Jointly Improving Parsing and Perception for Natural Language Commands through Human-Robot Dialog

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

*Paul G. Allen School of
Computer Science and Engineering
University of Washington
jdtho@cs.washington.edu

†Department of Computer Science
University of Texas at Austin

‡Department of Computer Science
Tufts University

***Abstract*—Natural language understanding in robots needs to be robust to a wide-range of both human speakers and human environments. Rather than force humans to use language that robots can understand, robots in human environments should dynamically adapt—continuously learning new language constructions and perceptual concepts as they are used in context. In this work, we present methods for parsing natural language to underlying meanings, and using robotic sensors to create multi-modal models of perceptual concepts. We combine these steps towards language understanding into a holistic agent for jointly improving parsing and perception on a robotic platform through human-robot dialog. We train and evaluate this agent on Amazon Mechanical Turk, then demonstrate it on a robotic platform initialized from conversational data gathered from Mechanical Turk. Our experiments show that improving both parsing and perception components from conversations improves communication quality and human ratings of the agent.**

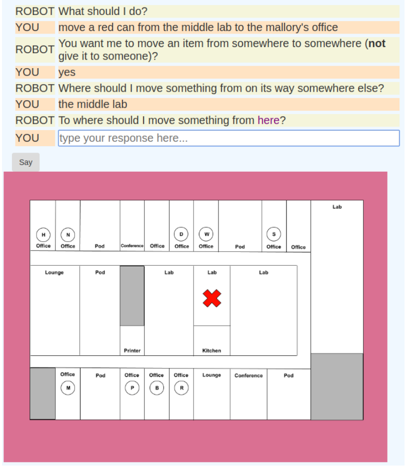
I. Introduction

Humans use natural language to articulate their thoughts and intentions to other people. As robots become ubiquitous across diverse human environments, such as homes, offices, factory floors, and hospitals, the need for smooth human-robot communication grows. The language we use to discuss these spaces varies, with domain-specific words and affordances in each (e.g., *turn on the living room lights*, *move the pallet a few feet to the north*, *notify me if the patient's condition changes*). Pre-programming robots with fixed language understanding components limits them, since different speakers and environments use and elicit different words. Robots should leverage human speaking partners as a source of additional learning signals for language understanding. Rather than force humans to use language that robots around them can understand, robots should dynamically adapt—continually learning new language constructions and perceptual concepts as they are used in context. The main thrust of this paper is to bring together methods for improving semantic understanding and grounded perceptual understanding.

Untrained human users providing natural language commands to robots expect world knowledge, perceptual knowledge, and semantic understanding from verbal robots. Translating human utterances to semantic meanings helps handle the synonymy of commands and words (e.g., *Bob* for *Robert*), compositionality (e.g., *Bob's office*, *the light mug*), and ambiguity (e.g., *light* in weight versus *light* in color). To learn a mapping between commands and their semantic forms, a *semantic parser* is often employed. Building a semantic parser requires expert annotation of a lexicon mapping words to their meanings, an ontology of relevant concepts in the world, and training examples of natural language sentences paired with composed meanings. In this work, we ameliorate this annotation effort by leveraging conversations that robots have with human users. We improve a parser built using sparse annotated resources using weak supervision from such conversations. This approach builds on past work, presented in full in [37].

To converse about the environment they share with humans, robots must gather and maintain world knowledge through perception. Some world knowledge can be modeled as static, such as the layout of a building, ownership relations between people and rooms (e.g., *Bob's office*), or assignments between patients and doctors in a hospital. Other world knowledge is perceptual, such as whether an object is a *mug*, where some movable objects were last seen, and whether an object can be picked up and moved somewhere else. Gathering correspondences between objects in the world and perceptual concepts, or predicates, applicable to those objects (e.g., *heavy, brown, mug*) is a time-consuming annotation effort if performed exhaustively. In this work, we instead extract this information from natural human-robot conversations on-the-fly.

We present a holistic system for jointly improving parsing and perception on a robotic system for natural language commands through human-robot dialog. This learning agent uses clarification questions in a conversation with a human partner to understand language commands. The agent induces additional training data for a semantic parser, similar to prior



ROBOT What should I do?
YOU move a red can from the middle lab to the mallory's office
ROBOT You want me to move an item from somewhere to somewhere (**not** give it to someone)?
YOU yes
ROBOT Where should I move something from on its way somewhere else?
YOU the middle lab
ROBOT To where should I move something from here?
YOU type your response here...

Fig. 1: Mechanical Turk web interface used to conduct our experiments. The user types a command to the learning agent, which replies with questions to clarify the command until the user confirms that the agent has correctly understood. In this conversation, the agent has just asked a confirmation question—whether it has understood the right location from where the *red can* should be taken. The colored text *here* refers to the pink-outlined map below the conversation window.

work [37], strengthening its parsing over time. The agent also uses opportunistic active learning [39] to ask questions about nearby objects to refine multi-modal perceptual concept models [38] on-the-fly during command dialogs.

