoming_
Attend to container nodes & **subcategory by CP**

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

About a linguistic representation: **domain-specific**

**Innate vs. derived?**
- Could be specified **innately**
Attend to container nodes & subcategorize by CP

About a linguistic representation: domain-specific

**Innate vs. derived?**

- Could be specified *innately*
- Could be *derived* from prior linguistic experience:
  - Uncontroversial to assume children learn to distinguish different types of CPs since the lexical content of CPs has substantial consequences for the semantics of a sentence.
  - Also, adult speakers are sensitive to the distribution of *that* versus null complementizers (Jaeger 2010).

...but still have to know this is the right thing to subcategorize.

*Pearl & Sprouse forthcoming*
Extract & track container node trigrams

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

*Pearl & Sprouse forthcoming*
Extract & track container node trigrams

Applied in different cognitive domains: **domain-general**


...though why trigrams instead of some other n-gram?
Calculate dependency probability from trigrams

<table>
<thead>
<tr>
<th>Innate</th>
<th>Derived</th>
<th>Domain-specific</th>
<th>Domain-general</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Calculate dependency probability from trigrams

Applied in different cognitive domains: domain-general

Likely innate
Main implications of this learner

(1) Even though there is an induction problem for these syntactic islands, it may not require Universal Grammar learning biases to solve it.

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

Learn from all *wh*-dependencies

*Parse data into phrase structure trees*

Attend to container nodes & subcategorize by CP

Extract & track container node trigrams

Calculate dependency probability from trigrams

*Pearl & Sprouse forthcoming*
Main implications of this learner

(2) Even if Universal Grammar learning biases are required, they are different from (and less specific than) the biases previously proposed.

In particular, while one bias also specifies a particular linguistic representation, there is no bias defining the “constraint”. This falls out from the other non-UG learning biases.

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

Learn from all *wh*-dependencies

Attend to container nodes & subcategorize by CP

Attend to bounding nodes (BNs)

Dependencies crossing 2+ BNs are not allowed

Pearl & Sprouse forthcoming
I. Potential induction problem:
Learning constraints on long-distance dependencies

II. Potential induction problem:
Learning English anaphoric one
English anaphoric *one*

Look - a red bottle!
English anaphoric one

Look - a red bottle!

Do you see another one?
English anaphoric *one*

Look - a red bottle!

Do you see another *one*?

Process: First determine the **antecedent** of *one* (what string *one* is referring to).
→ “red bottle”
English anaphoric *one*

Look - a red bottle!

Do you see another *one*?

Process: Because the antecedent ("red bottle") includes the modifier "red", the property RED is important for the referent of *one* to have.

→ referent of *one* = RED BOTTLE
English anaphoric *one*

Look - a red bottle!

Do you see another *one*?

Two steps:
1. Identify **syntactic** antecedent
2. Identify **semantic** referent (based on syntactic antecedent)

*Pearl & Mis submitted*
Anaphoric *one*: Syntactic category

Standard linguistic theory (Chomsky 1970, Jackendoff 1977) posits that *one* in these kind of utterances is a syntactic category smaller than an entire noun phrase (NP), but larger than just a noun (N⁰). This category is *N’. This category includes strings like “bottle” and “red bottle”.

![Diagram of NP structure]

\[
[\text{NP another } [\text{N’ } [\text{N⁰ bottle}]]] \quad \text{and} \quad [\text{NP another } [\text{N’ } \text{red } [\text{N’ } [\text{N⁰ bottle}]]]]
\]
Anaphoric one: Syntactic category

Standard linguistic theory (Chomsky 1970, Jackendoff 1977) posits that one in these kind of utterances is a syntactic category smaller than an entire noun phrase (NP), but larger than just a noun ($N^0$). This category is $N'$. This category includes strings like “bottle” and “red bottle”.

\[
\text{NP} \rightarrow \text{det} \rightarrow \text{N'} \rightarrow \text{n-phrase} \rightarrow \text{NP}
\]

\[
\text{NP} \rightarrow \text{det} \rightarrow \text{N'} \rightarrow \text{adj} \rightarrow \text{N'} \rightarrow \text{N}^0 \rightarrow \text{n-phrase} \rightarrow \text{NP}
\]
Anaphoric one: Syntactic category

Importantly, one is not $N^0$. If it was, it could only have strings like “bottle” as its antecedent, and could never have strings like “red bottle” as its antecedent.

\[ [\text{NP another } [\text{N'} [N^0 \text{bottle}]]] \]

\[ [\text{NP another } [\text{N'} \text{red } [\text{N'} [N^0 \text{bottle}]]]] \]
Anaphoric one: Interpretations based on syntactic category

If one was $N^0$, we would have a different interpretation of

“Look – a red bottle! Do you see another one?”

Because one’s antecedent could only be “bottle”, we would have to interpret the second part as “Do you see another bottle?” and the purple bottle would be a fine referent for one.

Since one’s antecedent is “red bottle”, and “red bottle” cannot be $N^0$, one must not be $N^0$.  

Pearl & Mis submitted
Anaphoric *one*: Adult knowledge

“Look – a red bottle! Look, there’s another *one!*”
≈ “Look – a red bottle! Look, there’s another *red bottle!*”

Target state:

Syntactic knowledge: category N’

Semantic knowledge: mentioned property ("red") is included in the linguistic antecedent (antecedent = “red bottle”)
Anaphoric *one*: Children’s knowledge

Lidz, Waxman, & Freedman (2003) [LWF] found that 18-month-olds have a preference for the red bottle in the same situation.

“Look – a red bottle! Do you see another one?”

LWF interpretation & conclusion:

Preference for the **red bottle** means the preferred syntactic antecedent is “red bottle”.

*Pearl & Mis submitted*
Anaphoric one: Children’s knowledge

Lidz, Waxman, & Freedman (2003) [LWF] found that 18-month-olds have a preference for the red bottle in the same situation.

“Look – a red bottle! Do you see another one?”

LWF interpretation & conclusion:
Preference for the red bottle means the preferred syntactic antecedent is “red bottle”.

LWF concluded that 18-month-old knowledge =

syntactic category of one = N’
syntactic antecedent when modifier is present (i.e., property is mentioned) includes modifier (e.g., “red”) = referent has modifier property

Learning period = completed by 18 months

Pearl & Mis submitted
Characterizing the induction problem: English anaphoric *one*

initial state:
- Knowledge: Syntactic categories exist, in particular N\(^0\), N’, and NP.
- Knowledge: Anaphoric elements like *one* take linguistic antecedents of the same category.
- Bias: Only direct evidence of *one* is useful.
- Bias: Only unambiguous evidence of *one* is useful (Baker 1978).

data intake:
All unambiguous *one* evidence in the input.

learning period:
Completed by 18 months (LWF 2003)

target state:
*One* is category N’ and its antecedent includes the mentioned modifier when present.
Behavior signal: Generate adult interpretation in utterances with mentioned modifier ("Look – a red bottle. Do you see another one?")
Anaphoric *one*: The available data

Acquisition: Children must learn the right syntactic category for *one*, and the right interpretation preference for *one* in situations with more than one option.
Anaphoric *one*: The available data

Acquisition: Children must learn the right syntactic category for *one*, and the right interpretation preference for *one* in situations with more than one option.

**Problem: Unambiguous data are rare (<0.25%: LWF 2003, 0.00%: Pearl & Mis submitted)**

Unambiguous (UNAMB) data:

“Look – a red bottle! Hmmm - there doesn’t seem to be another one here, though.”

![Image of a red and a purple bottle]

*one’s referent = BOTTLE? If so, *one*’s antecedent = “bottle”.*

But it’s strange to claim there’s not another *bottle* here.

So, *one*’s referent must be **RED BOTTLE**, and *one*’s antecedent = $[_{N'} \text{red}[_{N'}[_{N_0} \text{bottle}]]]$.

*Pearl & Mis submitted*
Anaphoric one: The available data

Acquisition: Children must learn the right syntactic category for one, and the right interpretation preference for one in situations with more than one option.

Problem: Most data children encounter are ambiguous.

