Simultaneous Intention Estimation and Knowledge Augmentation via Human-Robot Dialog

Sujay Bajracharya, Saeid Amiri
Cleveland State University
{s.bajracharya; s.amiri}@vikes.csuohio.edu

Jesse Thomason
University of Washington
jdtho@uw.edu

Shiqi Zhang
SUNY Binghamton
szhang@cs.binghamton.edu

***Abstract*—Robots have been able to interact with humans using natural language, and to identify service requests through human-robot dialog. However, few robots are able to improve their language capabilities from this experience. In this paper, we develop a dialog agent for robots that is able to interpret user commands using a semantic parser, while asking clarification questions using a probabilistic dialog manager. This dialog agent is able to augment its knowledge base and improve its language capabilities by learning from dialog experiences, e.g., adding new entities and learning new ways of referring to existing entities. We have extensively tested our dialog system using a mobile robot that is capable of completing object delivery and human guidance tasks. Experiments were conducted both in simulation and with human participants. Results suggest that our dialog agent performs better in both efficiency and accuracy when compared to one that does not learn from this experience and another that learns using a predefined strategy.**

Fig. 1. Segway-based mobile robot platform used in this research.

I. Introduction

Mobile robots have been extensively used to conduct tasks such as object delivery in the real world. Notable examples include the Amazon warehouse robots and the Relay robots from Savioke. However, these robots either work in human-forbidden environments, or have no interaction with humans except for obstacle avoidance. Researchers are developing mobile robot platforms that are able to interact with and provide services to people in everyday, human-inhabited environments [9, 28, 7, 3]. Some of the robot platforms can learn from the experience of human-robot interaction (HRI) to improve their language skills, e.g., learning new synonyms [26], but none of them learn entirely new entities. This work aims to enable the robots to simultaneously identify service requests through human-robot dialog, and learn from this experience to augment the robot's knowledge base (KB).

A robot dialog system typically includes the following four components: 1) A language understanding component for converting spoken or text-based language inputs into a formal representation that computers understand; 2) A state tracking component that estimates the current dialog state based on the language understanding; 3) A dialog management component that suggests a language action (e.g., clarification questions); and 4) A language generator that outputs spoken or text-based natural language. The dialog agent developed in this work includes the four components, and further supports dialog-based knowledge augmentation.

There are at least two types of dialog systems that can be distinguished based on their design purposes. The first type focuses on maximizing social engagement, e.g., Microsoft XiaoIce, where the dialog agent usually prefers extended conversations. We are concerned with the second type of dialog systems, often referred to as being goal-oriented, that aim at maximizing information gain. Goal-oriented dialog systems can be evaluated based on dialog efficiency and accuracy. In this setting, people prefer dialog agents that are able to accurately identify human intention using fewer dialog turns.

Goal-oriented dialog systems are necessary for language-based human-robot interaction because, in most cases, people cannot *fully* and *accurately* deliver information using a single dialog turn. Consider a service request of "*Robot, please deliver a coffee to the conference room!*" It is possible that the robot does not know which conference room the speaker is referring to, in which case it is necessary to ask clarification questions such as "*Where should I deliver a coffee?*" To further identify the service request, the robot might want to ask about the recipient as well: "*For whom is the delivery?*" Although such dialog systems have been implemented on robots, few of them can learn to improve their language capabilities or augment their KB from the experience of human-robot conversations in the real world (Section II).

This work focuses on dialog-based robot knowledge augmentation, where the agent must identify when it is necessary to augment its KB and how to do that, as applied to our Segway-based mobile robot shown in Figure 1. Partially observable Markov decision processes (POMDPs) [8] have been used to construct dialog managers to account for the uncertainty in language understanding [30]. We develop a dual-track POMDP-based dialog manager to help the agent maintain a confidence level of its current knowledge being able to support the dialog, and accordingly decide to whether augment its KB or not. After the agent becomes confident that new entities are necessary so as to make progress in the dialog, it decides where in the KB to add a new entity (e.g., as a new *item* or a new *person*) by analyzing the flow of the dialog. As a result, our dialog agent is able to decide when it is necessary to augment its KB and how to do so in a semantically meaningful way.

Our dialog system has been implemented both in simulation and on a Segway-based mobile robot that is able to interact with people using natural language and conduct guidance and delivery tasks. Experimental results show that our dialog system performs better in service request identification in both efficiency and accuracy, in comparison to baselines that use a static KB or update the KB using a predefined strategy. Experiments with human participants suggest that our knowledge augmentation component improves user experiences as well.

II. Related Work

This work is related to existing research on robots interpreting human language. We discuss applications where language inputs are used for providing service requests, introducing new concepts, and instructing a robot to conduct specific tasks.

