

Grounding Language in Robot Control Systems

Dieter Fox, Luke Zettlemoyer
Cynthia Matuszek, Nicholas FitzGerald, Liefeng Bo

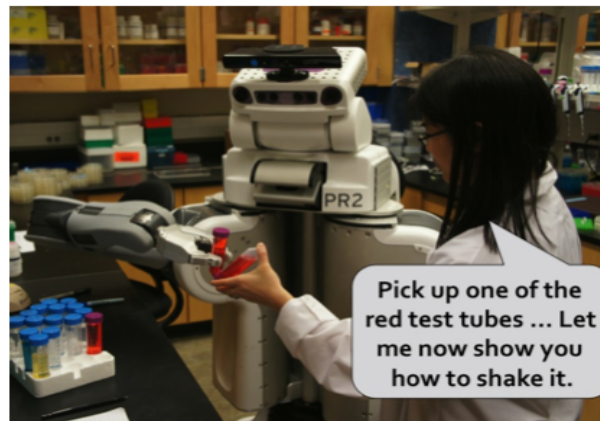
University of Washington

Funded by Intel Science and Technology Center for Pervasive Computing,
Robotics Collaborative Technology Alliance Program, and National Science
Foundation



Goal

- Enable teachable robots that can
 - interpret and execute upon rich human input
 - interactively learn objects, attributes, skills, tasks



Smart Wetlab Assistant

Interactive Grounding

- Parse language, gestures, gaze, and body motion into formal reasoning system
 - Semantic NLP style parsing of multi-modal input
 - Activity recognition
- Ground symbols in real world perception and actuation
 - Interactive object / attribute learning
 - Skill learning via interactive demonstration

Outline

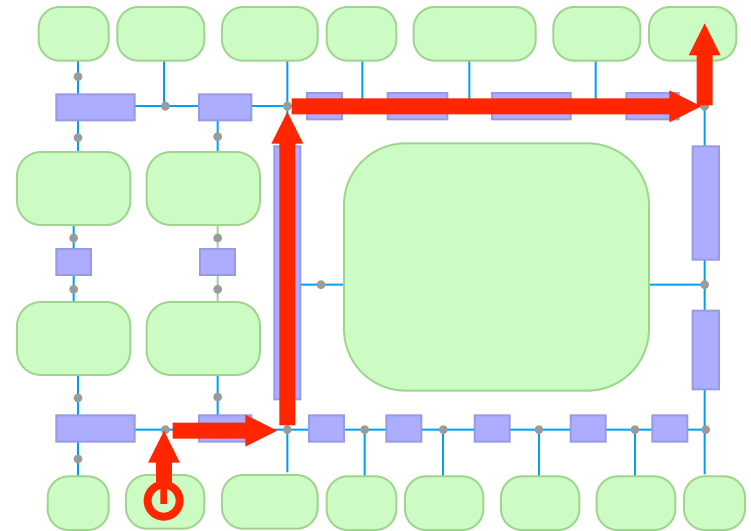
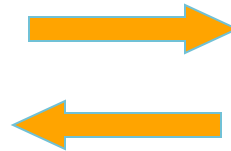
- Direction following
- Learning and grounding object attributes
- Discussion

Grounding Language in Robot Control

- Logic-based representations for robot control
[Beetz-etal, Lakemeyer-Haehnel-etal, Kress-Gazit-etal, Baral-etal, ...]
- Direction following in rich, simulated environments
[MacMahon-Kuipers]
- Ground parsed NLP in world and action models for direction following and forklift operation [Tellex-Kollar-Roy-etal]
- Learn to parse NLP for RoboCup and direction following (w/ minimal supervision) [Mooney-etal]
- Learning for semantic parsing [Zettlemoyer-etal, Liang-etal, ...]
- Language grounding for semantic mapping [Kruijff-etal]

Route Instruction Following

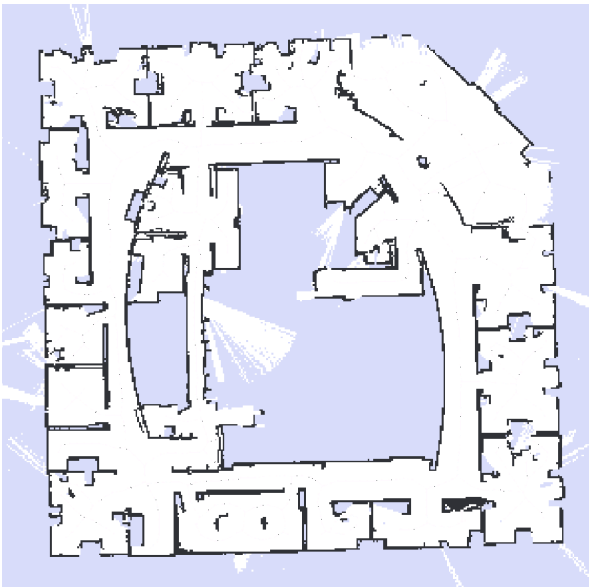
"Leave the room and turn right, take the first left, go past the meeting room and go right, then go to the end of the hall and turn left."



- Humans pretty bad at directions (~70% accurate)*
 - Missed turns, right/left confusion, ...
- Several sources of uncertainty
 - Map labeling errors, parsing, instructions are uncertain