Syntactically (SYN) ambiguous data:
“Look – a bottle! Oh, look – another one.”

one’s referent = BOTTLE
one’s antecedent = $\left[_{N'}\left[_{N_0} \text{bottle}\right]\right] \text{ or } \left[_{N_0} \text{bottle}\right]$?
Anaphoric *one*: The available data

Acquisition: Children must learn the right syntactic category for *one*, and the right interpretation preference for *one* in situations with more than one option.

Problem: Most data children encounter are ambiguous.
Semantically and syntactically (SEM-SYN) ambiguous:
“Look – a red bottle! Oh, look – another one.”

*one’s* referent = RED BOTTLE or BOTTLE?
*one’s* antecedent = \([_N^* \text{red}[N^*[_{NO} \text{bottle}]]]\) or \([_{NO}\text{bottle}]\) or \([NO\text{bottle}]\)?

*Pearl & Mis submitted*
Previous learning strategies

Update the initial state


How then?
Children have innate, domain-specific knowledge restricting the hypotheses about one: one cannot be syntactic category N0.

What about when there are multiple N’ antecedents?
[\text{red}_N[\text{red}_N[\text{bottle}]]] or [\text{red}_N[\text{bottle}]]?
(No specific proposal for this.)

Pearl & Mis submitted
Previous learning strategies

Update the initial state

Baker (1978) [DirectUnamb]

initial state

Knowledge: Syntactic categories exist, in particular $N^0$, $N'$, and NP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
Bias: Only direct evidence of one is useful.
Bias: Only unambiguous evidence of one is useful (Baker 1978).

+ (UG) Knowledge: one is not $N^0$.

Successful at solving induction problem w.r.t syntactic category.

Pearl & Mis submitted
Previous learning strategies

Update the initial state

Regier & Gahl 2004 [R&G]: Sem-Syn ambiguous data can be leveraged, in addition to using unambiguous data.
   “Look – a red bottle! Oh, look – another one!”

How?

Use innate domain-general statistical learning abilities (Bayesian inference) to track how often one’s referent has the mentioned property (e.g. red). If the referent often has the property (RED BOTTLE), this is a suspicious coincidence unless the antecedent really does include the modifier (“red bottle”) and one’s category is N’.

\[ \left[ \text{N'} \right. \text{red} \left[ \text{N'} \left[ \text{N0 bottle} \right] \right] \]
Previous learning strategies

Update the initial state

Regier & Gahl 2004 [R&G]

Initial state

Knowledge: Syntactic categories exist, in particular $N^0$, $N'$, and NP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
Bias: Only direct evidence of one is useful.
- **Bias**: Only unambiguous evidence of one is useful (Baker 1978).
+ **Bias**: Use Bayesian inference.

Successful at solving induction problem.

Pearl & Mis submitted
Previous learning strategies

Update the initial state

Pearl & Lidz 2009 [P&L]: Syn ambiguous data must not be leveraged, even if Sem-Syn and unambiguous data are used.
“Look – a bottle! Oh, look – another one!”

Why?
These data cause an “equal-opportunity” (EO) probabilistic learner to think one’s category is $N^0$.

[No bottle]

How?
P&L propose a domain-specific learning bias to ignore just these ambiguous data, though they speculate how this bias could be derived from an innate domain-general preference for learning when there is local uncertainty.
Previous learning strategies

Update the initial state

Pearl & Lidz 2009 [R&G in practice, Equal Opportunity = DirectEO]

initial state

Knowledge: Syntactic categories exist, in particular N₀, N’, and NP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
Bias: Only direct evidence of one is useful.

- Bias: Only unambiguous evidence of one is useful (Baker 1978).
+ Bias: Use Bayesian inference.

Not successful at solving induction problem.
Previous learning strategies

Update the initial state

Pearl & Lidz 2009 [R&G intended, P&L filtered = DirectFiltered]

initial state

Knowledge: Syntactic categories exist, in particular $N^0$, $N'$, and NP.
Knowledge: Anaphoric elements like *one* take linguistic antecedents of the same category.
Bias: Only direct evidence of *one* is useful.

- Bias: *Only unambiguous evidence of one is useful* (Baker 1978).
+ Bias: Use Bayesian inference.
+ (UG?) Bias: Ignore Syn ambiguous data.

Successful at solving induction problem.

Pearl & Mis submitted
A new strategy: 
Using indirect positive evidence

Pearl & Mis (2011, submitted) [+OtherPro]: Other words in the language can also be used anaphorically: *him, her, it,* ...

Look at the cute penguin. I want to hug *it.*

\[
[\text{NP the } [\text{N' cute } [\text{N' N0 penguin}] technological]]] \rightarrow [\text{NP it}]
\]

Look! A cute penguin. I want *one.*

\[
[\text{NP a } [\text{N' cute } [\text{N' N0 penguin}] technological]]] \rightarrow [\text{NP one}]
\]

Note: The issue of *one*'s category only occurs when *one* is used in a syntactic environment that indicates it is smaller than an NP (<NP).

*Pearl & Mis submitted*
A new strategy: Using indirect positive evidence

Pearl & Mis (2011, submitted) [+OtherPro]: Track how often the referent of the anaphoric element (*one, him, her, it*, etc.) has the property mentioned in the potential antecedent, using innate domain-general statistical learning abilities (*Bayesian inference*).

Important: This applies, even when the syntactic category is known.

Look at the cute penguin. I want to hug *it*.
Look! A cute penguin. I want *one*.

Is the referent cute? Yes!
So the antecedent includes the modifier “cute”.

*Pearl & Mis submitted*
A new strategy: Using indirect positive evidence

Pearl & Mis (2011, submitted) [+OtherPro]: Track how often the referent of the anaphoric element (one, him, her, it, etc.) has the property mentioned in the potential antecedent, using innate domain-general statistical learning abilities (Bayesian inference).

Important: This applies, even when the syntactic category is known.

Look at the cute penguin. I want to hug it.

Look! A cute penguin. I want one.

These kind of data points will always include the modifier in the antecedent, since the category of the pronoun is NP and so the antecedent is the entire NP. These data are unambiguous: The referent must have the mentioned property & the antecedent must include the modifier corresponding to that property.

Pearl & Mis submitted
A new strategy: Using indirect positive evidence

Pearl & Mis (2011, submitted) [+OtherPro]

initial state
  Knowledge: Syntactic categories exist, in particular N₀, N’, and NP.
  Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
- Bias: Only direct evidence of one is useful.
- Bias: Only unambiguous evidence of one is useful (Baker 1978).
+ Bias: Use Bayesian inference.
+ (UG?) Bias: Learn from other pronoun data.

Successful at solving induction problem?

Pearl & Mis submitted
Data set comparisons

**Unamb <NP**
“Look – a red bottle! Hmm – there doesn’t seem to be another one here, though.”

Learners: DirectUnamb, DirectFiltered, DirectEO, +OtherPro

**Sem-Syn Amb**
“Look – a red bottle! Oh, look – another one!”

Learners: DirectFiltered, DirectEO, +OtherPro

**Syn Amb**
“Look – a bottle! Oh, look – another one!”

Learners: DirectEO, +OtherPro

**Unamb NP**
“Look – a red bottle! I want one/it.”

Learners: +OtherPro

*Pearl & Mis submitted*
Information in the data

Understanding a referential expression

Includes both syntactic and semantic/referential information, since both are used to determine the linguistic antecedent.
Information in the data

“Look, a red bottle! Look, another one!”

Syntactic information

R = referential expression used
ex: “another one”

Pro = pronoun used in referential expression
ex: “one”

env = smaller than NP?
ex: yes
Information in the data

“Look, a red bottle! Look, another one!”

Syntactic information

C = syntactic category of pronoun used (= syntactic category of linguistic antecedent)
  ex: N'

det = antecedent includes determiner?
  ex: no

mod = antecedent includes modifier?
  ex: yes

Pearl & Mis submitted
Information in the data

“Look, a red bottle! Look, another one!”

Semantic/referential information

m = property mentioned in previous linguistic context
ex: yes

o-m = referent (object) in current context has mentioned property
ex: yes

i = mentioned property is included in antecedent?
ex: yes

Pearl & Mis submitted
Information in the data

“Look, a red bottle! Look, another one!”

