Asking For Help Using Inverse Semantics

Stefanie Tellex and Ross A. Knepper; Adrian Li, Thomas M. Howard, Daniela Rus, Nicholas Roy

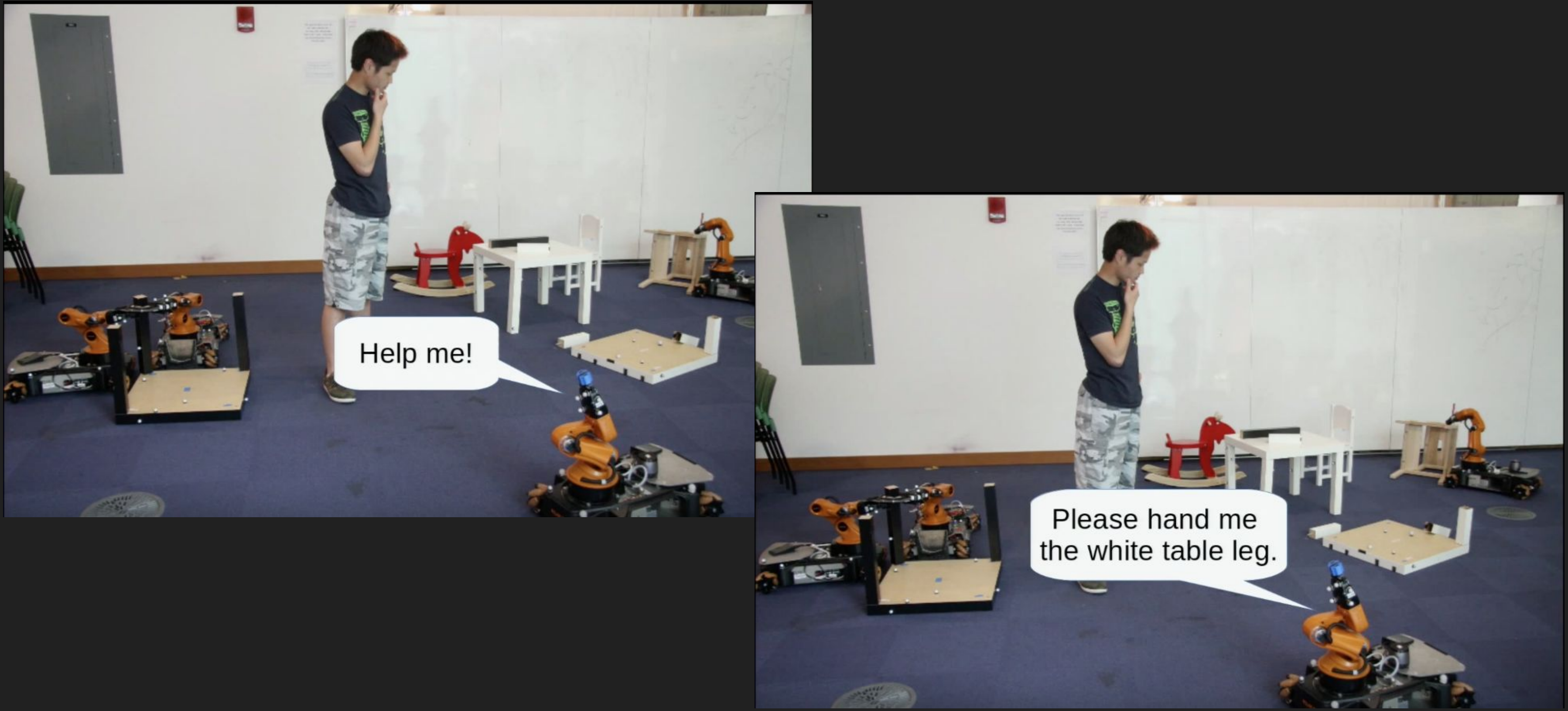
Presented by Lucas Schuermann

Motivation

- Robots fail often -> asking for help could assist in recovery
- Requesting human intervention is hard -> need to ask specific questions
- “I’m stuck. Move the table.” is bad
- “I’m stuck. Move the white table 1 meter to the left.” is good

- How can we generalize detection of failure states and resolution steps?
  - How does the robot determine what it wants the person to do?
- How can we create natural language directions for a human to resolve?
  - How does the robot ask the person to do it?