We evaluate this language understanding agent on Mechanical Turk with hundreds of users. Figure 1 shows the Mechanical Turk interface we created, with an example command and the beginning of a human-agent dialog. To demonstrate flexibility for understanding non-visual predicates, we also implement the agent on a physical robot with an arm [16], using the learning agent as a back-end to drive human-robot dialog.[1] As more training conversations are seen by the agent in Mechanical Turk, users are better able to communicate tasks to the agent. Users rate the agent more favorably for use in deployed tasks as more training conversations become available.

II. Related Work

Instructing robots through natural language is essential for humans and robots to cooperate in shared environments. Research in this space spans semantic parsing, robotic perception and grounding, and human-robot dialog.

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 using language guidance [19, 24, 25]. Recent methods perform semantic parsing translations using sequence to sequence [18, 14, 20] or sequence-to-tree [8] neural networks. One framework, based on generalized grounding graphs, aims to both understand human language requests about objects in the world and generate language requests regarding the shared environment [36]. Extensions of this framework can be used to memorize new semantic referents in a dialog, like *this is my snack* [30], or to reason about abstract sets and ordinality [29]. In this work, our agent can learn new referring expressions and novel perceptual concepts on-the-fly through dialog.

Mapping from a referring expression such as *the red cup* to an object referent in the world is an example of the *symbol grounding problem* [12]. *Grounded language learning* bridges internal represenations of information in a machine with natural language. Most work in this space has grounded language using visual perception [10, 21, 42, 23]. There has been some work on combining language with sensory modalities other than vision, such as audio [17] and haptic [4, 11] signals. A recent survey attempts to cover this broad space of multi-modal representations of objects and concepts in machine learning [1]. Some neural methods bypass any explicit modeling of language predicates and instead train end-to-end on tasks such as localizing an object in a given image given a target query in natural language [13], and translating human instructions directly to grounded behavior like route-following [26] or question answering [7]. In this work, we explicitly model language predicates that refer to spatial relations and categories in an office environment, as well as perceptual predicates that refer to properties of objects, the latter of which is an open set.

Some researchers gather data for this kind of perceptual grounding using interaction with a human interlocutor. This combination of dialog and perception affords new opportunities, such as the robot asking questions targeting weaknesses in its understanding [38]. Previous work on learning to ground object attributes and names using dialog framed the data gathering phase as a *20 Questions*-style [40] or *I Spy* [28, 38] game. Neural approaches have been used to train grounded robot-robot conversational agents [6], and may be applicable for human-robot dialog in future works. We do not use neural methods, which share a data hungry weakness and fail to converge to useful solutions when training examples are sparse, such as in human-robot dialogs where gathering environment-specific interactions is costly.

We present a robotic agent that understands requests for actions in natural language that include both domain knowledge (for example, a building floorplan) and environmental perceptual information (for example, learned properties of objects), using semantic parsing as an understanding step and learning multi-modal perceptual concept models using supervision elicited during conversations.

III. Conversational Agent

We implement and evaluate a conversational dialog agent that uses a semantic parser to translate human utterances into semantic meaning representations, then grounds those meaning representations using both a static knowledge base of facts about an office environment and perceptual concept models

[1] The demonstration video can be viewed at https://youtu.be/PbOfteZ_CJc

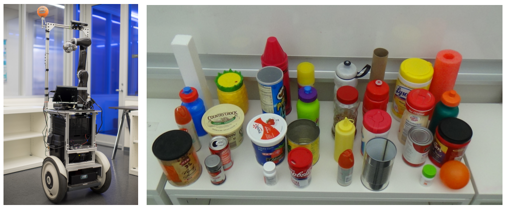


Fig. 2: The robot used in our experiment, and the objects explored by the robot for grounding perceptual predicates.

that consider multi-modal representations of physical objects.[2]

A. Semantic Parser

We use the Combinatory Categorial Grammar (CCG) formalism [35] in our lexicon to perform Cocke-Kasami-Younger (CKY) chart parsing [41] on input sentences, and use a unification-based grammar to add new lexical entries during training. We add ontological entries dynamically during conversations with human users (for example, when a new perceptual concept like *red* is used for the first time). In the interest of space, we refer the reader to other work in learning statistical parsers for details [22].

In order to assign a semantic meaning to a sequence of tokens, each token that is not skipped must have an entry in the lexicon from which its semantic meaning can be determined. We use word embeddings [27] to augment the lexicon at test time to attempt to recover from out-of-vocabulary words, an idea similar in spirit to previous work [2], but formally integrated into our parsing pipeline. This allows, for example, unseen word *grab* to use the lexical entry for the nearby (in embedding space), known word *take* at test time.

B. Multi-modal Perception

Once a command has been translated into a semantic form, *grounding* that semantic form to actions, objects, and rooms in the real world must take place before the robot can act on the command. For objects, perceptual concepts like *red* and *heavy* require considering sensory perception of physical objects. We build multi-modal concept models to connect robot perception to concept labels. We use multi-modal feature representations across various sensorimotor contexts by exploring those objects with a robot arm (Figures 2 and 3), as detailed in previous work [34, 38].

We connect these feature representations of objects to language labels by learning discriminative classifiers on the feature spaces for each perceptual language concept, as detailed in previous work [33, 38]. Relevant for this work, each trained classifier produces both a decision and a confidence (in $[0, 1]$) when evaluating a test object for a language concept (e.g., an object is *red* with confidence 0.8). These confidence values are also used to drive an opportunistic active learning strategy for improving concept models during conversations.

