

Data-driven HRI: Learning social behaviors by example from human-human interaction

Phoebe Liu, Dylan F. Glas, Takayuki Kanda, *Member, IEEE*, Hiroshi Ishiguro, *Member, IEEE*

Abstract—Recent studies in human-robot interaction (HRI) have investigated ways to harness the power of the crowd for the purpose of creating robot interaction logic through games and teleoperation interfaces. Sensor networks capable of observing human-human interactions in the real world provide a potentially valuable and scalable source of interaction data that can be used for designing robot behavior. To that end, we present here a fully-automated method for reproducing observed real-world social interactions with a robot. The proposed method includes techniques for characterizing the speech and locomotion observed in training interactions, using clustering to identify typical behavior elements and identifying spatial formations using established HRI proxemics models. Behavior logic is learned based on discretized actions captured from the sensor data stream, using a Naïve Bayesian classifier. Finally, we propose techniques for reproducing robot speech and locomotion behaviors in a robust way, despite the natural variation of human behaviors and the large amount of sensor noise present in speech recognition. We show our technique in use, training a robot to play the role of a shop clerk in a simple camera shop scenario, and we demonstrate through a comparison experiment that our techniques successfully enabled the generation of socially-appropriate speech and locomotion behavior. Notably, the performance of our technique in terms of correct behavior selection was higher than the success rate of speech recognition, indicating its robustness to sensor noise.

Index Terms—Human-robot interaction, data-driven learning, learning by imitation, social robotics, service robots.

I. INTRODUCTION

As robots become more prevalent in the modern era, the field of human robot interaction (HRI) provides the promise of integrating robots into everyday human life. These service robots are gaining presence in museums [1-4], offices [6, 7], elder care [8, 9], shopping malls [10, 11], and healthcare facilities [12]. The ability of the robots to socially integrate into those environments will be essential. For example, a shop

assistance robot needs to be able to greet customers, answer questions, give recommendations, guide to various products, and assist the customers in various situations.

One approach for designing interaction logic for a robot is to explicitly program the behaviors the robot should execute, the expected inputs from the environment, and the execution rules it should follow. However, this can be a difficult process, heavily dependent on the designer's ability to imagine a variety of social situations (for example, anticipating all of the questions people may ask the robot) and use their intuition to specify social behaviors and execution rules for the robot, which may be difficult to articulate. This process can be very labor intensive, and it becomes even more difficult to create robust interactions when natural variations of human behavior and errors due to sensor noise are considered.

We believe that a data-driven approach to interaction design could provide solutions to many of these problems. By directly capturing behavior elements such as utterances, social situations, and transition rules from a large number of real, *in-situ* human-human interactions, it may be possible to easily and automatically collect a set of behaviors and interaction logic that can be used in a robot. This would reduce the difficulty and effort of interaction design, and it could enable more robust interaction logic, since sensor errors and variation of behavior would be implicitly considered.

Thanks to recent advances in sensor technology, this idea of data-driven interaction design based on real-world interactions could soon become a realistic possibility. High-precision tracking systems are being deployed in public spaces, enabling passive collection of natural human interaction data [13], and technologies such as microphone arrays may soon provide usable sound source localization and speech recognition in noisy real-world environments [14]. Such technologies could allow enormous amounts of human behavior data to be collected effortlessly. For example, deploying sensor networks in a chain of retail stores could provide hundreds of thousands of example interactions in a matter of few months, which could be used to train a robot to perform the role of a shop clerk.

The possibility of effortless collection of large amounts of interaction data is what gives importance to this idea of data-driven interaction design. HRI researchers have recently begun to take advantage of the scalability of the web to train robots based on collected interaction data from the crowd [15, 16]. We believe that capturing human-human interactive behavior

Manuscript received February 17, 2015; This work was supported in part by JSPS KAKENHI Grant Number 25240042 and in part by the JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project. An early version of this work was presented in part at the 23rd IEEE Int. Symp. on Robot and Human Int. Communication, Edinburgh, Scotland, Aug. 2014[5]. The current paper describes several technical improvements over that system and presents an evaluation of its performance through a comparison experiment.

P. Liu, D.F. Glas, and H. Ishiguro are with the JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project and Hiroshi Ishiguro Laboratories (e-mail: phoebe@atr.jp; dylan@atr.jp; ishiguro@sys.es.osaka-u.ac.jp), and T. Kanda is with the Intelligent Robotics and Communication Laboratories (email: kanda@atr.jp), at the Advanced Telecommunications Research Institute International, Kyoto 619-0288, Japan.

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org> provided by the author.

through sensor networks will prove to be another powerful and scalable way to leverage the wisdom of the crowd to create interactive robots.

Our objective in this study is to provide a proof-of-concept of such a data-driven interaction design methodology and to provide observations and suggest directions for future development of this powerful concept. We present a fully-autonomous method for training a socially-interactive robot from observed examples of human-human interaction, wherein behavior contents and interaction logic are extracted directly from noisy sensor data without human intervention.

Included in this work are techniques for (a) identifying typical action elements from a set of example interactions, (b) reproducing observed human behaviors in a robot despite high amounts of sensor noise, and (c) robustly selecting context-appropriate behaviors for the robot to execute in live social interactions.

II. RELATED WORK

As mentioned above, our goal is to utilize the crowd, by capturing people's movement and speech during live human-human interaction and automatically generating interaction logic for reproducing the observed behaviors based on the set of passively-collected data. Such ideas of learning from data and using the crowd for learning have been explored in a number of different areas within the field of social robotics.

A. Creating Interaction Content

In designing interaction flows for social robots, several custom frameworks have been developed to explicitly break down interaction into subcomponents, such as state (input) and behavior actuation (output) components, and specify transition logic to direct the execution flow based on data from sensor inputs [17, 18]. Teleoperation interfaces have also been used to iteratively build interaction content over a period of time [19, 20]. In this work, we use sensors to capture interaction content directly from human interactions.

B. Learning from Data

In robotic tasks like manipulation, machine learning approaches such as *learning by demonstration* are often utilized to learn from a dataset of examples in order to reproduce a demonstrated task, as it is easier for humans, including non-robotic-experts, to input poses by moving an arm manually, than to explicitly specify them numerically. Some examples include trajectory following [21, 22] or joint motion replication [23]. Typically this is seen as a way to input sensory-motor patterns, but not cognitive and decision-making skills.

In social robotics, machine learning has been used to teach low-level behaviors, for example, to mimic gestures and movements [24], and to learn how to direct gaze in response to gestural cues [25]. In one example, pointing and gaze behaviors were recognized in an imitative game using a hidden Markov model [26]. The challenge in using a data-driven approach to learn an entire social interaction is the level of complexity that goes into decision-making process. The ways we act are often influenced by our intentions, and it is still an open question to

how we can extract intentions from only observed behaviors.

Data-driven dialogue systems have been demonstrated in robots which infer meanings from spoken utterances. Rybski *et al.* developed an algorithm which allowed a human to interact with a robot with a subset of spoken English language in order to train the robot on a new task [27]. Meena *et al.* used a data-driven chunking parser for automatic interpretation of spoken route directions for robot navigation [28].

Unlike other works, we focus on training examples based on real human-human interaction, with natural spoken dialogue.

C. Using the crowd for learning

With the advancement of high-precision tracking systems able to monitor real social environments [13, 29], it is becoming possible to collect large amounts of detailed interaction data with little effort. This suggests the possibility of using a "crowdsourcing" approach, like the distributed techniques used over the web to solve complex problems, e.g. users on Amazon's Mechanical Turk helping to annotate images for grasp planning [30].

The use of real human interaction data collected from sensors for learning interactive behaviors has been investigated in numerous works. The robot JAMES was developed to serve drinks in a bar setting, in which a number of supervised (i.e. dialog management) and unsupervised learning techniques (i.e. clustering of social states) have been applied to learn social interaction [31]. In contrast, we propose a completely unsupervised approach for both abstraction and clustering of social states as well as for robot behavior generation

In Young *et al.*'s work [32] [33], a person provides an example of an interactive locomotion style, which is used to teach the robot to generate interactive locomotive behaviors in real time according to that style. We also propose to use real human interaction to train the robot, but our focus is not only the robot's motion, but its speech as well.

Connectivity to the web has also changed the way interaction data can be collected. The Robot Management System framework was developed to make learning of manipulation and navigation tasks easier by collecting demonstrations from remote users through a browser as a game [16]. The Restaurant Game used annotated crowdsourced data to generate abstracted representation of data to automate game characters [34]. The Mars Escape online game used crowdsourcing to learn robot behaviors [15, 35, 36]. The idea was to use a data-driven approach to develop HRI behaviors from players of an online collaborative game to provide large amounts of training data and reproduce behaviors in a real autonomous robot.

Our work complements these approaches by considering a crowd-based data collection from sensors in a physical environment, where some new challenges include resolving recognition ambiguities due to sensor noise and natural variation of human behavior.

III. DATA COLLECTION

A. Sensor Environment

To collect human-human interaction data for our learning

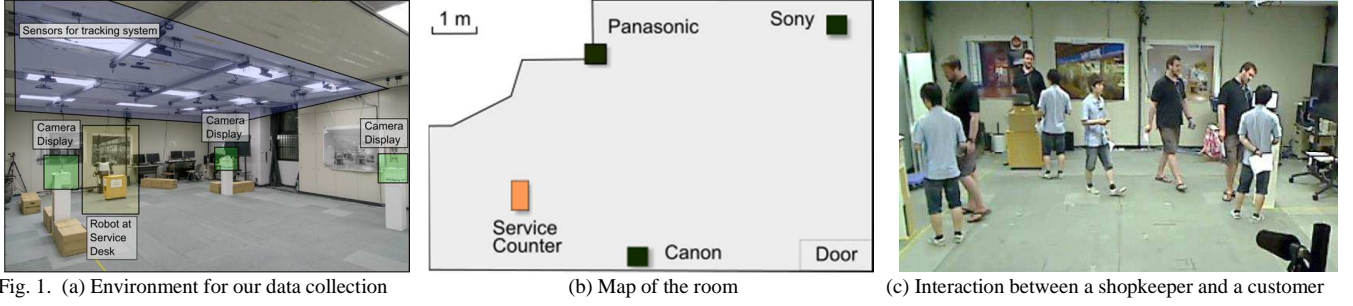


Fig. 1. (a) Environment for our data collection

(b) Map of the room

(c) Interaction between a shopkeeper and a customer

study, we prepared a data collection environment with a sensor network, including a human position tracking system and a set of handheld mobile phones to use for speech recognition, to capture participants' motion and speech.

The position tracking system consists of 16 ceiling-mounted Microsoft Kinect RGBD sensors, arranged in rows. Particle filters are used to estimate the position and body orientation of each person in the room based on point cloud data [13].

Ideally, we would like to collect people's speech passively. However, modern speech recognition technology is still not robust enough to use with ambient microphones when background noise exists in the environment [37, 38]. For that reason, we developed a smartphone application to capture speech directly from a hands-free headset, and use the Android speech recognition API to recognize utterances, sending the text to a server via Wi-Fi. The user wears a hands-free headset and touches anywhere on the mobile screen to indicate the beginning and end of their speech, so no visual attention is required, making it possible to conduct natural face-to-face interactions without breaking eye contact.

Although the study was conducted in Japan, we found a greater variety of tools available for analysis of English text, so the interactions in this study were carried out in English.

B. Training Interactions

We chose a shopping scenario in a camera shop setting, where we asked one person to role-play as a shopkeeper and one person as a customer. To create a set of training interactions, we set up three product displays, representing different digital camera models, in an 8m x 11m experiment space, shown in Fig. 1(a) and (b). Each product display had a feature sheet with a short list of the camera's relevant features, such as "optical zoom" or "megapixels". We also set up a service counter, where we instructed the shopkeeper to stand at the start of each interaction.

Participants were members of our laboratory and interacted with each other in English. Four fluent English speakers role-played as the shopkeeper. 10 participants, including 7 fluent English speakers, played the role of customer. Each customer took part in 10-20 interactions, for a total of 178 trials.