* Riesbeck, 1980; Macmahon, 2006

Topological / Semantic Mapping

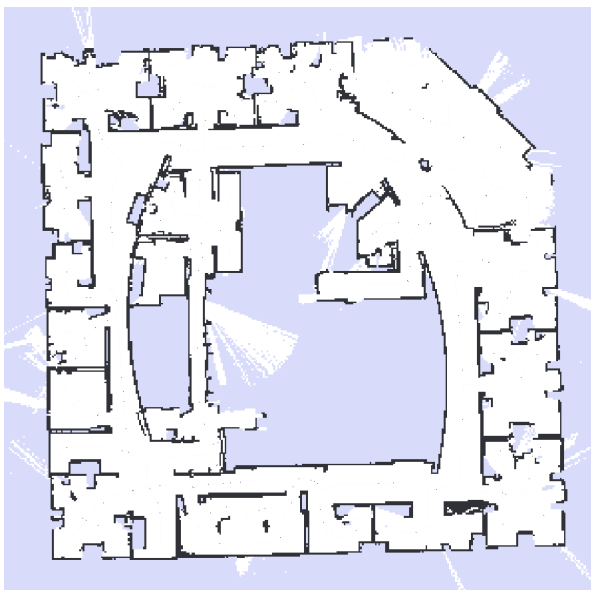


Occupancy grid map

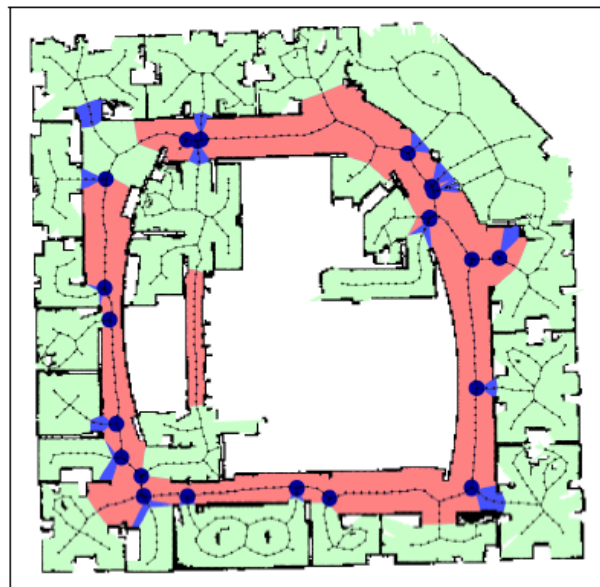
- Not sufficiently rich for communication with people or grounding natural language.
- Need to reason about topological structure and types of places.

Topological / Semantic Mapping

- Voronoi Random Field: CRF defined over Voronoi graph labels grid cells as room, hallway, junction, entry way.



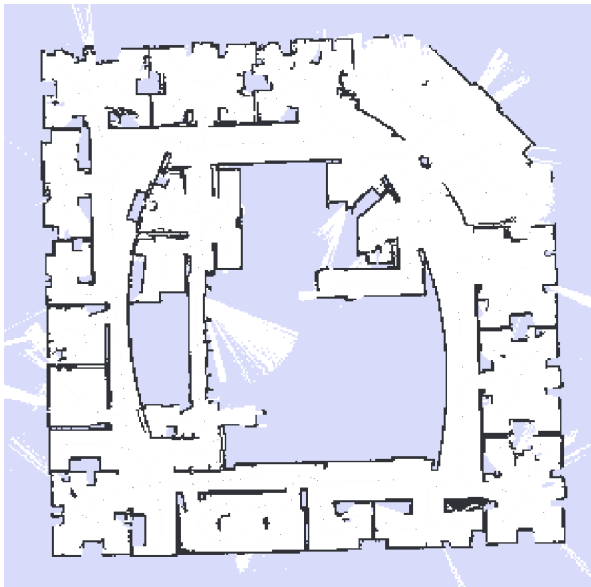
Occupancy grid map



Spatial labelling

$$p(\mathbf{x} | \mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c \in C} \Phi_c(\mathbf{x}_c, \mathbf{z}_c) = \frac{1}{Z(\mathbf{z})} \exp \left\{ \sum_{c \in C} \mathbf{w}_c^T \mathbf{f}_c(\mathbf{x}_c, \mathbf{z}_c) \right\}$$

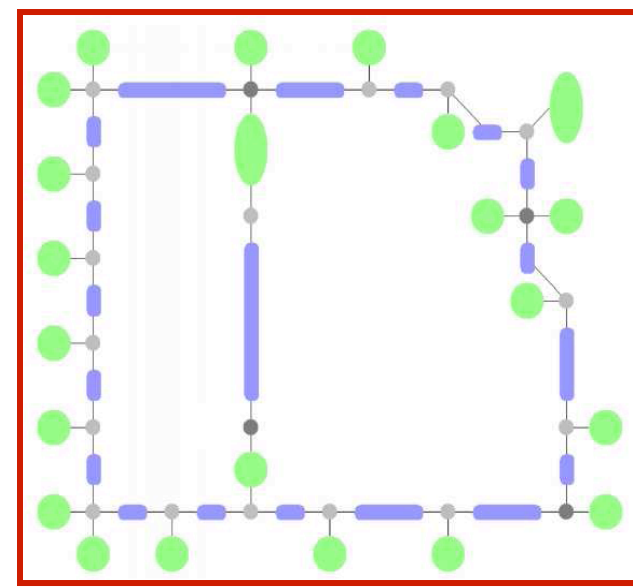
Topological / Semantic Mapping



Occupancy grid map



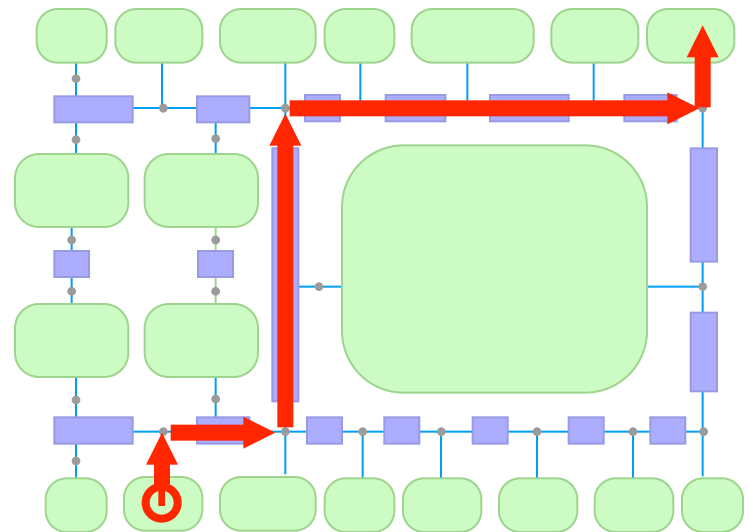
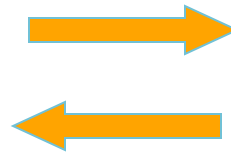
Spatial labelling



Topological map

Route Instruction Following

"Leave the room and turn right, take the first left, go past the meeting room and go right, then go to the end of the hall and turn left."

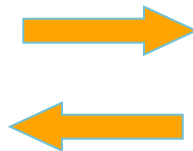


Statistical Machine Translation

- Source language: natural language directions
- Target language: path description language grounded in labeled map
- Learn to parse based on source / target pairs

[Wong-Mooney: HLTC-06]

"Go down the
hall and take the
second left."