Motivation



Prior Work

- Template-based approaches for generating requests
  - Failure state = [object] out of reach
  - Request = please hand me [object]
- NLP generating language - environment independent, given a symbolic representation of what you want to say
  - Robot has a situational remark -> generate statement
- NLP understanding language - given environment, input, what does the command mean in context?
  - Human says something -> robot does something

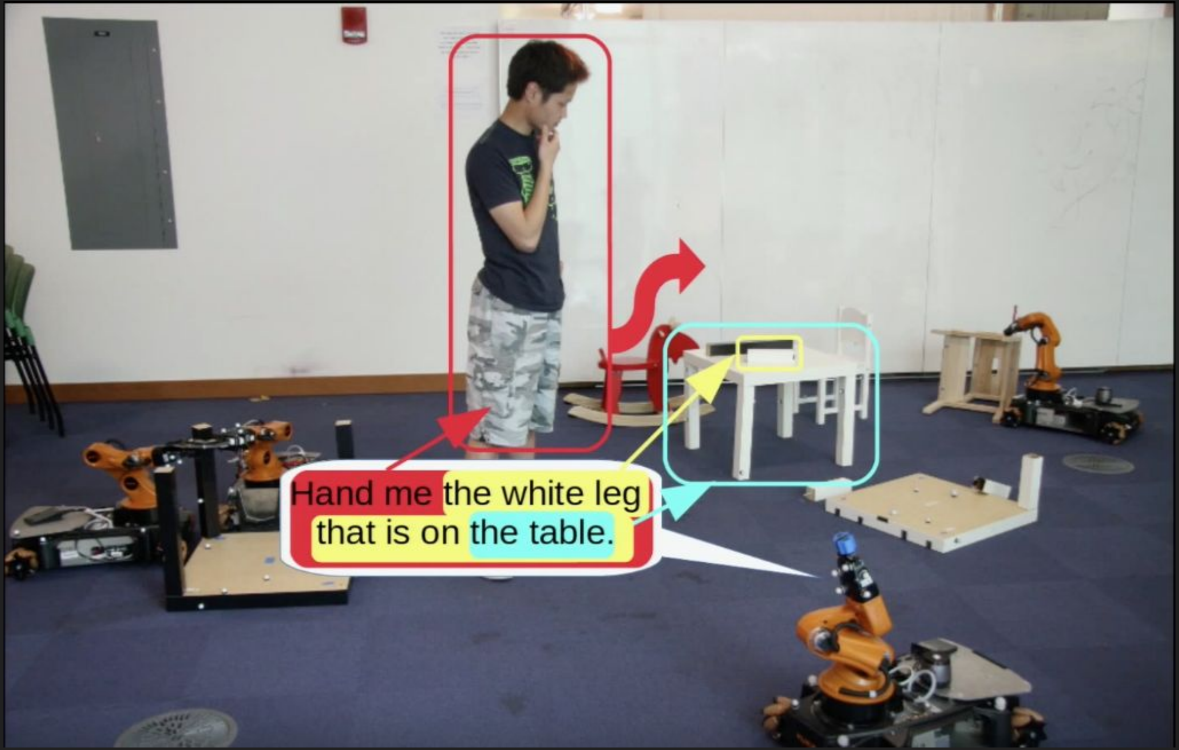
Solution

- Generate *specific* and *targeted* natural language requests that dynamically reflect
- Plan for collaboration to generate symbolic requests when necessary
- Bayesian approach to generate grounded language for robots by inverting previous understanding framework
- Build upon Dragan and Srinivasa (2012) towards a mathematical framework for human-robot interaction, representation of physical motion

