# Reducing Errors in Object-Fetching Interactions through Social Feedback

David Whitney[1], Eric Rosen[1], James MacGlashan[2], Lawson L.S. Wong[1], Stefanie Tellex[1]

***Abstract*— Fetching items is an important problem for a social robot. It requires a robot to interpret a person's language and gesture and use these noisy observations to infer what item to deliver. If the robot could ask questions, it would help the robot be faster and more accurate in its task. Existing approaches either do not ask questions, or rely on fixed question-asking policies. To address this problem, we propose a model that makes assumptions about cooperation between agents to perform richer signal extraction from observations. This work defines a mathematical framework for an item-fetching domain that allows a robot to increase the speed and accuracy of its ability to interpret a person's requests by reasoning about its own uncertainty as well as processing implicit information (implicatures). We formalize the item-delivery domain as a Partially Observable Markov Decision Process (POMDP), and approximately solve this POMDP in real time. Our model improves speed and accuracy of fetching tasks by asking relevant clarifying questions only when necessary. To measure our model's improvements, we conducted a real world user study with 16 participants. Our method achieved greater accuracy and a faster interaction time compared to state-of-the-art baselines. Our model is 2.17 seconds faster (25% faster) than a state-of-the-art baseline, while being 2.1% more accurate.**

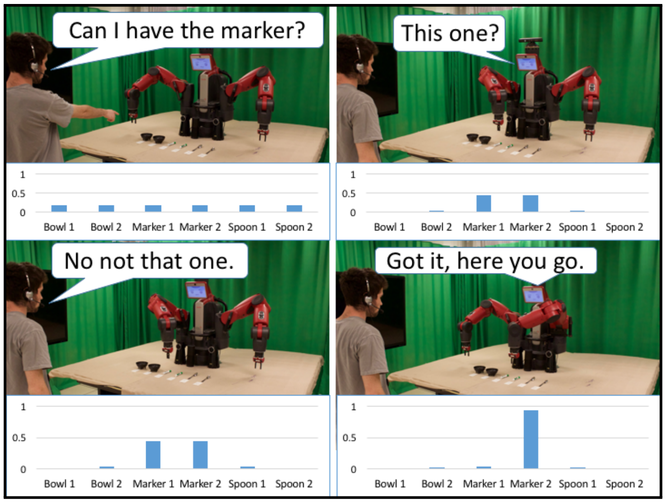

Fig. 1. Demonstration of our FETCH-POMDP model correctly fetching item for user. Note the robot's understanding of implicit information between panels three and four. This reasoning is not hard-coded into our system, but emerges from the solution of our POMDP.

## I. INTRODUCTION

Object retrieval tasks are common in life, and are representative of tasks expected of a social robot. Humans use both speech and pointing gestures to refer to specific objects. A mechanic repairing a car, for instance, may point and ask the robot to fetch a specific tool from the shelf. There will be times, however, where the robot will fail to understand, either due to errors in interpreting the person or from an genuinely ambiguous command. It would be beneficial if the robot could communicate its lack of understanding back to the human, asking questions only when needed.

The difficulty in this this task lies in the noise of natural language and gesture. Speech-to-text software often introduces transcription errors, and human body trackers perform far worse than human level. These problems lead to ambiguities for the robot. When the robot is uncertain, we want it to ask questions, but when confident, we want it to hand the item without bothering the user. Therefore we want to intelligently choose between information gathering actions and reward gathering actions. A POMDP is a natural framework to use to make these choices.

Existing approaches for object fetching have used batch-mode language understanding to map human language commands to robot action sequences [18]. These systems, however, do not allow for the robot to ask questions, and could not clarify ambiguity. In non-robotic domains, others have considered systems that explicitly modeled the beliefs of other agents, laying the groundwork for question asking [8]. Williams and Young [24] created a Speech Dialog System that allows agents to model the beliefs of others in order to ask questions of the user based on phone-based communication, which is very noisy. Because phone lines are very noisy, that system had a fixed question asking routine it followed after choosing the question subject. In human robot collaboration, the robot and human have multiple methods of communication.

To achieve a framework for the item-fetching domain that intelligently asks questions as well as extracts implicit information, we define the FEedback To Collaborative Hand-off Partially Observable Markov Decision Process (FETCH-POMDP). Our system determines a human's desired item by interpreting natural language input as well as pointing gestures, and can ask clarifying questions when confused. Our model is able to understand implicit meaning in the humans actions, known as implicatures.

Implicatures are the inferences a listener makes when bridging together a speaker's utterances and assumption that the speaker is acting cooperatively. For example, in Figure 1, the implicature is that the speaker wanted the other marker because there is only one marker left, the speaker said

[1] Department of Computer Science, Brown University, Providence, RI 02912, USA david_whitney@brown.edu, eric_rosen@brown.edu, lsw@brown.edu, stefie10@cs.brown.edu
[2] Cogitai Inc. james@cogitai.com

they wanted a marker, and the assumption exists that the speaker is not being deceitful about what they desire. This assumption of cooperation is what allows the robot to gather more information from the speakers utterance, making the interaction quicker and more efficient.

