Improving Grounded Natural Language Understanding through Human-Robot Dialog

Jesse Thomason∗, Aishwarya Padmakumar †, Jivko Sinapov‡, Nick Walker∗, Yuqian Jiang‡, Harel Yedidsion‡, Justin Hart‡, Peter Stone‡, and Raymond J. Mooney‡

Abstract—Natural language understanding for robotics can require substantial domain- and platform-specific engineering. For example, for mobile robots to pick-and-place objects in an environment to satisfy human commands, we can specify the language humans use to issue such commands, and connect concept words like red can to physical object properties. One way to alleviate this engineering for a new domain is to enable robots in human environments to adapt dynamically—continually learning new language constructions and perceptual concepts. In this work, we present an end-to-end pipeline for translating natural language commands to discrete robot actions, and use clarification dialogs to jointly improve language parsing and concept grounding. We train and evaluate this agent in a virtual setting on Amazon Mechanical Turk, and we transfer the learned agent to a physical robot platform to demonstrate it in the real world.

I. INTRODUCTION

As robots become ubiquitous across diverse human environments such as homes, factory floors, and hospitals, the need for effective human-robot communication grows. These spaces involve domain-specific words and affordances, e.g., turn on the kitchen lights, move the pallet six feet to the north, and notify me if the patient’s condition changes. Thus, pre-programming robots’ language understanding can require costly domain- and platform-specific engineering. In this paper, we propose and evaluate a robot agent that leverages conversations with humans to expand an initially low-resource, hand-crafted language understanding pipeline to reach better common ground with its human partners.

We combine bootstrapping better semantic parsing through signal from clarification dialogs [1], previously using no sensory representation of objects, with an active learning approach for acquiring such concepts [2], previously restricted to object identification tasks. Thus, our system is able to execute natural language commands like Move a rattling container from the lounge by the conference room to Bob’s office (Figure 5) that contain compositional language (e.g., lounge by the conference room) understood by the semantic parser and objects to be identified by their physical properties (e.g., rattling container). The system is initialized with a small seed of natural language data for semantic parsing, and no initial labels tying concept words to physical objects, instead learning parsing and grounding as needed through human-robot dialog.

Our contributions are: 1) a dialog strategy to improve language understanding given only a small amount of initial in-domain training data; 2) dialog questions to acquire perceptual concepts in situ rather than from pre-labeled data or past interactions alone (Figure 1); and 3) a deployment of our dialog agent on a full stack, physical robot platform.

We evaluate this agent’s learning capabilities and usability on Mechanical Turk, asking human users to instruct the agent through dialog to perform three tasks: navigation (Go to the lounge by the kitchen), delivery (Bring a red can to Bob), and relocation (Move an empty jar from the lounge by the kitchen to Alice’s office). We find that the agent receives higher qualitative ratings after training on information extracted from previous conversations. We then transfer the trained agent to a physical robot to demonstrate its continual learning process in a live human-robot dialog.

II. RELATED WORK

Research on the topic of humans instructing robots spans natural language understanding, vision, and robotics. Recent methods perform semantic parsing using sequence-to-sequence [3], [4], [5] or sequence-to-tree [6] neural networks, but these require hundreds to thousands of examples. In human-robot dialog, gathering information at scale for a given environment and platform is unrealistic, since each data point comes from a human user having a dialog interaction in the same space as a robot. Thus, our methods assume only a small amount of seed data.

Semantic parsing has been used as a language understanding step in tasks involving unconstrained natural language instruction, where a robot must navigate an unseen environment [7], [8], [9], [10], [11], to generate language requests regarding a shared environment [12], and to tie language to

A demonstration video can be viewed at https://youtu.be/PbOfteZ_CJc?t=5.
Fig. 2: User commands are parsed into semantic slots (left), which are grounded (center) using either a known map (for rooms and people) or learned concept models (for objects) to a distribution over possible satisfying constants (e.g., all rooms that can be described as an “office”). A clarification dialog (right) is used to recover from ambiguous or misunderstood slots (e.g., Q1), and to improve concept models on the fly (e.g., Q2).