Planning for Collaboration

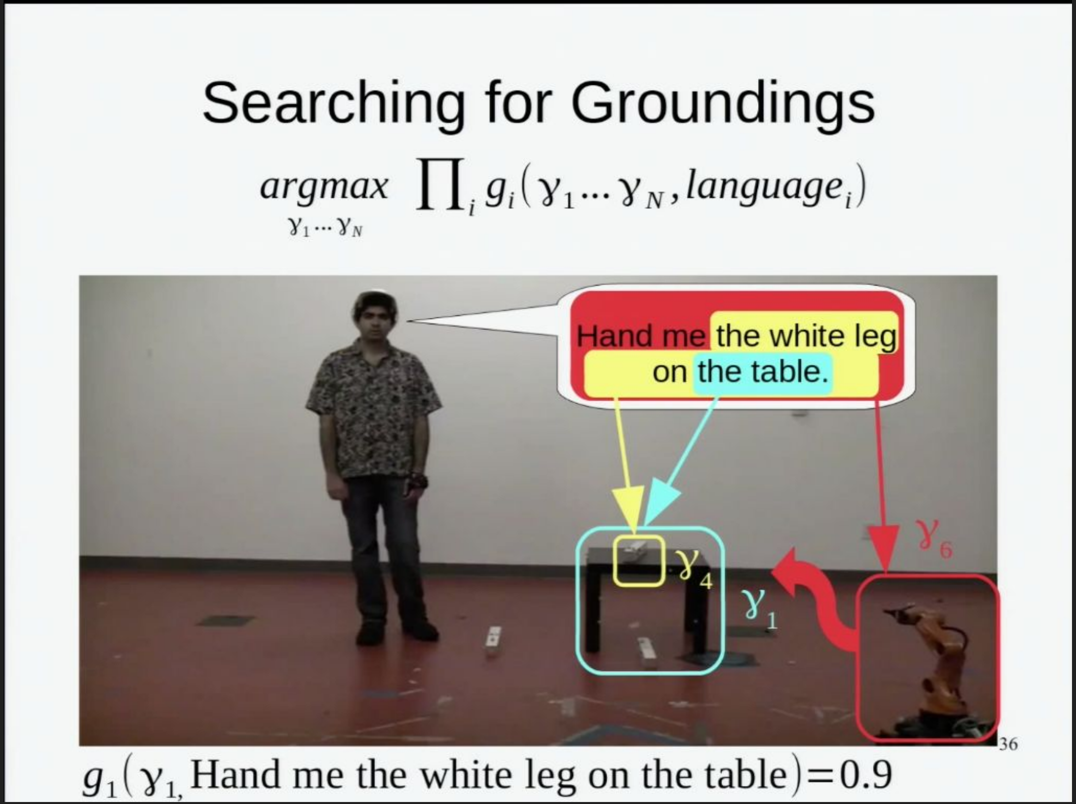
- Strips-style symbolic planner to assemble furniture (or perform any task)
    - Essentially a series of states
- Pre- and post- conditions for each action
    - Satisfied -> transition between states
- Given an action that failed a precondition, generate a symbolic request for help
- Remember: hard-coding a map from failure state to request to help would generate "hand me the white leg" rather than "hand me the white leg that is on the table"

Language Generation with Inverse Semantics



- Need a way to generate language specific to an environment
- Therefore, need to map between language and stuff that is in the environment

Forward Semantics: Model



Searching for Groundings

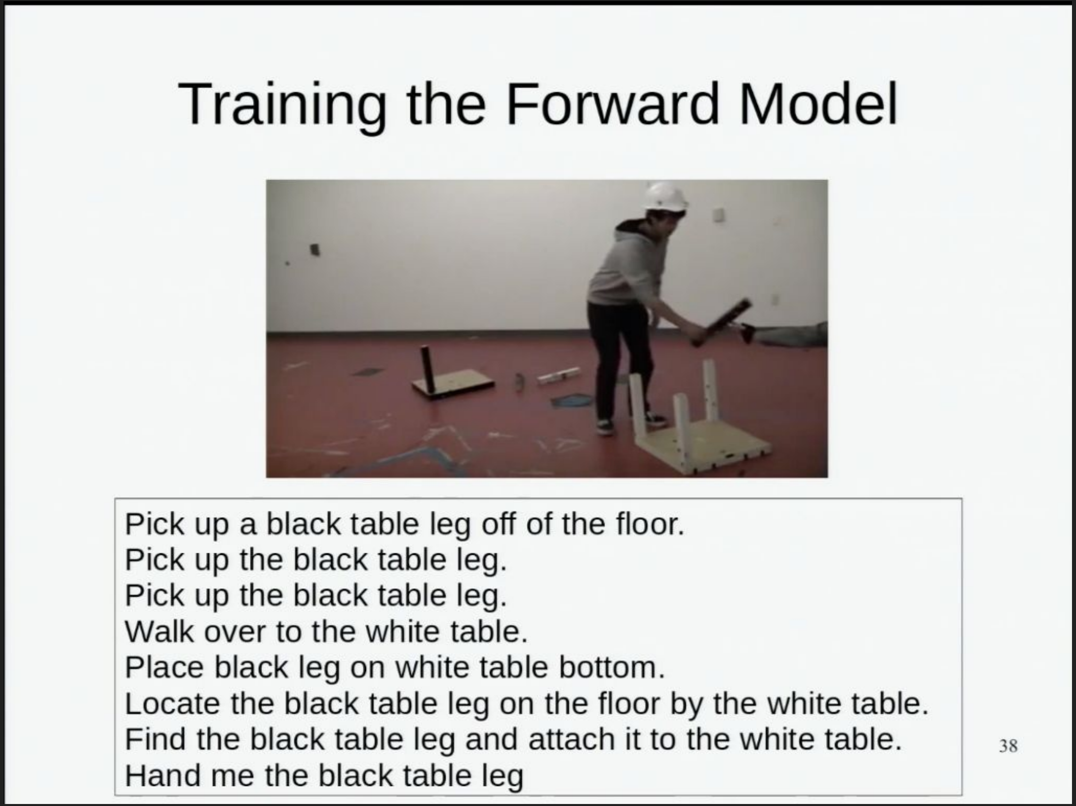
$$\underset{\gamma_1 \dots \gamma_N}{argmax} \prod_i g_i(\gamma_1 \dots \gamma_N, language_i)$$