We evaluate the speed and accuracy of our FETCH-POMDP through a real-world user study, comparing it to two state-of-the-art baselines. We had 16 users request items from our robot with either an ambiguous or unambiguous item configuration. We found that our FETCH-POMDP the most accurate method in an ambiguous environment, and the fastest in an unambiguous environment.

## II. Related Work

Early works in robot-question asking realized the potential of question asking to increase acccuracy but were limited by their rule based approaches [7].

Common methods of natural language processing treat speech as a serialized process and infer utterances through batch-mode approaches [12, 19, 14]. These methods typically do not take into account situational context or other agents' beliefs in order to correct failures. Our work looks to create an system that makes robotic inference of human desires an interactive process. An interactive decision process allows for certain language utterances to mean different things to the robot depending on the current situation, making richer communication channels between the agents.

In the learning from demonstration domain, researchers such as Cakmak and Thomaz [3] have investigated what questions are useful for learning new skills. Our work differs in that we are concerned with completing a known task, and focus on when to ask questions as opposed to what type of question to ask.

Vogel et al. [21] researched how implicatures allow agents to communicate more information than what is in the utterance, allowing quicker and smoother interactions. Implicatures arise in Decentralized POMDP's (Dec-POMDP) when agents model the state of other agents in order to maximize joint utility [21]. Due to the fact that in the FETCH-POMDP, the agent keeps a model of the desired object the human has in mind, implicatures naturally arise in the interactions.

POMDPs are used in many approaches for solving decision problems where the environment is noisy and not perfectly observable. For example, Hoey et al. [9] created a decision making system from a POMDP for a robot helping dementia patients wash their hands, where the agent must infer the human's actions and psychological state through noisy hand and towel tracking. Since agents keep track of states and personal histories internally, POMDPs have been a natural choice for modeling multi-agent settings [20, 23]. Gmytrasiewicz and Doshi [8] used a POMDP to handle a multi-agent setting more interactively than typical approaches. By augmenting the state space to include a limited construct of other agent's beliefs, each agent is able to reason over the states and actions of the other agents while solving for the optimal policy. Gmytrasiewicz and Doshi [8] prove how modeling the interaction as an Interactive-POMDP (I-POMDP) allow agents to independently compute optimal policies. However, Gmytrasiewicz and Doshi [8] state that an I-POMDP's belief-depth modeling of agents has to be limited because it is impossible to solve the infinite-recursive chain of beliefs. Our approach, in contrast, makes the simplifying assumption that only the last item referenced matters, rather than a fully inference of the other agent's belief. This assumption enables us to perform inference in real time.

Williams and Young [24] casted a spoken dialogue system as a POMDP. Williams and Young [24] show how the formalization as a POMDP gives a strong statistical model for determining an optimal policy between two speaking agents. Rather that needing to decide on a fixed heuristic for inferring observations such as confidence scoring, automated planning of long-run interactions, or parallel state hypotheses of the world, modeling the system as a POMDP allows a statistical approach to frame the optimal decisions. Williams and Young [24] discuss the potential for their framework to encompass more sophisticated interactions, yet they limit the scope of their trials to speech-related communication tasks. Our work differs from Williams and Young [24] by implementing a POMDP model onto a robot agent to perform the item-delivery task with a human using both speech and gesture. Furthermore, Williams and Young [24] always ask questions, regardless of context, and use the POMDP to have a policy on which questions to ask, while our work allows the robot to decide whether to ask questions at all.

Chai et al. [4] created a probabilistic model for human-robot interaction that allows a human to inform a robot of objects in the environment using natural language, and the robot to ask for clarification using both speech and gesture. The question asking policy, mapping state to action, is fixed. Our work differs in that our FETCH-POMDP generates its own policy based on its observations.

Wu et al. [25] addressed the item-fetching domain by formalizing a POMDP that allowed a robot agent to model the user's beliefs to calculate a policy based on multiple noisy communication modalities. However, Wu et al. [25]'s state space was very large, preventing quick inference and real-time calculation of policy.

Doshi and Roy [6] implemented a POMDP model to understand natural language in order to infer noisy communication and ambiguous word choice. By modeling the dialog manager as a POMDP, Doshi and Roy [6] balances between question-asking for ambiguity clarification with action-taking to fulfill the human's request. However, Doshi and Roy [6] state factorization does not include a representation of the human's belief's, which prevents their model from inferring implicatures. Our work differs by implementing a way to naturally infer implicit information from observations, as well as infer an extra modality of pointing.

## III. Technical Approach

We define a novel model, the FEedback-To-Collaborative-Handoff Partially Observable Markov Decision Process

(FETCH-POMDP) to solve our object fetching problem by intelligently selecting when to provide feedback based on its belief state.

Our problem is an item delivery problem. Imagine a person carrying out a task, such as assembling a piece of furniture or cooking a meal. To complete the task, they need something, such as a screwdriver or a whisk. They use language and gesture to instruct the robot what item they need. The robot observes their language $l$ and gesture $g$ and must select the correct item $i$ as quickly and accurately as possible.