In each trial, the customer was instructed to role-play in one of the following scenarios: (1) a need-based customer, who is looking for a camera with a specific feature (4 trials), (2) a curious customer, who is interested in multiple cameras (4 trials), or (3) a window-shopping customer, who prefers to browse around alone (2 trials). In order to help the participant to naturally role-play as a specific type of customer, we gave

the customer a different feature to look for each time. The shopkeeper was not informed of the chosen scenario, and was instructed to allow the customer to browse, to answer any questions the customer had, and to gently introduce products when appropriate, as shown in Fig 1(c).

Before the experiment, the participants were trained to use the Android phone and given a list of camera features to ask about. The shopkeeper was given a reference sheet containing a set of feature specifications for each camera. The practice trials were designed to help the participants become accustomed to using the Android phone and to illustrate the differences between the interaction scenarios.

The goal of the data collection was to capture repeatable interactions, so we restricted the scope of the scenario to focus on providing information about the cameras. For this reason, we asked the participants to keep the interactions simple by avoiding other topics, such as negotiating the price of the camera (e.g. "can you give me a better deal?").

Furthermore, we found it necessary to remind participants not to make up new information that did not exist in our scenario. For example, if a shopkeeper participant was asked "what kind of warranty policy do you have?", which was not defined in the scenario, they would have had to improvise an answer. These improvised responses would not be useful for learning because of inconsistency over time (in pre-trials, one shopkeeper participant said the store had a 1-year warranty policy on one occasion, but later said it was a 5-year warranty).

C. Example of human-human interaction

Within the defined scenario, the participants interacted in a free-form way, using natural conversational language, and a reasonable degree of variation in people's phrasing and terminology was observed. Table 1 illustrates this variety with transcripts from two example trials by the same participant: (1) a need-based customer looking for a camera with large memory storage, and (2) a curious customer interested in cameras with good battery life.

IV. PROPOSED TECHNIQUE

A. Overview

We implemented a fully unsupervised data-driven strategy to enable a service robot to reproduce human behaviors using only captured data from human-human interaction. Our approach represents interaction data via several abstractions, as follows:

- Customer **speech** is vectorized using Latent Semantic Analysis (LSA) and other text processing techniques (Sec. IV. B.1).

TABLE I. EXAMPLES FROM HUMAN-HUMAN INTERACTION

Example of a need-based customer	Example of a curious customer
S: (<i>Approaches customer</i>) Hi are you looking for anything in particular today? C: Yes I would like to... I am looking for a camera with good storage memory. S: (<i>Guides to Canon</i>) Ok the Canon Rebel XT _i can hold 10000 photos. C: Ok, that is very good. What about the price? S: This camera is \$400. C: I see. Is it heavy? S: Yes, very heavy. C: How much? S: Like, a kilogram. C: I see, that is very heavy. Well I will think about it. Thank you. (<i>Leaves shop</i>) S: Sure, no problem.	C: (<i>Goes to Sony</i>) Excuse me. S: (<i>Approaches customer</i>) Yes sir how can I help you? C: I am looking for a camera that I can use for a long time without changing the battery. S: (<i>Guides to Canon</i>) Ok we have a couple of options for that; over here is the Canon Rebel XT _i . It has a 7 hour battery life. C: I see, and other possibilities? S: (<i>Guides to Panasonic</i>) Other possibilities for long battery life are the Panasonic Lumix... this can run for 9 hours on standby. C: So this is longer. What's the difference between these two? S: This one is far worse in photo quality and it doesn't have a replaceable lens. C: I see, so probably I am more interested in the other model. I will think a little bit about it. Thank you very much. (<i>Leaves shop</i>) S: No problem sir.

- Shopkeeper **speech** is similarly vectorized, and it is then categorized into speech clusters representing lexically-similar, discrete utterances (Sec. IV. B.1).
- Customer and shopkeeper **trajectories** are segmented into stopped and moving segments, which are then clustered to identify typical stopping locations and typical motion trajectories (Sec. IV. B.2).
- An **interaction state** is defined based on the relative positions of the customer and shopkeeper, based on a set of two-person spatial formations taken from other HRI and proxemics work (Sec. IV.B.3).

We then analyze the training data to identify discrete **actions**, comprised of speech and/or movement of the customer or shopkeeper (Sec. IV.C.2), and we train a machine learning classifier to predict the appropriate shopkeeper action output which follows an observed customer action input.

The **input** (Sec. IV.C.3) to the classifier is the processed training data – a vector consisting of the customer's speech vector, spatial states for the customer and shopkeeper (Sec. IV.C.1), and the current interaction state of the customer and shopkeeper.

The **output** (Sec. IV.C.4) is a discretized shopkeeper action comprised of a speech cluster combined with a target interaction state.

The top part of Fig. 2 shows an overview of how the training data is processed to generate an input vector (“input”) and the corresponding shopkeeper action vector (“label”) for training the machine-learning classifier (Sec. IV.D.1-3).

During real-time operation, the sensor data are processed in the same way as they were during training – a vector is built by combining the LSA vectorization of the customer utterance with the spatial and interaction states abstracted from motion data. This vector is input to the trained classifier whenever a customer action is detected. A shopkeeper action is then predicted, and the speech and spatial formation of the predicted action are executed by the robot (Sec. IV.D.4).

The bottom part of Fig. 2 illustrates the processing of the sensor data as an input to generate robot behavior in real-time.

The following subsections will explain the details of these abstraction and vectorization processes, as well as the setup of the learning algorithm itself.

B. Abstraction

One challenge of using a data-driven approach to learn from human-human interaction is that human behavior occupies a very high-dimensional feature space, even considering only speech and locomotion (social behaviors such as gaze, gesture, and facial expression are not considered in the current study). In practice, however, the variation of human behavior occupies only a small manifold within this high-dimensional space – people usually perform actions in predictable ways and follow common patterns. We introduce here a number of abstraction techniques designed to capture these patterns, in order to reduce the dimensionality of the learning problem and diminish the effects of sensor noise.

First, we perform unsupervised **clustering** to identify sets of typical actions in the training data. Clustering is performed for speech data to deal with the large amounts of noise associated with speech recognition (Sec. IV.B.1), and also for motion trajectories observed by the tracking system, in order to identify typical stopping locations and motion paths in the environment (Sec. IV.B.2).

Next, we model each interaction as consisting of a sequence of stable **interaction states**, which last for several turns in a dialogue, recognizable by distinct spatial formations such as talking face-to-face or presenting a product. The modeling of interaction states helps to generate locomotion in a stable way, to specify robot proxemics behavior at a detailed level, and to provide context for more robust behavior prediction.

1) Speech Clustering

A great deal of variation was present in the speech captured in our training data, including alternative phrasings, *e.g.* “what is the price” versus “how much does it cost,” as well as speech recognition errors, *e.g.* “how much does the scammer cost” rather than “how much does this camera cost?” The challenge of speech processing is to represent these utterances in a way that preserves the similarity between phrases with similar semantic meaning.

The strategy for processing speech elements is shown in Fig. 3. As soon as an utterance was captured, **speech recognition** was performed. We then **extracted keywords** using a cloud-based service and created a vectorized representation of the speech results and keywords using **Latent Semantic Analysis (LSA)**.

Further processing was applied to shopkeeper's utterances

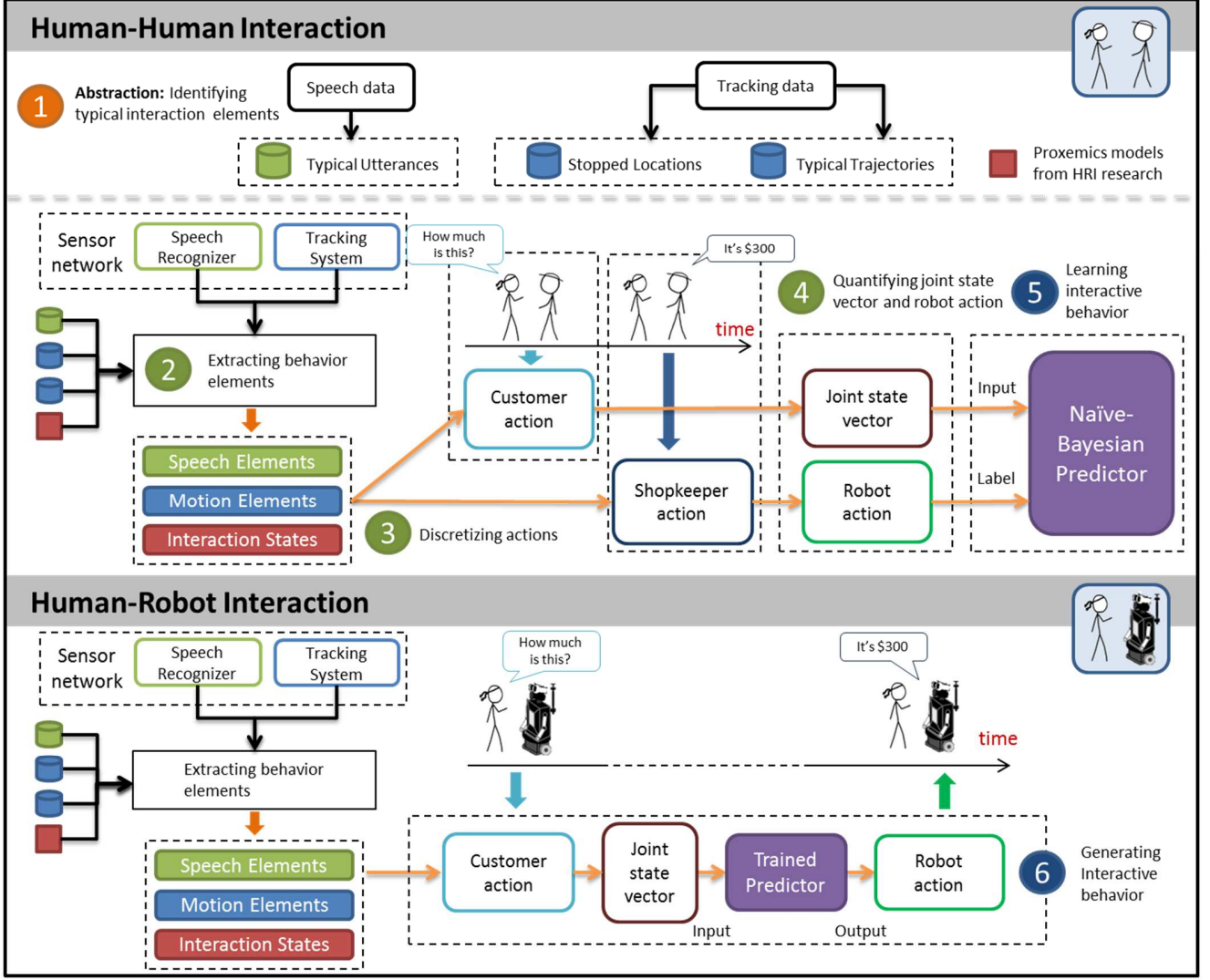


Fig. 2. Overall procedure for human-human interaction (data collection) and human-robot interaction (online)

only, with the goal of minimizing errors so that they could be used for generating robot speech. After vectorization of the utterances, we used unsupervised **clustering** to group them into clusters of similar utterances, and a **typical utterance** was then chosen from each cluster, to be used as content for synthesized speech output. Clustering was not applied to customer utterances, so that the information in the utterance vector could be kept for the purpose of prediction.

Speech recognition: For automatic speech recognition (ASR), we used the Google Speech API. An analysis of 400 utterances from the training interactions showed that 53% were correctly recognized, 30% had minor errors, e.g., “can it should video” rather than “can it shoot video,” and 17% were complete nonsense, e.g. “is the lens include North Florida.”

Keyword extraction: Phrases like “I am looking for a camera with large memory size” and “I am looking for a camera with large LCD size,” have different meanings despite lexical

similarity. To capture keywords in the phrases, we used AlchemyAPI¹, a cloud-based service for text analysis based on deep learning.