Hand me the white leg on the table.
γ_4
γ_1
γ_6
36

$g_1(\gamma_{1,}$ Hand me the white leg on the table$)=0.9$

- Tellex et al., 2011
- *Groundings* are objects, actions in the world.
- What *groundings* match the language?
- Parse sentence into tree
- Given semantics model, for each object, infer likelihood that *grounding* matches constraint set -> assign score. Maximize

Forward Semantics: Training

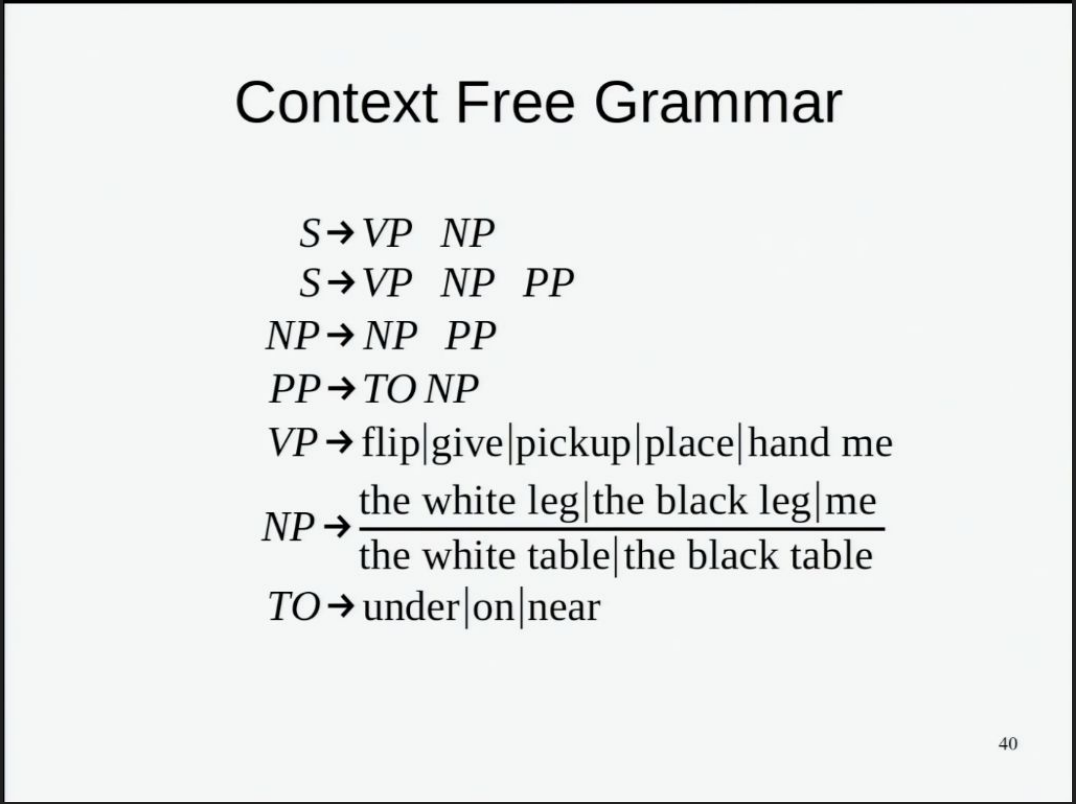


Training the Forward Model

Pick up a black table leg off of the floor.
Pick up the black table leg.
Pick up the black table leg.
Walk over to the white table.
Place black leg on white table bottom.
Locate the black table leg on the floor by the white table.
Find the black table leg and attach it to the white table.
Hand me the black table leg

- Train model with corpus of language -> parts/people etc.
- Mechanical turk watching motion captured videos, asked how to request action be performed
- Scene descriptions form set of associations with demonstrated behavior