Researchers have developed algorithms for learning to interpret natural language commands. Examples include the work of Matuszek et al. that has enabled a robot to learn to understand navigation commands [13]. Other examples include the work of "Tell me Dave" [15], and the work of Tellex et al. that focused on a graph-based language understanding approach [25]. Recent research enabled the co-learning of syntax and semantics of spatial language [23, 6]. Although the systems support the learning of language skills, they do not have a dialog management component (implicitly assuming perfect language understanding), making the methods not suitable for applications that require multi-turn communications.

Another research area that is related to this work focuses on learning dialog strategies. For instance, the NJFun system [22] modeled the dialog management problem using Markov Decision Processes (MDPs) [19], and learned the dialog strategy using Reinforcement Learning (RL) algorithms [24]. Recent research on Deep RL (DRL) has enabled an agent to interactively learn action policies in more complex domains [16], and DRL methods have been used to learn dialog strategies [29, 4]. The systems do not include a semantic parsing component. As a result, users can only communicate with their dialog agents using simple or predefined language patterns.

Mobile robot platforms have been equipped with semantic parsing and dialog management capabilities. After a task is identified in dialog, these robots are able to conduct service tasks using a task planner. One example is the dialog agent developed for the KeJia robot [12]. Integrated commonsense reasoning and probabilistic planning (CORPP) has been applied to dialog systems, resulting in a commonsense-aware robot dialog system [31]. Recent research further integrated multi-modal perception capabilities, such as locations and facial expressions, into dialog systems [11]. Although these systems enable a robot to identify human requests via dialog, they do not learn from such experiences.

Thomason et al. developed a dialog agent for a mobile service robot that is able to conduct service tasks such as human guidance and object delivery [26]. A dialog manager suggests language actions for asking clarification questions, and the agent is able to learn from human-robot conversations. Recent research further enabled the agent to learn to simultaneously improve the semantic parsing and dialog management capabilities [17]. These methods focus on learning to improve an agent's language capabilities, and do not augment its knowledge base in this process. This work builds on the dialog agent implemented by Thomason et al., and introduces a dual-track POMDP-based dialog manager and a strategy for augmenting the robot's knowledge base.

There are other dialog agents that specifically aim at knowledge augmentation through human-robot dialog. An example is the CMU CoBots that are able to learn procedural knowledge from verbal instructions via human-robot dialog [14], and learn task knowledge by querying the web [18]. Recently, the instructable agent developed by Azaria et al. is able to learn new concepts and new procedural knowledge through human-robot dialog [2]. Recent work enabled a mobile robot to ground new concepts using visual-linguistic observations, e.g., to ground new word "box" given a command of "move to the box" by exploring the environment and hypothesizing objects [27]. These agents are able to augment their knowledge bases through language-based interactions with humans. However, their dialog management components (if existing) are relatively weak and do not model the noise in natural language understanding.

She and Chai developed a robot dialog system that focuses on situated verb semantics learning [21]. Their dialog agent uses RL to learn a dialog management policy, and uses a semantic parser to process natural language inputs. However, their work specifically focused on learning the semantics of verbs, limiting the applicability of their knowledge augmentation approach.

The dialog agent developed in this work processes language inputs using a semantic parser to understand users' service requests, leverages a POMDP-based dialog manager to handle the uncertainty from the unreliable parser by asking clarification questions, and augments its knowledge base when the current knowledge is not sufficient to interpret the human request.

III. Dialog Agent

In this section, we present our dialog agent that integrates a decision-theoretic approach for dialog management under uncertainty, and an information-theoretic approach for knowledge management. Figure 2 shows an overview of our dialog system. Next, we present our dual-track controller for knowledge and dialog management, the language understanding component, and the representation of our knowledge base.

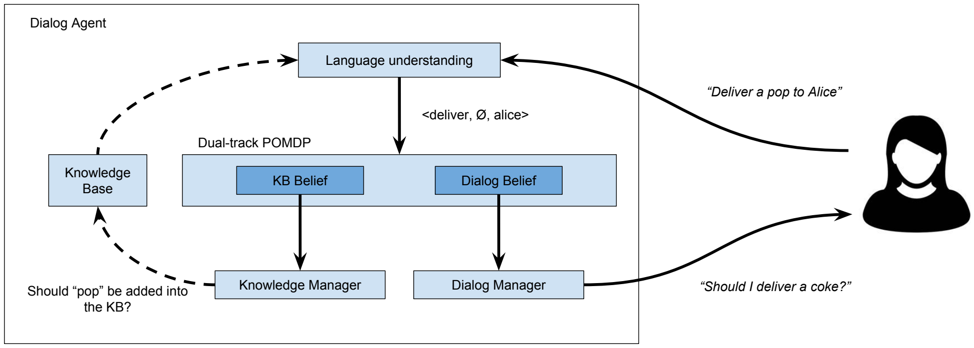

Fig. 2. A pictorial overview of our dialog system, including a semantic parser for language understanding, a knowledge base represented by Answer set programming (ASP) [5], and a dual-track POMDP for the management of knowledge base (KB) and dialog.