Latent Semantic Analysis: We created a vector to represent each utterance using Latent Semantic Analysis (LSA), a technique commonly used for classifying document similarity in text mining applications [39]. To achieve this, we performed several steps which are standard in text processing: we removed stop words, applied a Porter stemmer [40] to remove conjugations, enumerated n-grams (up to N=3), computed a term frequency – inverse document frequency (TF-IDF) matrix, and computed the singular-value decomposition of the TF-IDF matrix, truncating it to reduce the dimensionality of the space. The list of keywords returned for each utterance was separately processed using LSA, and those columns were added to the feature vector.

We chose the dimensionality for the truncated LSA matrix to

¹ <http://www.alchemyapi.com>

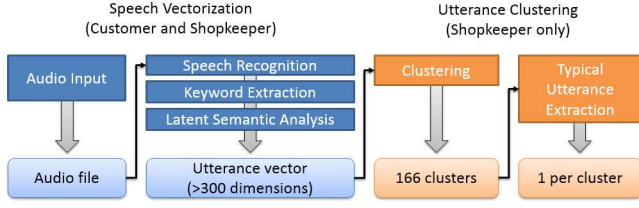


Fig. 3. The abstraction of speech elements into typical utterances

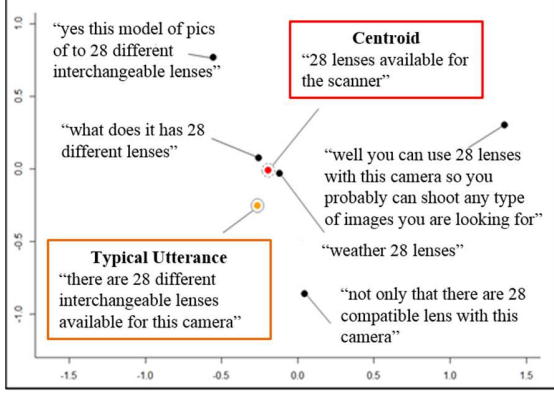


Fig. 4. An example of typical utterance selection from a shopkeeper speech cluster (ID 292). The utterance vectors have been collapsed to two-dimensional vector using multidimensional scaling (MDS) for visualization. The closest utterance to the centroid and the typical utterance chosen using our technique are shown

achieve a 50% “share” (percentage of cumulated singular values) as described in [41]. The numbers of dimensions and instances for each group are presented in Table 2.

Clustering of shopkeeper utterances: We used dynamic hierarchical clustering [9] to group the observed shopkeeper utterances into clusters representing unique speech elements. 166 clusters were obtained.

Typical utterance extraction: From each shopkeeper speech cluster, one utterance was selected for use in behavior generation. We found that simply choosing the utterance closest to the centroid of the cluster was often problematic – sometimes this vector was not actually lexically similar to other utterances in the cluster and contained many errors, as shown in Fig. 4.

We instead choose the utterance with the highest level of lexical similarity to the most other utterances in the cluster, as this utterance would be the least likely to contain random errors. For each utterance, we compute the cosine similarity of its term frequency vector with every other utterance in the same cluster, and we sum these similarity values. The utterance with the highest similarity sum is chosen as the typical utterance.

2) Motion Clustering

In the abstraction of motion elements, our primary objectives are (1) to identify common stopping locations in the social space, so that we can discretize our representations of people’s motion in the joint state vector, and (2) to identify typical trajectory shapes so that we can estimate people’s motion targets. We do so by analyzing and clustering the motion data to characterize the overall sets of stopping locations and motion trajectories that exist in the data.

Using the approach described by Guéguen [42], we analyzed the distribution of trajectories in the data set and selected 0.55

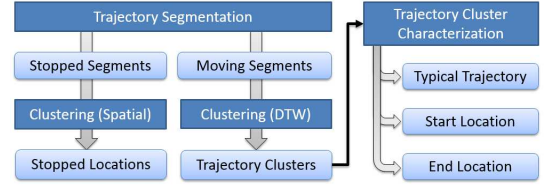


Fig. 5. The abstraction of motion elements into stopped locations and trajectory clusters

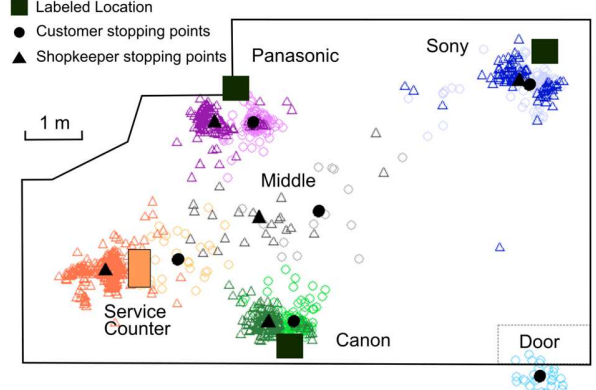


Fig. 6. Customer (O) and shopkeeper (Δ) stopped locations. Solid markers show the centroids of clusters of stopped segments which are marked as “stopped locations”. Customer and shopkeeper data are shown together for ease of comparison.

m/s as a threshold speed for trajectory segmentation. We then segmented all observed trajectories in the training data into “stopped” and “moving” segments, and clustered those segments to identify the typical **stopped locations** and **motion trajectories** present in the data set, as illustrated in Fig. 5.

Stopped location: The “stopped” segments were clustered spatially with k-means clustering to identify typical stopping locations, six for the customer and five for the shopkeeper. The centroid of each cluster was defined as a “stopped location”. Usually, these points corresponded to significant locations such as the cameras or service counter, so for ease of explanation we will refer to these points by the names shown in Fig. 6.

Trajectory clusters: We clustered the moving segments into 50 trajectory clusters, separately for shopkeeper and customer, using k-medoid clustering based on distances computed between trajectories using dynamic time warping (DTW).

The medoid trajectory for each cluster was designated as its “typical trajectory”, and the nearest stopped locations to the start and end points of that typical trajectory were identified. Fig. 7 shows some examples of the trajectory clusters.

3) Interaction States

We observed that the participants spent the majority of their time in a few static spatial formations, such as talking face-to-face or standing together at a camera. To capture this aspect of spatial behavior, we model each interaction as consisting of a series of interaction states characterized by common proxemic

TABLE II. DIMENSIONS FOR UTTERANCE VECTORS

	TF-IDF	LSA	
	Dimension	Dimensions	Instances
Customer Speech	7289	346	1194
Shopkeeper Speech	9181	353	1233

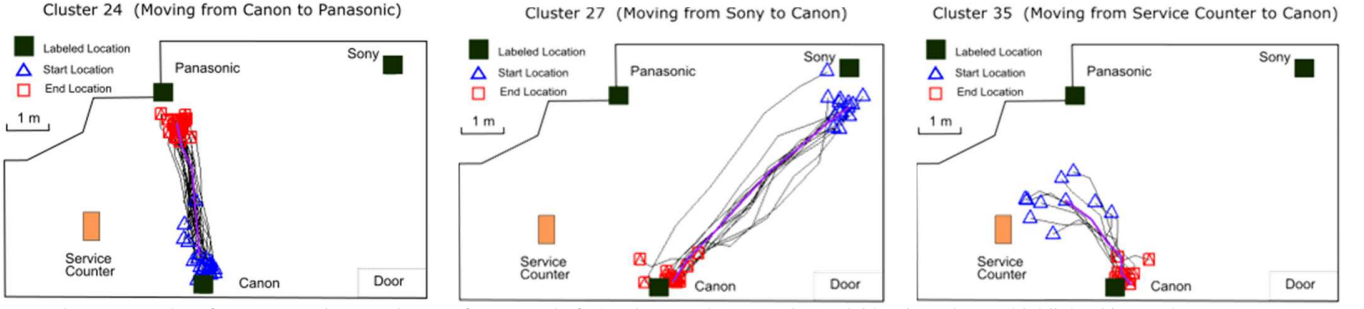


Fig. 7. Examples of customer trajectory clusters (from a total of 50 trajectory clusters). The medoid trajectories are highlighted in purple.

formations, such as talking face-to-face or presenting a product. The overall movement of the customer and shopkeeper can be seen as primarily serving as a means for transitioning between these interaction states. Fig. 8 presents example interaction state sequences observed in the training data.

HRI models have been developed for generating appropriate proxemics behavior in specific social situations such as initiating conversation [43] or presenting an object [44]. These models are useful abstractions, as they enable interaction states to be used not only to describe target destinations for movement, but also to specify proxemics constraints and other behavior at a detailed level for a robot.

In this work, we use three interaction states related to existing HRI models: *present object*, based on [44], *face-to-face*, based on interpersonal distance defined by Hall [45], and *waiting*, inspired by the modeling of socially-appropriate waiting behavior in [46]. Examples of these states are shown in Fig. 9.

We created rules for identifying each of these interaction states, based on the distance between the interactants and their locations. If both interactants were at stopping locations corresponding to the same camera, the interaction state was categorized as present object. If they were within 1.5m of each other but not at a camera, it was modeled as face to face, and if the shopkeeper was at the service counter while the customer was not, the interaction state was defined as waiting.

In addition, we also identified the current target for a particular interaction state. The *state target* for “present object” can be either Sony, Panasonic, or Canon, whereas the *state target* for the interaction states “face-to-face” and “waiting” is ‘none’.

C. Vectorization

When processing time-series sensor data for offline training or online interaction, these abstractions are used for creating vectorized representations of discrete customer and shopkeeper actions, as shown in Fig.10. First, **motion analysis** is performed

based on a comparison with typical trajectories. It is then possible to **discretize actions** based on detections of movement and speech. Each customer action is represented by a **joint state vector** describing the abstracted state of both participants at the time of that action, and each shopkeeper action is represented by a **robot action vector** containing the necessary information for a robot to reproduce that action later.

For all processes presented here, the sensor data is sampled at a constant rate of 1 Hz. Except where noted, the same techniques were applied to both the recorded training data and the live data from the online system.

1) Motion Analysis

We characterize a person’s motion using a vector containing three parameters: *current location*, *motion origin*, and *motion target*, corresponding to stopping locations from the clustering.

We identify whether a person is moving or stopped by applying the same speed threshold used in the offline trajectory analysis (Sec. IV.B.2). For stopped trajectories, *current location* is set to the nearest stopping location, and *motion origin* and *motion target* are “none”.

For moving trajectories, *current location* is “none” and *motion origin* is set to the most recent *current location*. For the customer, the *motion target* field must be estimated, although as we will explain, estimation is unnecessary for the shopkeeper.

Customer motion target: To estimate the customer’s motion target, we examine the similarity of the customer’s trajectory to the typical trajectories identified in clustering, similar to the approach used in [47]. We compute the spatiotemporal distance between the customer’s trajectory and each of the typical trajectories from the training data using DTW. The distance calculated for each trajectory cluster is then weighted according to the number of instances in that cluster, and probabilities are summed for trajectories that terminate at the same end location. The motion target is output once the probability of some result is above 50%, usually attained within 2-3 seconds.

Shopkeeper motion target: Estimation of the motion target through sensor data is unnecessary for the shopkeeper. Since we always know the robot’s target destination with certainty, based on the commands sent to the robot, the shopkeeper’s motion target in the training data should also reflect this knowledge of the intended motion target. In order to do so for the training data, we can determine the shopkeeper’s actual motion target at any time by looking ahead in time to observe their eventual destination, rather than relying on estimation

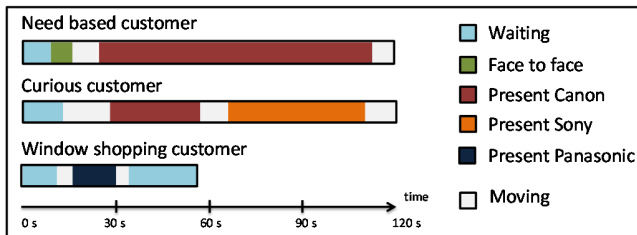
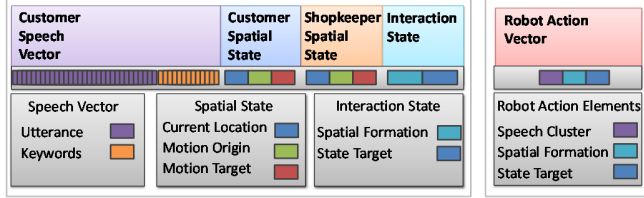


Fig. 8. Examples of sequences of interaction states from training data for the 3 customer scenarios: curious, need-based, and window-shopping



Fig. 9. Typical interaction states: (a) **Waiting**: One person is at a designated waiting area and interactants are not near each other, (b) **Face to face**: both people near and facing each other, but not near an object, (c) **Present object**: both people stopped near an object



(a) Features in joint state vector (b) Features in robot action vector
Fig. 10. Quantifying joint state vector and robot action vector

from the sensor data. By doing so, the shopkeeper motion target from the training data and from real-time data will be consistent.