A. Semantic Parser

The semantic parsing component takes in a sequence of words and infers a semantic meaning representation of the task. For example, a relocate task moves an item (patient) from one place (source) to another (goal) (Figure 2). The agent uses the Combinatory Categorial Grammar (CCG) formalism [34] to facilitate parsing.

Word embeddings [35] augment the lexicon at test time to recover from out-of-vocabulary words, an idea similar in spirit to previous work [36], but taken a step further via formal integration into the agent’s parsing pipeline. This allows, for example, the agent to use the meaning of known word take for unseen word grab at inference time.

B. Language Grounding

The grounding component takes in a semantic meaning representation and infers denotations and associated confidence values (Figure 2). The same semantic meaning can ground differently depending on the environment. For example, the office by the kitchen refers to a physical location, but that location depends on the building.

Perceptual concepts like red and heavy require considering sensory perception of physical objects. The agent builds multi-modal feature representations of objects by exploring them with a fixed set of behaviors. In particular, before our experiments, a robot performed a grasp, lift, lower, drop, press, and push behavior on every object, recording audio information from an onboard microphone and haptic information from force sensors in its arm. That robot also looked at each object with an RGB camera to get a visual representation. Summary audio, haptic, and visual features are created for each applicable behavior (e.g., drop-audio, look-vision), and these features represent objects at training
Feature representations of objects are connected to language labels by learning discriminative classifiers for each concept using the methods described in previous work [37], [31]. In short, each concept is represented as an ensemble of classifiers over behavior-modality spaces weighted according to accuracy on available data (so yellow weighs look-vision highly, while *rattle* weighs *drop-audio* highly). While the objects have already been explored (i.e., they have feature representations), language labels must be gathered on the fly from human users to connect these features to words.

Different predicates afford the agent different certainties. Map-based facts such as room types (*office*) can be grounded with full confidence. For words like *red*, perceptual concept models give both a decision and a confidence value in [0, 1]. Since there are multiple possible groundings for ambiguous utterances like *the office*, and varied confidences for perceptual concept models on different objects, we associate a confidence distribution with the possible groundings for a semantic parse (Figure 2).

<table>
<thead>
<tr>
<th>Role</th>
<th>Min Prob B Role</th>
<th>Question</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>(∅, ∅, ∅, ∅, ∅)</td>
<td>All action</td>
<td>What should I do?</td>
<td>Clarification</td>
</tr>
<tr>
<td>(walk, ∅, ∅, ∅, ∅)</td>
<td>action</td>
<td>You want me to go somewhere?</td>
<td>Confirmation</td>
</tr>
<tr>
<td>(deliver, ∅, p1, ∅, ∅)</td>
<td>patient</td>
<td>What should I deliver to p1?</td>
<td>Clarification</td>
</tr>
<tr>
<td>(relocate, ∅, ∅, ∅, ∅)</td>
<td>source</td>
<td>Where should I move something from on its way somewhere else?</td>
<td>Clarification</td>
</tr>
<tr>
<td>(relocate, o1, ∅, r1, r2)</td>
<td>-</td>
<td>You want me to move o1 from r1 to r2?</td>
<td>Confirmation</td>
</tr>
</tbody>
</table>

TABLE I: Samples of the agent’s static dialog policy π for mapping belief states to questions.

The belief state, B, is a mapping from semantic roles (components of the task) to probability distributions over the known constants that can fill those roles (*action*, *patient*, *recipient*, *source*, and *goal*). The belief state models uncertainties from both the semantic parsing (e.g., prepositional ambiguity in “pod by the office to the north”; is the pod or the office north?) and language grounding (e.g., noisy concept models) steps of language understanding.

The belief states for all roles are initialized to uniform probabilities over constants. We denote the beliefs from a single utterance, x, as B_x, itself a mapping from semantic roles to the distribution over constants that can fill them. The agent’s belief is updated with

\[
B(r, a) \leftarrow (1 - \rho)B(r, a) + \rho B_x(r, a),
\]

for every semantic role r and every constant a. The parameter \( \rho \in [0, 1] \) controls how much to weight the new information against the current belief.\(^5\)

After a belief update from a user response, the highest-probability constants for every semantic role in the current belief state B are used to select a question that the agent expects will maximize information gain. Table I gives some examples of the policy π.