A. Dialog and Knowledge Management

Markov decision process (MDP) is a general sequential decision-making framework that can be used for planning under uncertainty in action outcomes [19]. Partially observable MDP (POMDP) [8] generalizes the MDP framework by assuming the world being partially observable. POMDPs have been used for dialog management [30].

There are two POMDP-based control loops in our dialog agent, resulting in a *dual-track POMDP controller*. One track focuses on maintaining the dialog belief state, and accordingly suggesting language actions to the agent. The other track focuses on maintaining the belief of the current knowledge being (in)sufficient to complete the task, and accordingly suggesting knowledge augmentation.

1) Dialog Management Track: The dialog management POMDP includes the following components:

- $S : (S^T \times S^I \times S^R) \cup term$, where S^T is the set of task *types* (delivery and guidance in our case), S^I is the set of *items* used in the task, S^R is the set of recipients of the delivery, and *term* is the terminal state.
- $A : A^W \cup A^C \cup A^R$ is the action set. A^W consists of general "wh" questions, such as "*Whom is this delivery for?*" and "*What item should I deliver?*" A^C includes confirming questions that expect yes/no answers, and reporting actions A^R return the estimated human requests.
- $T : S \times A \times S' \rightarrow [0, 1]$ is the state-transition function. In our case, the dialog remains in the same state after question-asking actions, and reporting actions lead transitions to *term* deterministically.
- $R : S \times A \rightarrow \mathbb{R}$ is the reward function and the reward values are assigned as:

$$R(s,a) = \begin{cases} r^C, & \text{if } s \in S, a \in A^C \\ r^W, & \text{if } s \in S, a \in A^W \\ r^+, & \text{if } s \in S, a \in A^R, s \odot a \\ r^-, & \text{if } s \in S, a \in A^R, s \otimes a \end{cases}$$

  where r^C and r^W are the costs of confirming and general questions, in the form of negative, relatively small values; r^+ is a big bonus for correct reports; and r^- is a big penalty (negative) for wrong reports.
- $Z : Z^T \cup Z^I \cup Z^R \cup \{z^+, z^-\}$ is the set of observations, where Z^T, Z^I and Z^R include observations of *task type*, *item*, and *recipient* respectively. z^+ and z^- correspond to "*yes*" and "*no*". Our dialog agent takes in observations as semantic parses that have correspondence to elements in Z. Other parses, including the malformed ones, produce random observations (Section III-B).
- $O : S \times A \times Z \cup$ *inapplicable* is the observation function that specifies the probability of observing $z \in Z$ in state $s \in S$, after taking action $a \in A$. Reporting actions yield the *inapplicable* observations. Our observation function models the noise in language understanding, e.g., the probability of correcting recognizing z^+ ("*yes*") is 0.8. The noise model is heuristically designed in this work, though it can be learned from real conversations.

It should be noted that this POMDP model is dynamically revised (similar to [32]), when new entities (Section III-C) are added into the knowledge base. Solving this POMDP generates a policy π, which maps a belief to an action that maximizes long-term information gain.

2) Knowledge Management Track: In addition to the dialog management POMDP, we have a knowledge management POMDP that monitors whether the agent's knowledge is sufficient to support estimating human intentions. The knowledge management POMDP formulation is similar to that for dialog management but includes entities for unknown items and recipients.

The components of the knowledge management POMDP are shown below, where transition and observation functions are generated accordingly and hence not listed:

- $S^+ : S \cup \{(s^T, s^I, \hat{s}^R) \mid \forall s^T \in S^T, \forall s^I \in S^I\} \cup \{(s^T, \hat{s}^I, s^R) \mid \forall s^T \in S^T, \forall s^R \in S^R\}$ is the set of states. It includes all states in S along with the states corresponding to new entries that correspond to an unknown item $\hat{s}^I$ and an unknown recipient $\hat{s}^R$.
- $A^+ : A \cup \{\hat{a}^I, \hat{a}^R\} \cup \hat{A}^R$ is the set of actions including the actions in A, two actions for confirming the unknown

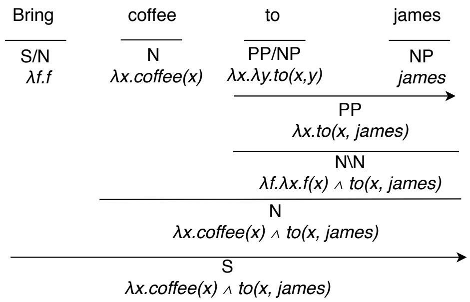

Fig. 3. An example of parsing a service request sentence using CCG semantic parsing and λ calculus.

item and *recipient* respectively, and $\hat{A}^R$, reporting actions that correspond to the states in S^+.