2) Discretizing Actions

Discrete “customer actions” and “shopkeeper actions” are defined when one of the participants speaks and/or begins moving to a new location. Speech actions are defined at the moment the speech recognition result is received, and motion actions are defined at the moment a motion target is determined. Customer and shopkeeper events are received within the same 1-second interval are classified as two separate events, so no event can contain both customer and shopkeeper speech.

3) Joint state vector (Input)

When a customer action is detected, the state of both interactants is recorded in a *joint state vector*. This vector will be used for training the predictor to identify the most appropriate robot action to perform. The features in the joint state vector are shown in Fig. 10 (a). It includes the customer speech vector (including LSA vectors for both the utterance and keywords, 346 dimensions in total), customer and shopkeeper spatial states (each consisting of *current location*, *motion origin*, and *motion target*), and interaction state (*spatial formation* and *state target*).

4) Robot action vector (Output)

When a shopkeeper action is detected, it is represented in a *robot action vector*, which can be translated later into commands for the robot. In our case we are concerned with reproducing only speech and locomotion, so the robot action vector contains two properties: speech (consisting of a *speech cluster*) and interaction state (*spatial formation* and *state target*), as shown in Fig.10 (b).

Robot Speech: This field contains information to enable the robot to reproduce a shopkeeper utterance. It is only populated if the shopkeeper action contains a speech component; otherwise, it is left blank.

Definition: Directly using the raw text output from speech recognition is not appropriate for generating robot speech, because often it contains speech recognition errors. For this reason, we record the ID of the shopkeeper speech cluster containing the detected speech. For example, if the recognized

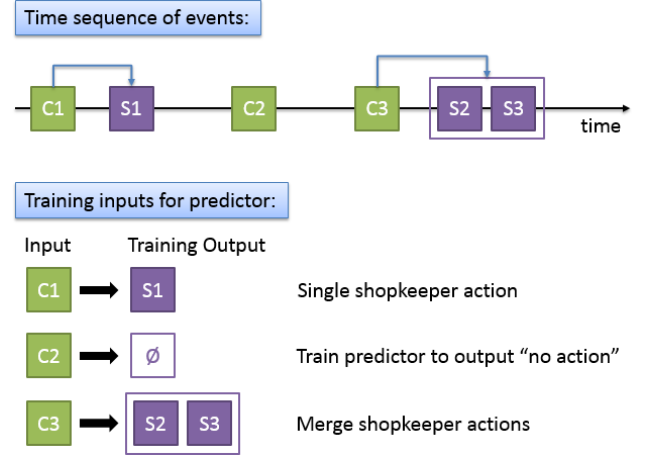


Fig. 11. Example time sequence of customer and shopkeeper actions.

utterance is “what does it has 28 different lenses”, cluster ID 292 would be chosen as the representative shopkeeper speech cluster, as illustrated in Fig. 4.

Generating robot behavior: As described in Sec. IV.B.1, a typical utterance is extracted from each shopkeeper speech cluster, which is expected to contain fewer random errors than a typical instance of recognized speech. To generate a robot speech behavior from a cluster ID, we use this typical utterance as the text to be sent to the robot’s speech synthesizer. In the above example, the chosen robot speech would be “there are 28 different interchangeable lenses available for this camera”.

Target Interaction State: Recall that the interaction state described in Sec. IV.B.3 encapsulates the proxemic formation of the two interactants at a given time. We can use this information to generate robot motion by recording the “target interaction state” of the shopkeeper.

Definition: If the shopkeeper is not moving at the time the action is detected, then the shopkeeper’s current interaction state is recorded. If the shopkeeper is moving, then we look ahead in time to determine the shopkeeper’s destination as described in Sec. IV.C.1. We then determine the “target interaction state” by evaluating the interactants’ spatial formation at the time when the shopkeeper arrives at the destination.

The interaction state is identified in the same way as described in Sec. IV.C.3, except that to accommodate the case where the shopkeeper is leading the customer and arrives first, we classify the target state as “present object” if either the customer’s *current location* or the customer’s *motion target* are the same object as the shopkeeper’s *current location*.

Generating robot behavior: Then, to generate a robot behavior in the online system we can simply compare the robot’s current location with the location necessary to achieve the target interaction state, and command it to move if necessary. For *waiting*, this target location will be the service counter; for *present object*, the target location will be the object of interest; and for *face-to-face*, the target location will not be a fixed location but rather a point in front of the customer. If the robot is not already at the target location, we command the robot to drive to a point near that location. The precise x, y position near the target location is determined by using the HRI

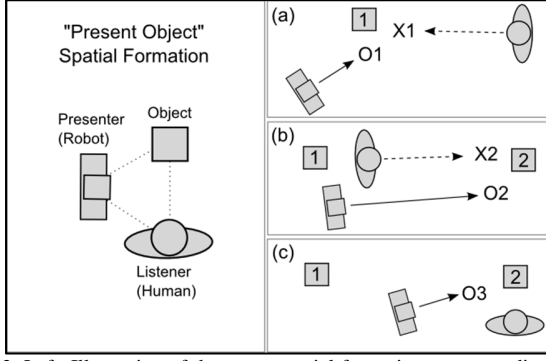


Fig. 12. Left: Illustration of the target spatial formation corresponding to the “present object” interaction state. Right: Examples of dynamic path planning to achieve the “present object” formation. “X” represents the projected future position of the robot, and “O” represents the calculated target position of the robot in response.

proxemics model associated with the target interaction state.

D. Learning and execution of interactive behaviors

To use machine learning to determine which robot behaviors should be performed in response to which human actions, we examine the discretized actions to **identify action pairs**, that is, sequential pairs of customer and shopkeeper actions, in the training data. For each action pair, we **train a predictor** using the joint state vector and robot action vector corresponding to the customer and shopkeeper actions. Finally, this predictor is used in the online phase to **generate robot behaviors** in response to detected customer actions.

1) Identifying Action Pairs

By examining the time sequence of detected actions (see Sec. IV.C.2), we identify correspondences between customer actions and subsequent shopkeeper actions. However, social interactions are not always cleanly divided into action-response pairs, e.g., when two customer actions or two shopkeeper actions occur in a row. Consecutive shopkeeper actions are combined according to a set of rules, and customer actions that are not followed by a shopkeeper action are associated with “no action” for purposes of training the predictor.

Fig. 11 shows an example time sequence of customer and shopkeeper actions. The first two, C1 and S1, illustrate the usual case of a customer action followed by a shopkeeper action, and these are paired as training inputs and outputs for the predictor. Customer action C2 is not followed by a shopkeeper action, so it is paired with “no action”. The third customer action is followed by two shopkeeper actions, which are then merged to produce a single shopkeeper action.

Recall that each robot action is comprised of an utterance (166 possibilities) and a target interaction state (5 possibilities). After merging shopkeeper actions, we translate each of the shopkeeper actions into a robot action vector, as described in Sec. IV.C.4. The final list of robot action vectors for our data set contained 467 distinct combinations of utterance and interaction state.

2) Modeling Delay

There is a natural delay time between customer actions and shopkeeper responses, and if the robot responds too quickly or too slowly, it is unnatural. To reproduce the delay time between

customer actions and responses from the shopkeeper, we calculated the average time delay between customer and shopkeeper actions from the training data corresponding to each robot action, and we constructed a lookup table mapping robot actions to average delay times.

For most robot actions, such as answering direct questions, the delay time was usually in the range of 0 - 2.5 seconds. For some behaviors longer pauses were observed. For example, when a customer entered and moved directly to the Sony camera while saying nothing, the system predicted that the robot should approach and offer assistance, after a delay of 17 seconds. If the customer performed another action during this time, the robot responded to that action. In this way, the robot was able to respond to long pauses which occurred, e.g., in the “window-shopping” scenarios.

3) Training the Predictor

Once all action pairs in the training data have been identified, we train a naïve Bayesian classifier, using the joint state vector for each customer action as a training input and the subsequent robot action vector corresponding to the shopkeeper action as its training class.

The naïve-Bayesian classifier is a generative classification technique, which uses the formula below to classify an instance that consists of a set of feature-value pairs.

$$a_{NB} = \arg \max_{a_j \in C} P(a_j) \prod_i P(f_i = v_i | a_j) \quad (1)$$

a_j , denotes a robot action, and f_i denotes a feature in the joint state vector. The naïve-Bayesian classifier picks a robot action, a_{NB} , that maximizes the probability of being classified to the robot action given the value v_i for each feature f_i .

Each feature f_i in the joint state vector is multidimensional, consisting of a set of terms t_{ik} . For example, the customer speech vector has 346 dimensions, whereas the customer spatial state only has 21 dimensions. Thus, we can rewrite the classifier equation to consider the partial matches between the values for each feature, as in Eq. (2), where the conditional probability of each term of each feature, given a robot action a_j , is computed in the training phase:

$$v_i = \{t_{i1}, t_{i2}, \dots, t_{im}\}$$

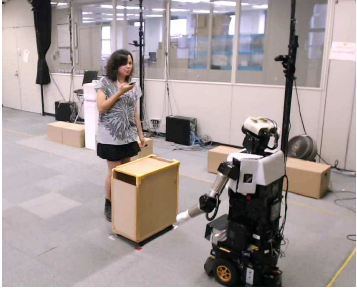
$$a_{NB} = \arg \max_{a_j \in C} P(a_j) \prod_i (\prod_k P(t_{ik} \text{ appears in } f_i | a_j))^{w_i} \quad (2)$$

We would like to give higher priority to values in the features that are more discriminative in classifying robot action. Gain ratio tells us how important a given feature in the joint state vector is. Therefore, w_i , calculated from the gain ratio of each feature, is added as the weighting factor for the classifier.

4) Generating Robot Behaviors

During live interaction between a human customer and the robot shopkeeper, the sensor network records the customer’s motion and speech at one-second intervals.

When a customer action is detected, we query the trained naïve Bayesian predictor, passing in the joint state vector corresponding to the social state at that time. The predictor will then output either the ID of one of the 467 robot actions, or it will predict “no action”. If a robot action is specified, the system waits for the time specified in the delay table corresponding to that action, and then commands are sent to the robot to move to



[Joint state vector]	
Customer Speech Vector	Utterance Vector: LSA vector representing “hello I’m looking for a camera that has interchangeable lenses do you have any?” Keyword Vector: LSA vector representing “camera, interchangeable lenses”
Customer Spatial State	Current location: Service Counter Motion Origin: None Motion Target: None
Shopkeeper Spatial State	Current location: Service Counter Motion Origin: None Motion Target: None
Interaction State	Spatial Formation: Face-to-face State Target: None
[Predicted robot action]	
Robot Speech	Speech Cluster ID: 170 (Typical utterance: “over here we have my favorite which is the Sony NEX 5 which is a mini SLR and has 28 replaceable lens.”)
Target Interaction State	Spatial Formation: Present object State Target: Sony

Fig. 13. Example of predictions in a live interaction.

a destination or speak an utterance.

When the robot action includes an interaction state of “*present object*” or “*face-to-face*”, the precise target position is computed according to that formation’s proxemics model. While in motion, the robot projects the future position of the customer and recalculates a target location according to the proxemics model every second until it arrives. Some examples of this calculation are illustrated in Fig. 12.

