Learning to Interpret Natural Language Commands through Human-Robot Dialog

Jesse Thomason
Shiqi Zhang, Raymond Mooney, Peter Stone

The University of Texas at Austin
Commanding Robots

• Autonomous robots in human environments

• Simplest to interact with via natural language
Our Task

• Command a robot operating in an office environment
• Robot autonomously wanders by default
• Robot can navigate to rooms and deliver items
System Goals

• Require little initial data
  – More domain independent

• Reason using composition
  – “Alice’s office”

• Robust to lexical variation
  – “bring”, “deliver”, “take”

• Execute the right action
  – Perform clarifications with user
Closest Previous Work

• Service robot that accepts commands (Kollar, 2013)
• Semantics match spans of words to known actions/people/locations
• Can learn new referring expressions through dialog
  
<table>
<thead>
<tr>
<th>Human</th>
<th>Go to Alice’s office</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot</td>
<td>Where is “Alice’s office”?</td>
</tr>
<tr>
<td>Human</td>
<td>Room 3</td>
</tr>
</tbody>
</table>

• This system would explicitly match “Alice’s office” to room 3
Closest Previous Work

• When system sees “Bob’s office”, will have to ask where that is
• Want to take advantage of compositionality instead
  – Reason about possessive marker “’s” and what entities “office” picks out
• Need a more powerful formalism for representing sentence semantics
  – Want to keep initial training data light
Helpful Previous Work

• Augment a semantic parser through conversation logs (Artzi, 2011)

<table>
<thead>
<tr>
<th>Human</th>
<th>I would like to fly out of boston arriving to new york and back from new york to boston</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Leaving boston (CONFIRM: from(fl1,BOS)) on what date? (ASK:λx.departdate(fl1,x))</td>
</tr>
</tbody>
</table>

• Key idea for us: use known system semantic meanings to guess human utterance word meanings
### Tag Token Sequence

<table>
<thead>
<tr>
<th>token(s)</th>
<th>syntax : semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bring</strong></td>
<td>((S/PP)/NP : \lambda x.\lambda P.(\text{action(bring)} \land \text{patient(bring},x) \land P(\text{bring})))</td>
</tr>
<tr>
<td><strong>coffee</strong></td>
<td>NP : coffee</td>
</tr>
<tr>
<td><strong>to</strong></td>
<td>PP/NP : \lambda x.\lambda y.(\text{recipient(y,}x))</td>
</tr>
<tr>
<td><strong>Bob</strong></td>
<td>NP : bob</td>
</tr>
</tbody>
</table>

University of Washington Semantic Parsing Framework (SPF); (Artzi, 2011)

Known possibilities for each token stored in a lexicon

Use Combinatory Categorial Grammar (CCG)-driven parsing
Construct Meaning Hierarchically

\[ S : action(\text{bring}) \land patient(\text{bring,coffee}) \land recipient(\text{bring,bob}) \]

\[ S/PP : \lambda P. (action(\text{bring}) \land patient(\text{bring,coffee}) \land P(\text{bring})) \]

\[ PP : \lambda y. (recipient(y,bob)) \]

\[ (S/PP)/NP : \lambda x. \lambda P. (action(\text{bring}) \land patient(\text{bring,x}) \land P(\text{bring})) \]

\[ NP : \text{coffee} \]

\[ PP/NP : \lambda x. \lambda y. (recipient(y,x)) \]

\[ NP : \text{bob} \]

\[ \text{bring} \]

\[ \text{coffee} \]

\[ \text{to} \]

\[ \text{Bob} \]
Tag Token Sequence – Missing Entry

bring  (S/PP)/NP : \( \lambda x.\lambda P.\) \((\text{action}(\text{bring}) \land \text{patient}(\text{bring}, x) \land P(\text{bring}))\)

java  ?

to  PP/NP : \( \lambda x.\lambda y.\) \((\text{recipient}(y, x))\)

