Announcements

- a9 due Tuesday, May 1\textsuperscript{st}, in class
- final exam Tuesday May 8\textsuperscript{th} in class
- final paper due Friday May 11\textsuperscript{th}, 11:59pm
- final review sometime next week
Schedule

- Last Week
  - Constructions, ECG
- This Week
  - Models of language learning
  - Embodied Construction Grammar learning
- Next Week
  - Open lecture
  - Wrap-Up
Questions

1. Why is learning language difficult?
2. What are the “paucity of the stimulus” and the “opulence of the substrate”?
3. What is Gold's Theorem?
4. How does the analyzer use the constructions to parse a sentence?
5. How can we learn new ECG constructions?
6. What are ways to re-organize and consolidate the current grammar?
7. What metric is used to determine when to form a new construction?
Difficulty of learning language

- What makes learning language difficult?
  - How many sentences do children hear?
  - How often are those sentences even correct?
  - Even when they're correct, how often are they complete?
  - How often are they corrected when saying something wrong?
  - How many possible languages are there?
Larger context

- War!
  - Is language innate?
  - Covered in book
Questions

1. Why is learning language difficult?
2. What are the “paucity of the stimulus” and the “opulence of the substrate”?
3. What is Gold's Theorem?
4. How does the analyzer use the constructions to parse a sentence?
5. How can we learn new ECG constructions?
6. What are ways to re-organize and consolidate the current grammar?
7. What metric is used to determine when to form a new construction?
Paucity and Opulence

- Poverty of the stimulus
  - Coined to suggest how little information children have to learn from
- Opulence of the substrate
  - Opulence = “richness”
  - Coined in response to suggest how much background information children have
Questions

1. Why is learning language difficult?
2. What are the “paucity of the stimulus” and the “opulence of the substrate”?
3. What is Gold's Theorem?
4. How does the analyzer use the constructions to parse a sentence?
5. How can we learn new ECG constructions?
6. What are ways to re-organize and consolidate the current grammar?
7. What metric is used to determine when to form a new construction?
Gold's Theorem

- Suppose that you have an infinite number of languages
  - language = “set of legal sentences”
- Suppose that for every language Ln, there is a bigger language Ln+1
  - makes every sentence, and then some
- There's some language Linfinity
  - contains all the sentences in all other grammars
- You can arrange data so that no one ever learns Linfinity
Questions

1. Why is learning language difficult?

2. What are the “paucity of the stimulus” and the “opulence of the substrate”?

3. What is Gold's Theorem?

4. How does the analyzer use the constructions to parse a sentence?

5. How can we learn new ECG constructions?

6. What are ways to re-organize and consolidate the current grammar?

7. What metric is used to determine when to form a new construction?
Analyzing “You Throw The Ball”

**FORM (sound)**
- t1 before t2
- t2 before t3
- “you”
- “throw”
- “the”
- “ball”
- “block”

**MEANING (stuff)**
- t2.thrower ↔ t1
- t2.throwee ↔ t3
- Addressed Addressee subcase of Human
- Schema Throw roles: thrower throwee throwee
- Schema Ball subcase of Object
- schema Block subcase of Object
Another way to think of the SemSpec

Discourse Element
- type = statement
- content
- speaker
- addressee

Referent Schema
- category = human
- gender = male
- count = one
- specificity = known
- resolved = harry2
- modifications

Self Motion Schema
- mover
- action
- direction

WalkX Schema
- agent
  - speed = slow
  - tense = past
  - aspect = complete

TL Schema
- landmark = Berkeley
- trajector

SPG Schema
- trajector
- goal
- source
- path
Analyzing in ECG

create a recognizer for each construction in the grammar
for each level $j$ (in ascending order)

repeat
  repeat
    for each recognizer $r$ in $j$
      for each position $p$ of utterance
        initiate $r$ starting at $p$
  until we don't find anything new
Recognizer for the Transitive-Cn

- an example of a level-1 construction is Red-Ball-Cn
- each recognizer looks for its constituents in order (the ordering constraints on the constituents can be a partial ordering)
Learning-Analysis Cycle (Chang, 2004)

1. Learner passes input (Utterance + Situation) and current grammar to Analyzer.


1. Learner updates grammar:
   a. Hypothesize new map.
   b. Reorganize grammar (merge or compose).
   c. Reinforce (based on usage).