In this example, the first target interaction state is “Present Camera 1”, shown in Fig. 12 (a). The robot projects the customer’s destination to be X1, so it computes a target destination to point O1. The next target interaction state is “Present Camera 2”. In Fig. 12 (b), the robot first projects the customer to be moving towards X2, so it begins moving towards point O2. However, in Fig. 12 (c), the customer chooses to move to a different location than predicted. The robot dynamically updates its path to move to point O3.

E. Example of Behavior Execution

Fig. 13 shows an example of a prediction from a live interaction with a robot. In this example, the customer approaching the shopkeeper at the service counter is detected as a customer action, and the predictor is queried with the joint state vector shown in the figure. The predicted robot action consists of an utterance with cluster ID 170 paired with an interaction state, “Present Sony”. The recorded delay time corresponding to “170-Present Sony” action is 2.75 seconds, so the system waits for that duration before executing an action. Because the current interaction state is “waiting” and the target interaction state is “Present Sony”, the robot starts moving to Sony. A speech command is sent to the robot containing the typical utterance from the selected speech cluster, which in this case causes the robot to speak, “over here we have my favorite which is the Sony NEX 5 which is a mini SLR and has 28 replaceable lens”.

V. EVALUATION EXPERIMENT

We conducted a comparison experiment to evaluate the

quality of the robot’s behavior in live interactions. Because we consider the proposed abstraction technique to be the main contribution which makes it possible to learn interactive behaviors despite high sensor noise, we compared two conditions: (a) *proposed*, using the abstraction techniques including clustering and interaction states described in Sec. IV, and (b) *without-abstraction*, a similar technique we developed that does not use our abstraction techniques.

A. Comparison system

We designed the *without-abstraction* system to be similar to other state-of-the-art data-driven techniques for generating interactive robot behaviors. For example, Admoni et al. [48] developed a system that matches observed data in real-time to the nearest example from human-human training data to select a robot behavior, following the idea that people learn to communicate by mimicking observed behavior in a given situation.

Thus, we created a modified version of our system which also uses the observed sensor data in real-time to find the most similar example from the training data. If our data were not susceptible to noise, the behavior generated by the *without-abstraction* system would have represented exactly what a human shopkeeper had done in a similar situation. The differences between the *proposed* and *without-abstraction* systems are described here and summarized in Table 3.

Speech elements: Speech is captured and processed using the same standard text processing techniques in both systems. However, no clustering is performed on the shopkeeper’s speech in the *without-abstraction* system, so shopkeeper utterances must be generated directly from the raw speech recognition results captured in the training data. Keyword extraction is also not used in the *without-abstraction* system, because its purpose is to assist with clustering of shopkeeper speech.

Motion elements: Our proposed technique uses the results from trajectory clustering to define stopping locations and to anticipate a person’s motion target. For the *without-abstraction* system, a person’s stopping location is represented by their raw

TABLE III. DIFFERENCES BETWEEN PROPOSED SYSTEM AND WITHOUT-ABSTRACTION SYSTEM

	Proposed system	Without-abstraction system
Clustering	<ul style="list-style-type: none"> Cluster shopkeeper speech Cluster motion data 	<ul style="list-style-type: none"> No clustering
Vectorization	<ul style="list-style-type: none"> Motion target prediction based on trajectory clusters Abstracted locations Interaction states used 	<ul style="list-style-type: none"> Motion target prediction based on mean motion direction Raw position data No interaction states
Predictor	<ul style="list-style-type: none"> Naïve-Bayesian predictor to select an abstracted action 	<ul style="list-style-type: none"> Nearest neighbor-predictor to select an instance to reproduce
Robot action generation	<ul style="list-style-type: none"> Motions generated based on target interaction state Utterances generated from shopkeeper speech clusters 	<ul style="list-style-type: none"> Motion generated directly from a shopkeeper motion instance Utterances generated directly from a shopkeeper speech instance

x, y position, rather than the nearest stopping point cluster. When moving, a person’s motion target is estimated based on their motion direction, rather than using our technique of comparison to the trajectory clusters. Finally, the set of possible motion targets is defined manually for the *without-abstraction* system, rather than using clustering results (we defined five points: the three cameras, the door, and the service counter).

To estimate a person’s motion target, the person’s mean motion direction θ_{motion_dir} is calculated over the last 3 seconds, and the *motion target* is calculated as the x, y position of the nearest object to the mean motion direction from their position in the environment.

$$\text{motion target} = \arg \min(\theta_{motion_dir} - \theta_{obj_n} : obj_n \in \text{all objects in environment}) \quad (3)$$

Feature vector: The stopping locations identified in the clustering phase are not available in the *without-abstraction* system, so feature vectors include the following 5 features: the customer’s and shopkeeper’s current x, y coordinates, the customer’s projected x, y motion target, the shopkeeper’s actual x, y motion target, and the LSA vector representation of the customer’s speech. Interaction state was not included in the feature vector for the *without-abstraction* system.

Prediction: The predictor from the *proposed* system cannot be used in the *without-abstraction* system – since shopkeeper utterances are not clustered, there is no set of discrete robot actions to be trained. Instead, we created a “nearest-neighbor predictor” – whenever a customer action is detected, the current raw feature vector is compared to the feature vectors from all customer actions in the training data. The best match is identified, and the subsequent shopkeeper action from the training data is returned as a robot action. Robot actions in this case have two properties: motion target (if moving), and utterance text (if speaking). A lookup table for delay time between the customer and shopkeeper actions was also created in the same way as the proposed system.

For our dataset, the set of customer action vectors consisted of 1636 entries in 330 dimensions, so a k-d tree [49], was used to speed up the nearest-neighbor comparisons.

Robot behavior generation: Robot behaviors are generated directly from the specific instance of shopkeeper behavior output by the nearest-neighbor predictor. For movement, the robot moves directly towards the x, y position where the shopkeeper had moved to in the matched instance, instead of using interaction state to generate the target. For speech, the robot speaks the exact phrase captured by speech recognition in the matched instance.

B. Hypotheses

In the comparison experiment, we made the following hypotheses about the effects of our abstraction techniques (clustering and modeling of interaction states) in the *proposed* system, compared with the *without-abstraction* system:

Speech clustering: Clustering of shopkeeper utterances will produce more correct utterance behaviors in the robot, because the act of clustering and our technique for typical utterance extraction will reduce the effect of noise in the captured utterances.

Stopping point clustering: Representing spatial locations based on abstracted stopping point clusters, rather than as raw positions, will lead to more efficient learning through abstraction. This will also be more robust to sensor noise, since the influence of noise is incorporated in the clustering step.

Trajectory clustering: Estimation of motion target will be more accurate when similarity to clustered trajectories is used, compared with raw extrapolation of velocity. This will lead to more appropriate responses to customer motion from the robot.

Interaction states: The modeling of movement in terms of transitions between long-term-stable interaction states will result in more reliable locomotion behaviors than reproducing individual movement events.

Based on these hypotheses, we chose to test the following predictions for the comparison between the *proposed* system and the *without-abstraction* system:

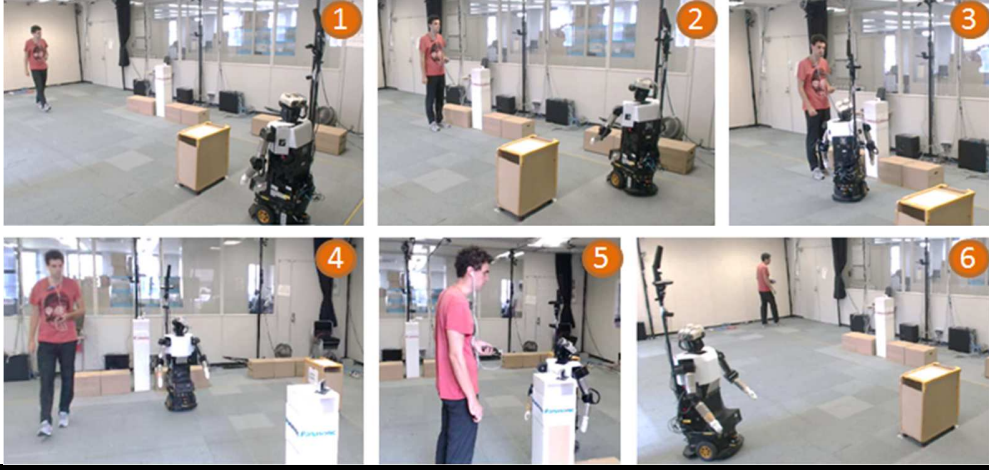
- **Correctness of wording:** The robot will produce more correct wording in the *proposed* system.
- **Consistency between speech and movement:** The robot’s speech and movement will be more consistent with each other in the *proposed* system.
- **Appropriateness of robot actions:** The robot will respond more appropriately to the customer’s actions in the *proposed* system.
- **Social-appropriateness:** The robot’s behaviors will be more socially-appropriate for its role as the shopkeeper in the *proposed* system.
- **Overall evaluation:** The overall evaluation of the robot’s behaviors will be better in the *proposed* system.
- **Robustness:** The *proposed* system will be more effective at generating appropriate robot behaviors even when recognition errors occur.

C. Experiment Setup

1) Participation

A total of 17 paid participants (11 male and 6 female, average age 34.42, s.d. 13.30) played the role of customer in the

TABLE IV. AN EXAMPLE OF THE ROBOT INTERACTING WITH THE CUSTOMER.



- | | |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | Customer walks into the shop
Robot greets the customer with “ <i>hi can I help you with anything</i> ” at service counter |
| (2) | Customer stops at Canon, and says “ <i>yes I’m looking for a camera with large memory storage</i> ”
Robot approaches customer at Canon while saying “ <i>yes we have Canon Rebel XT<i>i</i> I over here this camera has a very large storage memory it can store about 10000 photos</i> ” |
| (3) | Customer : “ <i>how much is it?</i> ”
Robot : “ <i>this is \$400</i> ”
Customer : “ <i>and what about the battery life?</i> ”
Robot : “ <i>7 hours</i> ”
[The robot answers a few more questions about Canon (e.g. color, weight)] |
| (4) | Customer walks to Panasonic
Robot follows the customer to Panasonic |
| (5) | Customer : “ <i>what is the LCD size?</i> ”
Robot : “ <i>a 3 inch touch screen</i> ”
Customer : “ <i>that sounds nice. I like it.</i> ”
Robot : “ <i>also this is very light only weighs 150 grams so you can fit right in your pocket</i> ”
[The robot answers a few more questions about Panasonic (e.g. color, optical zoom)] |
| (6) | Customer says “ <i>Thank you for your help. I will think about it.</i> ” and leaves the shop
Robot returns to the service counter while saying “ <i>no problem</i> ” |

experiments. All of them were fluent English speakers (9 North and South Americans, 7 Europeans, 1 Russian).

2) Environment

The experiment was conducted in the same camera shop setting used for the data collection, with three digital cameras displayed in an 8m x 11m experiment space. The same sensor network was used for tracking, and the participants communicated with the robot using an Android phone.

3) Robot Platform

For this experiment, we used Robovie 2, a humanoid robot with a 3-Degree-of-Freedom (DOF) head, two 4-DOF arms, a wheeled base, and a speaker that can output synthesized utterances.

Robovie is capable of moving at a speed of 0.7 m/s. For its motion planning, the dynamic window approach (DWA) was implemented to avoid obstacles [50].

Implicit behaviors were implemented into the robot, where the robot makes small arm and head movements while idling, speaking, and moving [43]. Automatic face-tracking of robot’s interaction partner was also implemented, and the robot followed the customer with its gaze during all interactions.

4) Procedure

We compared the robot’s performance between two conditions: *proposed* and *without-abstraction*, and each participant was asked to role-play for 8 trials in each condition.

As in our data collection, participants played each of the following roles: a need-based customer (3 trials), a curious customer (3 trials), and a window-shopping customer (2 trials). The order of the conditions was counterbalanced and the order of the trials within each condition was randomized.

As in our data collection, participants were asked to pretend to be a first-time customer in the camera shop for every trial and the participants performed scripted interactions before the experiment to become familiar with the Android phone interface and confirm their understanding of the instructions.