Bob  NP : \emph{bob}

Given semantic form, can guess about missing token syntax/semantics

<table>
<thead>
<tr>
<th>Human</th>
<th>bring java to bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot</td>
<td>what should I bring to bob?</td>
</tr>
<tr>
<td>Human</td>
<td>coffee</td>
</tr>
</tbody>
</table>

\( S : \text{action}(\text{bring}) \land \text{patient}(\text{bring, coffee}) \land \text{recipient}(\text{bring, bob}) \)
Tag Token Sequence – Missing Entry

bring (S/PP)/NP : \lambda x.\lambda P.(\text{action}(\text{bring}) \land \text{patient}(\text{bring},x) \land P(\text{bring}))
java ?
to PP/NP : \lambda x.\lambda y.(\text{recipient}(y,x))
Bob NP : bob

Given form:
\text{action}(\text{bring}) \land \text{patient}(\text{bring},\text{coffee}) \land \text{recipient}(\text{bring},\text{bob})
Lexicon entries that produce parts of this form:

bring :- (S/PP)/NP : \lambda x.\lambda P.(\text{action}(\text{bring}) \land \text{patient}(\text{bring},x) \land P(\text{bring}))
bring :- (S/NP)/NP : \lambda x.\lambda y.(\text{action}(\text{bring}) \land \text{recipient}(\text{bring},x) \land \text{patient}(\text{bring},y))
coffee :- NP : coffee
Bob :- NP : bob

Candidates for ‘java’ lexical entry:

:- (S/PP)/NP : \lambda x.\lambda P.(\text{action}(\text{bring}) \land \text{patient}(\text{bring},x) \land P(\text{bring}))
:- (S/NP)/NP : \lambda x.\lambda y.(\text{action}(\text{bring}) \land \text{recipient}(\text{bring},x) \land \text{patient}(\text{bring},y))
:- NP : coffee
:- NP : bob
Tag Token Sequence – Missing Entry

bring \quad (S/PP)/NP : \lambda x. \lambda P.(\text{action(\text{bring})} \land \text{patient(\text{bring}, x)} \land P(\text{bring}))

java \quad NP : \text{coffee}

to \quad PP/NP : \lambda x. \lambda y. (\text{recipient}(y, x))

Bob \quad NP : \text{bob}

With new lexicon entry, we can construct the correct semantic form

\[ S : \text{action(\text{bring})} \land \text{patient(\text{bring, coffee})} \land \text{recipient(\text{bring, bob})} \]
Meeting System Goals

• Require little initial data
  – Bootstrap parser with 5 expressions, 105 words

• Handle composition used by speakers
  – Use CCG-driven semantic parsing (Artzi, 2011)

• Robust to lexical variation
  – Incrementally train parser to obtain new words

• Execute the **right** action
  – Use dialog to clarify meanings with user (Kollar, 2013)
Mechanical Turk Experiment

• Users given one navigation and one delivery goal
  – Train/test goals chosen at random from possibilities
• Chat with robot’s dialog agent until goal is understood
Mechanical Turk Interface

**TASK TO COMPLETE**

Dave Daniel wants the item in slot 5.

<table>
<thead>
<tr>
<th>ROBOT</th>
<th>How can I help?</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOU</td>
<td>please bring the item in slot 5 to dave daniel</td>
</tr>
<tr>
<td>ROBOT</td>
<td>What should I bring to dave daniel?</td>
</tr>
<tr>
<td>YOU</td>
<td>a calendar</td>
</tr>
<tr>
<td>ROBOT</td>
<td>I'm sorry, but I couldn't pinpoint what you meant by that.</td>
</tr>
<tr>
<td>ROBOT</td>
<td>What should I bring to dave daniel?</td>
</tr>
<tr>
<td>YOU</td>
<td>a day planner</td>
</tr>
<tr>
<td>ROBOT</td>
<td>I'm sorry, but I couldn't pinpoint what you meant by that.</td>
</tr>
<tr>
<td>ROBOT</td>
<td>What should I bring to dave daniel?</td>
</tr>
<tr>
<td>YOU</td>
<td>calendar</td>
</tr>
<tr>
<td>ROBOT</td>
<td>You want me to bring calendar to dave daniel?</td>
</tr>
<tr>
<td>YOU</td>
<td>yes</td>
</tr>
<tr>
<td>ROBOT</td>
<td>I thought so</td>
</tr>
<tr>
<td>ROBOT</td>
<td>Happy to help</td>
</tr>
</tbody>
</table>