Because of noise in speech and gesture observations, the robot will not be able to infer $i$ from the initial speech and gesture of the human. We therefore want the robot to ask questions when, and only when, it is confused, so as to be accurate while not bothering the human unnecessarily. We need to balance between information gathering actions, like asking questions, and goal inducing actions, like fetching. Therefore, we model this problem as a POMDP.

### A. *POMDP Overview*

A Markov Decision Process (MDP) [1] is a decision problem formalism in which an agent observes the state of the environment and takes actions in discrete time steps. It is defined by the tuple $\langle S, A, R, T \rangle$, where $S$ is the set of environment states, $A$ is the set of actions, $R(s,a)$ is a reward function that specifies how much instantaneous reward is received for taking action $a$ in state $s$, and $T(s,a,s')$ is the transition function that defines the probability of the environment transitioning to state $s'$ after the agent takes action $a$ in state $s$. The goal of the agent is to find an action in any given state (a policy) that maximizes the expected future reward. In an MDP, it is assumed the agent knows the true state at each timestep. For many problems, this assumption is invalid. Partially Observable Markov Decision Processes (POMDPs) [10] extend MDPs to describe the case when the agent can only indirectly observe the underlying state at each time step from a set of observations $\Omega$. These observations are modeled as conditionally dependent on the true hidden state by the observation function $O(s,o)$, which defines the probability that the agent will observe observation $o$ in state $s$.

### B. *FETCH-POMDP Definition*

Solving POMDPs is very challenging; to make progress, we need to define a model with specific state representations and independence assumptions that enable us to define or learn model parameters and carry out efficient inference.

Our POMDP model for the item-delivery task is called the FEedback-To-Collobarative-Handoff POMDP, or FETCH-POMDP. The model is given by components $\langle I, S, A, R, T, O \rangle$.

- $I$ is a list of all items on the table, which we assume are known and fixed. Each item $i \in I$ has a known $(x,y,z)$ location on the table, and has a set of associated words $i.\texttt{vocab}$ that may be used to refer to itself.
- $S$: $i_d \in I$ is the human's desired item which is hidden. For convenience, we also include the last item the robot asked about (or `null` if none): $i_r \in I \cup \{\texttt{null}\}$. Note that $i_r$ is known and hence the state $(i_d, i_r)$ is mixed observable [15].
- $A$: We categorize actions as social feedback and physical actions. The physical actions consist of a wait action and a parametrized pick$(i)$ action. The wait action does nothing, and merely advances the time-step by one. A pick$(i)$ action finalizes the robot's selection of item $i$ as the user's desired object, and the interaction terminates. The social feedback actions consist of a parametrized point$(i)$ action. When the robot chooses to point at an item $i$, the robot moves its end effector in a pointing motion above item $i$, and asks "this one?" Because both the pick$(i)$ and point$(i)$ are parametrized by the items on the table, there are $2|I|+1$ total actions the agent can select at any time.
- $R(s,a)$: We provide a large positive reward for picking the correct item, a large negative reward for picking an incorrect item, and smaller negative rewards for wait and point. The costs of the different actions were initially set to correspond to the number of seconds it would take to complete said action, and were tuned from there using both simulated trials and a small pilot study. They were tuned to result in the shortest interaction time and the highest accuracy, regardless of social feedback paradigm.

| $a$ | $s$ | $R(s,a)$ |
|---|---|---|
| pick$(i)$ | $i = i_d$ | $+10$ |
| pick$(i)$ | $i \neq i_d$ | $-12.5$ |
| point$(i)$ | $*$ | $-6$ |
| wait | $*$ | $-1$ |

- $T(s,a,s') \equiv p(s' \mid s,a)$: Our transition function is deterministic. We assume that $i_d$, the desired object, remains fixed. We also assume that after the robot asks about item $i$, $i_r$ changes deterministically from `null` to $i$.[1] Littman [13] and others [2] have shown that deterministic POMDPs retain much of their expressive power compare to stochastic POMDPs. The focus of our model is to estimate the value of a hidden variable, not handle stochastic transitions. The complexity in our problem arises from our observation function.
- $O(s,o) \equiv p(o \mid s)$: Observations consist of the human's language, $l$ and gesture, $g$. To define the POMDP the robot needs a model of $p(o|s) = p(l,g|s)$. Most of the complexity of our model is captured in this observation model which is defined in the next section.

### C. *Observation Model*

Users may produce speech and gestures, which we consider as observations in our model. Each observation $o \in \Omega$ is a tuple of language $l$, and gesture $g$.

[1] We could model the transition of $i_r$ as being stochastic, to capture the possibility of the human not understanding the robot's question. In domains where the only method of communication is noisy, e.g. a phone-line [26], this is very important. In a domain like ours, where the human can both see and hear the robot with high fidelity, we were able to design our robot actions so the human understood the robot's question with near perfect accuracy.