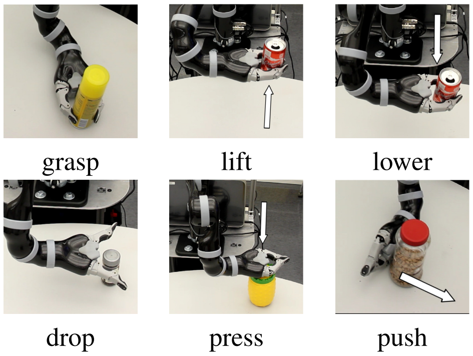


Fig. 3: The behaviors the robot used to explore the objects. In addition, the *hold* behavior (not shown) was performed after the *lift* behavior by holding the object in place.

C. Language Grounding

To execute a command, an utterance is first translated into a semantic form. Some forms must be instantiated in a particular context. For example, *the office by the kitchen* refers to a physical location in an environment, but the utterance means different such locations depending on where it is uttered, and must be *grounded* to the current environment.

Static facts such as room types (*office*) and relations (*owns(robert, room1)*) can be looked up in a provided floorplan, such that unambiguous noun phrases can be grounded with full confidence. For perceptual predicates, concept models return 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 create a confidence distribution over the possible groundings for a semantic parse. This confidence probability distribution is used as part of an update procedure for helping the agent understand the user's intent during dialog.

D. Dialog Policy

We implement a conversational dialog agent $\mathcal{A}$ for command understanding similar to that in previous work [37]. The differences between this agent and the previous one are: 1) grounding semantic parses in both static knowledge and perceptual knowledge; 2) dynamically adding new ontological predicates for novel perceptual concepts; 3) leveraging opportunistic active learning for refining perceptual concept models on-the-fly; and 4) semantic parser training from pairs of utterances and denotations.

a) Clarification Dialog Policy: Dialog begins with a human user commanding the robot to perform a task. The agent maintains a belief state modeling the unobserved true task in the user's mind, and uses the language signals from the user to infer it. The command is first parsed by the agent's semantic parser, then grounded against static and perceptual knowledge with denotation procedure, which results in a set of pairs of denotations of the semantic parser's understanding of the command and associated confidence values. Using these

[2] The source code for this conversational dialog agent, as well as the experiments described in the following section, can be found at https://github.com/thomason-jesse/grounded_dialog_agent

denotations and their confidence distribution, we update the agent's belief state, then engage in a clarification dialog to refine that belief.

The agent's belief state, $\mathcal{B}$, is a mapping from semantic roles (components of the task) to probability distributions over the ontological constants that can fill those roles (*action*, *patient*, *recipient*, *source*, and *goal*).

The belief state for the *action* role is initialized with uniform probabilities across three actions (*walk*, *deliver*, and *relocate*). The remaining role belief states are initialized with half of the probability mass on an *unknown* constant, $\varnothing$, indicating that the role is not known or is not necessary for the action the user has in mind, and the remaining half of the probability mass is distributed uniformly across all constants that can fill the role.

We call the collection of beliefs from a single utterance, x (a command or question answer), $\mathcal{B}_x$, a mapping from semantic roles to the distribution over constants that can fill them. We update the agent's belief based on new utterance x:

$$\mathcal{B}(r, a) \leftarrow (1 - \rho)\mathcal{B}(r, a) + \rho\mathcal{B}_x(r, a), \tag{1}$$

for every semantic role r and every constant a (for example, as in Figure 4). The parameter ρ controls how much to trust the new information versus the current belief (in our experiments, we set $\rho = 0.5$).

The dialog agent poses questions to the user regarding different semantic roles. The highest-probability constant for every semantic role in the current belief state $\mathcal{B}$, together with which among those roles has the least probability, are used to select a question. Table I gives some examples of the policy π.

For confirmation questions, the confirmed $\mathcal{B}_x$ constant(s) receive the whole probability mass for their roles, and ρ is set to 1 for the update in Equation 1, such that $\mathcal{B}$ reflects the confirmation. If a user denies a confirmation question, $\mathcal{B}_x$ is constructed with the constants in the denied question given zero probability weight for their roles, and other constants given a uniform weight, such that the update in Equation 1 reduces the belief only for denied constants. A conversation concludes when the user has confirmed every semantic role.

b) Detecting Perceptual Words and Synonyms: When describing objects in the real world, humans can use words the agent has never heard before. Some of these are perceptual concepts—words that need to be grounded in the physical world. In prior work, a stopword list is used to remove non-content words, and all content words in human descriptions of objects are considered perceptual concept words [38, 39].

In this work, if one of the neighboring (among the nearest 3) words of unknown word x_i (in word-embedding distance) has a semantic form involving a perceptual predicate, we ask the user whether the unseen word x_i is also perceptual in nature. The question posed is: *I haven't heard the word 'x_i' before. Does it refer to properties of things, like a color, shape, or weight?*. If the user answers *yes*, we attempt to discover whether x_i is a synonym of an already known perceptual concept, such as one of the identified neighbors.