**Items available to robot:**

1. Coffee cup
2. Burger
3. Cell phone
4. Trash can
5. Calendar
Large-Scale Experiment

• Tested in 4 phases
• ~50 users received test goals, ~50 train goals
  – Unique users in each phase
• System incrementally trained via train goal conversations only
Mechanical Turk Dialog Turns

Average Turker Turns for Success

- **Navigation task turns**
- **Delivery task turns**

<table>
<thead>
<tr>
<th>Batch</th>
<th>Navigation Task</th>
<th>Delivery Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch 0</td>
<td>2.7</td>
<td>18</td>
</tr>
<tr>
<td>Batch 1</td>
<td>2.5</td>
<td>5.6</td>
</tr>
<tr>
<td>Batch 2</td>
<td>3.4</td>
<td>5.2</td>
</tr>
<tr>
<td>Batch 3</td>
<td>3.5</td>
<td>3.8</td>
</tr>
</tbody>
</table>

* denotes a significant difference.
Mechanical Turk Survey Responses

The robot understood me

The robot frustrated me

Graph showing the average responses across batches:
- Batch 0: Strongly Agree 2.2, Somewhat Agree 2.2
- Batch 1: Strongly Agree 2.6, Somewhat Agree 2.0
- Batch 2: Strongly Agree 2.7, Somewhat Agree 1.9
- Batch 3: Strongly Agree 2.8, Somewhat Agree 1.5

* Indicates a significant difference in responses.
Robot Experiment

• Same setup, but real robot and fewer users
  – Users type to robot to mimic Mechanical Turk setup
• 10 users in initial test batch
• System interacted freely with people on the floor for four days as training (34 conversations in total)
• 10 users in the second test batch, after retraining
Office Robot Dialog Completion

- **Navigation task completion**
- **Delivery task completion**

<table>
<thead>
<tr>
<th>Batch</th>
<th>Navigation Completion</th>
<th>Delivery Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch 0</td>
<td>90%</td>
<td>20%</td>
</tr>
<tr>
<td>Batch 1</td>
<td>90%</td>
<td>60% +</td>
</tr>
</tbody>
</table>
Office Robot Survey Responses

The robot understood me

The robot frustrated me

<table>
<thead>
<tr>
<th>Strongly Agree</th>
<th>Somewhat Agree</th>
<th>Somewhat Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>3</td>
<td>2.5</td>
<td>1.6</td>
</tr>
<tr>
<td>Batch 0</td>
<td>Batch 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average:

Batch 0: 2.9 *
Batch 1: 1.5 +
Conclusions

• Lexical acquisition reduces dialog lengths for multi-argument predicates like delivery
• Causes users to perceive the system as more understanding
• Leads to less user frustration
• Allows improving language understanding without large, annotated corpora
Learning to Interpret Natural Language Commands through Human-Robot Dialog

Jesse Thomason
Shiqi Zhang, Raymond Mooney, Peter Stone

The University of Texas at Austin
Related Work

• Command processing has taken many forms
• Specify tasks step-by-step (Meriçli, 2014)
  – Assumes particular words in particular order
• Specify low-level action sequences (Misra, 2014; Tellex, 2011)
  – Uses a parser trained on a huge corpus
• Map language to action specifications (Matuszek, 2013)
  – Cannot learn new words/expressions
Future Work

• Perceptual grounding (‘blue’, ‘left of’)
• Predicate invention (‘ruddy’)
• Learning a multi-objective dialog policy that trades off learning and user satisfaction