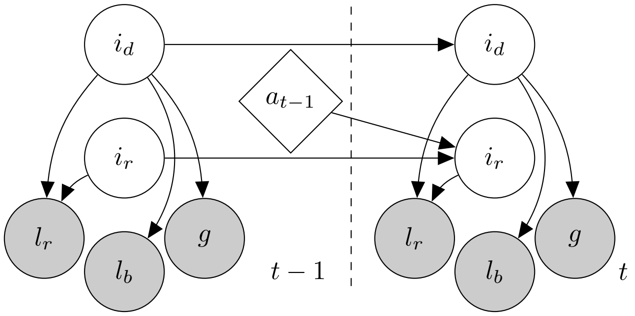

Fig. 2. A graphical model of our FETCH-POMDP. Hidden variables are white, observed variables are gray.

- Language: Let $l$ be the string of words the user has said.[2] We split $l$ into two portions: The response utterance $l_r$ consisting of positive/negative response words, and the base utterance $l_b$ consisting of all other words. Either of these two strings may be empty ($\epsilon$). To determine which words the user spoke are part of $l_r$, we compare each word in $l$ to a list of positive and negative responses. The positive responses $r_p$ are the words { 'yes', 'yeah', 'sure', 'yup' } and the negative responses $r_n$ are { 'no', 'nope', 'other', 'not' }.
- Gesture: $g$ is the pointing vector, measured from the user's head to the user's wrist[3]. If no pointing is detected, $g$ has value $\texttt{null}$.

The entire observation calculation is given as follows:

$$p\left(o \,|\, s\right) = p\left(l_b, l_r, g \,|\, i_d, i_r\right) \tag{1}$$

We assume the three observation components are conditionally independent given the state.

$$p\left(o \,|\, s\right) = p\left(l_b \,|\, i_d, i_r\right)\; p\left(l_r \,|\, i_d, i_r\right)\; p\left(g \,|\, i_d, i_r\right) \tag{2}$$

As can be seen in our graphical model (Fig. 2), $l_b$ and $g$ do not depend on $i_r$, as the response of the human is captured in $l_r$. Therefore

$$p\left(o \,|\, s\right) = p\left(l_b \,|\, i_d\right)\; p\left(l_r \,|\, i_d, i_r\right)\; p\left(g \,|\, i_d\right) \tag{3}$$

We will now describe each portion of Equation 3.

*1) Language Component:* The probability of the base utterance is $p\left(l_b \,|\, i_d\right)$. It is calculated according to a smoothed unigram speech model. This unigram model, also called a bag-of-words model, considers the probability of each word independently. Each utterance $l_b$ is broken down into its individual words $w \in l$:

$$p\left(l_b \,|\, i_d\right) = \begin{cases} p_l\; \prod_{w \in l_b} p\left(w \,|\, i_d\right)\;, & l_b \neq \epsilon \\ 1 - p_l\;, & l_b = \epsilon \end{cases} \tag{4}$$

The probability of each word (within the product term) is:

$$p\left(w \,|\, i_d\right) = \frac{\mathbb{I}[w \in i_d.\texttt{vocab}] + \alpha}{|i_d.\texttt{vocab}| + \alpha\; |\texttt{words}|}\;, \tag{5}$$

where $\mathbb{I}[w \in i_d.\texttt{vocab}]$ is one if $w$ appears in the vocabulary of $i_d$, and zero otherwise. $|i_d.\texttt{vocab}|$ is the number of words in the vocabulary of $i_d$. $|\texttt{words}|$ is the total size of the vocabulary. $\alpha$ is the smoothing parameter, which guarantees the probability of a word can never be zero. Also, $p_l$ is the probability an utterance is made. We empirically chose $\alpha = 0.2$ and $p_l = 0.95$ based on simulation trials and the small pilot study.

Next we consider the probability of the response, $p\left(l_r \,|\, i_d, i_r\right)$. We make another conditional independence assumption, so that each word $u$ in $l_r$ is independent.

$$p\left(l_r \,|\, i_d, i_r\right) = \begin{cases} p_l\; \prod_{u \in l_r} p\left(u \,|\, i_d, i_r\right)\;, & l_r \neq \epsilon \\ 1 - p_l\;, & l_r = \epsilon \end{cases} \tag{6}$$

To calculate $p(u|s)$, we must consider three possibilities for the state: $i_r = i_d$, $i_r \neq i_d$, and $i_r = \texttt{null}$. If $i_r = i_d$, then it is very likely that the user will respond with a positive utterance, and very unlikely that they will respond with a negative utterance. If $i_r \neq i_d$, then the opposite is true. If $i_r = \texttt{null}$, then no question has been asked, so both types of responses are equally likely. The mathematical representation of $p\left(u \,|\, s\right)$ is governed by the following conditional probability table:

TABLE I

CONDITIONAL PROBABILITY TABLE FOR $p\left(u \,|\, i_d, i_r\right)$

| | $u \in r_p$ | $u \in r_n$ |
|---|---|---|
| $i_r = i_d$ | 0.99 | 0.01 |
| $i_r \neq i_d$ and $i_r \neq$ null | 0.01 | 0.99 |
| $i_r =$ null | 0.5 | 0.5 |

The 0.99 and 0.01 values correspond to our assumption that the human is cooperating with the robot and will respond truthfully to questions. The 0.5 values come from the fact that if no question has been asked, either response type is equally likely.