For updates based on confirmation question responses, the confirmed \( B_x \) constant(s) receive the whole probability mass for their roles (i.e., \( \rho = 1 \)). If a user denies a confirmation, \( B_x \) is constructed with the constants in the denied question given zero probability for their roles, and other constants given uniform probability (so Equation 1 reduces the belief only for denied constants). A conversation concludes when the user has confirmed every semantic role.

**D. Learning from Conversations**

The agent improves its semantic parser by inducing training data over finished conversations. Perceptual concept models are augmented on the fly from questions asked to a user, and are then aggregated across users in batch.

a) **Semantic Parser Learning From Conversations:** The agent identifies utterance-denotation pairs in conversations by pairing the user’s initial command with the final confirmed action, and answers to questions about each role with the confirmed role (e.g., *robert’s office* as the *goal* location \( r_1 \)), similar to prior work [1]. Going beyond prior work, the agent then finds the latent parse for the pair: a beam of parses is created for the utterance, and these are grounded to discover those that match the target denotation. The agent then retrains its parser given these likely, latent utterance-semantic parse pairs as additional, weakly-supervised examples of how natural language maps to semantic slots in the domain.

b) **Opportunistic Active Learning:** Some unseen words are perceptual concepts. If one of the neighboring words of unknown word \( x_i \) is associated with a semantic form involving a perceptual concept, the agent asks: I haven’t heard the word ‘\( x_i \)’ before. Does it refer to properties of things, like a color, shape, or weight? If confirmed, the agent ranks the nearest neighbors of \( x_i \) by distance and sequentially asks the user whether the next nearest neighbor

---

3That is, at inference time, while all objects have been explored, the language concepts that apply to them (e.g., *heavy*) must be inferred from their feature representations.

4Half the mass of non-*action* roles is initialized on the ∅ constant, a prior indicating that the role is not relevant for the not-yet-specified action.

5We set \( \rho = 0.5 \) for clarification updates.


$t_p$ is a synonym of $x_i$. If so, new lexical entries are created to allow $x_i$ to function like $t_p$, including sharing an underlying concept model (e.g., in our experiments, tall was identified as a synonym of the already-known word long). Otherwise, a new concept model is created for $x_i$ (e.g., in our experiments, the concept red).

We introduce opportunistic active learning questions [2] as a sub-dialog, in which the agent can query about training objects local to the human and the robot (Figure 5). This facilitates on the fly acquisition of new concepts, because the agent can ask the user about nearby objects, then apply the learned concept to remote test objects (Section IV-C).

IV. EXPERIMENTS

We hypothesize that the learning capabilities of our agent will improve its language understanding and usability. We also hypothesize that the agent trained in a simplified world simulation on Mechanical Turk can be deployed on a physical robot, and can learn non-visual concepts (e.g., rattling) on the fly that could not be acquired in simulation.

A. Experiment Design

The agent (and corresponding robot) can perform three high-level tasks: navigation (the agent goes to a location), delivery (the agent takes an object to a person), and relocation (the agent takes an object from a source location to a goal location). We denote 8 (randomly selected) of the 32 objects explored in prior work [38] as test objects and the remaining 24 as training objects available for active learning queries. We randomly split the set of possible task instantiations (by room, person, and object arguments) into initialization (10%), train (70%), and test sets (20%).

a) Initialization Phase: Sixteen users (graduate students) were shown one of each type of task (from the initialization set) and gave two high-level natural language commands for each (initial and rephrasing). We used a subset of these utterances as a scaffold on which to build a seed language-understanding pipeline: an initial lexicon and a set of 44 utterance-semantic parse pairs, $D_0$.

b) Training Procedure: The initial pipeline is used by a baseline agent $A_1$; we denote its parser $P_1$ trained on $D_0$, and denote untrained concept models for several predicates $P_{e,1}$. That is, the initial lexicon contains several concept words (like yellow), but no labels between objects and these concepts. All learning for the parsing and perception modules arises from human-agent conversations.