A = antecedent
ex: “red bottle”
(depends on both syntactic information of det and mod, and semantic/referential information from i.)

O = intended object (learner can usually observe this)
ex: RED BOTTLE
The online probabilistic learning framework

semantic/referential knowledge

When an object has the property mentioned in the potential antecedent (o-m=yes), track the probability that the property is included in the antecedent (i=yes):

\[ p_{incl} = p(i=yes \mid o-m=yes) \]

Two values: (i=yes or i=no)
The online probabilistic learning framework

syntactic knowledge

When the syntactic environment indicates the category is smaller than NP (env=<NP), track the probability that the syntactic category is N’ (C=N’): 

\[ p_{N'} = p(C=N' \mid env=<NP) \]

Two values: (C=N’ or C=N0)
The online probabilistic learning framework

General form of online update equations for $p_x$ (adapted from Chew 1971):

$$p_x = \frac{\alpha + data_x}{\alpha + \beta + totaldata_x}, \alpha = \beta = 1$$

A very weak prior

total informative data seen w.r.t $x$

After every informative data point encountered:

$$data_x = data_x + \phi_x$$

Incremented by probability that data point suggests $x$ is true

$$totaldata_x = totaldata_x + 1$$

One informative data point seen

Pearl & Mis submitted
Corpus analysis & learner input

Brown/Eve corpus (CHILDES: MacWhinney 2000)

17,521 utterances of child-directed speech, 2874 referential pronoun utterances

<table>
<thead>
<tr>
<th>Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unamb &lt;NP</td>
<td>0.00%</td>
</tr>
<tr>
<td>Sem-Syn Amb</td>
<td>0.66%</td>
</tr>
<tr>
<td>Syn Amb</td>
<td>7.52%</td>
</tr>
<tr>
<td>Unamb NP</td>
<td>8.42%</td>
</tr>
<tr>
<td>Uninformative</td>
<td>83.4%</td>
</tr>
</tbody>
</table>

Pearl & Lidz 2009: Children learn one’s representation between 14 and 18 months.

Based on estimates of the number of utterances children hear from birth until 18 months (Akhtar et al., 2004), we can calculate the data distribution in their input (36,500 referential pronoun utterances total).
Corpus analysis & learner input

Brown/Eve corpus (CHILDES: MacWhinney 2000)

17,521 utterances of child-directed speech, 2874 referential pronoun utterances

<table>
<thead>
<tr>
<th>Category</th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unamb &lt;NP</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sem-Syn Amb</td>
<td>0.66%</td>
<td>0</td>
<td>242</td>
<td>242</td>
</tr>
<tr>
<td>Syn Amb</td>
<td>7.52%</td>
<td>0</td>
<td>0</td>
<td>2743</td>
</tr>
<tr>
<td>Unamb NP</td>
<td>8.42%</td>
<td>0</td>
<td>0</td>
<td>3073</td>
</tr>
<tr>
<td>Uninformative</td>
<td>83.4%</td>
<td>36500</td>
<td>36258</td>
<td>33515</td>
</tr>
</tbody>
</table>

Pearl & Lidz 2009: Children learn one’s representation between 14 and 18 months.

Based on estimates of the number of utterances children hear from birth until 18 months (Akhtar et al., 2004), we can calculate the data distribution in their input (36,500 referential pronoun utterances total).
Measures of success: Children’s behavior

In addition to directly assessing $p_{incl}$ and $p_{N'}$, we can measure how often a learner would reproduce the behavior in the LWF experiment ($p_{beh}$).

Look – a red bottle!

Do you see another one?

Pearl & Mis submitted
Testing assumptions about what behavior means

Does target behavior in the LWF experiment mean the learner has the target representation for \textit{one in general} (as measured by $p_{incl}$ and $p_N$)?

Signal: $p_{beh}$ is high only when $p_{incl}$ and $p_N$ are both high.

Does the target behavior in the LWF experiment mean the learner has the target representation for \textit{one at the time the behavior is being produced}? \textit{$p_{rep|beh}$}: Given that the learner has looked at the red bottle, what is the probability that the learner has the target knowledge representation (N', “red bottle”) while doing so?

Signal: $p_{rep|beh}$ is high (irrespective of $p_{incl}$ and $p_N$).
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th>DirectUnamb</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 ($&lt;0.01$)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 ($&lt;0.01$)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 ($&lt;0.01$)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
</tr>
</tbody>
</table>

Since the input data include no Unambiguous <NP data, and those are the only data the DirectUnamb learner learns from, it learns nothing.
## Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th>DirectUnamb</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{\text{incl}}$</td>
<td>0.50 ($&lt;0.01$)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 ($&lt;0.01$)</td>
</tr>
<tr>
<td>$p_{\text{beh}}$</td>
<td>0.56 ($&lt;0.01$)</td>
</tr>
<tr>
<td>$p_{\text{rep}</td>
<td>\text{beh}}$</td>
</tr>
</tbody>
</table>

It is at chance for having the **target syntactic** and **semantic** representation.

It is only slightly above chance at producing the **observed toddler behavior**, and when it does, it is **unlikely** to have the target representation when doing so.

*Pearl & Mis submitted*
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th>DirectUnamb</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
</tr>
</tbody>
</table>

Implication:
This is an induction problem if only unambiguous <NP data are relevant.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectFiltered learner (R&G, P&L’s filtered)

This learner believes a mentioned property should be included in the antecedent and one is N’ when it is smaller than NP, which is similar to previous findings by R&G & P&L.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{\text{incl}}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_N$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{\text{beh}}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{\text{rep</td>
<td>beh}}$</td>
<td>0.23 (&lt;0.01)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectFiltered learner (R&G, P&L’s filtered)

In addition, it is likely to generate the observed toddler behavior, and have the target representation when doing so.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectEO learner (P&L’s EO)

The learner does not believe the mentioned property should be included in the antecedent, and prefers one to be $N^0$ when it is smaller than NP.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
</tr>
<tr>
<td>$p_N$'</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectEO learner (P&L’s EO)

This causes the learner to be at chance at generating the observed toddler behavior, and unlikely to have the target representation when generating that behavior.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
<td>&gt;0.99 (0.01)</td>
</tr>
<tr>
<td>$p_N$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
<td>0.37 (0.04)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
<td>0.01 (0.01)</td>
</tr>
</tbody>
</table>

The +OtherPro learner robustly decides the antecedent should include the mentioned property.

However, the learner has a moderate dispreference for believing one is $N'$ when it is smaller than NP.

This is therefore not the target representation.

Pearl & Mis submitted
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
<td>&gt;0.99 (0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
<td>0.37 (0.04)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
<td>0.01 (0.01)</td>
</tr>
</tbody>
</table>

However...this learner still generates the observed toddler behavior with high probability, and has the target representation when doing so.

Pearl & Mis submitted
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
<td>&gt;0.99 (0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
<td>0.37 (0.04)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
<td>0.01 (0.01)</td>
</tr>
</tbody>
</table>

Why?

The learner believes very strongly that the mentioned property must be included in the antecedent.

Only one representation allows this: $[N', \text{red}[N'[N_0 \text{bottle}]]$
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_N'$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
<td>0.37 (0.04)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
<td>0.01 (0.01)</td>
</tr>
</tbody>
</table>

Why?

So, because the antecedent includes the mentioned property, it and the referential pronoun referring to it (one) must be N’ in this context - even if the learner believes one is not N’ in general.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91 (&lt;0.01)</td>
<td>0.10 (0.05)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{N^r}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98 (&lt;0.01)</td>
<td>0.18 (0.03)</td>
<td>0.37 (0.04)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88 (&lt;0.01)</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87 (&lt;0.01)</td>
<td>0.01 (0.01)</td>
</tr>
</tbody>
</table>

Take away point:

A learner using an indirect positive evidence strategy can generate target behavior without reaching the target state – instead, this learner has a context-sensitive representation (depending on whether a property was mentioned).
Learning strategies & induction problems

Using indirect positive evidence:
Generate observed target behavior without having target state knowledge

What does this mean for the induction problem?

target state:
One is category N’ and its antecedent includes the mentioned modifier when present.
Behavior signal: Generate adult interpretation in utterances with mentioned modifier
(“Look – a red bottle. Do you see another one?”)  