*2) Gesture Component:* Gesture is measured as a pointing vector starting at the head of the user and moving through the user's wrist. (see Fig. 3). We assume a user points directly at their desired item $i_d$, with a Gaussian noise term on the angle with mean zero and standard deviation $\sigma$. Let $\theta_{i_d}$ be the angle between the observed pointing vector and an ideal pointing vector directly pointing at $i_d$. From our pilot study, we determined $\sigma = 0.15$ radians and $p_g = 0.1$ resulted in the fastest interaction time and highest accuracy.

$$p\left(g \,|\, i_d\right) = \begin{cases} p_g\; \mathcal{N}\left(\theta_{i_d}\,;\, 0, \sigma^2\right) & g \neq \texttt{null} \\ 1 - p_g & g = \texttt{null} \end{cases} \tag{7}$$

To determine if a gesture was made, we created a threshold for our gesture function. If $\theta_i > 0.3$ radians for all objects $i$ on the table, then we considered the user to not currently be pointing, and $g = \texttt{null}$.

[2] $l$ is obtained by transcribing microphone input using CMU Pocketsphinx, a speech-to-text software [5].

[3] $g$ is obtained using a Microsoft Kinect and OpenNI's skeleton tracker software [16].

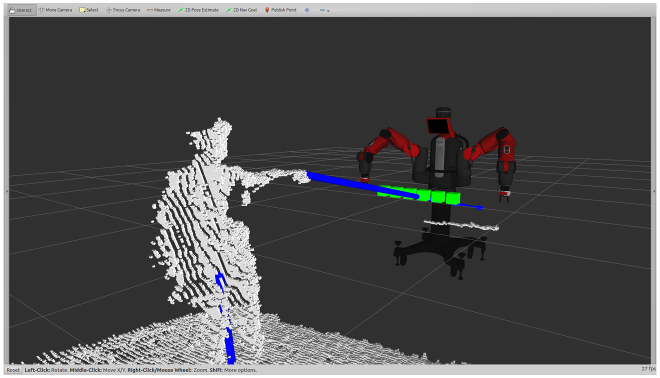
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
27 fps

Fig. 3. Screen-capture of RViz visualization of user pointing at item. The blue vectors represent the calculated pointing vectors from each arm. The left arm is down at the user's side, and the right arm is pointing at item four.

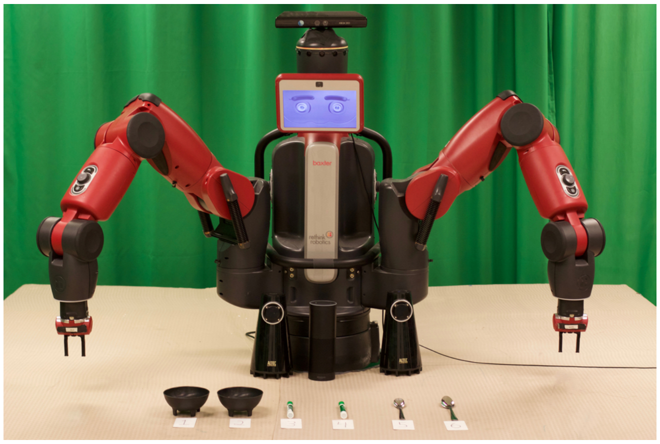

Fig. 4. User's view of robot, with items arranged in the ambiguous configuration.

### D. Solving the POMDP

Our observation space for language is countably infinite, and our observation space for gesture is continuous. An enormous observation space makes solving the POMDP challenging. We solve it by using an approximate solver, sparse sampling [11], on the resulting belief MDP for the POMDP. All POMDPs can be converted into a corresponding belief MDP, which is an MDP where every belief in the original POMDP is a state. The state space of the belief MDP is therefore continuous [10]. The solution to the belief MDP is identical to the original POMDP [10].

Sparse sampling finds an approximate solution to the MDP by constructing a probabilistic decision tree of a finite depth $d$, where each node is a state-action pair, and chooses the action whose branch has the highest expected reward. To construct the tree, the algorithm samples a finite number $n$ of observations from each node, and treats these finite observations as the total observation space of each node. This type of solver is called a receding horizon planner, because the planner can only consider states up to $d$ actions away. Therefore the solver's accuracy increases as $d$ and $n$ increase. Of course, as $d$ and $n$ increase, runtime also increases.

From our simulations and pilot study, we found $d = 2$ and $n = 10$ lead to appropriate action choices while running quickly enough to enable real-time communication.

As mentioned earlier, sparse sampling must be able to sample observations.

*1) Sampling Language:* We model the sampled $l_b$ is a single word sampled from the observation function. We do the same for $l_r$. We constrained the length of the samples to one word to speed up calculations. In our simulations, we did not find this constraint affected performance.

*2) Sampling gesture:* Gesture is sampled from the observation function for gesture. The simulated human will directly point at the desired item, with an added noise term sampled from the Gaussian distribution described in III-C.2.

## IV. EVALUATION

The goal for our system is to perform robot-to-human object hand-off as quickly and accurately as possible. We define the speed of the interaction as the time the human begins the request to the time the robot decides to pick an item. We report accuracy as whether the robot decided to hand over the item the human desired.

