The Neural Basis of Thought and Language

Week 14
Administrivia

- Final exam review session Monday
  - 7-9pm in 310 Soda
- Final in class next Thursday, May 8th
- Be there on time!
- Format:
  - closed books, closed notes
  - short answers, no blue books
- Final paper due on bSpace on Monday, May 11
Questions

1. How does the analyzer use the constructions to parse a sentence?
2. How can we learn new ECG constructions?
3. What are ways to re-organize and consolidate the current grammar?
4. 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
- Schema Ball subcase of Object
- Schema Block subcase of Object

**Schema**
- Throw
  - thrower
  - throwee
- Ball
  - subcase of Object
- Block
  - subcase of Object
Do not forget the SemSpec!
create a recognizer for each construction in the grammar
for each level j (in ascending order)

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)
Questions
1. How does the analyzer use the constructions to parse a sentence?
2. How can we learn new ECG constructions?
3. What are ways to re-organize and consolidate the current grammar?
4. What metric is used to determine when to form a new construction?
1. Learner passes input (Utterance + Situation) and current grammar to Analyzer.


3. Learner updates grammar:
   a. Hypothesize new map.
   b. Reorganize grammar (merge or compose).
   c. Reinforce (based on usage).
Usage-based Language Learning

(Utterance, Situation) → Analyze → Partial Analysis → Comprehension

Constructions

Reorganize

(Hypothesize) → (Utterance, Situation) → Generate → Utterance → Production

(Comm. Intent, Situation)
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"

lexical constructions

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
New Data: “You Throw The Ball”
Relational Mapping Scenarios

throw ball $\leftrightarrow$ throw.throwee $\leftrightarrow$ ball

put ball down $\leftrightarrow$ put.mover $\leftrightarrow$ ball

down.tr $\leftrightarrow$ ball

Nomi ball $\leftrightarrow$ possession.possessor $\leftrightarrow$ Nomi
possession.possessed $\leftrightarrow$ ball
Questions

1. How does the analyzer use the constructions to parse a sentence?
2. How can we learn new ECG constructions?
3. What are ways to re-organize and consolidate the current grammar?
4. 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.throwee = Block

throw before ball

Throw.aspect = ongoing

throw the block

THROW.throwee = Object

throw before-thing

THROW-OBJECT

throwing the ball

Throw.throwee = Ball

throw before-thing

Throw.aspect = ongoing
Resulting Construction

**construction** THROW-OBJECT

**constructional**

**constituents**

\( t : \text{THROW} \)

\( o : \text{OBJECT} \)

**form**

\( t_f \) before \( o_f \)

**meaning**

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

\[ \text{throw the ball} \]
\[ \text{throw before ball} \]
\[ \text{ball before off} \]
\[ \text{ball off} \]
\[ \text{THROW-BALL-OFF} \]

Throw.throwee = Ball
Motion m
m.mover = Ball
m.path = Off

Motion m
m.mover = Ball
m.path = Off
Resulting Construction

construction THROW-BALL-OFF

constructional

constituents

\( t : \text{THROW} \)
\( b : \text{BALL} \)
\( o : \text{OFF} \)

form

\( t_f \text{ before } b_f \)
\( b_f \text{ before } o_f \)

meaning

evokes \text{MOTION as} m

\( t_m \text{-throwee } \leftrightarrow b_m \)
\( m \text{-mover } \leftrightarrow b_m \)
\( m \text{-path } \leftrightarrow o_m \)
Questions

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

2. How can we learn new ECG constructions?

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

4. What metric is used to determine when to form a new construction?
Size Of Grammar

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

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

- Size of each construction $c$ is:

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

where

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

**construction** THROW-BALL

**constructional constituents**
- t : THROW
- b : BALL

**form**
- $t_f$ before $b_f$

**meaning**
- $t_m$.throwee ↔ $b_m$

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

$$size(\text{THROW-BALL}) = 2 + 2 + (2 + 3) = 9$$
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:

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

where

- $c$ is a construction used in the analysis of $d$
- $\text{weight}_c \approx$ relative frequency of $c$,
- $|\text{type}_r|$ = number of ontology items of type $r$ used,
- $\text{height}_d$ = height of the derivation graph,
- $\text{semfit}_d$ = semantic fit provide by the analyzer
Minimum Description Length

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

- **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** $\approx 1/\text{likelihood}$
  $\approx 1/P(D|G)$
  - favor simpler analyses
    (fewer, more likely constructions)
  - based on derivation length + score of derivation
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.