We rank the nearest neighbors of x_i by distance and sequentially ask the user whether the next nearest neighbor 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 perceptual concept model. For example, in our experiments, previously unseen word *tall* was added as a synonym for the known word *long*. If no synonym is identified, a new ontological concept is created to represent x_i. For example, in our experiments, the color concept word *red* was added with a new ontological predicate to represent it.

c) Opportunistic Active Learning during Conversation: We introduce opportunistic active learning questions as a sub-dialog routine for the agent, in which it can query about objects *local* to the human and the robot (e.g. objects in the room where the conversation is happening) to refine its perceptual concept models before applying them to the *remote* test object items (e.g. items that are physically in a different room but being discussed in the conversation), a strategy employed for object selection from language descriptions in previous work [39].

Objects in the nearby *active training set* can be labeled by asking the human questions during a conversation about whether particular predicates apply to them. Prior work established that an agent asking questions about both on-topic (used in the current conversation) and off-topic (irrelevant to the current user's needs) predicates outperformed an agent that only asked about predicates in the current human description [39]. We allow our agent to ask both on- and off-topic questions, moving to off-topic ones only if there are no more useful on-topic labels to query. Our question selection strategy is similar to that described in this prior work [39]. Using this sub-dialog, the agent is able to query the user for labels on the active training objects, O_{tr}, to improve its perceptual classifiers before continuing its clarification conversation and possibly selecting a described object in the *active test set*, O_{te}.

E. Learning from Conversations

Past work retrains a semantic parser from conversations an agent has with a human [37]. In this work, we expand on this retraining procedure and make a distinction between denotation parses and latent semantic parses. That is, we differentiate between *robert's office* and its denotation, r_1, where in the strategy presented in previous work, the denotation would be treated as the latent form of *robert's office*, weakening its ability to generalize from induced data and creating more reliance on a robust initial lexicon. We first induce utterance-denotation pairs from conversations, then induce latent semantic forms that connect those utterances and denotations.

For details on inducing utterance-denotation pairs from conversations, we refer to previous work [37]. In this work, given an utterance-denotation pair, we discover a semantic parse that can be derived from the input utterance and has a denotation matching the known one for that utterance. We formulate this training and finding of the latent semantic form similar to past work on learning statistical, compositional

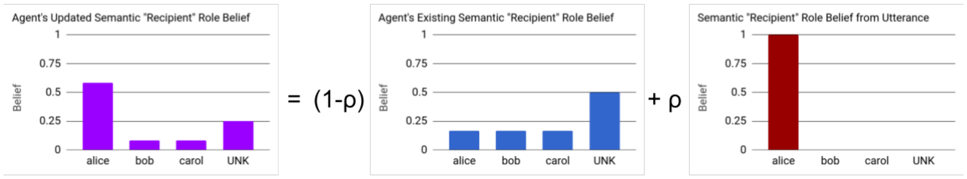


Fig. 4: Example belief update for the *recipient* role. This could arise from the question *To whom should I bring something?* being answered *alice*.

B **max per role** (***action*, *patient*, *recipient*, *source*, *goal***)	**Min Prob** B **Role**	**Question**	**Type**
$(\varnothing, \varnothing, \varnothing, \varnothing, \varnothing)$	All	What should I do?	Clarification
(walk, $\varnothing, \varnothing, \varnothing, r_1$)	*action*	You want me to go somewhere?	Confirmation
(deliver, $\varnothing, p_1, \varnothing, \varnothing$)	*patient*	What should I deliver to p_1?	Clarification
(relocate, $\varnothing, \varnothing, \varnothing, \varnothing$)	*source*	Where should I move something from on its way somewhere else?	Clarification
(relocate, $o_1, \varnothing, r_1, r_2$)	-	You want me to move o_1 from r_1 to r_2?	Confirmation

TABLE I: Samples of the agent's static dialog policy π for mapping belief states (left) to questions (right). In the Mechanical Turk experiments described in Section IV, constants like people (p_1), objects (o_1), and rooms (r_1, r_2) were represented pictorially, with pronouns (*this person, this, here, there*) in place of their variables in the sentence shown.

semantic parsers [22]. At a high level, a beam of parses is created for the utterance, and these are grounded to discover which matches the target denotation (selecting among these parses the one with highest joint confidence between parsing and grounding). After inferring these latent parses, we train the parser on the discovered utterance-semantic parse pairs.

IV. Experiments

We evaluate our agent with hundreds of users through the Mechanical Turk interface, asking human users who instruct it 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*). After training the agent using data from conversations it had with users on Mechanical Turk, we instantiate the trained agent on a physical robot.

A. Experiment Design

We deploy the agent in a simulated office environment populated by rooms, people, and object items. We fix 8 of the 32 objects explored in prior work [34] as possible arguments to the tasks for our experiments (selected at random), and use the remaining 24 as training objects available for opportunistic active learning queries for learning concept models. We randomly split the set of possible tasks into initialization (10%), train (70%), and test sets (20%).

a) Initialization Phase: Sixteen users (graduate students at the university across several fields) engaged with a faux-agent using the web interface. They were shown one of each type of task, drawn from the initialization set, and gave two high-level commands for each (the faux-agent simply asked the user to rephrase each high-level command once, with not following clarification dialog). We used these commands as a scaffold on which to build an ontology, lexicon, and initial utterance-semantic parse pairs. Of them, 44 pairs, D_0, were used to train an initial parser.