Inverse Semantics: Model

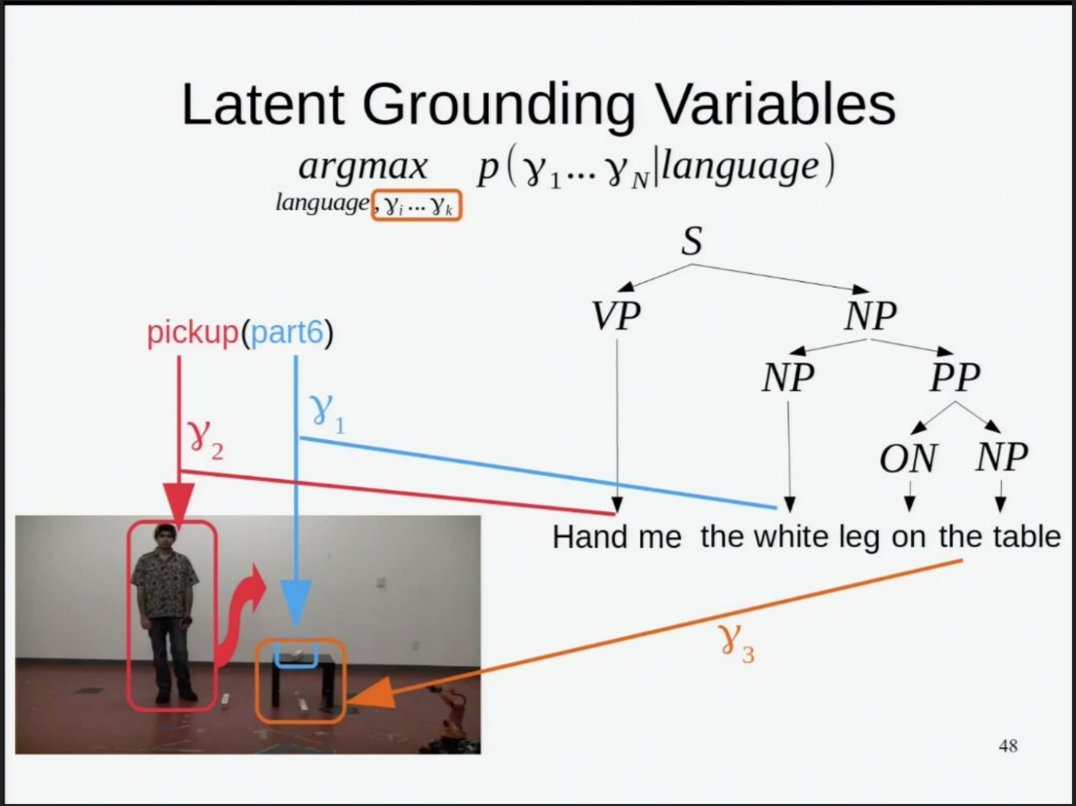


Context Free Grammar

$$
\begin{aligned}
S &\rightarrow VP \quad NP \\
S &\rightarrow VP \quad NP \quad PP \\
NP &\rightarrow NP \quad PP \\
PP &\rightarrow TO\; NP \\
VP &\rightarrow \text{flip}|\text{give}|\text{pickup}|\text{place}|\text{hand me} \\
NP &\rightarrow \frac{\text{the white leg}|\text{the black leg}|\text{me}}{\text{the white table}|\text{the black table}} \\
TO &\rightarrow \text{under}|\text{on}|\text{near}
\end{aligned}
$$

- What language specifies the groundings?
- Search space for optimization is a context free grammar
- Wide variety of phrases that the robot can construct w.r.t. environment