- $Z^+ = Z \cup \{\hat{z}^I, \hat{z}^R\}$ is the augmented observation set.

At runtime, we maintain belief distributions for both tracks of POMDPs. Belief b of dialog POMDP is used for sequential decision making and dialog management, and belief b^+ of knowledge POMDP is only used for language augmentation purposes, i.e., determining when it is necessary to augment the KB (Section III-D). When observations are made (observation $z \in Z$), both beliefs are updated using the Bayes update rule:

$$b'(s') = \frac{O(s', a, z) \sum_{s \in S} T(s, a, s') b(s)}{pr(z|a, b)}$$

where s is the state, a is the action, $pr(z|a, b)$ is the normalizing factor, and z is the observation. In our dialog system, observations are made based on the language understanding using a semantic parser.

Remarks: Fundamentally, one can merge multiple POMDPs to unify the action selection process. We use a two-track controller in this work, because the knowledge track is not used for action selection. We believe separating the knowledge and dialog tracks reduces the complexity of the entire framework, and we leave formal analysis to future work.

B. Semantic Parsing for Language Understanding

In order to understand natural language and make observations for POMDPs, we use a semantic parser that builds on the Semantic Parsing Framework (SPF) described in [1]. The input of the semantic parser is natural language from human users, and the output is a list of possible parses for a given sentence. Using the semantic parser, the request in natural language is transformed to a formal representation that computers understand.

Figure 3 shows an example of the parser recognizing a sentence. It can reason over the ontology of the known words when it parses a sentence, e.g., *james:pe* and *coffee:it*. The dialog manager can use this information to translate from words to corresponding observation for the question asked by the robot. If the language understanding fails (e.g., producing parses that are malformed or do not comply with the preceding questions), then a random observation from Z will be made for the unknown part of the request.

C. Domain Knowledge Representation

We use Answer Set Prolog (ASP) [5], a declarative language, for knowledge representation. The agent's knowledge base (KB), in the form of an ASP program, includes rules in the form of:

$$l_0 \leftarrow l_1, \ \cdots, \ l_n, \ \text{not } l_k, \ \cdots, \ \text{not } l_{n+k}$$

where l's are literals that represent whether a statement is true. The right side of a rule is the *body*, and the left side is the *head*. A rule without a body is called a fact.

The KB of our agent includes a set of *objects* in ASP: {*alice, sandwich, kitchen, office1, delivery,* $\cdots$}, where *delivery* specifies the task type. A set of *predicates*, such as {*recipient, item, task, room*}, are used for specifying a category for each object.

Using the above predicates and objects, we can use rules to fully specify tasks, such as "*deliver a coke to Alice*":

$$task(delivery).$$
$$item(coke).$$
$$recipient(alice).$$

One can easily include more information into the ASP-based KB, such as rooms, positions of people, and a categorical tree of objects. This ASP-based KB is mainly used for responding to information queries, and task planning purposes, where the query and/or task are specified by our dialog agent.

D. Our Algorithm for Simultaneous Intention Estimation and Knowledge Augmentation

In this subsection, we first define a few terms and functions, and then introduce the main algorithm for simultaneously estimating human intention (service requests, in our case) and augmenting the agent's knowledge base. We use *entropy* to measure the uncertainty level of the agent's belief distribution:

$$H(b) = -\sum_{i=0}^{n-1} b(s_i) \cdot \log b(s_i)$$

where H is the entropy function.

When the agent is (un)confident about the state, the entropy value is (high) low. In particular, a uniform belief distribution corresponds to the highest entropy level. We use entropy for two purposes in our algorithm.

Rewording Service Request: If the belief entropy is higher than threshold h (meaning the agent is highly uncertain about the dialog state), we encourage the human user to state the entire request in one sentence. Otherwise, the dialog manager decides the flow of the conversation.