(Utterance, Situation)

Analyze

Semantic Specification, Constructional Analysis

Constructions

Reorganize

Hypothesize
Questions

1. Why is learning language difficult?
2. What are the “paucity of the stimulus” and the “opulence of the substrate”?
3. What is Gold's Theorem?
4. How does the analyzer use the constructions to parse a sentence?
5. How can we learn new ECG constructions?
6. What are ways to re-organize and consolidate the current grammar?
7. What metric is used to determine when to form a new construction?
Usage-based Language Learning

Acquisition

Reorganize

Hypothesize

Production

(Utterance, Situation)

Constructions

(Comm. Intent, Situation)

Analyze

Partial Analysis

Hypothesize

Generate

Comprehension

Acquisition

Production

Utterance
Basic Learning Idea

- The learner’s current grammar produces a certain analysis for an input sentence
- The context contains richer information (e.g. bindings) that are unaccounted for in the analysis
- Find a way to account for these meaning relations (by looking for corresponding form relations)
Initial Single-Word Stage

FORM (sound)

- "you"
- "throw"
- "ball"
- "block"

MEANING (stuff)

- schema Addressee subcase of Human
- schema Throw roles: thrower throwee
- schema Ball subcase of Object
- schema Block subcase of Object

lexical constructions

you

throw

ball

block
New Data: “You Throw The Ball”

**FORM**
- “you”
- “throw”
- “the”
- “ball”

**MEANING**
- **Addressee**
  - schema Addressee
  - subcase of Human
- **Throw**
  - roles: thrower, throwee
- **Ball**
  - schema Ball
  - subcase of Object
- **Block**
  - schema Block
  - subcase of Object

**SITUATION**
- Self
- Addressee
  - Throw
  - thrower
  - throwee
- Ball
Relational Mapping Scenarios

**Throw ball**

- **Form relation**: $A_f \rightarrow A \rightarrow A_m$  
- **Role-filler**: $B_f \rightarrow B \rightarrow B_m$

**Roles**:  
- $A_f \rightarrow A_m$  
- $B_f \rightarrow B_m$

**Throw throwee ↔ ball**

- **Form relation**: $A_f \rightarrow A \rightarrow A_m$  
- **Role-filler**: $B_f \rightarrow B \rightarrow B_m$

**Put ball down**

- **Form relation**: $A_f \rightarrow A \rightarrow A_m$  
- **Role-filler**: $B_f \rightarrow B \rightarrow B_m$

- **Put mover ↔ ball**
  - **Form relation**: $A_f \rightarrow A \rightarrow A_m$  
  - **Role-filler**: $B_f \rightarrow B \rightarrow B_m$

- **Put down.tr ↔ ball**

**Nomi ball**

- **Possession possessor ↔ Nomi**
- **Possession possessed ↔ ball**
Questions

1. Why is learning language difficult?
2. What are the “paucity of the stimulus” and the “opulence of the substrate”?
3. What is Gold's Theorem?
4. How does the analyzer use the constructions to parse a sentence?
5. How can we learn new ECG constructions?
6. What are ways to re-organize and consolidate the current grammar?
7. What metric is used to determine when to form a new construction?
Merging Similar Constructions

**throw** the **block**

**throw** before **block**

**throw** before **Object**

**throw** before **ball**

**throw**-**ing** the **ball**

**throw** before-**s** **ing**

**THROW**.throwee = Block

**THROW**.throwee = **Object**

Throw.throwee = Ball

Throw.aspect = ongoing
Resulting Construction

construction THROW-OBJECT

constructional

constituents

\( t : \text{THROW} \)

\( o : \text{OBJECT} \)

form

\( t_f \) before \( o_f \)

meaning

\( t_m, \text{throwee} \leftrightarrow o_m \)
Composing Co-occurring Constructions