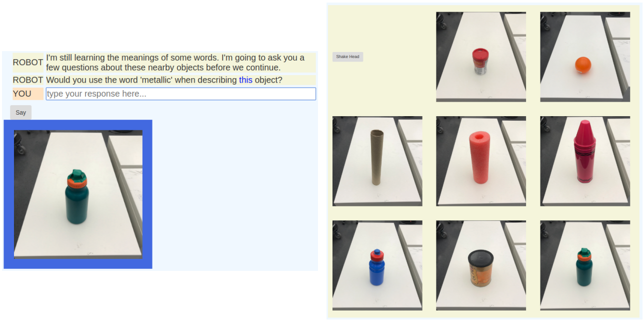
b) Training Procedure: We use these initial parsing resources to create a baseline agent $\mathcal{A}_1$ with a parser $\mathcal{P}_1$ trained only on the initialization pairs D_0 mentioned above and concept models for several predicates $P_{c,1}$, but with no initial object examples against which to train them. All learning for the parser and perception modules arises naturally from conversations the agent has with humans.

We divide the training procedure into three phases, each associated with 8 objects from the active training set, which can be queried about during conversations by the agent. Between phases, the parser and perception models are retrained. Each phase i is carried out by agent $\mathcal{A}_i$, after which parser $\mathcal{P}_{i+1}$ and concept predicates $P_{c,i+1}$ are trained to instantiate agent $\mathcal{A}_{i+1}$.

After three training phases, agent $\mathcal{A}_4$ with parser $\mathcal{P}_4$ and perception models $P_{c,4}$ is tested by interacting with users trying to accomplish tasks from the unseen test set of tasks. We also test an ablation agent, $\mathcal{A}_4^*$, with parser $\mathcal{P}_1^*$ and perception models $P_{c,4}$ (trained perception with simply initialized parser).

c) Performance Metrics: Quantitatively, we measure the *semantic f-score* of each user on each task. This metric is a measure of the agreement between the task the user confirmed and the task they were instructed to convey, and is used as a measure of how close users came to conveying the correct task [5]. The metric is defined as the harmonic mean of the precision and recall between the sets T_U, the set of pairs of roles and constants the user confirmed, and T_G, the gold task specification pairs.

We also consider user's answers to survey questions about whether they would use the agent for the three tasks in the real world (Figure 7). Each questions was answered on a 7-point Likert scale: *Strongly Disagree* (0), *Disagree* (1), *Slightly Disagree* (2), *Neutral* (3), *Slightly Agree* (4), *Agree* (5), *Strongly Agree* (6). Users are also able to provide optional, free-form text feedback as a part of completing this survey,



ROBOT I'm still learning the meanings of some words. I'm going to ask you a few questions about these nearby objects before we continue.
ROBOT Would you use the word 'metallic' when describing this object?
YOU type your response here...

Fig. 5: Web interface for the agent asking whether a predicate applies to an object (Left), and for positive/negative examples (Right) (e.g., *could you show me one you would use the word 'red' when describing, or shake your head if there are none?*).

and their responses provide some anecdotal insight into their experiences with the agent.

B. Mechanical Turk Evaluation

Workers connect to our web interface and engage in three conversations, then fill out a survey about their experience using the agent. To avoid biasing workers towards certain words (linguistic priming), we present tasks by describing the target state of the world after the task is completed. For example, for the navigation task, the prompt is: *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 visually marking the target. Figure 1 shows a clarification and confirmation question for the *goal* semantic role of the delivery task. The agent also asks questions in sub-dialogs regarding whether perceptual concept words apply to objects. Figure 5 shows an example of such a yes/no question and of the panel used for open-ended positive/negative example questions (asking, among available objects, which one a word does or does not apply to).

a) Human Intelligence Tasks (HITs): We run 50 HITs at a time on our server. For the train condition, we run two batches per fold of active training set objects, for a total of $2 \times 3 \times 50 = 300$ workers. For the test conditions—without parser or perception training ($\mathcal{A}_1$), with perception training only ($\mathcal{A}_4^*$), and with parser and perception training ($\mathcal{A}_4$)—we run three batches of 50 workers each for a total of $3 \times 50 = 150$ workers.

In addition to removing workers who timed out (2 hour limit) or had exceedingly long conversations (30 human dialog turns), we *vet* the remaining set of workers by removing repeat workers and workers who confirmed *navigation* commands with the agent for all three target tasks[3].

Table II gives a breakdown of the numbers of workers who engage with our HITs through different experimental conditions. Only workers who submit the HIT with the correct survey code (i.e. actually use the interface) are considered for training the system (for the train condition) and evaluation (for the test condition). For training the parser, conversations are only included if the worker confirms the correct task. The low number of workers that complete the tasks given that they submitted the HIT at all gives a sense of how difficult the HIT is compared to others on Mechanical Turk.

b) Quantitative Performance Results: Figure 6 gives quantitative measures of the agent's performance in the untrained condition ($\mathcal{A}_1$), the trained condition where *only* the perception modules are updated based on user conversations ($\mathcal{A}_4^*$), and the trained condition where *both* the parsing and perception modules are updated based on user conversations ($\mathcal{A}_4$).