To evaluate our system, we conducted a user study where users used language and gesture to instruct the robot to hand them a particular item. We had two physical configurations of the items, ambiguous and unambiguous. In each physical layout we tested three robot interaction paradigms. In paradigm one, the robot never gave social feedback. This is equivalent to an improved version of the model from our previous work [22]. In paradigm two, the robot always asked at least one question about the item it considered most likely until it was 95% confident of its answer. This is comparable to the PODMP model described in [26], where the system determined what piece of information to ask about via a POMDP solution, but had a fixed question asking routine. In paradigm three, the robot intelligently asked questions according to the found solution of the FETCH-POMDP. We report the speed and accuracy at this task across all combinations of physical layouts and interaction paradigms.

Our motivation for these physical configurations is to test the two ends of the spectrum for needing social feedback. When the environment is unambiguous, the robot should be able to intelligently infer that it does not need to be asking lots of questions, but as the environment becomes ambiguous, the robot will intelligently infer the need to ask questions.

Our evaluation aimed to assess the effectiveness of our autonomous system at increasing the speed and accuracy of our human-robot interaction. Specifically we had the following two hypotheses:

- **H1:** In the unambiguous configuration, our autonomous system will be at least as accurate as the two baselines, faster than always-social feedback, and at least as fast as non-social-feedback.
- **H2:** In the ambiguous configuration, our autonomous system will be more accurate than no-social feedback, and as accurate as always-social-feedback. Our system

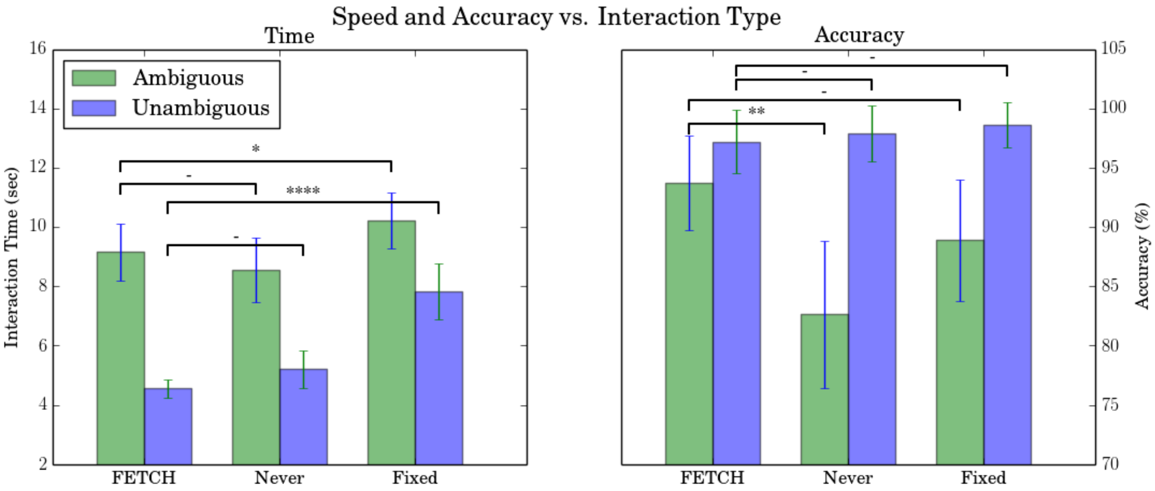

Fig. 5. Average interaction time and accuracy for users. Error bars represent 95% confidence intervals.

will be faster than always-social and at least as fast as no-social-feedback.

The dependent variables are the accuracy and elapsed time measures recorded with each trial (see Sec. IV-B). The independent variables are what interaction paradigm was used by the robot, and the physical configuration of the items on the table. The null hypothesis for H1 is that all interaction paradigms will have have same accuracy and elapsed time in the unambiguous layout configuration. The null hypothesis for H2 is that all interaction paradigms will have have same accuracy and elapsed time in the ambiguous layout configuration.

### A. Physical Setup

Each user stood in front of a Baxter robot with six items spread across a table directly in front of the robot, as shown in Figure 4. The items were two black plastic bowls, two green expo markers, and two silver metal spoons. The two bowls, two markers, and two spoons are identical except for their locations. The Kinect was mounted on the robot's head

In the unambiguous layout, the items were spread far apart from one another along a large arc in front of the robot (inter-item distance of 45 cm), and the user stood 1.22 m away from the objects, at the minimum range for the Kinect. The items were spread to cover the entire reachable span of the robot. Identical items were placed far apart from one another, so as to be easily distinguishable using pointing gestures. In the ambiguous layout, the items were in a line at the center of the table, and the user stood 3.2 m away, just inside the Kinect's maximum range of 4 m. Identical items were placed next to each other (i.e. bowls next to bowls and spoons next to spoons) with an inter-item distance of 15 cm, making pointing less effective at distinguishing items. Any closer and the robot's pointing action would have become uninterpretable. Half of the users had the items in the ambiguous layout, and half had the unambiguous layout.