Inverse Semantics: Model



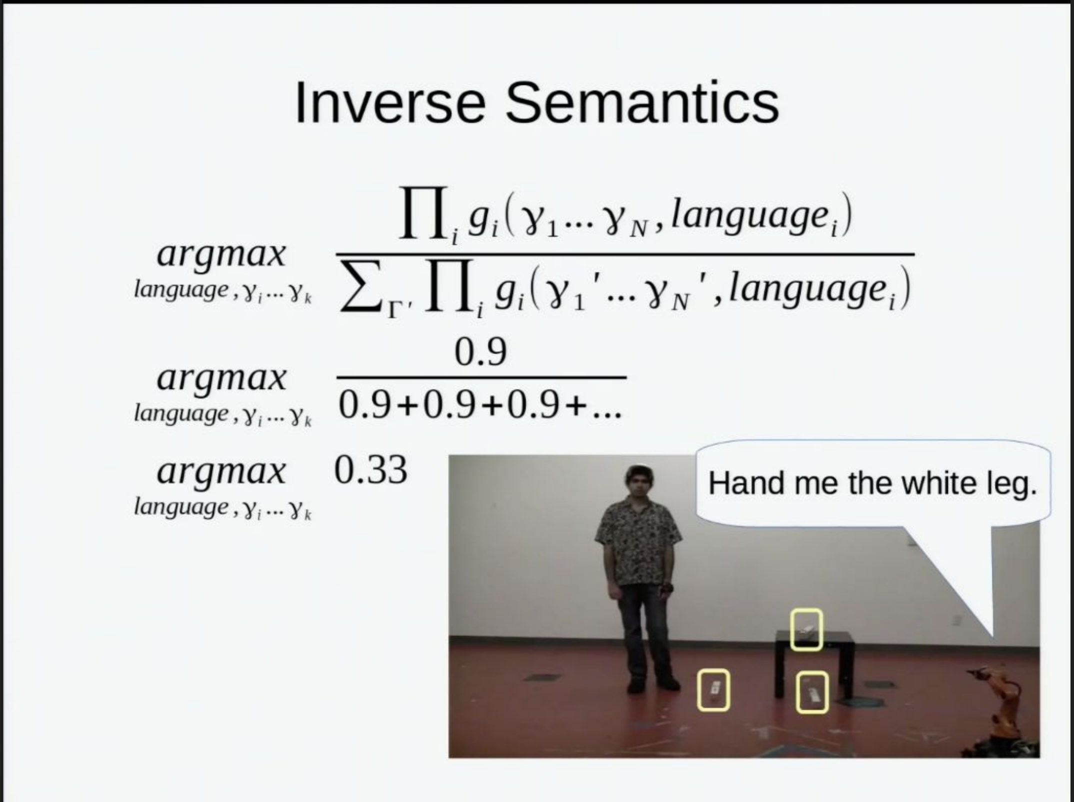
Latent Grounding Variables

$$\underset{language,\ \boxed{\gamma_i \dots \gamma_k}}{argmax} \quad p(\gamma_1 \dots \gamma_N | language)$$

Hand me the white leg on the table

- Given resolution action as symbolic request from planner, map to *groundings*
- Generate candidate phrase from grammar -> assign score depending on match to *groundings*
- To optimize such that disambiguating clauses are generated, include *groundings* not in request

Inverse Semantics: Model



Inverse Semantics

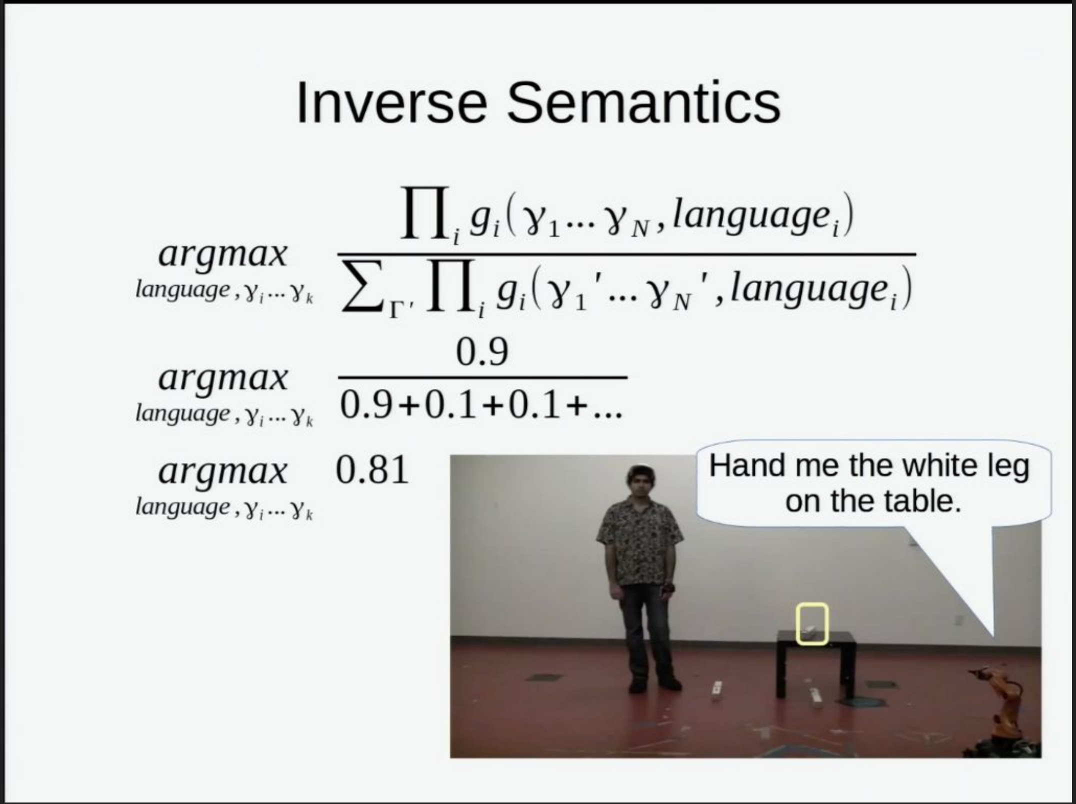
$$\underset{language,\gamma_i\ldots\gamma_k}{\operatorname{argmax}} \frac{\prod_i g_i(\gamma_1\ldots\gamma_N, language_i)}{\sum_{\Gamma'}\prod_i g_i(\gamma_1'\ldots\gamma_N', language_i)}$$