(go (hall) (4junction 1)
(hall) (3junction lt 0)
(room))

Source Language: NL

Target Language: Formal

- Path descriptions readily transformed to robot actions
- Trained on >1,000 steps, tested on 14 routes, 71% success

Key Limitations

- Ground directly into the map, no target concepts such as **while** or **counting**
- Parser must be able to produce many **possible** groundings:

"Take the second left."



```
1: (go (hall) (4junction 1)
      (hall) (3junction lt 0) (room))
2: (go (room) (4junction 1)
      (room) (3junction lt 0) (room))
3: (go (hall) (4junction 1)
      (hall) (3junction rt 1) (room)
      (3junction lt 0) (room))
...
```

- Even worse:
"go to the end of the hall,"
"keep turning right until you can't any more", ...
- Extremely hard to learn concepts such as *counting*

Grounding in Control System

“Go left to the end of the hall.”

```
(do-sequentially  
  (turn-left current-loc)  
  (do-until  
    (or  
      (not (exists forward-loc))  
      (room forward-loc))  
    (move-to forward-loc)))
```

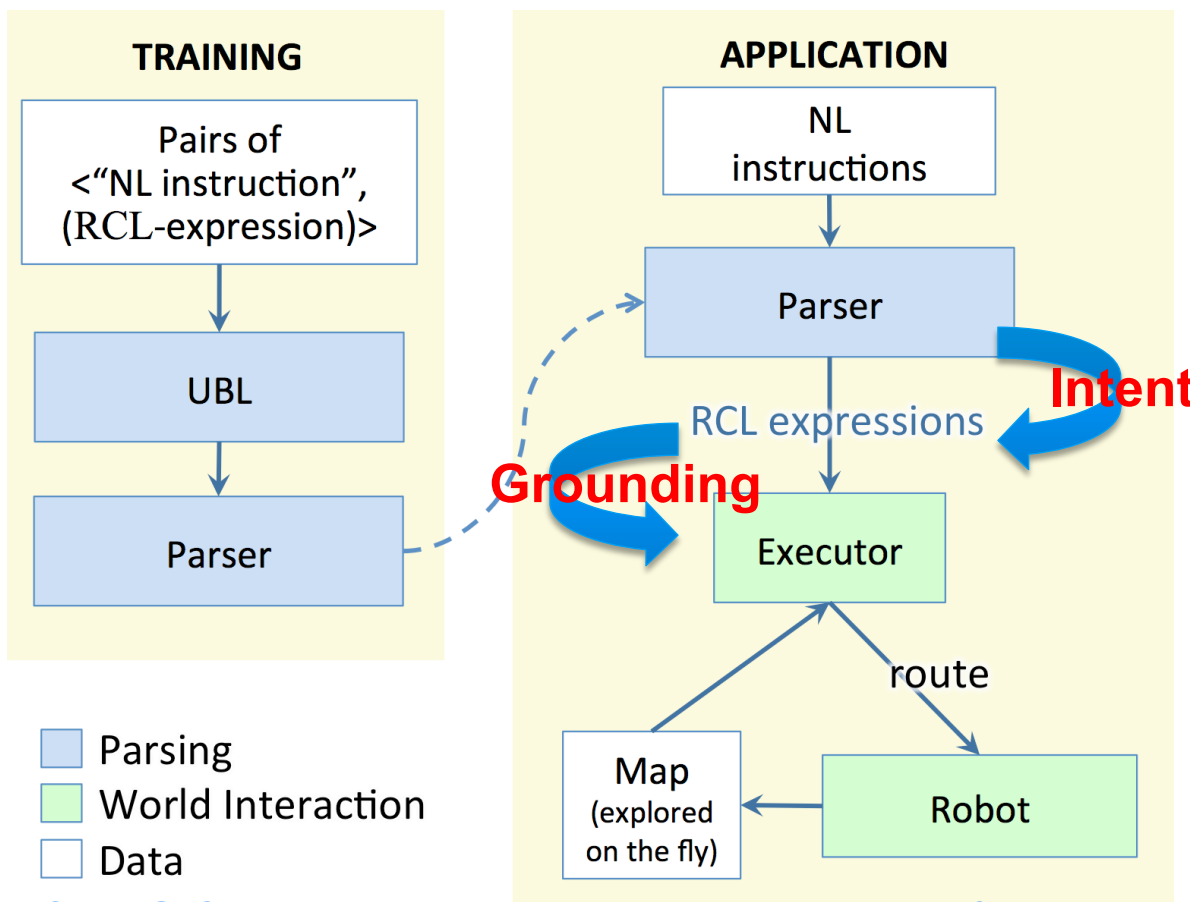
“Go to the third junction and take a right.”

```
(do-sequentially  
  (do-n-times 3  
    (do-sequentially  
      (move-to forward-loc)  
      (do-until  
        (junction current-loc)  
        (move-to forward-loc))))  
  (turn-right current-loc))
```

- Assumptions: robot can execute actions, knows places, and determine conditionals

Grounding System

- Formal robot control language (lambda-calculus)



- RCL expresses procedural intent
- Disambiguation performed against map on-line as robot navigates

Categorial Combinatory Grammars

- Capture both **syntax** and **semantics** of language
- Parse sentences to expressions in **lambda calculus**
- **Lexical entries** such as