Entropy Fluctuation: We use the concept of entropy to define *entropy fluctuation* (EF):

$$f(\mathbf{b}) = sign\Big(H(\mathbf{b}[1]) - H(\mathbf{b}[0])\Big) \oplus sign\Big(H(\mathbf{b}[2]) - H(\mathbf{b}[1])\Big)$$

where $\mathbf{b}$ is a belief queue that includes the three most recent dialog beliefs, $f(\mathbf{b})$ outputs *true*, if there is an EF in the last

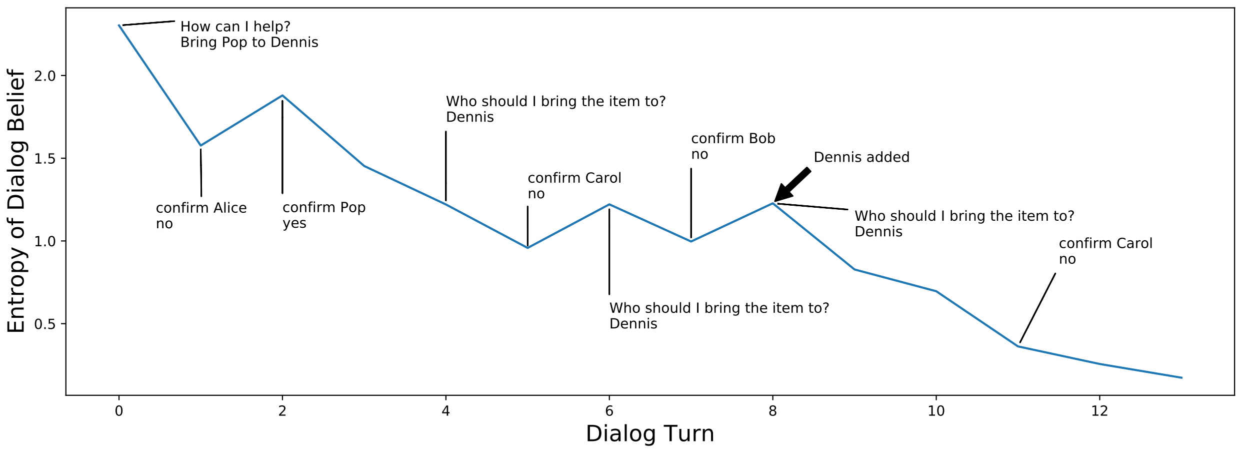

Fig. 4. In this example the user requests a *Pop* to an novel recipient, *Dennis*, who is not yet included in the system's ontology. The dialog system makes observation for *Pop* but does not understand *Dennis*, so it selects another person at random as an observation (in this case, *Alice*). The users denies this incorrect person choice, then confirms *pop*. The conversation continues with the system trying to figure out who the delivery is for, but it cannot yet reason about *Dennis*, who is absent from the ontology. When the number of EFs crosses a specified threshold (5 in this case), *Dennis* is added as a new ontological object. After that, the system asks some final confirmation questions to clear any confusion for example there may be some belief in *Carol* which is cleared once the user replies no. The entropy continues to decrease until request is identified successfully.

Algorithm 1 Simultaneous Intention Estimation and Knowledge Augmentation

Require: $\tau_b, h, \Delta, \mathcal{M}, \mathcal{M}^+$, and a POMDP *solver*

Initialize b, b^+ with uniform distributions
Initialize EF counter $\delta \leftarrow 0$
Initialize queue **b** of size 3 with $\{b, b, b\}$
repeat
  if $Pr(s^R = \hat{s}^R) > \tau_b$, where $s^R \in b^+$ **then**
    Add a new recipient entity in KB
  else if $Pr(s^I = \hat{s}^I) > \tau_b$, where $s^I \in b^+$ **then**
    Add a new item entity in KB
  else if $\delta > \Delta$ **then**
    Add (item or recipient) entity that is more likely
  if $f(\mathbf{b}, t)$ **is** *true* **then**
    $\delta \leftarrow \delta + 1$
  if $H(b) > h$ **then**
    $a \leftarrow$ "*Please reword your service request*"
  else
    $a \leftarrow \pi(b)$
  $o \leftarrow parse(human\ response)$
  Update b based on observation o and action a
  b.enqueue(b)
  if $a \in A_c$ **then**
    $b^+ \leftarrow update(b^+)$
until s **is** *term*
return the request based on the last (reporting) action, and the (possibly updated) knowledge base.

three beliefs (i.e., entropy of the second last is the highest or lowest among the three), and $\oplus$ is the *xor* operator. Figure 4 shows an example of EFs in a dialog.

Algorithm 1 shows the main operation loop of the dialog system. τ_b is a probability threshold; h is an entropy threshold; Δ is a threshold over the number of EFs; and $\mathcal{M}$ and $\mathcal{M}^+$ are POMDPs for dialog and knowledge management respectively.

The algorithm starts by initializing the two belief distributions uniformly. δ, which counts the number of EFs, is initialized to 0. If the marginal probability over $\hat{s}^R$ (or $\hat{s}^I$) of knowledge belief b^+ is higher than threshold τ_b, or the number of EFs is higher than Δ, we add a new entity into the KB. If the entropy of dialog belief is higher than h, then the agent asks for rewording the original service request. Otherwise, we use the dialog POMDP to maintain the dialog flow. Finally, the knowledge belief is only updated by confirming questions, which are able to invalidate agent hypothesis of unknown entities. The algorithm returns the request and the updated knowledge base.