The link between observed behavior and underlying knowledge representation may not be so clearcut.

Pearl & Mis submitted
Learning strategies & induction problems

Using indirect positive evidence:
Generate observed target behavior without having target state knowledge

What does this mean for the induction problem?

target state:
*One* is category $N'$ and its antecedent includes the mentioned modifier when present.

Behavior signal: Generate adult interpretation in utterances with mentioned modifier
(“Look – a red bottle. Do you see another one?”)

+Behavior signal: Recognize ungrammaticality of utterances where *one* is used as an
$N^0$, like *“Jack sat by the side of the road and Lily sat by the one of the river.”*

Children may achieve this later than 18 months.

Pearl & Mis submitted
Learning strategies & induction problems

Using indirect positive evidence:
Generate observed target behavior without having target state knowledge

What does this mean for the induction problem?

target state:

*One* is category *N’* and its antecedent includes the mentioned modifier when present.

[Stage 1] Behavior signal: Generate adult interpretation in utterances with mentioned modifier ("Look – a red bottle. Do you see another one?")

[Stage 2] Behavior signal: Recognize ungrammaticality of utterances like *
*”Jack sat by the side of the road and Lily sat by the one of the river.’”

Maybe there are (at least) two stages of acquisition?
Motivating UG

What kind of biases does the +OtherPro learner use, if we want to achieve stage 1?

initial state: Two new biases

Knowledge: Syntactic categories exist, in particular $N^0$, $N'$, and NP.

Knowledge: Anaphoric elements like *one* take linguistic antecedents of the same category.

+ Bias: Use Bayesian inference.

+ Bias: Learn from other pronoun data.
Motivating UG

What kind of biases does the +OtherPro learner use, if we want to achieve stage 1?

Bias to use Bayesian inference:
   innate, domain-general statistical learning ability (not UG)
Motivating UG

What kind of biases does the +OtherPro learner use, if we want to achieve stage 1?

Bias to learn from other pronoun data:

- concerns language data, so clearly domain-specific

innate or derived?
Motivating UG

What kind of biases does the +OtherPro learner use, if we want to achieve stage 1?

Bias to learn from other pronoun data:
concerns language data, so clearly domain-specific

innate or derived?
If innate, then this is a UG bias.

If so, this is a specific proposal for the contents of UG that is less specific than Baker’s proposal and doesn’t involve limiting the data intake like the DirectFiltered strategy.
Motivating UG

What kind of biases does the +OtherPro learner use, if we want to achieve stage 1?

Bias to learn from other pronoun data:
   concerns language data, so clearly domain-specific

innate or derived?
   Could be derived from prior linguistic experience with pronouns (and noticing overlapping syntactic environments for “one” and other referential pronouns.)

If so, this is a non-UG learning strategy that will produce the desired behavior. This then takes away support for UG that comes from this induction problem characterization.
The big picture:
Making an argument from acquisition for UG

**Universal Grammar**: a theory of linguistic knowledge that is explicitly motivated by the **existence** of induction problems during acquisition and the **solutions** to those problems.

**Existence**
Requires a specific characterization that defines initial state, data intake, learning period, and target state

**Solutions**
Here: Exploring an indirect positive evidence learning strategy as a general approach, and applying it to two different induction problems. We can then examine the biases involved.
Making progress on UG

I. Potential induction problem:

- Learning constraints on long-distance dependencies

- Target state knowledge indicated by adult judgment behavior.
- Indirect positive evidence strategy can generate this behavior.
- Strategy may involve UG biases, but if so, they’re much less specific than those previously proposed.
Making progress on UG

• Target state knowledge thought to be indicated by 18-month-old behavior... but may not actually be (potential recharacterization of induction problem).
• Indirect positive evidence strategy can generate this behavior, though.
• Strategy may involve a UG bias, but if so, it’s much less specific than what was previously proposed.
• May mean there are two stages of knowledge acquisition.

II. Potential induction problem:
  Learning English anaphoric one
Empirically investigating UG

Empirical investigation of UG involves drawing on multiple research methods to

(1) make sure we’re all worried about the same problem, and

(2) make headway on the UG debate by providing a formal mechanism for evaluating induction problem solutions.
Thank you!

Jon Sprouse  
Diogo Almeida  Max Bane  Misha Becker  Bob Berwick
Sue Braunwald  Ivano Caponigro  Alexander Clark  Bob Frank
LouAnn Gerken  Norbert Hornstein  Greg Kobele  Jeff Lidz
Colin Phillips  William Sakas  Morgan Sonderegger  Mark Steyvers
Virginia Valian  Ming Xiang  Charles Yang

Computational Models of Language Learning seminar, UC Irvine 2010
Audiences at:
CogSci 2011
UChicago 2011 workshops on
  Language, Cognition, and Computation &  
  Language, Variation, and Change

Computation of Language Laboratory  
UC Irvine

This work was supported in part by NSF grant BCS-0843896.
Extra Material
Why learning from container node trigrams works

For each island-spanning dependency, there is at least one extremely low probability container node trigram in the dependency.

Complex NP island
\[ \text{start-IP-VP-NP-CP}_{\text{that}}-\text{IP-VP-end} \]

Subject island
\[ \text{start-IP-VP-CP}_{\text{null}}-\text{IP-NP-PP-end} \]

Whether island
\[ \text{start-IP-VP-CP}_{\text{whether}}-\text{IP-VP-end} \]

Adjunct island
\[ \text{start-IP-VP-CP}_{\text{if}}-\text{IP-VP-end} \]

These trigrams are never observed in the input – which is crucially different than being observed rarely. Thus, these islands are worse than dependencies involving trigrams that are rarely seen (e.g., dependencies with \( CP_{\text{that}} \)) and even longer dependencies that involve more frequent trigrams (e.g., triply embedded object dependencies using \( CP_{\text{null}} \)).
The empirical necessity of trigrams

Not unigrams
A unigram model will successfully learn Whether and Adjunct islands, as there are container nodes in these dependencies that never appear in grammatical dependencies ($\text{CP}_{\text{whether}}$ and $\text{CP}_{\text{if}}$)....but it will fail to learn Complex NP and Subject islands, as all of the container nodes in these islands are shared with grammatical dependencies.

Complex NP:
- *IP-VP-NP-CP_{that}-IP-VP

Subject:
- *IP-VP-CP_{null}-IP-NP-PP

Whether:
- IP-VP-CP_{\text{whether}}-IP-VP

Adjunct:
- IP-VP-CP_{\text{if}}-IP-VP

*Pearl & Sprouse forthcoming*
The empirical necessity of trigrams

Not bigrams
At least for Subject islands, there is no bigram that occurs in a Subject island violation but not in any grammatical dependencies. The most likely candidate for such a bigram is IP-NP...However, sentences such as *What, again, about Jack impresses you?* or *What did you say about the movie scared you?* suggest that a gap can arise inside of NPs, as long as the extraction is of the head noun (what), not of the noun complement of the preposition.

Complex NP: \( \text{IP-VP-NP-CP}_{\text{that}}-\text{IP-VP} \)
Subject: \( \ast \text{IP-VP-CP}_{\text{null}}-\text{IP-NP-PP} \)
Whether: \( \text{IP-VP-CP}_{\text{whether}}-\text{IP-VP} \)
Adjunct: \( \text{IP-VP-CP}_{\text{if}}-\text{IP-VP} \)

*Pearl & Sprouse forthcoming*
Parasitic gaps

The learner can’t handle parasitic gaps, which are dependencies that span an island (and so should be ungrammatical) but which are somehow rescued by another dependency in the utterance.

*Which book did you laugh [before reading __]?
Which book did you judge ___true [before reading ___parasitic]?

Adjunct island

*What did [the attempt to repair ___] ultimately damage the car?
What did [the attempt to repair ___parasitic] ultimately damage ___true?

Complex NP island

Pearl & Sprouse forthcoming
Parasitic gaps

Why not? The current learner would judge the parasitic gap as ungrammatical since it is inside an island, irrespective of what other dependencies are in the utterance.