$$\underset{language,\gamma_i\ldots\gamma_k}{\operatorname{argmax}} \frac{0.9}{0.9+0.9+0.9+\ldots}$$

$$\underset{language,\gamma_i\ldots\gamma_k}{\operatorname{argmax}} \; 0.33$$

- Want to penalize a non-specific request for help
- Numerator is high, a *grounding* matches the language
- Denominator is high, many *groundings* in the scene match the language
- Low score

Inverse Semantics: Model



Inverse Semantics

$$\underset{language,\gamma_i\ldots\gamma_k}{\operatorname{argmax}} \frac{\prod_i g_i(\gamma_1\ldots\gamma_N, language_i)}{\sum_{\Gamma'}\prod_i g_i(\gamma_1'\ldots\gamma_N', language_i)}$$

$$\underset{language,\gamma_i\ldots\gamma_k}{\operatorname{argmax}} \frac{0.9}{0.9+0.1+0.1+\ldots}$$

$$\underset{language,\gamma_i\ldots\gamma_k}{\operatorname{argmax}} 0.81$$

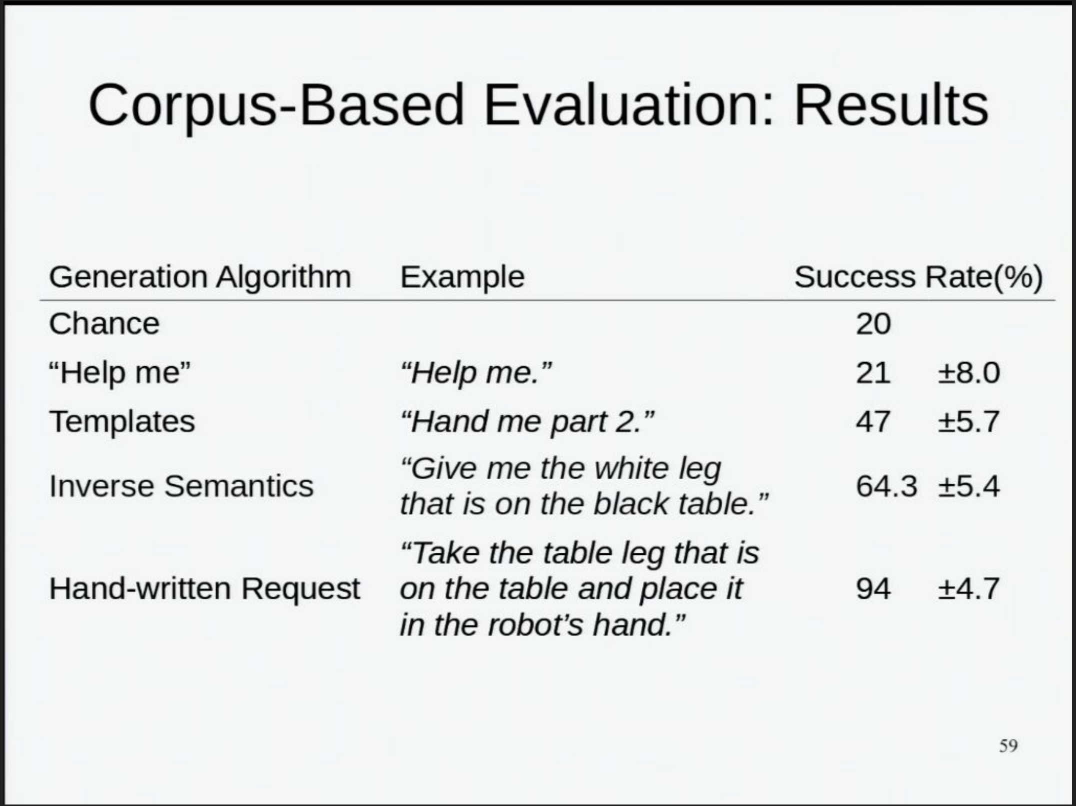
Hand me the white leg on the table.