### B. Experimental Procedure

We want each item to be selected an equal number of times with each interaction paradigm, so we gave each user a fixed list of items to select. The ordering of the list was shuffled. The user requested the item from the robot using natural language and gesture, and was instructed to treat the robot as they would a person. The interaction began following a countdown given from the experimenter, and ended when the robot told the user which item it thought was desired. We had 16 users in total. Each user conducted 54 trials, 18 with no social feedback, 18 with intelligent social feedback, and 18 with always-social-feedback. For each of the interaction paradigms, every item was selected as the desired item 3 times. For each trial, two variables were measured; length of trial, and correctness of the robot's prediction.

### C. Statistical Analysis

Note that this study partially follows a within subjects design. All users perform trials with all interaction paradigms, but only perform trials with one of the two item configurations. We would have preferred to conduct a full within-subjects design study, but doubling the trials for each user would have meant a average study time of an hour per user, which would have led to user fatigue. Interaction paradigm efficacy was more susceptible to individual differences in pilot studies, so we chose for those variables to be tested within subjects.

Because our interaction paradigms were measured within-subjects, we tested for significance with the Wilcoxon signed-rank test, a non-parametric statistical hypothesis test. It is similar to the paired Student's t-test, but does not assume the data is normally distributed [17].

### D. Results

Overall all systems were accurate, detecting the correct item with 88.4% accuracy in the ambiguous configuration and with 97.9% accuracy in the unambiguous configuration.

Overall mean interaction time was $9.31\,\mathrm{s}$ in the ambiguous configuration and $5.86\,\mathrm{s}$ in the unambiguous configuration.

We found the results of our experiments confirmed our hypotheses. In the ambiguous configuration, our model was not significantly slower than no social feedback ($p = 0.06$), with an average difference of $0.59\,\mathrm{s}$, but was significantly faster than always asking social feedback ($p = 0.03$), with an average difference of $1.05\,\mathrm{s}$. There was no significant difference in accuracy between our model and the always asking policy ($p = 0.14$), but our model was significantly more accurate then not asking ($p = 0.003$), with an average improvement of 11.1%.

In the unambiguous configuration, our model was significantly faster than always asking ($p = 3.62 \times 10^{-22}$) with an average difference of $3.28\,\mathrm{s}$, and not significantly faster than not asking ($p = 0.89$). All paradigms in the unambiguous configuration had average accuracies above 97%, with no significant difference between them. See Fig. 5 for a graph of these results.

Combining the two physical configurations together, we found FETCH-POMDP was significantly faster than never asking by 5.21% ($p = .014$), while being just as fast ($0.03\,\mathrm{s}$ faster on average). When combining the physical configurations, FETCH-POMDP was significantly faster than the fixed asking policy by $2.17\,\mathrm{s}$, or 25% faster ($p = 1.7 \times 10^{-17}$), while also being more accurate (2.1% more accurate on average). Each user completed a qualitative survey after performing all the trials. When asked about what they thought the robot understood, all users correctly inferred that the robot understood pointing and basic name descriptions of items. Interestingly, 6 users, or 38%, thought the robot could also understand prepositional phrases such as "to the left of x". Our system does not understand prepositional phrases, but this suggests question asking improves the perceived competence of the robot.

## V. DISCUSSION

We were surprised that FETCH-POMDP was more accurate than the fixed feedback policy in the ambiguous configuration. We had hypothesized that fixed feedback would be the most accurate, since asking more questions should remove more confusion. We found, however, that asking too many questions risked speech-to-text failures that would confuse the system. One mistake we repeatedly saw during trials, for instance, was misinterpreting the word 'yes' as the word 'hand.' The more questions the system asked, the higher the chance of a transcription error. This is why the fixed feedback policy had a lower average accuracy than FETCH-POMDP in the ambiguous configuration. In the unambiguous configuration, the pointing observations were so much stronger that the fixed feedback model rarely needed to ask more than one question, so transcription error did not noticeably affect accuracy.

Another surprising result was that FETCH-POMDP was on average faster than no feedback in the unambiguous configuration. This is because the system was usually able to infer the correct item from its initial observations, but occasionally would be unsure. With FETCH-POMDP, the robot was able to ask a question, resolve the ambiguity, and pick the desired item. Without social feedback, the robot could only wait. The human wouldn't immediately realize the robot needed more observations, so the interaction would come to a standstill. These outlier interactions can be seen in Figure 6.

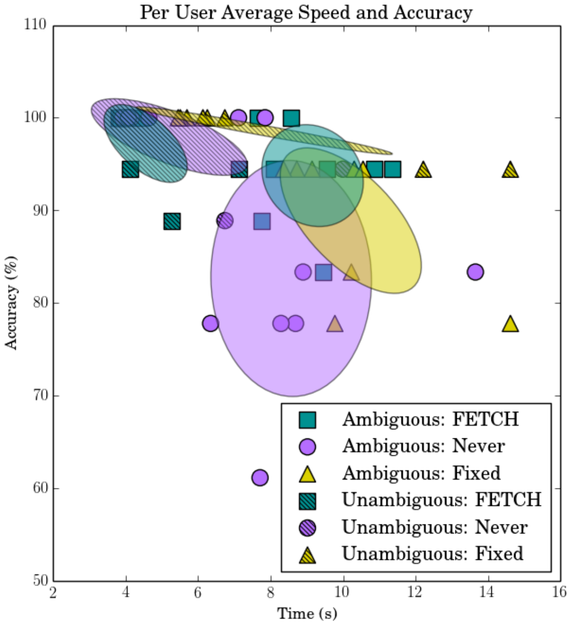

Fig. 6. Average accuracy and time for each user across each interaction paradigm. Each point represents the average accuracy and trial time for a interaction type for a single user. Ellipses represent Gaussian distribution fitted to points to one standard deviation. Note how the FETCH-POMDP ellipses (shown in green), are farthest to the top and left, with the smallest standard deviations.