*Which book did you laugh [before reading ___]?  
Which book did you judge ___true [before reading ___parasitic]?  

Adjunct island

*What did [the attempt to repair ___] ultimately damage the car?  
What did [the attempt to repair ___parasitic] ultimately damage ___true?  

Complex NP island

This may be able to be addressed in a learner that is able to combine information from multiple dependencies in an utterance (perhaps because the learner has observed multiple dependencies resolved in utterances in the input).

Pearl & Sprouse forthcoming
Across-the-board constructions

A similar problem occurs for across-the-board constructions.

Which book did you [ [read ___ ] and [then review ___]]? dependency for both gaps: IP-VP-VP

*Which book did you [[read the paper] and [then review ___]]? dependency for gap: IP-VP-VP

*Which book did you [[read ___ ] and [then review the paper]]? dependency for gap: IP-VP-VP

Again, this may be able to be addressed in a learner that is able to combine information from multiple dependencies in an utterance (perhaps because the learner has observed multiple dependencies resolved in utterances in the input).

Pearl & Sprouse forthcoming
Some cross-linguistic issues

High probability trigrams that may be ungrammatical

Rizzi (1982): reports situations in Italian where simply doubling a grammatical sequence of trigrams leads to ungrammaticality...

\[ IP-VP-CP_{wh}-IP-VP \]

but

\[ *IP-VP-CP_{wh}-IP-VP-CP_{wh}-IP-VP-IP-VP \]

But these involve the same trigrams, so the learner in Pearl & Sprouse (forthcoming) will treat both the same (either grammatical or ungrammatical). If humans do have different judgments of these, then this cannot be accounted for by this learning algorithm.

Pearl & Sprouse forthcoming
Complementizer *that*

That-trace effects

*Who do you think that ___ read the book?*

Who do you think ___ read the book?

The current learning strategy captures this distinction.
Complementizer *that*

That-trace effects

...but the current learning strategy will also generate a preference for object gaps without *that* compared to object gaps with *that*. (object *that*-trace effect)

What do you think that he read ___? [prefers this one]
What do you think he read ___?

Interestingly, Cowart 1997 finds an object *that-trace* effect, but it is much smaller than the subject *that-trace* effect

The model generates an asymmetrical dispreference when using adult-directed corpora, which contain more instances of *that* (5.40 versus 2.81). This could be taken to be a developmental prediction of the current algorithm: Children may disprefer object gaps in embedded *that-CP* clauses more than adults, and this dispreference will weaken as they are exposed to additional tokens of *that* in utterances containing dependencies.

*Pearl & Sprouse forthcoming*
English anaphoric one
Information in the data: Unamb <NP

“Look, a red bottle! Hmm – there isn’t another one here though!”

R = “another one”
Pro = “one”
m = yes
env = <NP
o-m = yes

C = N’
det = no
mod = yes
i = yes

A = “red bottle”
O = RED BOTTLE
Information in the data: Sem-Syn ambiguous

"Look, a red bottle! Look – another one!"

R = “another one”
Pro = “one”
m = yes
env = <NP
o-m = yes

C = N’ or N⁰?
det = no
mod = yes or no?
i = yes or no?

A = “red bottle” or “bottle”?
O = RED BOTTLE

Pearl & Mis submitted
Information in the data: Syn ambiguous

“Look, a bottle! Look – another one!”

R = “another one”
Pro = “one”       m = no
env = <NP        o-m = N/A

C = N’ or N^0?
det = no
mod = no       i = N/A

A = “bottle”
O = BOTTLE
"Look, a red bottle! I want it."

- **R** = “it”
- **Pro** = “it”
- **env** = NP
- **m** = yes
- **o-m** = yes
- **C** = NP
- **det** = yes
- **mod** = yes
- **i** = yes
- **A** = “a red bottle”
- **O** = RED BOTTLE

---

*Pearl & Mis submitted*
The online probabilistic framework: Updating $p_{incl}$

$\Phi_{incl}$

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unamb &lt;NP</td>
<td>1</td>
<td>Property definitely included</td>
</tr>
<tr>
<td>Unamb NP</td>
<td>1</td>
<td>Property definitely included</td>
</tr>
<tr>
<td>Syn Amb</td>
<td>N/A</td>
<td>Not informative for $p_{incl}$</td>
</tr>
<tr>
<td>Sem-Syn Amb</td>
<td>$\frac{rep_1}{rep_1 + rep_2 + rep_3}$</td>
<td>Probability property is included</td>
</tr>
</tbody>
</table>

$rep_1 = p_N \cdot \frac{m}{m+n} \cdot p_I$

Category = N', choose N' with modifier, property is included

$rep_2 = p_N \cdot \frac{n}{m+n} \cdot (1 - p_{incl}) \cdot \frac{1}{s}$

Category = N', choose N' without modifier, property is not included, choose object with property by chance

$rep_3 = (1 - p_N) \cdot (1 - p_{incl}) \cdot \frac{1}{s}$

Category = N⁰, property is not included, choose object with property by chance

Pearl & Mis submitted
## The online probabilistic framework: Updating $p_{N'}$

<table>
<thead>
<tr>
<th>Case</th>
<th>Formula</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unamb &lt;NP</td>
<td>$\Phi_{N'} = 1$</td>
<td>Category definitely N'</td>
</tr>
<tr>
<td>Unamb NP</td>
<td>N/A</td>
<td>Not informative for $p_{N'}$</td>
</tr>
<tr>
<td>Syn Amb</td>
<td>$\frac{rep_4}{rep_4 + rep_5}$</td>
<td>Probability category is N'</td>
</tr>
<tr>
<td>Sem-Syn Amb</td>
<td>$\frac{rep_1 + rep_2}{rep_1 + rep_2 + rep_3}$</td>
<td>Probability category is N'</td>
</tr>
</tbody>
</table>

### Detailed Formulas:

- $rep_1 = p_{N'} \cdot \frac{m}{m + n} \cdot p_I$
  - Category = N', choose N' with modifier, property is included

- $rep_2 = p_{N'} \cdot \frac{n}{m + n} \cdot (1 - p_{incl}) \cdot \frac{1}{s}$
  - Category = N', choose N' without modifier, property is not included, choose object with property by chance

- $rep_3 = (1 - p_{N'}) \cdot (1 - p_{incl}) \cdot \frac{1}{s}$
  - Category = N', property is not included, choose object with property by chance

*Pearl & Mis submitted*
The online probabilistic framework: Updating $p_{N'}$

<table>
<thead>
<tr>
<th>Category</th>
<th>$\Phi_{N'}$</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unamb &lt;NP</td>
<td>1</td>
<td>Category definitely $N'$</td>
</tr>
<tr>
<td>Unamb NP</td>
<td>N/A</td>
<td>Not informative for $p_{N'}$</td>
</tr>
<tr>
<td>Syn Amb</td>
<td>$\frac{rep_4}{rep_4 + rep_5}$</td>
<td>Probability category is $N'$</td>
</tr>
<tr>
<td>Sem-Syn Amb</td>
<td>$\frac{rep_1 + rep_2}{rep_1 + rep_2 + rep_3}$</td>
<td>Probability category is $N'$</td>
</tr>
</tbody>
</table>

$rep_4 = p_{N'}^{*} \cdot \frac{n}{m + n}$

$rep_5 = 1 - p_{N'}$

Category = $N'$, choose $N'$ without modifier

Category = $N^0$
Example updates

Start with $p_{N'} = p_{incl} = 0.50$, $m = 1$, $n = 2.9$, $s = 10$

[from Pearl & Lidz 2009]

One Unamb <NP data point: $p_{N'} = 0.67$, $p_{incl} = 0.67$

One Unamb NP data point: $p_{N'} = 0.50$, $p_{incl} = 0.67$

One Sem-Syn Amb data point: $p_{N'} = 0.59$, $p_{incl} = 0.53$

One Syn Amb data point: $p_{N'} = 0.48$, $p_{incl} = 0.50$

_Pearl & Mis submitted_
Corpus analysis & learner input

Brown/Eve corpus (CHILDES: MacWhinney 2000): starting at 18 months

17,521 utterances of child-directed speech, 2874 referential pronoun utterances

<table>
<thead>
<tr>
<th></th>
<th>Baker</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unamb &lt;NP</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sem-Syn Amb</td>
<td>0.66%</td>
<td>0</td>
<td>242</td>
<td>242</td>
</tr>
<tr>
<td>Syn Amb</td>
<td>7.52%</td>
<td>0</td>
<td>0</td>
<td>2743</td>
</tr>
<tr>
<td>Unamb NP</td>
<td>8.42%</td>
<td>0</td>
<td>0</td>
<td>3073</td>
</tr>
<tr>
<td>Uninformative</td>
<td>83.4%</td>
<td>36500</td>
<td>36258</td>
<td>33515</td>
</tr>
</tbody>
</table>

Free parameters:
m=1, n=2.9 (from corpus estimates done by P&L)

s (concerns number of salient properties learner is considering):
Child may only be aware of a few salient properties or may consider all known properties (# of adjectives known by 16 months \(\approx 49\) (MacArthur CDI: Dale & Fenson 1996). Use range from 2 to 49.

