Learning to Interpret Natural Language Commands through Human-Robot Dialog

Jesse Thomason, Shiqi Zhang, Raymond Mooney and Peter Stone

Presented by Siliang Lu
Task

Develop a dialog agent for mobile robots understanding human instructions through semantic parsing
Semantic parser

- $\lambda$ - calculus
- Combinatory categorical grammar (CCG)

NP: Noun phrase
N: Noun

Mallory Morgan’s office

NP: the($\lambda y$.office($y$) $\wedge$ possesses(mallory, $y$) $\wedge$ person(mallory)))

NP/N: $\lambda P.(the(\lambda y.(P(y) $\wedge$ possesses(mallory, $y$) $\wedge$ person(mallory))))$  N: office

NP: mallory  (NP/N)|\NP: $\lambda x.\lambda P.(the(\lambda y.(P(y) $\wedge$ possesses($x$, $y$) $\wedge$ person($x$))))$  office

Mallory Morgan

NP: Noun phrase
N: Noun
Belief state

• Three components:
  • Each component is a histogram of confidences over possible assignments
  • Action: walking and bringing items -> [0,1]
  • Recipient -> (people, room, items) U null
  • Patient -> (people, room, items) U null

Multiple meaning hypotheses:

Expression: go to the office
Logical form: action(walk) ^ recipient(walk, the (\lambda y. office(y))))
Updating the Belief state

• For open-ended statement (update all hypotheses):

\[ \text{conf}(c = H_{i,c}) \leftarrow \text{conf}(c = H_{i,c}) \left( 1 - \frac{\alpha}{k} \right) + \frac{\alpha}{k} \]

\( \alpha \) – threshold of confidence

• For unmentioned arguments:

\[ \text{conf}(c = \overline{H}_{j,c}) \leftarrow \gamma \text{conf}(c = \overline{H}_{j,c}) \]

\( \gamma \) – decay parameter
Responding

Reduce to discrete state:

\[ T_c = \arg\max_{t \in A_c} (\text{conf}(c = t)) \]

\( T_c \): the top candidate arguments

<table>
<thead>
<tr>
<th>( S' )</th>
<th>Role Request</th>
<th>Text</th>
<th>( \pi(S') )</th>
<th>Initiative</th>
</tr>
</thead>
<tbody>
<tr>
<td>(action, patient, recipient)</td>
<td>all</td>
<td>Sorry I couldn’t understand that. Could you reword your original request?</td>
<td>user</td>
<td></td>
</tr>
<tr>
<td>(unknown, unknown, unknown)</td>
<td>action</td>
<td>What action did you want me to take involving ( T_{\text{patient}} ) and ( T_{\text{recipient}} )?</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>(walk, ( \emptyset ), unknown)</td>
<td>recipient</td>
<td>Where should I walk?</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>(bring, unknown, ( T_{\text{recipient}} ))</td>
<td>patient</td>
<td>What should I bring to ( T_{\text{recipient}} )?</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>(walk, ( \emptyset ), ( T_{\text{recipient}} ))</td>
<td>confirmation</td>
<td>You want me to walk to ( T_{\text{recipient}} )?</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>(bring, ( T_{\text{patient}} ), ( T_{\text{recipient}} ))</td>
<td>confirmation</td>
<td>You want me to bring ( T_{\text{patient}} ) to ( T_{\text{recipient}} )?</td>
<td>system</td>
<td></td>
</tr>
</tbody>
</table>
Learning from conversations

Template-based lexical generation procedure (GENLEX)

• For each utterance paired with a logical form seen during training:

\[ \text{GENLEX}(x, z; \Lambda, \theta) \]
Learning from conversations

Template-based lexical generation procedure (GENLEX)

• For each utterance paired with a logical form seen during training:

I want a flight to New York.
\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{NYC}) \]

Create lexemes

I want a flight
flight
flight to new

flight

(to, NYC)

(I want, {})

(flight to new, {to, NYC})

...
Experimental setup

• Mechanical Turk
• Segbot Experiment
• Task:
  • Navigation
  • Delivery
• Survey
Figure 4: **Left**: Average Mechanical Turk survey responses across the four test batches. **Right**: Mean user turns in Mechanical Turk dialogs where the correct goal was reached. Means in underlined bold differ significantly ($p < 0.05$) from the batch 0 mean.
**Result (Segbot)**

Table 2: Average Segbot survey responses from the two test groups and the proportion of task goals completed. Means in bold differ significantly ($p < 0.05$). Means in italics trend different ($p < 0.1$).

<table>
<thead>
<tr>
<th>Survey Question</th>
<th>Init Test</th>
<th>Test</th>
<th>Trained Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Likert [0-4]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tasks Easy</td>
<td>3.8</td>
<td>3.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robot Understood</td>
<td>1.6</td>
<td></td>
<td>2.9</td>
<td></td>
</tr>
<tr>
<td>Robot Frustrated</td>
<td>2.5</td>
<td></td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>Use Navigation</td>
<td>2.8</td>
<td></td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>Use Delivery</td>
<td>1.6</td>
<td></td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>Goals Completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigation</td>
<td>90</td>
<td></td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Delivery</td>
<td>20</td>
<td></td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>