go to $\vdash S / NP : \lambda x. moveTo(x)$

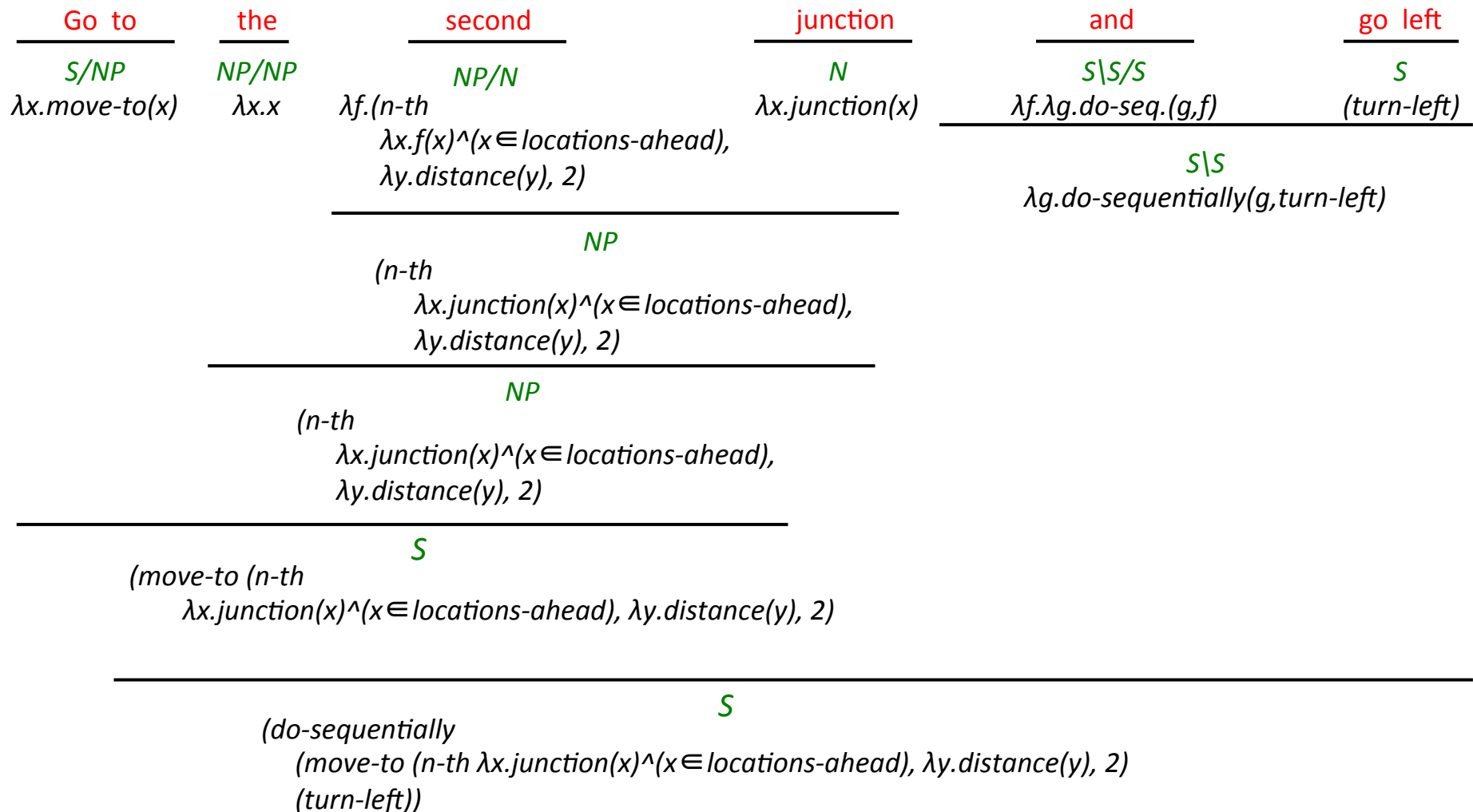
junction $\vdash N : \lambda x. junction(x)$

along with **combinatory rules** define space of parses.

- **Probabilistic CCG** defines log-linear model over sentence x , parse y , logical form z

$$p(y, z \mid x; \theta, \Lambda) = \frac{e^{\theta \cdot \phi(x, y, z)}}{\sum_{y', z'} e^{\theta \cdot \phi(x, y', z')}} \quad [Clark-Curran: EMNLP-03]$$

Example CCG Parse



Learning Probabilistic CCGs

- **Input:** Example pairs of sentences and logical forms
- **Output:** PCCG lexicon and feature weights
- **Structure learning:** Generate lexical items from examples via combination or splitting rules
- **Parameter estimation:**

Expected feature counts
given commands x_i and
target meanings z_i

Expected feature counts
given commands x_i

$$\frac{\partial \log(p(z_i | x_i; \theta, \Lambda))}{\partial \theta_j} = E_{p(y|x_i, z_i; \Theta, \Lambda)} [\phi_j(x_i, y, z_i)] - E_{p(y, z|x_i; \Theta, \Lambda)} [\phi_j(x_i, y, z)]$$
- Data driven updates, add lexical items only when involved in generating most likely parse of formula

Experimental Results

- Training corpus:
 - ~1000 route instructions
- Testing: novel instructions on novel maps
 - 1000 short instructions (1 clause)
 - 200 longer routes (avg. 5 clauses)
- How often would robot reach goal **by intended path?**

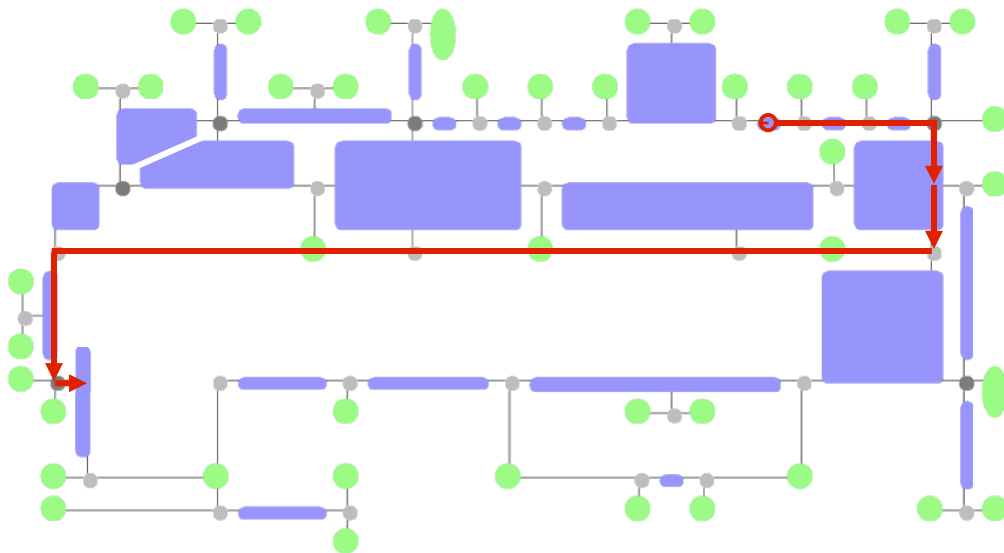
Successes:

Short	924/1000	92.4%
Long	125/200	62.5%

- Also collecting Mechanical Turk dataset

Example Parse

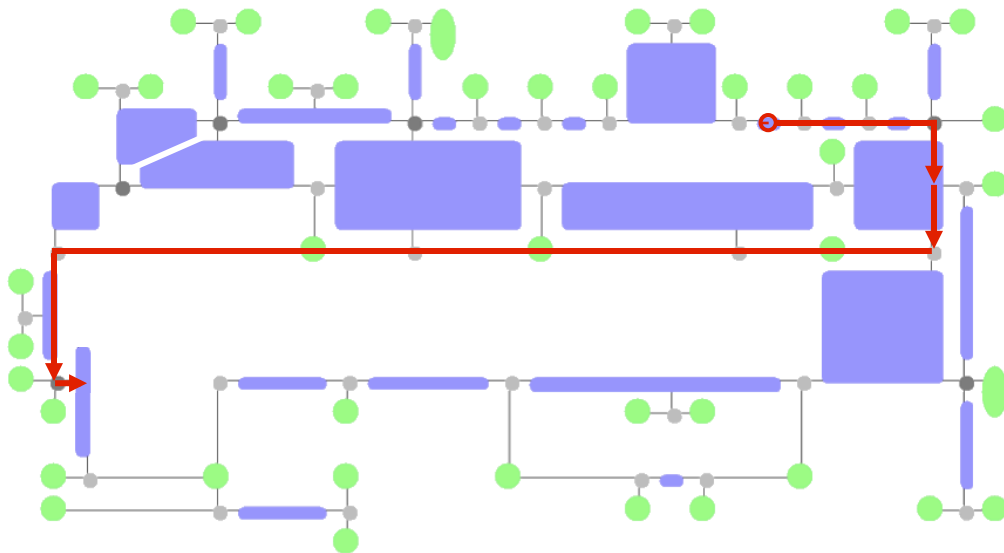
- Go past two junctions and turn right, go forward to the 3-way intersection, take the first right, go straight through the second junction then go left, and turn left again.



```
(do-sequentially
  (do-sequentially
    (do-n-times 2
      (do-sequentially
        (do-until
          (junction current-loc)
          (move-to forward-loc))
        (move-to forward-loc))))
    (turn-right current-loc))
  (do-until
    (junction3 current-loc)
    (move-to forward-loc))
  (turn-right current-loc)
  (do-sequentially
    (do-n-times 2
      (do-sequentially
        (do-until
          (junction current-loc)
          (move-to forward-loc))
        (move-to forward-loc))))
    (turn-left current-loc))
  (turn-left current-loc))
```

Example Parse

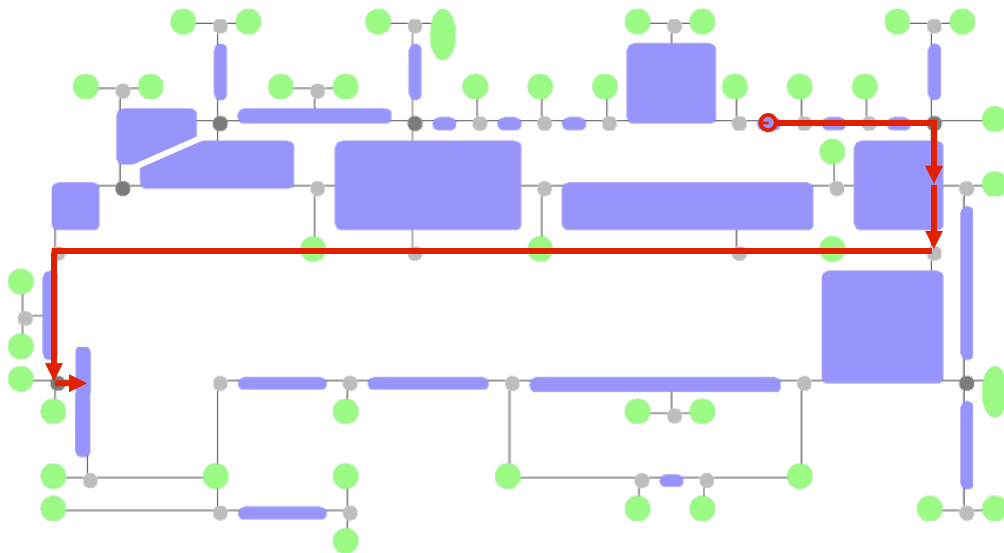
- Go past two junctions and turn right, go forward to the 3-way intersection, take the first right, go straight through the second junction then go left, and turn left again.



```
(do-sequentially
  (do-sequentially
    (do-n-times 2
      (do-sequentially
        (do-until
          (junction current-loc)
          (move-to forward-loc))
        (move-to forward-loc)))
    (turn-right current-loc))
  (do-until
    (junction3 current-loc)
    (move-to forward-loc))
  (turn-right current-loc)
  (do-sequentially
    (do-n-times 2
      (do-sequentially
        (do-until
          (junction current-loc)
          (move-to forward-loc))
        (move-to forward-loc)))
    (turn-left current-loc))
  (turn-left current-loc))
```

Example Parse

- Go past two junctions and turn right, go forward to the 3-way intersection, take the first right, go straight through the second junction then go left, and turn left again.



```
(do-sequentially
  (do-sequentially
    (do-n-times 2
      (do-sequentially
        (do-until
          (junction current-loc)
          (move-to forward-loc))
        (move-to forward-loc))))
    (turn-right current-loc))
  (do-until
    (junction3 current-loc)
    (move-to forward-loc))
  (turn-right current-loc)
  (do-sequentially
    (do-n-times 2
      (do-sequentially
        (do-until
          (junction current-loc)
          (move-to forward-loc))
        (move-to forward-loc))))
    (turn-left current-loc))
  (turn-left current-loc))
```

Outline

- Direction following
- Learning and grounding object attributes
- Discussion

Learning Attributes



- Handle sentences about novel things
 - “These are the limes”
- No longer assuming underlying concepts already exist