When add a new entity, the agent explicitly asks the users to help specify the name of the unknown item (or person). The KB is updated along with the lexicon for the semantic parser. The index for the unknown item or person is associated with the newly added entry. With the new knowledge added to KB, model has to be adjusted in size so that the dialog can be continued properly, and the belief b is replaced with b^+ which provides a reinitialization to continue the conversation.

E. An Illustrative Example

Table I is an illustrative example of one of the trials in the human participant experiment. The semantic parser failed to correctly parse the request, because it does not know the meaning of *get*. As a result, the dialog agent made a random guess that it is "*Alice requesting hamburger*". The robot asks again for clarification what it should deliver. The user responds "*coffee*", making the robot confused between *coffee* and *hamburger*. The robot asked a confirmation question about "*hamburger*", and received a negative response. Although the

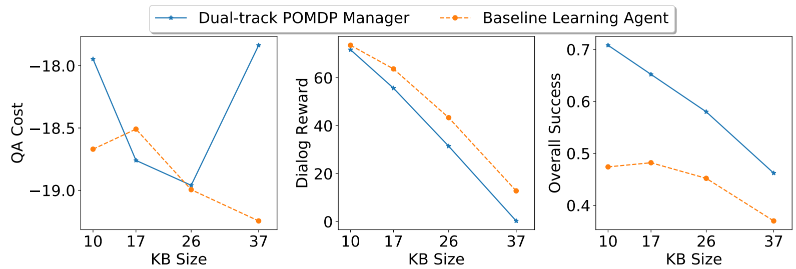

Fig. 5. Comparison between our dual-track POMDP-based agent, and a baseline learning agent that augments its knowledge base (KB) if the dialog does not terminate in $N = 5$ turns. The x-axis is the "KB size", measured by the number of states in dialog state space. Results of "QA Cost" and "Dialog Reward" are only based on dialog management, whereas we consider both dialog and knowledge management in "Overall Success".

TABLE I
AN EXAMPLE DIALOG FROM A HUMAN PARTICIPANT.

Robot	**Human**
How can I help you?	Get me coffee
What item should I bring?	Coffee
Do you want me to deliver hamburger?	No
Who should I bring the item to?	Nate
Is this delivery for ellen?	No
It seems I do not know the person you are talking about. Please write their name so I can learn it.	Nate
Who should I bring the item to?	Nate
Is this delivery for Nate?	Yes
What item should I bring?	Coffee
Execute: Robot brings coffee for Nate and the dialog is over.	

user was explicitly guided to select the recipient among the people, this participant used the pronoun "me", an unknown word to the robot. When the robot heard "*Nate*" the first time, it could not understand, and made a random guess that it is "*Ellen*". As a result, the agent confirmed "*Ellen*" afterwards, and was invalidated by the user. So it kept asking for clarifications until it adds the person's name to its knowledge and became confident about the request.

IV. EXPERIMENTS

For the experiments, we used a Segway-based robot platform, pictured in Figure 1, which is equipped with a screen that displays questions and responses to the user. For the purposes of our experiments, the participants used a wireless keyboard and the mounted screen to communicate with the robot. The dialog agent was implemented using Robot Operating System (ROS) [20]. In simulation experiments, we model the uncertainty in natural language understanding by adding noise to the observations. For instance, the agent can correctly recognize "coffee" in 0.8 probability, and this probability decreases given more items in the KB. When the user verbalizes the entire request, the agent receives a sequence of three (unreliable) observations on *task*, *item* and *recipient* in a row. The costs of confirming questions is $R^C = -1$, and the cost of wh-questions is $R^W = -1.5$. In each trial, a request is randomly selected with 50% chance of including an unknown item or person. POMDPs are solved using an off-the-shelf system [10].

Experiments were designed to evaluate the following hypotheses: 1) In comparison to a baseline that augments the KB after the number of turns becomes higher than a threshold, our algorithm reduces the QA cost while maintaining similar success rates. 2) A dialog agent that learns from human-robot conversations creates a better user experience, in comparison to the ones that use static KBs.

Evaluation metrics used in the experiments consist of: **QA Cost**, the total cost of QA actions; **Overall Success**, where a trial is successful, if the service request is correctly identified *and* (if needed) the KB is corrected augmented; and **Dialog Reward**, where both QA cost and bonus/penalty are considered. Focusing on the knowledge augmentation accuracy, we also use **F1 score** as a harmonic average of the precision and recall in knowledge management.