Pearl & Mis submitted
Measures of success: LWF children’s behavior

In addition to directly assessing $p_{incl}$ and $p_{N'}$, we can measure how often a learner would reproduce the behavior in the LWF experiment ($p_{beh}$).

$$p_{beh} = \frac{rep_1 + rep_2 + rep_3}{rep_1 + 2 \times rep_2 + 2 \times rep_3}$$

2 choices

$s = 2$

Any outcome where learner looks at red bottle

Additional two outcomes where learner looks at other bottle

Category = $N'$, antecedent = “red bottle”

Category = $N'$, antecedent = “bottle”

Category = $N^0$, antecedent = “bottle”

$rep_1 = p_{N'} \times \frac{m}{m+n} \times p_{incl}$

$rep_2 = p_{N'} \times \frac{n}{m+n} \times (1 - p_{incl}) \times \frac{1}{s}$

$rep_3 = (1 - p_{N'}) \times (1 - p_{incl}) \times \frac{1}{s}$

Pearl & Mis submitted
Testing LWF’s assumption about what behavior means

In addition to directly assessing the learner’s behavior, we can assess LWF’s assumption that target behavior indicates the children have the target representation for *one*.

Is it possible to get target behavior in the LWF experiment without having the target representation for *one* in general (as measured by $p_{incl}$ and $p_{N'}$)?

Is it possible to get target behavior in the LWF experiment without having the target representation for *one* at the time the behavior is being produced?

$$p_{rep|beh} = \frac{rep_1}{rep_1 + rep_2 + rep_3}$$

the probability the look to the red bottle is because the learner has the target representation ($N'$, “red bottle”) given that the learner looks at the red bottle

*Pearl & Mis submitted*
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 2, 5, 7, 10, 20, 49 \)

<table>
<thead>
<tr>
<th>( \text{DirectUnamb} )</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{\text{incl}} )</td>
<td>0.50</td>
</tr>
<tr>
<td>( p_{N'} )</td>
<td>0.50</td>
</tr>
<tr>
<td>( p_{\text{beh}} )</td>
<td>0.56</td>
</tr>
<tr>
<td>( p_{\text{rep}\mid \text{beh}} )</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Since the input data include no Unambiguous <NP data, and those are the only data the Baker learner learns from, it learns nothing.

It is at chance for having the target syntactic and semantic representation.

It is only slightly above chance at producing the observed toddler behavior, and when it does, it unlikely to have the target representation when doing so.

Implication: This is an induction problem if only unambiguous <NP data are relevant.

*Pearl & Mis submitted*
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.
s = 2, 5, 7, 10, 20, 49

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.34-0.38 (0.03-0.05)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
</tr>
</tbody>
</table>

The learner robustly decides the antecedent should include the mentioned property.

However, the learner has a moderate dispreference for believing one is N’ when it is smaller than NP.

This is therefore not the target representation.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 2, 5, 7, 10, 20, 49 \)

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{incl} )</td>
<td>0.50 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{N'} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.34-0.38 (0.03-0.05)</td>
</tr>
<tr>
<td>( p_{beh} )</td>
<td>0.56 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{rep</td>
<td>beh} )</td>
<td>0.23 (&lt;0.01)</td>
</tr>
</tbody>
</table>

However...this learner still generates the observed toddler behavior (not what LWF would expect) with high probability, and has the target representation when doing so (is what LWF would expect).

Why? Because the learner believes so strongly that a mentioned property must be included in the antecedent, the only representation that allows this (e.g., \([N', \text{red}[N'_{\text{N}_0 \text{bottle}}]]\)) overpowers the other potential representations’ probabilities. Thus, the +OtherPro learner will conclude the antecedent includes the mentioned property, and so it and the referential pronoun referring to it (one) must be \( N' \text{ in this context} \) - even if the learner believes \( one \) is not \( N' \) in general.

*Pearl & Mis submitted*
## Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 7, 10, 20, 49 \)

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{\text{incl}} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.91-0.99 (&lt;0.01)</td>
<td>( &gt;0.99 ) (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{N'} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.98-0.99 (&lt;0.01)</td>
<td>0.37-0.38 (0.04-0.05)</td>
</tr>
<tr>
<td>( p_{\text{beh}} )</td>
<td>0.56 (&lt;0.01)</td>
<td>0.88-0.99 (&lt;0.01)</td>
<td>( &gt;0.99 ) (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{\text{rep</td>
<td>beh}} )</td>
<td>0.23 (&lt;0.01)</td>
<td>0.87-0.99 (&lt;0.01)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectFiltered learner (R&G, P&L’s filtered)

Variability, depending on the value of \( s \), which determines how suspicious a coincidence it is that the intended object just happens to have the mentioned property.

When \( s = 7 \) or above, this learner believes a mentioned property should be included in the antecedent and \textit{one is N’} when it is smaller than NP, which is similar to previous findings by R&G & P&L. In addition, it is likely to generate the observed toddler behavior, and have the target representation when doing so.

\[ \text{Pearl & Mis submitted} \]
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 5 \)

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{\text{incl}} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.68 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_N' )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.94 (&lt;0.01)</td>
<td>0.36 (0.04)</td>
</tr>
<tr>
<td>( p_{\text{beh}} )</td>
<td>0.56 (&lt;0.01)</td>
<td>0.70 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{\text{rep</td>
<td>beh}} )</td>
<td>0.23 (&lt;0.01)</td>
<td>0.58 (&lt;0.01)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectFiltered learner (R&G, P&L’s filtered)

Variability, depending on the value of \( s \), which determines how suspicious a coincidence it is that the intended object just happens to have the mentioned property.

However, when \( s=5 \), the learner is less sure the mentioned property should be included in the antecedent, which causes the learner to be less likely to generate the observed toddler behavior, and only slightly above chance at having the target representation when generating that behavior.

_Pearl & Mis submitted_
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 2 \)

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{incl} )</td>
<td>0.50 ((&lt;0.01))</td>
<td>0.02 ((&lt;0.01))</td>
<td>( \approx 0.99 ((&lt;0.01)) )</td>
</tr>
<tr>
<td>( p_N' )</td>
<td>0.50 ((&lt;0.01))</td>
<td>0.34 ((&lt;0.01))</td>
<td>0.34 (0.03)</td>
</tr>
<tr>
<td>( p_{beh} )</td>
<td>0.56 ((&lt;0.01))</td>
<td>0.50 ((&lt;0.01))</td>
<td>( \approx 0.99 ((&lt;0.01)) )</td>
</tr>
<tr>
<td>( p_{rep</td>
<td>beh} )</td>
<td>0.23 ((&lt;0.01))</td>
<td>(&lt;0.01 ) ((&lt;0.01))</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectFiltered learner (R&G, P&L’s filtered)

Variability, depending on the value of \( s \), which determines how suspicious a coincidence it is that the intended object just happens to have the mentioned property.