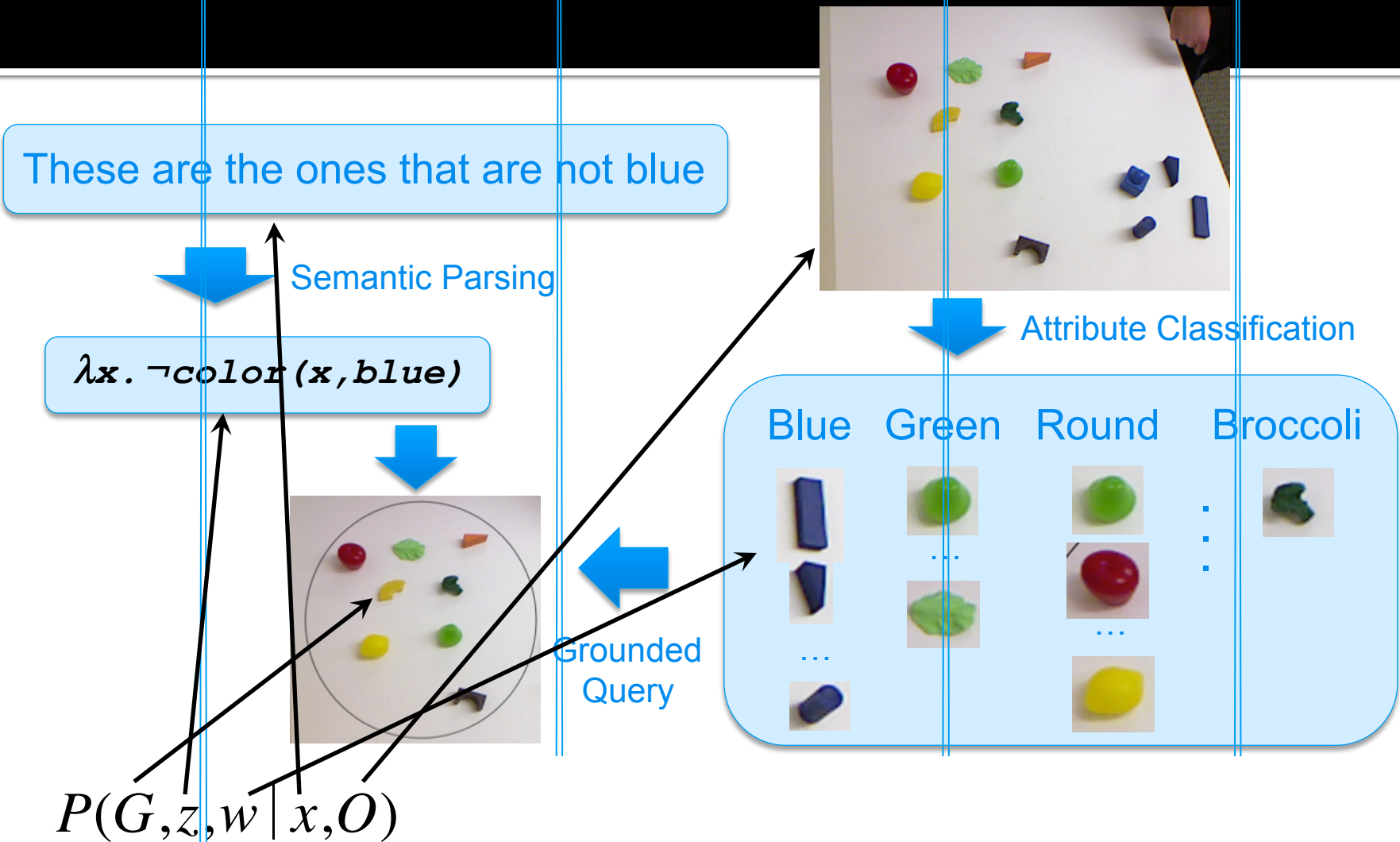


- Requires
 - Perception models
 - Classifiers: green; round
 - Language model
 - How words relate to these detectors

$$\lambda x. \text{green}(x) \wedge \text{round}(x)$$

- Need a joint model for learning these together

Joint Language / Perception Model



Joint Language / Perception Model

These are the ones that are not blue

Semantic Parsing

$\lambda x. \neg \text{color}(x, \text{blue})$



Grounded Query



Attribute Classification



$$P(G, z, w | x, O) = \underbrace{P(z | x)}_{\text{Parsing Model}} \prod_{o \in O} \prod_{c \in C} \underbrace{P(w_{o,c} | o)}_{\text{Vision Model}} \underbrace{P(G | z, w)}_{\text{Grounding Query}}$$

Joint Probability

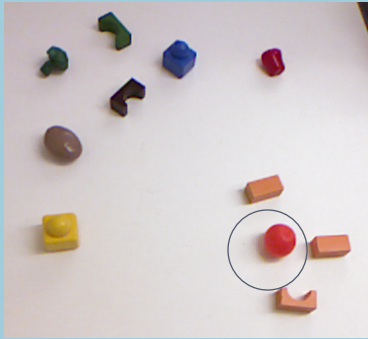
Parsing Model

Vision Model

Grounding Query

Model Learning

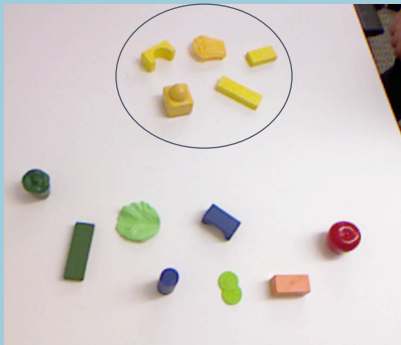
1: Initialization



“This is an orange ball.”

$\text{obj-color}(x, \text{color-orange}) \wedge \text{obj-shape}(x, \text{shape-round})$

2: Training



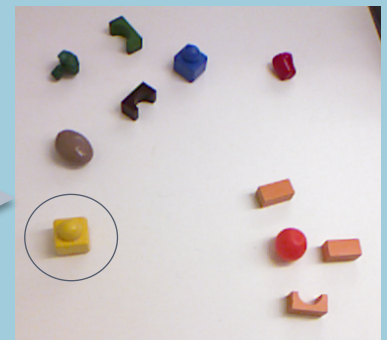
“All of these toys are yellow.”

3: Testing



“It’s the yellow one.”

$\text{obj-color}(x, \text{color-NEW})$



Why Joint Learning?

- Language helps determine attribute relations
- Language is ambiguous: “This is *<new-word>*.”
 - New color attribute?

“This is red.”
 - New shape attribute?

“This is round.”
 - Synonym?

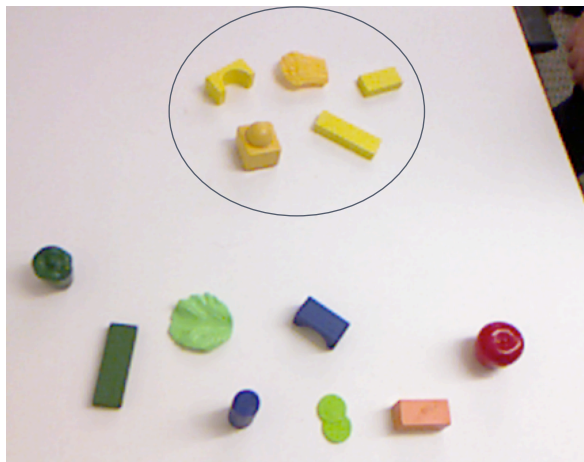
“This is peach.”
 - No attribute at all

“This is toy.”