For *navigation* (Figure 6a), little changes, possible due to the low number of semantic roles (2) involved. For *delivery* (Figure 6b), the score increases most when we retrain the perception module (the *patient* argument, the physical object, becomes easier to select). For *relocation* (Figure 6b), the score only increases when we retrain both the parsing and perception modules (this is consistent with the roles in this task: two locations on the map, referring to which becomes easier with a better parser, and an object in the real world, referring to which becomes easier with better perception modules).

c) User Survey Results: Across all three tasks, we see a slight increase in user ratings of usability between the untrained condition and the trained parsing and perception module condition. The improved parser may affect users' perception of the agent as a whole, regardless of its performance on individual tasks, making the ratings users give to the usability of these three tasks co-dependent.

We track the responses on the survey's open response text box as repeat users finish HITs in different conditions, obtaining qualitative feedback from users whose data we otherwise discard as they repeat the task. For example, one user participated in HITs across two learning phases and then one testing condition. The user first experience agent $\mathcal{A}_2$ (one phase of training), and wrote: *Ugh. I can never figure out how to get it to understand that red and white container with the snap lid! It always goes for the soda can instead. Argh. ...* The second time, with agent $\mathcal{A}_3$, the user wrote: *A good day for Mr.Robot. It's nice to have progress...* Finally, with agent $\mathcal{A}_4$, used with test set tasks, the user wrote: *Wow. It's made some progress. It was a lot easier to parse this time...*

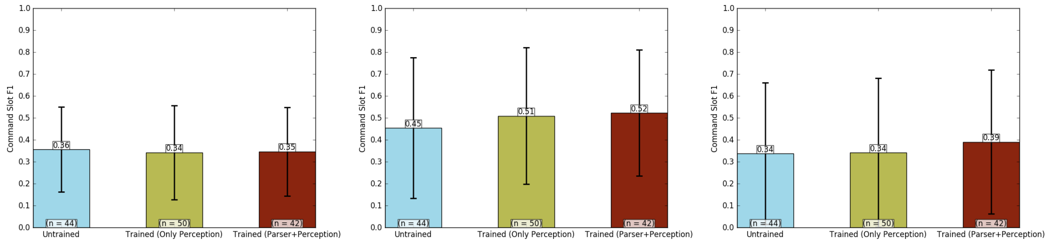
d) Learned Perceptual Concept Models: The agent acquires new perceptual concept models (25 in total), and synonym words for existing concepts, during the three phases of training. The learned concept models are noisy, given that Mechanical Turk workers are sometimes inattentive in the long HIT.[4] Nonetheless, these learned models quantitatively and qualitatively improve user experience with the agent. Table 8 shows the learned perceptual concept model for *can* on test objects.

[3] On inspection, these workers identify, during the navigation task (the first dialog), that the robot will advance to the next phase once a command is confirmed, and they continue issuing navigation commands because these are fast to resolve.

[4] For example, nine workers labeled a uniformly yellow mustard container as a positive example for *red*.

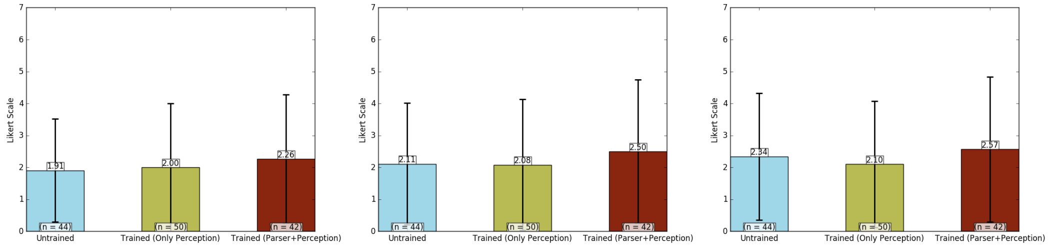
Condition	**Number of Workers**					
	Submitted HIT	Completed Tasks	Vetted	Nav. Correct	Del. Correct	Rel. Correct
Train (A_1, A_2, A_3)	297	162	113	36	44	18
Untrained (A_1)	150	67	44	17	22	10
Test* (A_4^*)	148	83	50	20	29	10
Test (A_4)	143	79	42	16	23	10

TABLE II: Breakdown of the number of workers in our experiment. We here count only workers that **submitted** the HIT with the correct code. Workers that **completed** all tasks and the survey finished the HIT entirely. **Vetted** workers' data was kept for evaluation. The Train condition (A_1, A_2, A_3 agents) draws from the training set of tasks, while the Untrained (A_1 untrained agent), Test* (A_4^* agent with trained perception and untrained parser), and Test (A_4 agent with trained parser and perception) conditions draw from the test set of tasks.



(a) Navigation Semantic Slot F1 (b) Delivery Semantic Slot F1 (c) Relocation Semantic Slot F1

Fig. 6: The average semantic slot f scores between the semantic roles in the target task and the task confirmed by the user.