- Want to reward request with specificity
- Numerator is high, a *grounding* matches the language
- Denominator is low, many *groundings* in the scene no longer match the language
- High score!
- This is one of the core paper contributions

Evaluation

- Looking to demonstrate that *specific requests* generated with this method help humans understand what task to perform better than baseline
- Base: “help me”, human left to look at situation
- New: “give me the white leg on the black table”
- Looking for a statistically-significant improvement
- Test (1): mechanical turk. Generate descriptions of action. Give user a number of videos of different actions. Check to see if user can identify which video performs the requested action.
- Test (2): keep humans behind screen. Have robot ask for help. Identify ability of human to resolve issue. Measure qualitative ease of solving issue.

Evaluation

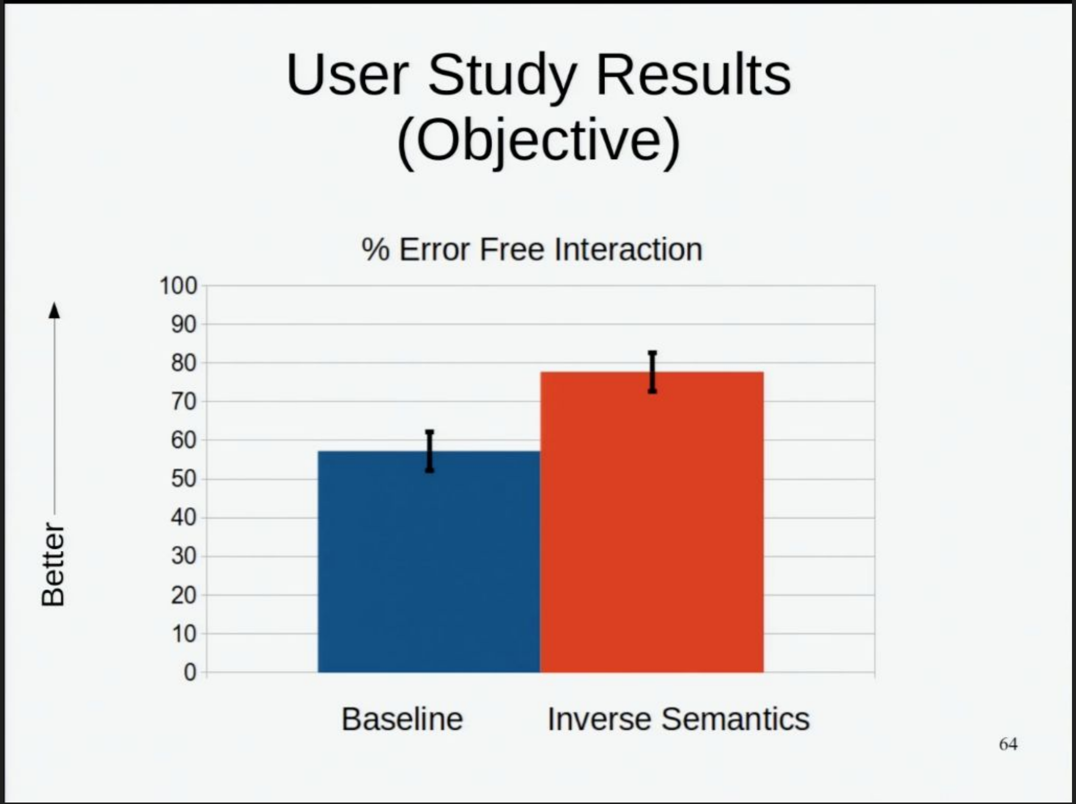


Corpus-Based Evaluation: Results

Generation Algorithm	Example	Success Rate(%)	
Chance		20	
“Help me”	*“Help me.”*	21	±8.0
Templates	*“Hand me part 2.”*	47	±5.7
Inverse Semantics	*“Give me the white leg that is on the black table.”*	64.3	±5.4
Hand-written Request	*“Take the table leg that is on the table and place it in the robot’s hand.”*	94	±4.7

- Hand-crafted is upper bound
- Error ephemerally attributed to “model of human understanding”
- Results are significant but there is still substantial room for improvement
- Templates approach (old, easy) works surprisingly well

Evaluation



User Study Results (Objective)

- Statistically quite significant result!
- When using specific requests, humans generally could perform the correct intervention on the first try
- People could eventually figure out what to do, result more ambiguous if not looking at first try only

Future Work

- Robot can still be nonspecific due to language limitations
  - “Near the table” should be “left” or “right” “of the table”.
- -> Improve model of semantics with new and larger data sets, more modern techniques
- Users can still not particularly well understand requests
  - Robots should have another chance to formulate a question
- -> Improve flow with multi-turn human interactions. Person can ask questions to clarify.
- Words sometimes can’t fully articulate a problem
- -> Apply same approach to both language and gesture for requests

Video Demo

https://www.youtube.com/watch?v=2Ts0W4SiOfs