A. Experiments in Simulation

Figure 5 shows the results of comparing our dual-track POMDP-based agent with a baseline that augments its KB, if human intention cannot be identified in $N = 5$ turns (Hypothesis-1). As the domain size increases, our dialog agent performs consistently better than the baseline in Overall Success. From Figure 5-Left, we see our agent asks more questions in larger domains, until it finds the domain (with KB Size 37) is too large and it is not worth more questions. From Figure 5-Middle, we see both methods produce lower dialog reward given larger domains, because it is harder to get confident that new knowledge is needed to achieve the dialog goal in larger domains.

We further evaluated the effect of the EF threshold Δ to the system performance. Intuitively, a higher EF threshold makes it more difficult to introduce new entities to the knowledge base, resulting in more conservative behaviors. Figure 6 shows the results, where we can see a small value of Δ encourages the agent to add new knowledge, even if it is not very confident about the necessity of doing that. Results of such behaviors are reflected in the ""F1 score" subfigure" on the right. Given a higher Δ threshold, the agent is more conservative in knowledge augmentation, and has higher F1 scores. The downside of a higher Δ is that the agent failed in more cases, where knowledge augmentation is needed but did not occur.

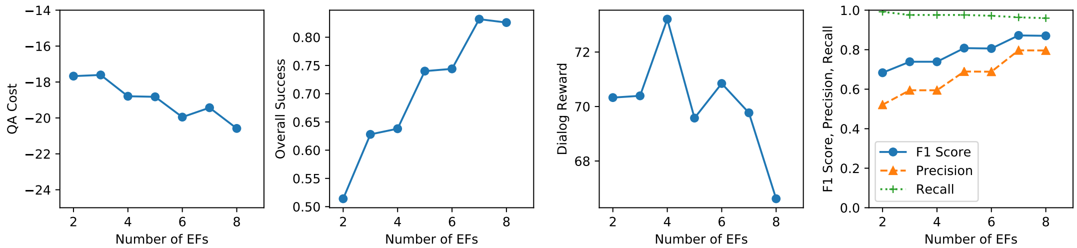

Fig. 6. Simulation experiment on a fixed size domain, with a varying EF threshold. As the value of EF threshold increases, the agent becomes more conservative in adding new knowledge. It increases the length of the conversation, while producing higher F1 scores at the same time.

Fig. 7. *Left*: Segway RMP110 used in the experiment. *Right*: List of the items and recipients that robot participants used to ask for delivery. Two items and two people were unknown to the robot.

TABLE II
RESULTS OF THE HUMAN PARTICIPANT EXPERIMENT.

	Q1	Q2	Q3	Q4
Average score using dual-track POMDP	3.42	2.50	**1.50**	**2.50**
Average score of baseline	3.33	1.83	2.17	1.75

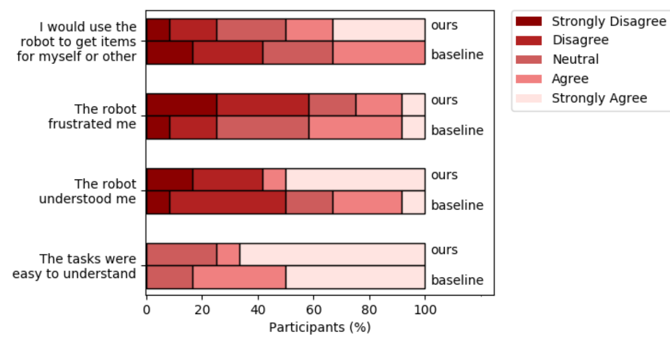

Fig. 8. Results of survey from participants. The survey consisted of four statements with Likert-scale responses.

B. Experiments with Human Participants

Twelve students of ages 19-30 volunteered to participate in an experiment where they could ask the robot to conduct delivery tasks. Two of each of the the item and recipient lists were unknown to the robot, resulting in only about 49% (i.e., $1-\frac{5}{7}\times\frac{5}{7}$) of the service requests that can be identified without requiring knowledge augmentation. The participants were not aware of this setting, and arbitrarily chose any combination of an item and a recipient to form a delivery task. Each participant conducted the experiment in two trials. In one, the robot used the dual-track POMDPs, and in the other, the robot used a baseline dialog agent with a static KB. The delivery items and recipients were shown in Figure 7.

At the end of the experiment, participants were required to fill out a survey form indicating their qualitative opinion including the following items. The response choices were 0 (Strongly disagree), 1 (Somewhat disagree), 2 (Neutral), 3 (Somewhat agree), and 4 (Strongly agree).

1) How easy the tasks were defined;
2) How well the robot understood the human request;
3) Whether robot frustrated the participant or not; and
4) Participant's willingness to use the robot in the future.

Table II shows the average scores. We see that, with p-value threshold 0.1, our dual-track POMDP approach makes significant improvements in response for Q3 ("robot frustrated me") and Q4 ("Will use the robot"). There is no significance difference observed in responses to the other two questions. Figure 8 presents more details from the survey papers.