- \textit{throw the ball}
  - \textit{throw before ball}
  - \textit{ball before off}
  - \textit{ball off}

- \textit{ball off}

- \textit{Throw.throwee = Ball}

- \textit{THROW-BALL-OFF}

- \textit{Motion m}
  - m.mover = Ball
  - m.path = Off

- \textit{THROW-BALL-OFF}

- \textit{ball before off}

- \textit{ball before off}

- \textit{ball before off}
Resulting Construction

coloration THROW-BALL-OFF

colorational constituents
t : THROW
b : BALL
o : OFF

form
t_f before b_f
b_f before o_f

meaning
evokes MOTION as m
t_m.throwee ↔ b_m
m.mover ↔ b

Questions

1. Why is learning language difficult?
2. What are the “paucity of the stimulus” and the “opulence of the substrate”?
3. What is Gold's Theorem?
4. How does the analyzer use the constructions to parse a sentence?
5. How can we learn new ECG constructions?
6. What are ways to re-organize and consolidate the current grammar?
7. What metric is used to determine when to form a new construction?
Minimum Description Length

- Choose grammar $G$ to minimize $\text{cost}(G \mid D)$:
  - $\text{cost}(G \mid D) = \alpha \cdot \text{size}(G) + \beta \cdot \text{complexity}(D \mid G)$
  - Approximates Bayesian learning;
    $\text{cost}(G \mid D) \approx 1/\text{posterior probability} \approx 1/P(G \mid D)$

- **Size of grammar** = $\text{size}(G) \approx 1/\text{prior} \approx 1/P(G)$
  - favor fewer/smaller constructions/roles; isomorphic mappings

- **Complexity of data given grammar** $\approx 1/\text{likelihood}$
  $\approx 1/P(D \mid G)$
  - favor simpler analyses
    (fewer, more likely constructions)
  - based on derivation length + score of derivation
Size Of Grammar

- Size of the grammar $G$ is the sum of the size of each construction:

$$\text{size}(G) = \sum_{c \in G} \text{size}(c)$$

- Size of each construction $c$ is:

$$\text{size}(c) = n_c + m_c + \sum_{e \in c} \text{length}(e)$$

where

- $n_c$ = number of constituents in $c$, $e \in c$
- $m_c$ = number of constraints in $c$,
- $\text{length}(e) = \text{slot chain length of element reference } e$
Example: The Throw-Ball Cxn

construction THROW-BALL

constructional constituents
  t : THROW
  b : BALL

form
  t₇ before b₇

meaning
  tₐ.throwee ↔ bₐ

size(c) = n_c + m_c + \sum_{e \in c} \text{length}(e)

size(THROW-BALL)
= 2 + 2 + (2 + 3) = 9
Complexity of Data Given Grammar

- Complexity of the data \( D \) given grammar \( G \) is the sum of the analysis score of each input token \( d \):

\[
\text{complexity}(D \mid G) = \sum_{d \in D} \text{score}(d)
\]

- Analysis score of each input token \( d \) is:

\[
\text{score}(d) = \sum_{c \in d} \left( \text{weight}_c + \eta \cdot \sum_{r \in c} \mid \text{type}_r \mid \right) + \text{height}_d + \text{semfit}_d
\]

where

- \( c \) is a construction used in the analysis of \( d \)
- \( \text{weight}_c \approx \) relative frequency of \( c \),
- \( \mid \text{type}_r \mid = \) number of ontology items of type \( r \) used,
- \( \text{height}_d = \) height of the derivation graph,
- \( \text{semfit}_d = \) semantic fit provided by the analyzer
Final Remark

• The goal here is to build a cognitive plausible model of language learning

• A very different game that one could play is unsupervised / semi-supervised language learning using shallow or no semantics
  - statistical NLP
  - automatic extraction of syntactic structure
  - automatic labeling of frame elements

• Fairly reasonable results for use in tasks such as information retrieval, but the semantic representation is very shallow