After the 8 trials in one condition were completed, the participant answered a questionnaire. The procedure was repeated with the remaining condition (*without-abstraction* or *proposed*). At the end of the experiment, the participants were interviewed to gain a deeper understanding of their opinions.

Examples of interactions from the experiment using the *proposed* system can be seen in the video attachment.

D. Measurement

1) Questionnaire

The participant rated the following items on a 1-7 scale (1 being very negative and 7 being very positive for the respective items) in a written questionnaire:

- Correctness of the wording of the robot’s utterance:
- Consistency of the robot’s speech and movement

- Appropriateness of the robot’s response to the participant’s action
- Social appropriateness of the robot’s behaviors as its role as the shopkeeper
- Overall evaluation

In the experiment, the robot may give an answer to the customer’s question that makes sense, but may not necessarily be accurate. For example, if the customer asks “how much is this camera”, the robot may respond with “\$600” instead of the correct answer, “\$300”. Because knowledge of these errors could affect the participant’s evaluation of the robot, we informed participants about any informational errors the robot made before they filled out the questionnaire in each condition.

2) Interaction analysis

For the robustness evaluation, we conducted a detailed action-by-action analysis of the robot’s behavior by asking a coder, blind to the experimental conditions, to examine each action (speech or movement) made by the participant, and to judge whether the robot’s response to that action was appropriate. The coder was shown examples of acceptable and unacceptable behavior in order to calibrate expectations. Examples of unacceptable behavior included answering a question incorrectly, or failing to guide a customer to a camera when asked to do so. From this evaluation, we calculated *behavior correctness* for each condition, for each participant.

A separate evaluator examined all of the customer speech events in each trial and recorded the number of correct and incorrect speech recognition results. We defined *ASR correctness* by whether the sentence-level meaning of the ASR result was understandable or not. Though some ASR results contained word errors, they were judged as “correct” if the utterance itself was still understandable on a sentence-level. For example, given that the customer said “thanks a lot”, the ASR result “thanks a lots” would be considered correct, whereas “insulet” would be considered as incorrect. Further analysis of the speech recognition accuracy can be found in the Appendix.

The *ASR correctness* was then compared with the *behavior correctness* to evaluate the robustness of the behavior generation technique to recognition noise.

E. Results

1) Observations

It was quite fun for us to watch the robot acting autonomously – since the learned rules created some interesting variations of behavior, we never knew exactly what how the robot would respond to any situation. Most of the robot’s behaviors were executed well - the robot was able to move with the customer to appropriate locations and answer most questions correctly. Although it did make some errors, it was often able to recover and continue the interaction. Many of the participants commented that they really enjoyed the interactions. Table 4 shows an interaction example from the experiment.

If the customer was looking for a particular camera feature (e.g. interchangeable lens), the robot usually responded correctly, guiding them to a camera with that feature and introducing the camera. The robot also answered most questions about camera features correctly, even though the

customers asked in different ways. For example, one customer asked, “this one comes in red, right?”, and another customer asked “what color do you have for this?”, and the robot was able to answer appropriately by saying “we have red and silver available” to both customers. Likewise, the robot correctly gave the weight of the camera in response to “how much does this camera weigh?” and “excuse me is this camera heavy?”

Sometimes the robot responded correctly despite speech recognition errors. The robot gave correct answers to questions such as the following (correct phrasing in brackets): “Amanda [um, and uh,] how much does it weigh?”, “I’m sorry does this camera have optimism [optical zoom]?” “How many car what color is coming? [how many, er, what colors does this come in?]”, “Skewes me [excuse me] what color does a scammer [this camera] come in?”, “how much does a camel [this camera] weigh?”, and “I say in the is it a popular vote [I see, and uh, is it a popular model?]”. Many recognition errors were fairly common, such as “scammer” or “camel” for “camera”, and “OCD” for “LCD”, and the system appears to have learned to treat these words as synonyms.

Sometimes it failed to respond correctly due to speech recognition errors. For example, when a customer asked “could you tell me how much this Lumix costs?”, the word “Lumix” was recognized as “LINE X”, and the robot responded, “yes, sir.” Then the customer rephrased his question, “could you tell me how much this is?” and the robot answered correctly. When a customer repeated or rephrased their question, the robot usually responded correctly the second time.

The robot’s utterances sometimes contained minor errors, as can be seen in the example in Table 4, although some of these mistakes sounded phonetically correct. For example, the robot sometimes said “my I help you?” when the customer entered the shop, yet none of the customers noticed the mistake.

The robot was also able to respond to the customer’s motion – when a customer entered and immediately approached the service desk, the robot would greet them immediately, whereas if they walked to one of the cameras first, it would often let them browse for a while before speaking.

When the customer thanked the shopkeeper and left, the robot would respond with phrases such as “no problem” or “you are welcome” and returned back to the service counter. In cases when the customer left the shop without talking to the shopkeeper (i.e. a window-shopping customer), the robot thanked the customer for visiting the shop. Three participants commented that the robot was polite in greeting and saying goodbye.

The robot was usually able to move together with the customer or follow them to a camera, and two participants responded that they liked the fact that the robot followed them to different cameras. Occasionally it misinterpreted a person’s motion and moved to the wrong camera, but in such cases it usually corrected itself in the following action. If the customer asked a question about a camera while the robot was in another place, it usually moved to the customer’s location while answering the question, in order to reconstruct the target interaction state learned during training.

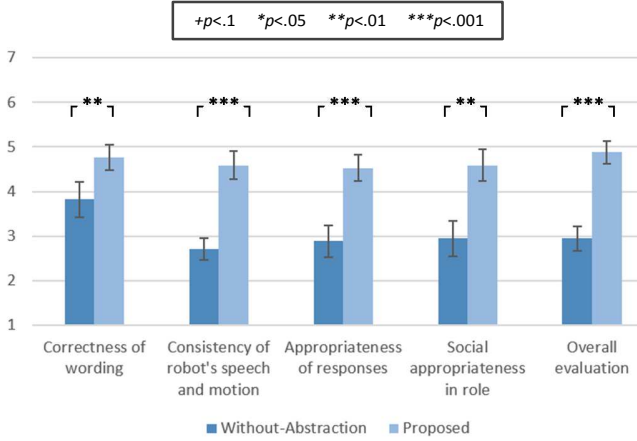


Fig. 14. Evaluation results of robot behaviors between conditions

The use of proxemics models based on the interaction state to control the robot's positioning relative to the customer also seemed to work effectively. Two participants commented that the positioning of the robot was very good, and the robot had a good idea of personal space.

2) Questionnaire

Fig. 14 shows questionnaire results from the participants. To compare each rating between the *proposed* condition and the *without-abstraction* condition, we conducted a repeated-measures ANOVA for each of the five questions.

This analysis found significant differences between the conditions for all ratings: "Correctness of wording" ($F(1,16)=9.660, p=.007$), "Consistency of robot's speech and motion" ($F(1,16)=26.947, p<.001$), "Appropriateness of responses" ($F(1,16)=20.564, p<.001$), "Social appropriateness in role" ($F(1,16)=14.222, p=.002$), and "Overall evaluation" ($F(1,16)=48.944, p<.001$).

These results support our hypothesis that the participant would perceive the overall behavior to be better with our proposed system. The results also support our predictions for the correctness of the wording, consistency in the robot's speech and motion, appropriateness of responses to the customer's actions, and the social appropriateness of the robot in its role as the shopkeeper.

3) Interaction Analysis

The results of the interaction analysis are shown in Fig. 15. We conducted a repeated-measures ANOVA comparing *behavior correctness* between the *proposed* and *without-abstraction* conditions. The results showed *behavior correctness* to be significantly higher in the *proposed* condition ($F(1,16)=97.507, p<.001$). This result further supports our hypothesis regarding appropriateness of responses to the customer's actions.

As some of the appropriateness judgments are subjective, we confirmed the consistency of the coder's evaluations by asking a second coder to independently rate 10% of the same interactions. Their results were compared, and a Cohen's Kappa value of 0.76 was calculated, indicating good interrater reliability, so we consider the coder's ratings to have consistency. Next, we compared *behavior correctness* and *ASR correctness* for each condition with a repeated-measures ANOVA. In the *proposed* condition, the *behavior correctness*

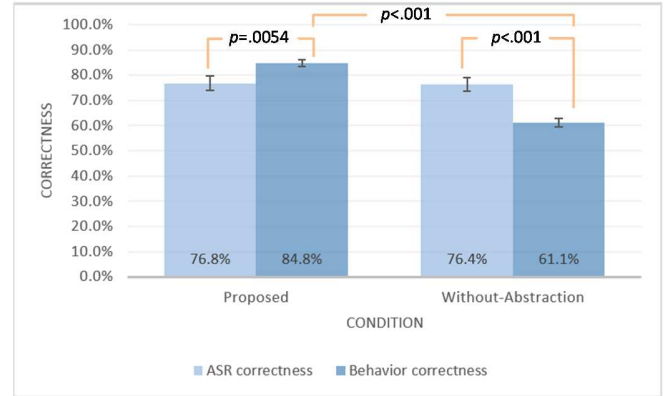


Fig. 15. Comparison of ASR correctness and robot behavior correctness.

was significantly higher than *ASR correctness* ($F(1,16)=10.669, p=.0054$). In the *without-abstraction* condition, the *behavior correctness* was significantly lower than *ASR correctness* ($F(1,16)=30.356, p<.001$). Incidentally, no significant difference was found in *ASR correctness* between conditions ($F(1,16)=.035, p=.854$).

These results confirm our hypothesis that behavior generation in the *proposed* condition is more robust to recognition errors than in the *without-abstraction* condition. We consider this to be an important result, as recognition errors and sensor noise constitute some of the major challenges to data-driven interaction design.

4) Qualitative Analysis

To better understand the nature of our system's performance, we investigated the specific causes of behavior incorrectness. Thus, of the total 1281 robot behaviors observed in the *proposed* system, we analyzed the 201 robot behaviors that were judged as incorrect by the coder. In Table 5, we present a qualitative analysis of the errors observed in our *proposed* system, including the possible causes for socially-inappropriate robot behaviors, examples of these errors, and their frequency of occurrence. The results are derived from open-coding and observation from video data and participant feedback in the evaluation experiment. The possible causes are:

Lack of repeatability: Some customer behaviors in the human-human interaction were either only observed once or not observed at all in the training data, thus it was difficult for the robot to learn to behave well. Questions such as comparison between two cameras did not often occur in our training data. For this reason, we could not collect enough examples to train the robot well to answer such questions. The robot sometimes answered these questions correctly, but it was usually a pleasant surprise when it did.

We believe that the performance of the system will improve if more data can be collected, and would help the robot answer questions such as comparison between two cameras.

Error in ASR: Certain ASR errors would trigger the robot to behave inappropriately, e.g. when an entire sentence gets misrecognized (i.e. "it's expensive" as "sixpence") or when a word about the camera feature gets misrecognized (i.e. "yes how many colors does this camera come in" as "how many calories does this camera come in").

TABLE V. COMMON CAUSES FOR BEHAVIOR INCORRECTNESS OF THE ROBOT IN THE PROPOSED SYSTEM

Causes	Examples	Freq.
Lack of repeatability	• A customer compares current camera with another camera (e.g. "so is this one better than the Sony camera?")	54
Error in ASR	• Misrecognized customer's utterance (e.g. "yes how many colors does this camera come in" misrecognized as "how many calories does this camera come in")	44
Lack of history representation	• A customer already indicated wanting to be left alone, yet Robovie sometimes offered to help several times in a row	23
Error in motion target estimation	• A customer says "that's great", and Robovie repeats an utterance that had already been said previously	17
Error in "farewell" behavior	• Robovie mistakenly estimates the customer to be leaving the shop, when the customer is not planning to leave yet	10
Ambiguous shopkeeper behavior	• When a customer leaves the shop, the robot returns to service counter without saying farewell (i.e. does not say "thank you for coming")	9
Embodiment	• Robovie addresses the customer with "yes sir"	7
Error in timing	• A customer asks the robot a question from across the room (in the training data, most customers first said "excuse me" to the human shopkeeper in order to call them over, before asking a product-related question)	7
Unexpected customer behavior	• A customer says something new before waiting to hear Robovie's response	6
Miscommunication	• A customer asks about something outside the scenario scope, to which the robot has not learned a response.	5
Missing information	• A customer asks for clarification, such as "Can you repeat that please?" or "did you say 10000 photos"	4
Error due to speech clustering	• A customer asks "Yes how much is this camera over there?" while pointing to or gazing to another camera	1
Other	• Robovie responds with "yeah mazzy s ball night hours 515"	14
	• The robot failed to respond due to hardware or operational errors (e.g., network failures, customer forgets to press button on smartphone after speaking)	

Since our system was trained with real ASR data, it was usually robust to ASR errors. However, some ASR errors were more frequent than others. For example, ASR misrecognized "color" as "kara" on several occasions, but only misrecognized "color" as "calories" on one occasion. In this case, when the customer asked about the camera's color, the robot would respond correctly to the misrecognized word "kara", but not to the misrecognized word "calories".