During trials, many users used prepositional phrases in order to describe items, such as "Hand me the spoon to the left of the bowl." Although the language model in this work did not account for referential phrases, the agent was able to use intelligent social feedback to figure out what the human desired. This may explain why many users reported that they thought the robot did understand prepositional phrases. Methods exist to interpret referential language, but problems in understanding will still occur. Our model will help correct those mistakes, regardless of the exact method of state estimation and language understanding.

## VI. CONCLUSION

This work shows how social feedback improves human robot communication, and how POMDPs are effective methods of generating this feedback. The FETCH-POMDP's ability to intelligently balance between clarifying uncertainty with speed allows for realistic interactions between a social robot and a human. This ability allows for realistic interactions with human users, which affords natural collaborations over tasks between humans and robots.

Using multimodal observations to model a hidden state of the human from noisy signals allows for richer state extraction than either modality alone. The FETCH-POMDP's framework allows extensions to make a more sophisticated model of the agent's hidden states. This lends itself to a more general framework that can model agent's mental states in more generalized interactions.

In future work, we would like to extend our model to more actions, like both pick and place. This capability would expand the functional use of the robot, and bring it closer to something that can actually be used in a real workplace. We would also like to extend our observation models, improving on language and adding other modalities. Currently we do not consider the grammatical parse of the human's speech. We would like the model to understand prepositional phrases ("on the left", "nearest to me"). This would allow the robot to understand how items are spatially related to other items through language. Additional modalities, such as eye-tracking, would increase the accuracy of the system. As stronger signal extraction methods are implemented into the FETCH-POMDP model, we believe larger amounts of items could be selected from with reasonable accuracy and speed.

## VII. ACKNOWLEDGEMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Grants No. W911NF-15-1-0503 and D15AP00102.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

Lawson L.S. Wong was supported by a Croucher Foundation Fellowship.

## REFERENCES

[1] R. Bellman. A Markovian decision process. *Indiana University Mathematics Journal*, 6:679–684, 1957.

[2] B. Bonet. Deterministic POMDPs revisited. In *UAI*, 2009.

[3] Maya Cakmak and Andrea L Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012.

[4] Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40. ACM, 2014.

[5] CMU Sphinx. CMU sphinx. http://cmusphinx.sourceforge.net/, 2016.

[6] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[7] Terrence Fong, Charles Thorpe, and Charles Baur. Robot, asker of questions. *Robotics and Autonomous systems*, 42(3): 235–243, 2003.

[8] P.J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multi-agent settings. *Journal on Artificial Intelligence Research*, 24:49–79, 2005.

[9] J. Hoey, P. Poupart, A. von Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[10] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1–2):99–134, 1998.

[11] M. Kearns, Y. Mansour, and A.Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49(2–3):193–208, 2002.

[12] T. Kollar, S. Tellex, D. Roy, and N. Roy. Grounding verbs of motion in natural language commands to robots. In *ISER*, 2010.

[13] M.L. Littman. *Algorithms for sequential decision making*. PhD thesis, Brown University, 1996.

[14] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *ISER*, 2012.

[15] S.C.W. Ong, S.W. Png, D. Hsu, and W.S. Lee. Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[16] OpenNI Tracker. OpenNI tracker. http://wiki.ros.org/openni_tracker, 2014.

[17] Sidney Siegel. Nonparametric statistics for the behavioral sciences. 1956.

[18] S. Tellex, T. Kollar, S. Dickerson, M.R. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, 2011.

[19] S. Tellex, T. Kollar, S. Dickerson, M.R. Walter, A.G. Banerjee, S. Teller, and N. Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4):64–76, 2011.

[20] B. Thomson and S. Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

[21] A. Vogel, C. Potts, and D. Jurafsky. Implicatures and nested beliefs in approximate decentralized-POMDPs. In *ACL*, pages 74–80, 2013.

[22] David Whitney, Miles Eldon, John Oberlin, and Stefanie Tellex. Interpreting multimodal referring expressions in real time. In *International Conference on Robotics and Automation*, 2016.

[23] J.D. Williams and S. Young. Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2116–2129, 2007.

[24] J.D. Williams and S. Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[25] E. Wu, Y. Han, D. Whitney, J. Oberlin, J. MacGlashan, and S. Tellex. Robotic social feedback for object specification. In *AAAI Fall Symposium on AI for Human-Robot Interaction*, 2015.

[26] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010.