Experimental Evaluation

- 142 scenes
- 6 colors and 6 shape attributes
- ~1,000 NL sentences from Mechanical Turk
- Ground truth formulas and classifier assignments



What is the Parent Saying?

Watch the video, then **describe what the parent is saying to the child**, in complete sentences.



- Pretend you are a parent teaching a child about something.
- The question is:

How does the parent describe this group of objects?

Your answer should be the sentence(s) the parent said while pointing to these things.

Submit

“All of these are yellow toys.”

Showing HIT 1 of 3

Next HIT

$\lambda x. \text{obj-color}(x, \text{color-yellow})$

What is the Parent Saying?

Watch the video, then **describe what the parent is saying to the child**, in complete sentences.



- Pretend you are a parent teaching a child about something.
- The question is:

How does the parent describe this group of objects?

Your answer should be the sentence(s) the parent said while pointing to these things.

“Here are some blue shapes.”

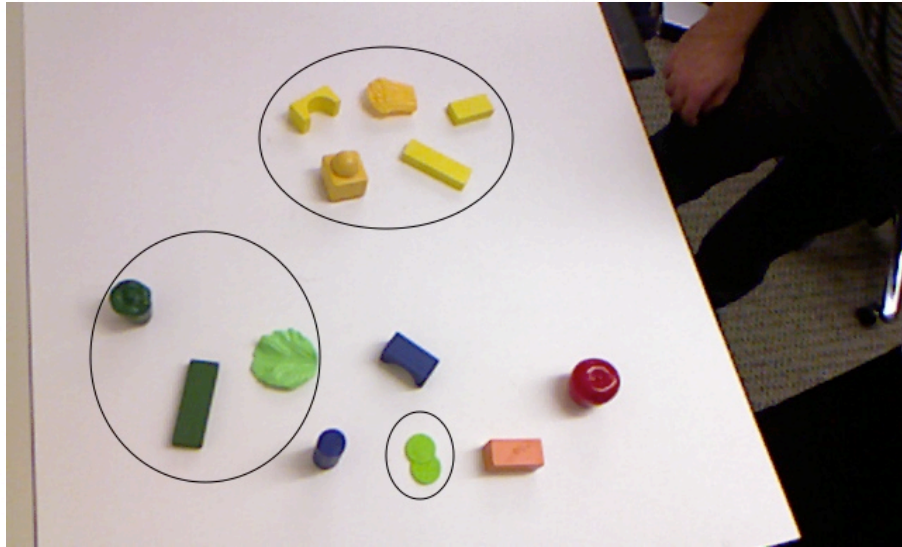
Showing HIT 1 of 3

Next HIT

$$\lambda x. \text{obj-color}(x, \text{color-blue})$$

Describe the Circled Objects

Look at the image, then describe *only* the circled objects.



- Answer this question:

How would you describe the objects that are circled (to distinguish them from the rest)?

- Using complete English sentences
- Describing the *objects themselves* (not their placement)
- [Click here](#) to review instructions

Your answer should be the description of *only* those objects:

“These are all the green and yellow objects.”

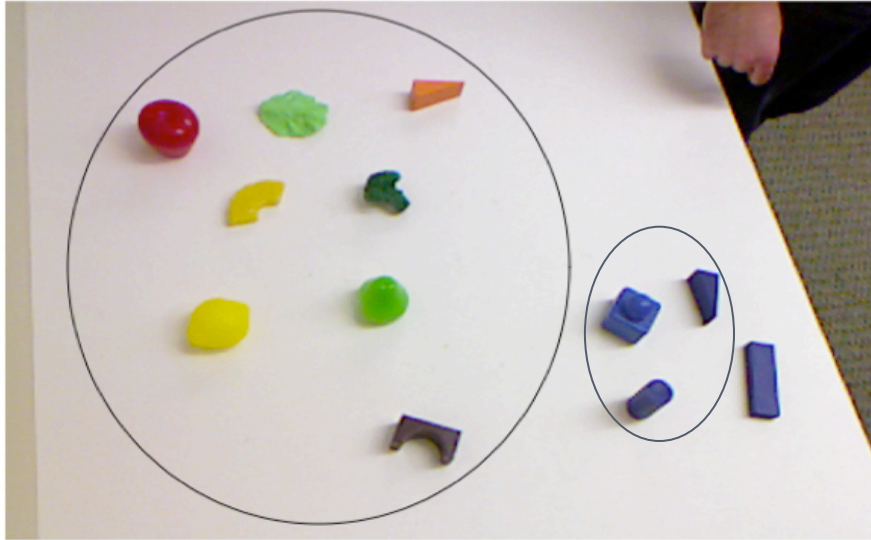
Showing HIT 1 of 3

Next HIT

$\lambda x. \text{obj-color}(x, \text{color-green}) \vee \text{obj-color}(x, \text{color-yellow})$

Describe the Circled Objects

Look at the image, then describe *only* the circled objects.



- Answer this question:

How would you describe the objects that are circled (to distinguish them from the rest)?

- Using complete English sentences
- Describing the *objects themselves* (not their placement)
- [Click here](#) to review instructions

Your answer should be the description of *only* those objects:

Submit

“This is everything but the blue rectangle.”