We divide the training procedure into three phases, each associated with 8 different objects from the active training set of 24. The perceptual concept models are retrained on the fly during conversations as questions are asked (e.g., as in Figure 1). The parsing model is retrained between phases. Each phase $i$ is carried out by agent $A_i$, and training on all phase conversations yields agent $A_{i+1}$ using concept models $P_{i+1}$ and parser $P_{c,i+1}$. In each phase of training, and when evaluating agents in different conditions, we recruit 150 Mechanical Turk workers with a payout of $1 per HIT.

c) Testing and Performance Metrics: We test agent $A_3$ with parser $P_3$ and perception models $P_{e,3}$ against unseen tasks and denote it Trained (Parsing+Perception). We also test an ablation agent, $A^*_3$, with parser $P^*_3$ and perception models $P_{e,4}^*$ (trained perception models with an initial, baseline parser with parsing rules only added for new concept model words), and denote it Trained (Perception). These agents are compared against the baseline agent $A_1$, denoted Initial (Seed).

We measure the number of clarification questions asked during the dialog to accomplish the task correctly. This metric should decrease as the agent refines its parsing and perception modules, needing to ask fewer questions about the unseen locations and objects in the test tasks. We also consider users’ answers to survey questions about usability. Each question was answered on a 7-point Likert scale: from Strongly Disagree (1) to Strongly Agree (7).

B. Mechanical Turk Evaluation

We prompt users with instructions like: Give the robot a command to solve this problem: The robot should be at the X marked on the green map, with a green-highlighted map marking the target. Users are instructed to command the

---

6Commands that would introduce rare predicates were dropped.

7An experimenter performed the annotations to create these resources in about two hours.

8Empirically, parser $P_4$ overfits the training data, so we evaluate with $P_3$. For $A^*_3$, this is not a concern since the initial parser parameters $P^*_1$ are used.
robot to perform a **navigation**, **delivery**, and **relocation** task in that order. The simple, simulated environment in which the instructions were grounded reflects a physical office space, allowing us to transfer the learned agent into an embodied robot (Section IV-C). Users type answers to agent questions or select them from menus (Figure 3). For **delivery** and **relocation**, target objects are given as pictures. Pictures are also shown alongside concept questions like *Would you use the word ‘rattling’ when describing this object?*

Table II gives measures of the agents’ performance in terms of the number of clarification questions asked before reaching the correct task specification to perform. For both **navigation** and **relocation**, there is a slight decrease in the number of questions between the **Initial** agent and the **Trained (Parsing+Perception)** agent. The **Trained (Perception)** agent which only re trains and adds new concept models from conversation history sees slightly worse performance across tasks, possibly due to a larger lexicon of adjectives and nouns (e.g., *can* as a descriptive noun now polysemous with *can* as a verb—*can you...*) without corresponding parsing updates. None of these differences are statistically significant, possibly because comparatively few users completed tasks correctly, necessary to use this metric.\(^9\)

Table III gives measures of the agents’ performance in terms of qualitative survey prompt responses from workers. Prompts were: *I would use a robot like this to help navigate a new building*, *I would use a robot like this to get items for myself or others*, and *I would use a robot like this to move items from place to place*. Across tasks, the **Trained (Parsing+Perception)** agent novel to this work is rated as more usable than both the **Initial** agent and the **Trained (Perception)** agent that updated only its concept models from training conversations.

\(^9\)Across agents, an average of 42%, 39%, and 9.5% workers completed **navigation**, **delivery**, and **relocation** tasks correctly, respectively. A necessary step in future studies is to improve worker success rates, possibly through easier interfaces, faster HITs, and higher payouts.

\(^10\)https://cloud.google.com/speech/

\(^11\)http://www.cstr.ed.ac.uk/projects/festival/

\(^12\)Demonstration video: https://youtu.be/PbOfteZ_CJc?t=5.