V. CONCLUSIONS AND FUTURE WORK

We introduced a dialog agent that simultaneously supports human intention identification, and knowledge augmentation on an as-needed basis. Experiments show that our dual-track POMDP controller enables the agent to do dialog and knowledge management. In comparison to a baseline that augments its knowledge base after a fixed number of turns, our dialog agent consistently produces higher dialog overall success. Experiments with human participants show that humans felt less frustrated and more willing to use our agent. This dialog agent can be particularly useful to robot platforms that work in open domains, where pre-programing a knowledge base is impractical. This includes places like offices, factories, and hospitals where the space of items and people may change as a function of time.

In the future, we plan to enable the agent to augment knowledge bases with more complex structures, e.g., to model subclasses of *item*. Such structures make knowledge management more difficult. Another direction is to enable the agent to learn and merge synonyms via human-robot dialog, and remove incorrect (or unnecessary) concepts on an as-needed basis. In this process, the agent may want to actively introduce "redundant" dialog turns for knowledge acquisition and clarification purposes. Finally, we plan to experiment with more human participants and in environments with more dynamics.

References

[1] Yoav Artzi. Cornell SPF: Cornell semantic parsing framework. *arXiv preprint arXiv:1311.3011*, 2013.

[2] Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. Instructable intelligent personal agent. In *AAAI*, pages 2681–2689, 2016.

[3] Yingfeng Chen, Feng Wu, Wei Shuai, Ningyang Wang, Rongya Chen, and Xiaoping Chen. Kejia robot–an attractive shopping mall guider. In *International Conference on Social Robotics*, pages 145–154. Springer, 2015.

[4] Heriberto Cuayáhuitl. Simpleds: A simple deep reinforcement learning dialogue system. In *Dialogues with Social Robots*, pages 109–118. Springer, 2017.

[5] Michael Gelfond and Yulia Kahl. *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press, 2014.

[6] Ze Gong and Yu Zhang. Temporal spatial inverse semantics for robots communicating with humans. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[7] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrová, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.

[8] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[9] Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, et al. BWIbots: A platform for bridging the gap between AI and human–robot interaction research. *The International Journal of Robotics Research*, 36(5-7):635–659, 2017.

[10] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, volume 2008. Zurich, Switzerland., 2008.

[11] Dongcai Lu, Shiqi Zhang, Peter Stone, and Xiaoping Chen. Leveraging commonsense reasoning and multimodal perception for robot spoken dialog systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6582–6588. IEEE, 2017.

[12] Dongcai Lu, Yi Zhou, Feng Wu, Zhao Zhang, and Xiaoping Chen. Integrating answer set programming with semantic dictionaries for robot task planning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4361–4367. AAAI Press, 2017.

[13] 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, 2013.

[14] Çetin Meriçli, Steven D. Klee, Jack Paparian, and Manuela Veloso. An interactive approach for situated task specification through verbal instructions. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1069–1076. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[15] Dipendra K Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research*, 35(1-3):281–300, 2016.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[17] Aishwarya Padmakumar, Jesse Thomason, and Raymond J. Mooney. Integrated learning of dialog strategies and semantic parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 547–557, 2017.

[18] Vittorio Perera, Robin Soetens, Thomas Kollar, Mehdi Samadi, Yichao Sun, Daniele Nardi, René van de Molengraft, and Manuela Veloso. Learning task knowledge from dialog and web access. *Robotics*, 4(2):223–252, 2015.

[19] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[20] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[21] Lanbo She and Joyce Chai. Interactive learning of grounded verb semantics towards human-robot communication. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1634–1644, 2017.

[22] Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133, 2002.

[23] Michael Spranger and Luc Steels. Co-acquisition of syn-

tax and semantics: an investigation in spatial language. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1909–1915. AAAI Press, 2015.

[24] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[25] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, volume 1, page 2, 2011.

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

[27] Mycal Tucker, Derya Aksaray, Rohan Paul, Gregory J Stein, and Nicholas Roy. Learning unknown groundings for natural language interaction with mobile robots. In *International Symposium on Robotics Research (ISRR)*, 2017.

[28] Manuela Veloso, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal. Cobots: robust symbiotic autonomous mobile service robots. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 4423–4429. AAAI Press, 2015.

[29] Jason D Williams and Geoffrey Zweig. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*, 2016.

[30] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

[31] Shiqi Zhang and Peter Stone. CORPP: commonsense reasoning and probabilistic planning, as applied to dialog with a mobile robot. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1394–1400. AAAI Press, 2015.

[32] Shiqi Zhang, Piyush Khandelwal, and Peter Stone. Dynamically constructed (PO)MDPs for adaptive robot planning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3855–3862, 2017.