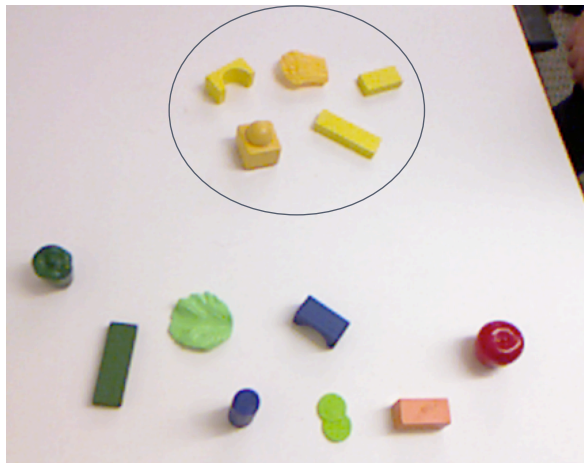
Showing HIT 1 of 3

Next HIT

$\lambda x. \neg(\text{obj-color}(x, \text{color-blue}) \wedge \text{obj-shape}(x, \text{shape-rect}))$

Experimental Evaluation

- 142 scenes
- 6 colors and 6 shape attributes
- ~1,000 NL sentences from Mechanical Turk
- Ground truth formulas and classifier assignments
 - 20 splits into
 - 30% training items for initialization phase (3 colors, 3 shapes)
 - 55% training items for teaching phase (3 new colors, 3 new shapes)
 - 10% test cases with new colors+shapes
 - Precision = 0.85; Recall = 0.8
 - Precision: are identified objects actually described?
 - Recall: how many described objects are identified?



Attribute Grounding

		Lexeme						
		NEW0	NEW1	NEW2	NEW3	NEW4	NEW5	null
NL Token	red	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	green	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	blue	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	thing	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	cube	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	that	0.00	0.00	0.00	0.00	0.00	0.00	0.18
	arch	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	triangle	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	toys	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Examples

Different natural language describes the same object:

“This toy is blue in color.”

$\lambda x. \text{obj-color}(x, \text{color-blue})$

“This is blue color rectangular toy.”

$\lambda x. \text{obj-color}(x, \text{color-blue})$
 $\wedge \text{obj-shape}(x, \text{shape-rect})$



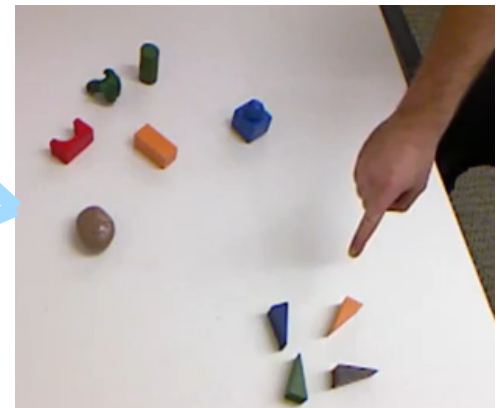
Different natural language has the same meaning:

“This objects are all triangular in shape.”

$\lambda x. \text{obj-shape}(x, \text{shape-triangle})$

“All of these are triangles.”

$\lambda x. \text{obj-shape}(x, \text{shape-triangle})$



Failure cases

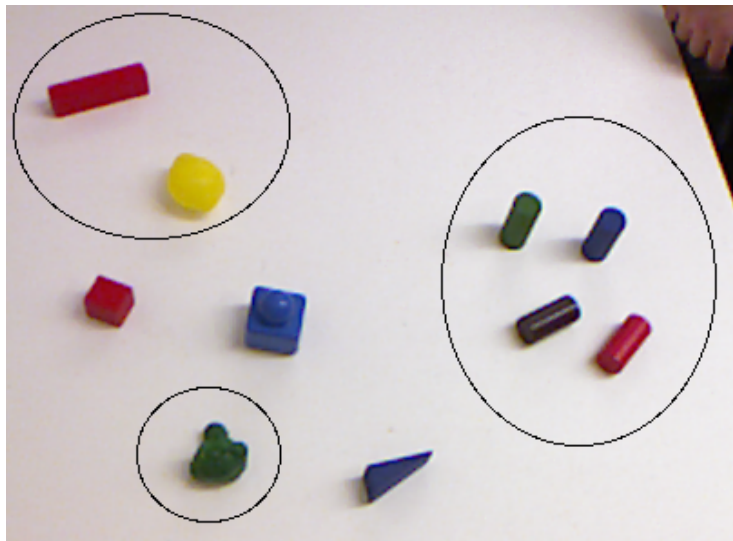
- Bad parses:

“This is a red,
toy triangle.”

$\lambda x. \text{obj-shape}(x, \text{shape-triangle})$
 $\wedge \text{obj-shape}(x, \text{shape-rect})$

Two different
shapes instead of
a shape and a
color

- Bad classification:



Cylinders (lengthwise)
look like rectangles;
cylinders (upright) look
like cubes

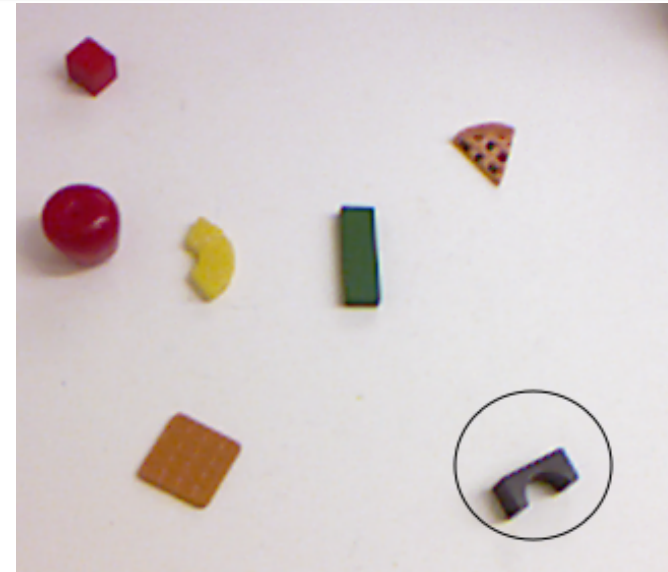
Humans made the same
errors, even with pre-
segmentation view

Failure cases

- Incorrect human input:

- Visual errors
- Typos

“This is a blue toy shaped like a half-pipe.”



It's brown. (This confused the classifiers, too)

- Unexpected human input:



“This object is a fake piece of green lettuce. Do not try to eat!”

Outline

- Direction following
- Learning and grounding object attributes
- Discussion

Limitations and Future Work

- No guarantee that program is valid / executable
Execute distribution over RCL programs?
- Add gesture, gaze, and speech as input / context
- More complex objects, scenes, and attributes
- Teach skills and multi-step tasks via annotated demonstration
 - “This is how to stack an object”
 - Speech helps with segmentation, goal definition, ...
- Compile models into Bayesian estimation

Discussion

- Perception is becoming more and more capable
- Need expressive framework to parse and represent rich human input
- Learning to ground in interactive settings

Advances in semantic NLP, computer vision, machine learning, activity recognition, robotics, control, Bayesian reasoning and estimation