(a) *I would use a robot like this to help navigate a new building.* (b) *I would use a robot like this to get items for myself or others.* (c) *I would use a robot like this to move items from place to place.*

Fig. 7: Survey prompt responses about usability.

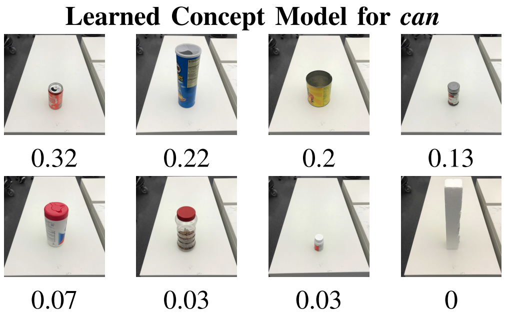


Fig. 8: The perceptual concept model learned for *can* after training from conversations with human users. The numbers below each object in the active test set represent their (rounded) portion of the distribution of probability based on outputs from the *can* concept model.

C. Physical Robot Implementation

For our embodied demonstration[5] (Figure 9), we use the BWIBot [15, 16], equipped with a Kinova MICO arm (for pointing to and manipulating objects), an Xtion ASUS Pro camera (for viewing objects on tabletop surfaces), a Hokuyo lidar (for navigation and obstacle avoidance), a Blue Snowball microphone, and a speaker (for conducting the dialog). For this demonstration, speech transcripts are provided by the Google Speech API[6] and speech synthesis is performed with the Festival Speech Synthesis System.[7] Tabletop perception, required for both the dialog interaction and the execution of the resulting command, is implemented with RANSAC [9] plane fitting and Euclidean clustering as provided by Point Cloud Library [31].

[5] The demonstration video can be viewed at https://youtu.be/PbOfteZ_CJc.

[6] https://cloud.google.com/speech/

[7] http://www.cstr.ed.ac.uk/projects/festival/

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.

R: I am getting the objects on the table into focus. . . Okay, I see them.

H: 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. *The human picks up a red water bottle*

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 container half-full of beans*

R: I see.

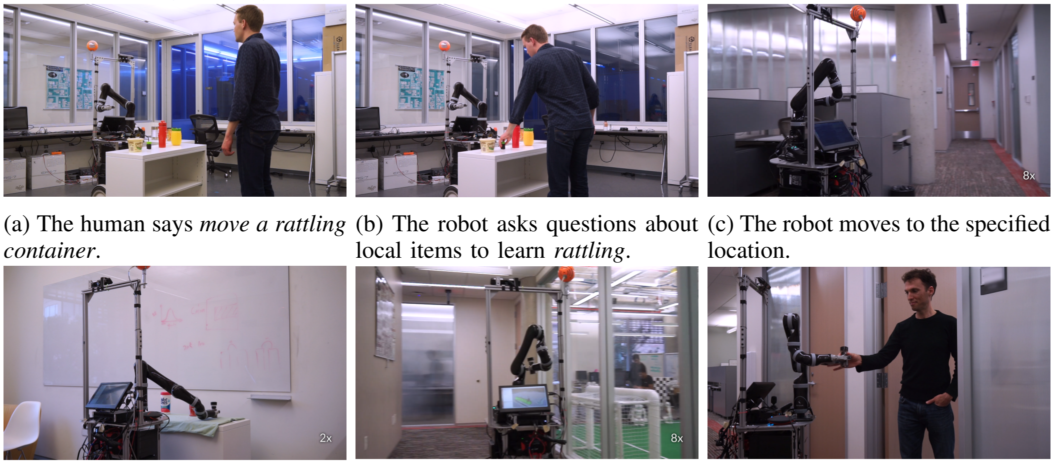
R: Thanks. Now back to business.

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?

H: Yes.



(a) The human says *move a rattling container*.

(b) The robot asks questions about local items to learn *rattling*.

(c) The robot moves to the specified location.

(d) The robot decides which item is a *rattling container* and grasps it.

(e) The robot navigates to the specified destination room.

(f) The robot hands over the item at the specified destination.

Fig. 9: The agent learns a new word, *rattling*, which requires perception using the auditory sensing modality, and uses this new concept model to correctly identify and move the target item.

V. Conclusion

In this article, we presented a holistic system for jointly improving semantic parsing and grounded perception on a robotic system for interpreting natural language commands during human-robot dialog. We show, via a large-scale Mechanical Turk experiment, that users are better able to communicate tasks and rate the system more usable after this dialog-based learning procedure. We embody this learning agent in a physical robot platform to demonstrate its learning abilities for the non-visual word *rattling*.

We currently use the penultimate layer of the VGG network [32] as a sensorimotor context space for *look*ing at objects, and in the future could similarly use auto-encoders over object representations [3] to provide a reduced feature vector representing a *learned* feature space for every sensorimotor context. There may be room for leveraging transfer learning between similarly-deployed robots (for example, in different hospitals) to increase the amount of human-robot language data available. This augmentation could enable learning more data-hungry, but less brittle, neural parsing methods [8].

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.