When \( s=2 \), the learner is sure the mentioned property should not be included in the antecedent, and prefer one to be \( N^0 \) when it is smaller than NP. This causes the learner to be at chance for generating the observed toddler behavior, and very unlikely to have the target representation when generating that behavior.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\[ s = 2, 5 \]

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{incl} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.02, 0.68 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{N'} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.34, 0.94 (&lt;0.01)</td>
<td>0.34-0.36 (0.03-0.04)</td>
</tr>
<tr>
<td>( p_{beh} )</td>
<td>0.56 (&lt;0.01)</td>
<td>0.50, 0.70 (&lt;0.01)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{rep</td>
<td>beh} )</td>
<td>0.23 (&lt;0.01)</td>
<td>&lt;0.01, 0.58 (&lt;0.01)</td>
</tr>
</tbody>
</table>

What’s going on?

If the suspicious coincidence isn’t strong enough, Sem-Syn ambiguous data don’t help the learner increase \( p_{incl} \) – in fact, they cause \( p_{incl} \) to drop. Because both \( p_{incl} \) and \( p_{N'} \) are used to calculate \( \phi_{incl} \) and \( \phi_{N'} \), a very low \( p_{incl} \) can eventually drag \( p_{N'} \) down.

Ex: \( s=2 \)
If the first 20 data points are Sem-Syn ambiguous data points, \( p_{incl} = 0.12 \) and \( p_{N'} = 0.48 \).
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 2, 5, 7, 10 \)

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{incl} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.02-0.96 (&lt;0.01)</td>
<td>&lt;0.01-0.38 (&lt;0.01-0.18)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{N^r} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.34-0.99 (&lt;0.01)</td>
<td>0.14-0.25 (&lt;0.01-0.06)</td>
<td>0.34-0.37 (0.03-0.04)</td>
</tr>
<tr>
<td>( p_{beh} )</td>
<td>0.56 (&lt;0.01)</td>
<td>0.50-0.98 (&lt;0.01)</td>
<td>0.50-0.53 (&lt;0.01-0.04)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{rep</td>
<td>beh} )</td>
<td>0.23 (&lt;0.01)</td>
<td>&lt;0.01-0.95 (&lt;0.01)</td>
<td>&lt;0.01-0.11 (&lt;0.01-0.11)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectEO learner (P&L’s EO)
Variability, depending on the value of \( s \), which determines how suspicious a coincidence it is that the intended object just happens to have the mentioned property.

When \( s \) is less than 10, the learner does not believe the mentioned property should be included in the antecedent, and prefers one to be \( N^0 \) when it is smaller than NP.
This causes the learner to be at chance at generating the observed toddler behavior, and unlikely to have the target representation when generating that behavior.

This is similar to what P&L previously found.

*Pearl & Mis submitted*
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

$s = 20, 49$

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.99 (&lt;0.01)</td>
<td><strong>0.93-0.99</strong> (&lt;0.01-0.03)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.99 (&lt;0.01)</td>
<td><strong>0.34-0.37</strong> (0.05)</td>
<td>0.37-0.38 (0.04-0.05)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.98-0.99 (&lt;0.01)</td>
<td><strong>0.79-0.94</strong> (0.02-0.07)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>0.98-0.99 (&lt;0.01)</td>
<td><strong>0.72-0.94</strong> (0.02-0.11)</td>
</tr>
</tbody>
</table>

Other learning strategies: DirectEO learner (P&L’s EO)
Variability, depending on the value of $s$, which determines how suspicious a coincidence it is that the intended object just happens to have the mentioned property.

However, when $s$ is 20 or 49, the learner strongly believes the mentioned property should be included in the antecedent, though it still prefers one to be $N^0$ when it is smaller than NP. This causes the learner to be likely to generate the observed toddler behavior, and likely to have the target representation when generating that behavior.

This is different from what P&L found, and more like the +OtherPro learner results.

Pearl & Mis submitted
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\[ s = 20, 49 \]

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{incl} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.99 (&lt;0.01)</td>
<td>0.93-0.99 (&lt;0.01-0.03)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{N'} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.99 (&lt;0.01)</td>
<td>0.34-0.37 (0.05)</td>
<td>0.37-0.38 (0.04-0.05)</td>
</tr>
<tr>
<td>( p_{beh} )</td>
<td>0.56 (&lt;0.01)</td>
<td>0.98-0.99 (&lt;0.01)</td>
<td>0.79-0.94 (0.02-0.07)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{rep</td>
<td>beh} )</td>
<td>0.23 (&lt;0.01)</td>
<td>0.98-0.99 (&lt;0.01)</td>
<td>0.72-0.94 (0.02-0.11)</td>
</tr>
</tbody>
</table>

What’s going on?

The flip side of what we saw with the R&G learner. If the suspicious coincidence is very strong, Sem-Syn ambiguous data help the learner increase \( p_{incl} \) (and \( p_{N'} \)) – in fact, they become almost as powerful as Unambiguous <NP data. Because both \( p_{incl} \) and \( p_{N'} \) are used to calculate \( \phi_{incl} \) and \( \phi_{N'} \), a very high \( p_{incl} \) can bolster \( p_{N'} \), and overpower the effect of the troublesome Syn ambiguous data.

Pearl & Mis submitted
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

\( s = 2, 5, 7, 10, 20, 49 \)

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{incl} )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.02-0.99 (&lt;0.01)</td>
<td>&lt;0.01-0.99 (&lt;0.01-0.18)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_N' )</td>
<td>0.50 (&lt;0.01)</td>
<td>0.34-0.99 (&lt;0.01)</td>
<td>0.14-0.37 (&lt;0.01-0.06)</td>
<td>0.34-0.38 (0.03-0.05)</td>
</tr>
<tr>
<td>( p_{beh} )</td>
<td>0.56 (&lt;0.01)</td>
<td>0.50-0.99 (&lt;0.01)</td>
<td>0.50-0.94 (&lt;0.01-0.07)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>( p_{rep</td>
<td>beh} )</td>
<td>0.23 (&lt;0.01)</td>
<td>&lt;0.01-0.95 (&lt;0.01)</td>
<td>&lt;0.01-0.94 (&lt;0.01-0.11)</td>
</tr>
</tbody>
</table>

Why isn’t the +OtherPro learner as susceptible to changing s values?

Unambiguous NP data only ever increase \( p_{incl} \), no matter what the value of \( s \). So, because there are so many of them, they can overwhelm the effect of Sem-Syn ambiguous data on \( p_{incl} \) (whether \( s \) is low or high). This helps keep \( p_N' \) from plummeting, though it still drops due to the troublesome Syn ambiguous data in the learner’s intake.
Learner results: Strategy comparison

Averages over 1000 simulations, standard deviations in parentheses.

$s = 2, 5, 7, 10, 20, 49$

<table>
<thead>
<tr>
<th></th>
<th>DirectUnamb</th>
<th>DirectFiltered</th>
<th>DirectEO</th>
<th>+OtherPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{incl}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.02-0.99 (&lt;0.01)</td>
<td>&lt;0.01-0.99 (&lt;0.01-0.18)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{N'}$</td>
<td>0.50 (&lt;0.01)</td>
<td>0.34-0.99 (&lt;0.01)</td>
<td>0.14-0.37 (&lt;0.01-0.06)</td>
<td>0.34-0.38 (0.03-0.05)</td>
</tr>
<tr>
<td>$p_{beh}$</td>
<td>0.56 (&lt;0.01)</td>
<td>0.50-0.99 (&lt;0.01)</td>
<td>0.50-0.94 (&lt;0.01-0.07)</td>
<td>&gt;0.99 (&lt;0.01)</td>
</tr>
<tr>
<td>$p_{rep</td>
<td>beh}$</td>
<td>0.23 (&lt;0.01)</td>
<td>&lt;0.01-0.95 (&lt;0.01)</td>
<td>&lt;0.01-0.94 (&lt;0.01-0.11)</td>
</tr>
</tbody>
</table>

Take away points:

An indirect positive evidence learning strategy has a beneficial impact on learning anaphoric one – it makes the learner’s behavior robust, no matter how suspicious a coincidence the Sem-Syn ambiguous data are (or aren’t).

A learner using an indirect positive evidence strategy can generate target behavior without reaching the target state – instead, this learner has a context-sensitive representation (depending on whether a property was mentioned).

Pearl & Mis submitted
Other induction problem characterizations

A different target state

target state
*One is category N’ and its antecedent includes the modifier.*

Just learning about the syntactic representation of *one* when it is smaller than NP.