Lack of history representation: The lack of interaction history modeling sometimes caused Robovie to repeat himself. Sometimes, when a window-shopping customer asked to be left alone, Robovie would respond with "no problem" and continue letting the customer browse, but since the system contained no long-term history, Robovie sometimes offered to help several times in a row. Though such cases were observed quite a few times (i.e. 15 times), participants did not seem to be mind at all. In fact, one participant thought the robot was being a very eager shopkeeper.

In another example illustrating lack of history representation, a customer asked "What about the Canon camera?" just after asking about the color of the Sony camera. Robovie could not answer correctly, since such question is implicitly referring to the previous question. This exchange is quite complicated but only happened once in the evaluation.

Error in motion target estimation: Sometimes the system would misrecognize the motion target of the customer. When the robot misrecognized the customer's motion target as the door (i.e. leaving the shop), the robot might say "thank you for coming" even when the customer was not planning to leave the shop yet. Sometimes when the robot misinterpreted the customer's motion, it would move to the wrong camera, but in such cases it usually corrected itself in the following action.

Sometimes it may be difficult for the robot to estimate the customer's motion target. For example, as the customer enters the shop, it is unclear whether the customer is going to Canon or to the service counter. However, regardless of whether the robot is able to correctly estimate the customer's motion target, it would still greet the customer appropriately since it learned

the customer's motion origin is more important than its motion target.

Error in "farewell" behavior: In most interactions, the robot acknowledged the customer leaving the shop, e.g. by saying, "thank you for coming." However, in a small percentage of cases, the robot said nothing when the customer left. We speculate that such behavior was learnt from a variety of situations where the human shopkeeper did not verbally acknowledge the customer, e.g., the shopkeeper had already said goodbye, but the customer continued to browse around before leaving; the shopkeeper smiled or nodded to the leaving customer instead of a verbal farewell; or the shopkeeper recognized that a window-shopping customer wanted to be left alone and thus did not verbally acknowledge the leaving customer. As a result, sometimes the robot would not acknowledge or say anything to a leaving customer, but would just return to the service counter.

Ambiguous shopkeeper behavior: There were few instances that it was ambiguous whether the robot behavior was actually right. For example, some human shopkeepers would use phrases like "yes sir". Since we did not track the gender of the customer, the robot would learn such phrases, despite whether the customer was female or male.

Embodiment: An interesting phenomenon is that customers sometimes acted differently towards the robot than they did towards the human shopkeeper. In the training data, when the human shopkeeper was waiting by the service counter, the customer would usually say "excuse me" first to call the shopkeeper over before asking a question. In the evaluation, customers often asked a question to the robot directly, even from across the room (perhaps because they were speaking to it through the smartphone). These combinations of spatial state and utterance were not observed in the human-human interaction data, so the robot sometimes did not always respond in an acceptable way. For example, it often approached the customer, but did not answer the respective question.

Error in timing: Turn-taking is a notoriously difficult problem, and sometimes the customer and robot would speak at

the same time. Once a customer action (utterance) is detected, the robot may be triggered to take an action. If the customer speaks again without waiting for the robot to respond, the robot sometimes interrupts the customer while he is speaking.

Unexpected customer behavior: A customer may ask a question outside the scenario scope, such as a feature that has not been defined for that camera. Since there are no training examples to handle these questions, the classifier would usually choose the most talked-about feature of that camera as the output behavior for the robot. In our scenario, the robot usually responded with the price of the camera if the customer asked about a non-existent feature.

Miscommunication: There were some situations where a customer asked the robot to repeat its utterance, and the robot was unable to do so. Most of the time, the robot spoke understandable and correct utterances, but some customers just wanted a confirmation. In some instances, Robovie would synthesize its speech in a very robotic way (i.e. “10000 photos” synthesized as “one zero zero zero zero photos”), and some customers wanted the robot to repeat for clarification, a situation for which no examples existed in the training data.

Missing information: Sometimes the customer may stand at one camera and ask about a feature of a different camera (e.g. “what about the price of that camera?”), while gazing or pointing to the referred camera. Since the robot does not know where “that” is, it would often answer with the price of the camera at the customer’s current location. If reliable sensing of gaze direction and pointing gestures were available, it might be possible to address this problem by representing that multimodal information in the feature vector.

Error due to speech clustering: Some clusters were too noisy to produce sensible speech. For example, the speech cluster ID 179 contains 3 shopkeeper utterances, which are all very dissimilar from each other and nonsensical. As a result of this bad cluster, the typical utterance chosen was “yeah Mazzy s ball night hours 515”. However, such instances were rare, and we only found one instance where such a cluster was chosen.

Other: Sometimes the robot may fail to respond appropriately or not respond at all due to errors in any of these problems: network connectivity between Google Speech Recognition engine and our system, hardware, software bugs, or the participant forgets to press the button on the Android phone to signal the robot that they started or stopped talking.

VI. DISCUSSION

A. Contribution

In this study, we showed a proof of concept that a purely data-driven approach can be used to reproduce social interactive behaviors with a robot based on example human-human interactions. We demonstrated that by collecting interaction data including natural variation in human behaviors and typical recognition errors, the clustering of the participants’ motion and speech, enabled the robot to respond in a natural way to such variations. We saw the robot respond appropriately when people with different speech styles or accents interacted with the robot. This could be an advantage of our approach over grammar-based speech systems, which would have difficulty extracting the meaning from speech recognition results

containing errors.

By learning from natural human behaviors, the robot learnt lifelike variation in its behaviors. Explicitly programming multiple phrasings of utterances requires time and effort, but our system implicitly learned to use a variety of synonymous phrases, such as “yes it’s very good in low light” and “and if you like to shoot in the dark this is really good”, which can help keep interactions interesting and lifelike.

Another merit is that our system naturally learned when speech was location-specific or generalizable to different locations. For example, “Show me a camera with good optical zoom” has the same meaning regardless of where it is spoken, whereas “How much does this cost?” is highly dependent upon the current interaction state target, as each camera is a different price. The robot was able to derive probabilistically how to handle these situations correctly.

The robot learned to mimic the interaction styles of the shopkeepers, such as the casual nature of their speech. We noticed one human shopkeeper in our training interactions spoke quite casually (e.g. “okay find me if you want”) and used slang words (e.g. “600 bucks”) at times. As a result, the robot learned to mimic that casual speech for some interactions. Likewise, we asked the human shopkeeper to appear busy and only approach the customer when appropriate. As a result, the robot adapted to a more passive interaction behavior, and waited at the service counter when the customer entered the shop. It could be interesting to explore further how the differences in personality, interaction style, and other personal traits can be modeled and captured from data.

B. Validation of the Model

We believe evaluating how appropriate the robot’s action was (i.e. *behavior correctness*) was more important than evaluating how accurately the model was able to exactly replicate a specific example from the training data. Nevertheless, as a reference to understanding the nature of the system, we evaluated the accuracy of our predictor with a 10-fold cross-validation, in which the model predicted a robot action vector out of 467 possible actions from the training examples, and the predicted robot vectors were compared with the actual state vectors of the shopkeeper actions from the training data. The average accuracy was 26.0%.

Even though the predictor indicates a low accuracy, it often predicts socially-appropriate behaviors. One reason for this is that, as a result from clustering, similar shopkeeper’s actions can be clustered into different groups even when they have the same meaning or are interchangeable. For example, shopkeeper behaviors at the Panasonic camera saying “5X optical zoom” and “it has 5 times optical zoom” had the same meaning, but they were respectively clustered into cluster ID 253 and cluster ID 183. When a customer asked “how much optical zoom does this have” the predictor would output 253, while a customer asking “can you tell me about the optical zoom?” predicted cluster 183, even though either cluster would be a correct and socially-appropriate response to either question.

C. Assumptions

There are a number of assumptions implicit in our system design. For example, we assumed that this is a one-on-one

interaction where each customer action is followed (optionally) by a shopkeeper's action. We also specified some parameters for our scenario (i.e. number of speech clusters, location of products, number of discretized states), which are needed to tune machine-learning techniques. These problems are not unique to our scenario, as thresholds must be chosen for clustering to work in any problem space, and a finite number of states must be specified to discretize continuous sensor data. We have not yet discovered a good mechanism for choosing these parameters in an automated way for our technique.

We used spatial formations to define 'interaction states' for our scenario. We believe the concept of spatial formation is generalizable, and can be applied to other domains as well. The spatial formations we used are common proxemics formations that characterize the relative positioning between different entities, which have also been adapted into existing HRI models.

D. Generalizability and Scalability

We believe that this data-driven approach is capable of covering a wide domain of tasks. We can expect our technique to work well with domains that share similar characteristics with ours, i.e. where a limited number of typical, repeatable interactions can be anticipated between the service provider and the visitor. For example, a museum guide moves around to different exhibits and answers a visitor's questions about an exhibit; or an information booth clerk answers the visitor's questions about a department store. For other scenarios where interaction is multimodal and speech or spatial data is not sufficient, we may need to adapt the system to include data from different modalities. For example, we can imagine incorporating skeleton tracking data from a Kinect sensor into our system to train an exercise coach robot.

Our current approach was demonstrated to work well for a scenario with a limited number of concepts, which we believe could be scaled up to some degree with more data. The amount of training data is dependent on the number of social behaviors that need to be reproduced, the variability of the customer actions that trigger those behaviors, and the reliability of sensing. Hence, the training effort scales linearly with the number of the behaviors to be learned, such as when the number of cameras on display increases.

The one-step lookahead approach we use might be sufficient for scenarios with highly repeatable interactions that focus on simple questions and answers, such as an information-booth robot or a museum guide, but for more involved interactions it will inevitably become necessary to structure interactions in a more complex way. Extending our current system to include interaction history would seem to be an important consideration for future work. Modeling and remembering different attributes of a person may also be important in an interaction, including everything from name, age, and gender (the robot occasionally said "thank you, sir", to female participants) to dynamic variables like emotional and psychological state, attention target, and goals. In some cases it might be sufficient simply to add these states to the joint state vector to improve prediction, but in many cases it will be important to introduce new behavior

models, for example, treating the occurrence of a person's name in speech data in a special way, in order to enable more complex interactive behavior.

E. Tradeoff between Variation and Robustness

There is an inherent trade-off between the variation of the shopkeeper responses and the robustness to sensor noise afforded by clustering similar behaviors. That is, choosing a large number of robot action clusters will lead to more variation in its behaviors, but will increase the likelihood of noise corrupting those behaviors. With our data, we found that 166 clusters preserved a fair amount of variation in the shopkeeper utterances, while providing reasonable robustness to noise. For example, multiple clusters with the same general meaning represent different ways the robot can explain the color of Canon (e.g. "well the also comes in grey red and brown so you have a choice of color is this" and "intense grey red and brown colors"). In high-noise situations, it might make sense to reduce the number of clusters in order to make it easier to reject utterances corrupted by noise. In that case some of these variations would be lost, and the robot might only be able to describe the camera's color in one way. Conversely, in a situation where a greater amount of training data was available, we could choose a higher number of clusters, thus capturing even more natural variations of the spoken utterances while still rejecting noise.