### C. Physical Robot Implementation

The browser-interfaced, Mechanical Turk agent enabled us to collect controlled training data, but our end goal is a human-robot interface in a physically shared environment. To establish that the agent and learning pipeline are robust and efficient enough to operate on real hardware in a live setting, we complement our Mechanical Turk evaluation with a demonstration of an embodied robot agent (Figure 5).

We use the BWIBot [39], [40], which can perceive and manipulate objects (Xtion ASUS Pro camera, Kinova MICO arm), navigate autonomously (Hokuyoi lidar), record and transcribe human speech (Blue Snowball microphone, Google Speech API\(^10\)), and verbalize audio responses (Festival Speech Synthesis System\(^11\)). Tabletop perception is implemented with RANSAC [41] plane fitting and Euclidean clustering as provided by the Point Cloud Library [42].

The agent is trained on Mechanical Turk conversations, transferring learned linguistic (e.g., *lounge by the conference room*) and perceptual (e.g., object classes like *can*) knowledge across platforms from simple simulation to real world application. In a live human-robot dialog, an experimenter tells the agent to *move a rattling container from the lounge by the conference room to bob’s office*, requiring the agent to select correct rooms and to learn the new, audio-grounded word *rattling* from the human user.\(^12\)

### V. Conclusion

This paper proposes a robotic agent that leverages conversations with humans to expand small, hand-crafted language understanding resources both for translating natural language commands to abstract semantic forms and for grounding those abstractions to physical object properties. We make
R: What should I do?
H: Move a rattling container from the lounge by the conference room to Bob’s office.
R: Does rattling refer to a property of an object?
H: Yes.
R: Does rattling mean the same thing as empty?
H: No.
R: I am still learning the meanings of some words.
R: Show me an object you could use the word rattling when describing, or say “none of them.”
H: Watch. This one. The human picks up a red water bottle.
R: I see.
R: Show me an object you could not use the word rattling when describing, or say “all of them.”
H: Watch. This one. The human picks up a container half-full of beans.
R: I see.
R: Show me an object you could use the word rattling when describing, or say “none of them.”
H: Watch. This one. The human picks up a small, full pill jar.
R: I see.
R: Show me an object you could not use the word rattling when describing, or say “all of them.”
H: Watch. This one.
R: I am getting the objects on the table into focus... Okay, I see them.
R: Show me an object you could use the word rattling when describing, or say “none of them.”
H: This one. The human picks up a small, full pill jar.
R: I see.
R: Thanks. Now back to business.
R: You want me to move an item from 3.514 to somewhere?
H: Yes.
R: I should move something from 3.514 to 3.510? [room numbers]
H: Yes.

Fig. 5: The Trained (Parsing+Perception) agent continues learning on the fly to achieve the specified goal.

several key assumptions, and promising areas of future work involve removing or weakening those assumptions. In this work, the actions the robot can perform can be broken down into tuples of discrete semantic roles (e.g., patient, source), but, in general, robot agents need to reason about more continuous action spaces, and to acquire new, previously unseen actions from conversations with humans [15]. When learning from conversations, we also assume the human user is cooperative and truthful, but detecting and dealing with combative users is necessary for real world deployment, and would improve learning quality from Mechanical Turk dialogs. Making a closed world assumption, our agent has explored all available objects in the environment, but detecting and exploring objects on the fly using only task relevant behaviors [43], [44] would remove this restriction. Finally, dealing with complex adjective-noun dependencies (e.g., a fake gun is fake but is not a gun) and graded adjectives (e.g., a heavy mug weighs less than a light suitcase) is necessary to move beyond simple, categorical object properties like can.

We hope that our agent and learning strategies for an end-to-end dialog system with perceptual connections to the real world inspire further research on grounded human-robot dialog for command understanding.

ACKNOWLEDGEMENTS

This work was supported by a National Science Foundation Graduate Research Fellowship to the first author, an NSF EAGER grant (IIS-1548567), and an NSF NRI grant (IIS-1637736). This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CNS-1305287, IIS-1637736, IIS-1651089, IIS-1724157), TxDOT, Intel, Raytheon, and Lockheed Martin.