References

[1] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *arXiv preprint arXiv:1705.09406*, 1705, 2017.

[2] Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2747–2753, July 2016.

[3] Benjamin Burchfiel and George Konidaris. Generalized 3d object representations using bayesian eigenobjects. In *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

[4] Vivian Chu, Ian McMahon, Lorenzo Riano, Craig G McDonald, Qin He, Jorge Martinez Perez-Tejada, Michael Arrigo, Naomi Fitter, John C Nappo, Trevor Darrell, et al. Using robotic exploratory procedures to learn the meaning of haptic adjectives. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3048–3055. IEEE, 2013.

[5] Rodolfo Corona, Jesse Thomason, and Raymond J. Mooney. Improving black-box speech recognition using semantic parsing. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP-17)*, November 2017.

[6] Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[7] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Li Dong and Mirella Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 33–43, 2016.

[9] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL http://doi.acm.org/10.1145/358669.358692.

[10] Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. Learning distributions over logical forms for referring expression generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[11] Yang Gao, Lisa Anne Hendricks, Katherine J Kuchenbecker, and Trevor Darrell. Deep learning for tactile understanding from visual and haptic data. In *International Conference on Robotics and Automation (ICRA)*, pages 536–543. IEEE, 2016.

[12] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[13] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12–22, 2016.

[15] Piyush Khandelwal, Fangkai Yang, Matteo Leonetti, Vladimir Lifschitz, and Peter Stone. Planning in Action Language $\mathcal{BC}$ while Learning Action Costs for Mobile Robots. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2014.

[16] Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, J. K. Aggarwal, Raymond Mooney, and Peter Stone. Bwibots: A platform for bridging the gap between ai and human–robot interaction research. *The International Journal of Robotics Research (IJRR)*, 36, February 2017.

[17] Douwe Kiela and Stephen Clark. Multi- and cross-modal semantics beyond vision: Grounding in auditory perception. In *Proceedings of the 2015 Conference on Emperical Methods in Natural Language Processing*, pages 2461–2470, Lisbon, Portugal, 2015.

[18] Tomáš Kočiskỳ, Gábor Melis, Edward Grefenstette, Chris Dyer, Wang Ling, Phil Blunsom, and Karl Moritz Hermann. Semantic parsing with semi-supervised sequential autoencoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, November 2016. Association for Computational Linguistics.

[19] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, HRI '10, pages 259–266, 2010.

[20] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 2017 Conference of the Association for Computational Linguistics (ACL)*, 2017.

[21] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206, 2013.

[22] Percy Liang and Cristopher Potts. Bringing machine learning and compositional semantics together. *Annual Review of Linguistics*, 1(1):355–376, 2015.

[23] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of

language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

[24] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. In *International Symposium on Experimental Robotics (ISER)*, 2012.

[25] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*, pages 403–415. Springer International Publishing, 2013.

[26] Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of AAAI*, 2016.

[27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada, 2013.

[28] Natalie Parde, Adam Hair, Michalis Papakostas, Konstantinos Tsiakas, Maria Dagioglou, Vangelis Karkaletsis, and Rodney D. Nielsen. Grounding the meaning of words through vision and interactive gameplay. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1895–1901, Buenos Aires, Argentina, 2015.

[29] Rohan Paul, Jacob Arkin, Nicholas Roy, and Thomas M. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Proceedings of Robotics: Science and Systems*, Ann Arbor, Michigan, June 2016.

[30] Rohan Paul, Andrei Barbu, Sue Felshin, Boris Katz, and Nicholas Roy. Temporal grounding graphs for language understanding with accrued visual-linguistic context. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4506–4514, 2017. doi: 10.24963/ijcai.2017/629. URL https://doi.org/10.24963/ijcai.2017/629.

[31] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[33] Jivko Sinapov, Connor Schenck, and Alexander Stoytchev. Learning relational object categories using behavioral exploration and multimodal perception. In *IEEE International Conference on Robotics and Automation*, 2014.

[34] Jivko Sinapov, Priyanka Khante, Maxwell Svetlik, and Peter Stone. Learning to order objects using haptic and proprioceptive exploratory behaviors. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[35] Mark Steedman and Jason Baldridge. Combinatory categorial grammar. In Robert Borsley and Kersti Borjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*. Wiley-Blackwell, 2011.

[36] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. In *Proceedings of Robotics: Science and Systems (RSS)*, Berkeley, California, 2014.

[37] Jesse Thomason, Shiqi Zhang, Raymond Mooney, and Peter Stone. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1923–1929, July 2015.

[38] Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond Mooney. Learning multi-modal grounded linguistic semantics by playing “I spy”. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3477–3483, July 2016.

[39] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J. Mooney. Opportunistic active learning for grounding natural language descriptions. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL-17)*, volume 78, pages 67–76. Proceedings of Machine Learning Research, November 2017.

[40] Adam Vogel, Karthik Raghunathan, and Dan Jurafsky. Eye spy: Improving vision through dialog. In *Association for the Advancement of Artificial Intelligence*, pages 175–176, 2010.

[41] Daniel Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10:189–208, 1967.

[42] Larry Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *Computer Vision and Pattern Recognition (CVPR)*, December 2013.