Baker’s original proposal:
initial state includes *UG knowledge* that *one* is not $N^0$. 

Pearl & Mis submitted
Other induction problem characterizations

A different target state

target state
*One is category N’ and its antecedent includes the modifier.*

Just learning about the syntactic representation of *one* when it is smaller than NP.

Foraker et al.’s proposal:
Use Bayesian inference on the available syntactic data only, given domain-specific knowledge of complements and modifiers.
Modifiers & complements

Syntactic modifier: not “conceptually evoked by its head noun”, indicates noun string is $N'$
Ex: “the ball with dots” (I like the one with dots.)

Syntactic complement: “conceptually evoked by its head noun”, indicates noun string is $N^0$
Ex: “the side of the road” (*I waited by the one of the road.*)

---

**Modifier:** “with dots”  
Sister of $N'$

**Complement:** “of the road”  
Sister of $N^0$

Pearl & Mis submitted
The Foraker et al. learning strategy

Foraker et al. 2009

initial state
  Knowledge: Syntactic categories exist, in particular N⁰, N’, and NP.
  Knowledge: Anaphoric elements like *one* take linguistic antecedents of the same category.
+ Bias: Only syntactic data are useful.
+ Bias: Use Bayesian inference.
+ Bias: Learn from all linguistic elements that take complements or modifiers.
+ Knowledge: Complements conceptually evoke their head noun while modifiers do not.
+ Knowledge: Syntactic category N⁰ is sister to a complement, not a modifier.

This strategy was successful at learning *one* is category N’ (not N⁰) from child-directed speech data.

Pearl & Mis submitted
Foraker et al. bias types

Foraker et al. 2009

initial state

Knowledge: Syntactic categories exist, in particular N⁰, N’, and NP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.

+ Bias: Only syntactic data are useful.

This bias could be derived from the target knowledge only pertaining to the syntactic representation.
Foraker et al. bias types

Foraker et al. 2009

initial state
   Knowledge: Syntactic categories exist, in particular N0, N’, and NP.
   Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
+ (non-UG) Bias: Only syntactic data are useful.
+ Bias: Use Bayesian inference.

This bias is likely innate and domain-general.

Pearl & Mis submitted
Foraker et al. bias types

Foraker et al. 2009

initial state

Knowledge: Syntactic categories exist, in particular N\(^0\), N\(^\prime\), and NP.
Knowledge: Anaphoric elements like *one* take linguistic antecedents of the same category.

+ (non-UG) Bias: **Only syntactic data are useful.**
+ (non-UG) Bias: **Use Bayesian inference.**
+ Bias: Learn from all linguistic elements that take complements or modifiers.

This indirect positive evidence bias is clearly **domain-specific**. It could be specified **innately**, though it could possibly be **derived** by noticing salient properties of nominal phrases.

_Pearl & Mis submitted_
Foraker et al. bias types

Foraker et al. 2009
initial state
  Knowledge: Syntactic categories exist, in particular N⁰, N’, and NP.
  Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
  + (non-UG) Bias: Only syntactic data are useful.
  + (non-UG) Bias: Use Bayesian inference.
  + (UG?) Bias: Learn from all linguistic elements that take complements or modifiers.
  + Knowledge: Complements conceptually evoke their head noun while modifiers do not.

Knowing complements evoke their head nouns while modifiers do not is domain-specific knowledge that is not obviously derivable.
Foraker et al. bias types

Foraker et al. 2009

initial state

Knowledge: Syntactic categories exist, in particular N⁰, N’, and NP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.

+ (non-UG) Bias: Only syntactic data are useful.
+ (non-UG) Bias: Use Bayesian inference.
+ (UG?) Bias: Learn from all linguistic elements that take complements or modifiers.
+ (UG) Knowledge: Complements conceptually evoke their head noun while modifiers do not.

+ Knowledge: Syntactic category N⁰ is sister to a complement, not a modifier.

Knowing N⁰ is sister to complement is also domain-specific knowledge that is not obviously derivable.
Foraker et al. bias types

Foraker et al. 2009

initial state
  Knowledge: Syntactic categories exist, in particular N⁰, N’, and NP.
  Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
+ (non-UG) Bias: Only syntactic data are useful.
+ (non-UG) Bias: Use Bayesian inference.
+ (UG?) Bias: Learn from all linguistic elements that take complements or modifiers.
+ (UG) Knowledge: Complements conceptually evoke their head noun while modifiers do not.
+ (UG) Knowledge: Syntactic category N⁰ is sister to a complement, not a modifier.

Upshot: This form of the induction problem leads to a different proposal for the contents of UG, even when Bayesian inference is used.

Pearl & Mis submitted
Other induction problem characterizations

A different initial & target state: Alternate theoretical representations

\[ N^0, N', \text{and NP} \quad \text{vs.} \quad N^0, N', \text{NP, and DP} \]
Other induction problem characterizations

A different initial & target state: Syntactic categories N^0, N', NP, DP

initial state

Knowledge: Syntactic categories exist, in particular N^0, N', NP, and DP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
Bias: Only direct evidence of one is useful.
Bias: Only unambiguous evidence of one is useful.

target state

Knowledge: In utterances like “Look, a red bottle! Look, another one!”, one is category NP and so its antecedent includes the modifier (“red”).
Other induction problem characterizations

A different initial & target state: Syntactic categories N^0, N', NP, DP

What an indirect positive evidence strategy like +OtherPro would do

initial state

Knowledge: Syntactic categories exist, in particular N^0, N', NP, and DP.
Knowledge: Anaphoric elements like one take linguistic antecedents of the same category.
- **Bias**: Only direct evidence of one is useful.
- **Bias**: Only unambiguous evidence of one is useful.
+ (non-UG) **Bias**: Use Bayesian inference
+ (UG?) **Bias**: Learn from other pronoun data.
Other induction problem characterizations

A different initial & target state: Syntactic categories $N^0$, $N'$, NP, DP

What an indirect positive evidence strategy like $+\text{OtherPro}$ would do

(1) Syn ambiguous data still ambiguous between two categories ($N^0$ and $N'$), and Bayesian inference causes learner to prefer the hypotheses that includes fewer strings, which is still the $N^0$ category. ($N'$ includes noun +complement strings)

Syn ambiguous data still cause $p_{N'}$ to drop, though perhaps not as fast.
Other induction problem characterizations

A different initial & target state: Syntactic categories N⁰, N’, NP, DP

What an indirect positive evidence strategy like +OtherPro would do

(2) Sem-Syn ambiguous data still ambiguous between three antecedents. When s is high enough (>5), the suspicious coincidence still causes the learner to increase $p_{incl}$.

Sem-Syn ambiguous data still cause $p_{incl}$ to increase when the suspicious coincidence is strong enough.
Other induction problem characterizations

A different initial & target state: Syntactic categories $N^0$, $N'$, NP, DP

What an indirect positive evidence strategy like +OtherPro would do

(3) Unambiguous <NP data still indicate antecedent that includes modifier – it’s just that the category label is NP (rather than $N'$).

$p_{incl}$ and $p_{NP}$ both increase.

Unambiguous <NP data still cause $p_{incl}$ and the category that includes the modifier (NP) to increase.
Other induction problem characterizations

A different initial & target state: Syntactic categories N⁰, N’, NP, DP

What an indirect positive evidence strategy like +OtherPro would do

(4) Unambiguous NP data still indicate antecedent that includes modifier – it’s just that the category label is DP (rather than NP).

p_{incl} still increases.

Unambiguous NP data still cause p_{incl} to increase.
Other induction problem characterizations

A different initial & target state: Syntactic categories N₀, N’, NP, DP

What an indirect positive evidence strategy like +OtherPro would do

Given that the updates from the different data types are effectively the same, the overall outcome should be similar: \( p_{incl} \) should be high while \( p_{NP} \) should be low. (Note: \( p_{N'} \) should also be very low, since no data cause it to increase.)

Non-target context-dependent representation.

\( p_{incl} = \text{high}, \ p_{NP} = \text{low} \)

LWF experiment: target behavior (and target representation when displaying that behavior) because of \( p_{incl} \).