It could also be possible to consider sampling more than one typical utterance from a cluster to use for robot speech. This could lead to a greater degree of lifelike variation in the robot's speech, but it would also increase the risk of ASR errors corrupting the spoken utterances.

F. Behavior Modeling in HRI

The current study used existing HRI proxemics models in order to create generalizable behavior templates that could be recognized and reproduced, such as the present-object formation. These models provide generalizable structural elements which can be helpful in learning complicated interactive behaviors. It would be useful to incorporate similar HRI models describing aspects of behavior such as gesture and gaze. During the interaction, some of the participants pointed to another camera, and says "what about that one?", but the robot was not sure which camera the participants were referring to. Some participants also commented that when the robot was trying to guide them, they were not sure where the robot was moving to at first. By incorporating pointing and gaze HRI models, the robot can better resolve ambiguities [51, 52].

Models of the structure of conversation would also be useful tools for extending this work into more complex domains. Some work has explored the use of generic dialogue patterns in HRI [53, 54], and it is plausible that some kind of templates could be used to help structure data-driven HRI, especially if utterances could be analyzed semantically. It would also be valuable to explore ways of incorporating models of turn-taking [55, 56], and models governing gaze cues and interaction distance for multiparty interaction [57, 58].

G. Embodiment of the robot

One question to be considered in this work is how well the

translation of experience from human-human to human-robot interaction can be achieved, given that the robot is embodied as a robot, rather than a human. After all, one could argue that learning to be a human is not necessarily the same as learning to be a robot. Regarding this point, we did observe a few cases where the human-robot interaction differed in some qualitative ways from human-human interaction. For example, one participant talked to the robot in keywords rather than sentences, as if it were a search engine. Some people seemed to treat the robot like a machine and never made eye contact with it. Several participants asked the robot to repeat itself when its speech synthesis was hard to understand. These differences resulted in situations that differed slightly from the training data – e.g., the humans never had difficulty pronouncing their speech, so the system never learned how to repeat and clarify statements.

In most cases, even when differences were observed, such as people not making eye contact with the robot, the difference did not cause any communication problems. The only real problem we observed regarding the dialog flow was the robot’s failure to repeat its utterances when asked. We believe specific cases like these are due to a few known issues, e.g. low-quality speech synthesis or speech recognition errors. Such problems are limited and can be expected to decrease as the associated technologies improve. A possible way to handle miscommunication such as a clarification request as an extension to our current system could be to encode the customer’s clarification request to a special behavior pattern. Without changing other parts of system, this special behavior pattern could trigger the robot to repeat its previous utterance when it detects the customer asks for clarification. While it is important to keep such differences in mind, we believe this work has demonstrated that the use of human-human interactions holds great potential as a source for generating realistic social behaviors in robots.

VII. CONCLUSION

We have presented a fully-autonomous method that enabled a robot to reproduce socially interactive behavior solely from examples of human-human interactions. Both behavior contents and execution logic are derived directly from observed data captured by a sensor network. We believe this is the first work in the field of social robotics to address this difficult problem. As such, our focus was not on any particular element of the system, but rather on demonstrating the effectiveness of our proposed system as a whole. Our evaluation shows that the robot’s behavior using our *proposed* system was rated more highly in a variety of measures than a version of the system that did not use clustering or interaction states. Furthermore, the proposed system showed robustness to sensor noise, achieving an 84.8% behavior correctness rate despite a speech recognition accuracy rate of only 76.8%.

This study has provided a proof-of-concept that interaction can be performed in a data-driven way, directly from observations of human-human interactions. This was made possible through a combination of abstractions: the empirical identification of the typical behavior patterns in the training data, combined with a set of generalizable HRI models

TABLE VI. WORD ACCURACY AND UTTERANCE ACCURACY

	Data Collection		Evaluation
	Customer (119 utterances)	Shopkeeper (123 utterances)	Customer (461 utterances)
Word Accuracy	79.81 %	76.62 %	87.31 %
Utterance Accuracy	37.82 %	30.89 %	64.43 %

specifying spatial formations. Although the interaction scenario we used was somewhat simple, we have suggested many directions in which this work could be extended to capture more complex elements of interactions, and we believe many of the techniques for interpreting sensor data, applying HRI proxemics models, and reproducing human behaviors in a robot despite large amounts of sensor noise will be applicable to other scenarios. This study highlights the importance of behavior modeling in HRI to provide structures useful in interpreting collected sensor data and generating robot behaviors.

Perhaps most importantly, the scalability of this approach gives it the potential to transform the way social behavior design is conducted in HRI. Once passive collection of interaction data becomes practical, even a single sensor network installation could provide enormous amounts of example interaction data over time, an invaluable resource for the collection and modeling of social behavior. We believe that with today’s trends towards big-data systems and cloud robotics, techniques like this will become essential methods for generating robot behaviors in the future.

APPENDIX

To complement our evaluation of ASR correctness, we also evaluated the output quality of the ASR system based on common metrics of word and utterance accuracy. We used measurements of accuracy rather than the error rate, in order to enable easier comparisons with our other metrics, *ASR correctness* and *behavior correctness*. Word Accuracy is defined as

$$\text{Word Accuracy} = 1 - \frac{S+D+I}{N} \quad (3)$$

where S is the number of incorrect words substituted, D is the number of words deleted, I is the number of extra words inserted, and N is the number of words in the correct transcript. Utterance Accuracy is defined as

$$\text{Utterance Accuracy} = 1 - \frac{N_e}{N_T} \quad (4)$$

where N_e is the number of utterances containing any errors and N_T is the total number of utterances.

The results are shown in Table 6. We speculate the reason why customer’s utterance accuracy was much higher during evaluation than during data collection is because customer participants spoke much more clearly to the robot than to the human shopkeeper.

ACKNOWLEDGMENT

We would like to thank Clément Congard for his help in conducting the experiment and editing the video. We would also like to thank members of our lab who participated in our data collection.

ETHICAL APPROVAL

This research was conducted in compliance with the standards and regulations of our company's ethical review board, which requires every experiment we conduct to be subject to a review and approval procedure according to strict ethical guidelines.

REFERENCES

- [1] T. Kanda, M. Shiomi, L. Perrin, T. Nomura, H. Ishiguro, and N. Hagita, "Analysis of people trajectories with ubiquitous sensors in a science museum," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 4846-4853.
- [2] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, pp. 3-55, 1999.
- [3] I. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: a five year experiment," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003, pp. 3636-3641 vol.3.
- [4] M. Bannet, F. Faber, D. Joho, M. Schreiber, and S. Behnke, "Towards a humanoid museum guide robot that interacts with multiple persons," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, 2005, pp. 418-423.
- [5] P. Liu, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "How to train your robot - teaching service robots to reproduce human social behavior," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, 2014, pp. 961-968.
- [6] L. Takayama, E. Marder-Eppstein, H. Harris, and J. M. Beer, "Assisted driving of a mobile remote presence system: System design and controlled user evaluation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1883-1889.
- [7] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, and A. C. Schultz, "Designing robots for long-term social interaction," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 1338-1343.
- [8] A. M. Sabelli, T. Kanda, and N. Hagita, "A conversational robot in an elderly care center: An ethnographic study," in *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, 2011, pp. 37-44.
- [9] P. Langfelder, B. Zhang, and S. Horvath, "Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R," *Bioinformatics*, vol. 24, pp. 719-720, 2008.
- [10] C. Gharpure and V. Kulyukin, "Robot-assisted shopping for the blind: issues in spatial cognition and product selection," *Intelligent Service Robotics*, vol. 1, pp. 237-251, 2008/07/01 2008.
- [11] H.-M. Gross, H. Boehme, C. Schroeter, S. Müller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley, "TOOMAS: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 2005-2012.
- [12] J. Mumm and B. Mutlu, "Human-robot proxemics: Physical and psychological distancing in human-robot interaction," in *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, 2011, pp. 331-338.
- [13] D. Brscic, T. Kanda, T. Ikeda, and T. Miyashita, "Person Tracking in Large Public Spaces Using 3-D Range Sensors," *Human-Machine Systems, IEEE Transactions on*, vol. 43, pp. 522-534, 2013.
- [14] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *Signal Processing Magazine, IEEE*, vol. 29, pp. 127-140, 2012.
- [15] C. Breazeal, N. DePalma, J. Orkin, S. Chernova, and M. Jung, "Crowdsourcing Human-Robot Interaction: New Methods and System Evaluation in a Public Environment," *Journal of Human-Robot Interaction*, vol. 2, pp. 82-111, 2013.
- [16] F. Toris, D. Kent, and S. Chernova, "The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing," *Journal of Human-Robot Interaction*, vol. 3, pp. 25-49, 2014.
- [17] D. F. Glas, S. Satake, T. Kanda, and N. Hagita, "An Interaction Design Framework for Social Robots," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, 2011.
- [18] G. Skantze and S. Al Moubayed, "IrisTK: a statechart-based toolkit for multi-party face-to-face interaction," in *Proceedings of the 14th ACM international conference on Multimodal interaction*, 2012, pp. 69-76.
- [19] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "An affective guide robot in a shopping mall," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, La Jolla, California, USA, 2009, pp. 173-180.
- [20] K. Zheng, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "Designing and Implementing a Human-Robot Team for Social Interactions," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 2012.
- [21] M. Nicolescu and M. J. Mataric, "Task learning through imitation and human-robot interaction," *Models and mechanisms of imitation and social learning in robots, humans and animals: behavioural, social and communicative dimensions*, 2005.
- [22] S. P. Chatzis and Y. Demiris, "Nonparametric mixtures of Gaussian processes with power-law behavior," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, pp. 1862-1871, 2012.
- [23] M. Ogino, H. Toichi, Y. Yoshikawa, and M. Asada, "Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping," *Robotics and Autonomous Systems*, vol. 54, pp. 414-418, 2006.
- [24] B. M. Scassellati, "Foundations for a Theory of Mind for a Humanoid Robot," Massachusetts Institute of Technology, 2001.
- [25] Y. Nagai, "Learning to comprehend deictic gestures in robots and human infants," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, 2005, pp. 217-222.
- [26] S. Calinon and A. Billard, "Teaching a humanoid robot to recognize and reproduce social cues," in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, 2006, pp. 346-351.
- [27] P. E. Rybski, J. Stolarz, K. Yoon, and M. Veloso, "Using dialog and human observations to dictate tasks to a learning robot assistant," *Intelligent Service Robotics*, vol. 1, pp. 159-167, 2008.
- [28] R. Meena, G. Skantze, and J. Gustafson, "A Data-driven Approach to Understanding Spoken Route Directions in Human-Robot Dialogue," in *INTERSPEECH*, 2012.
- [29] W. Yan and D. A. Forsyth, "Learning the behavior of users in a public space through video tracking," in *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, 2005, pp. 370-377.
- [30] A. Sorokin, D. Berenson, S. S. Srinivasa, and M. Hebert, "People helping robots helping people: Crowdsourcing for grasping novel objects," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 2117-2122.
- [31] M. E. Foster, S. Keizer, Z. Wang, and O. Lemon, "Machine learning of social states and skills for multi-party human-robot interaction," in *Proceedings of the workshop on Machine Learning for Interactive Systems (MLIS 2012)*, 2012, p. 9.
- [32] J. E. Young, E. Sharlin, and T. Igarashi, "Teaching robots style: designing and evaluating style-by-demonstration for interactive robotic locomotion," *Human-Computer Interaction*, vol. 28, pp. 379-416, 2013.
- [33] J. E. Young, T. Igarashi, E. Sharlin, D. Sakamoto, and J. Allen, "Design and evaluation techniques for authoring interactive and stylistic behaviors," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 3, p. 23, 2014.
- [34] J. Orkin and D. K. Roy, "Understanding Speech in Interactive Narratives with Crowdsourced Data," in *AIIDE*, 2012.
- [35] S. Chernova, J. Orkin, and C. Breazeal, "Crowdsourcing HRI through Online Multiplayer Games," presented at the AAAI Fall Symposium Series, 2010.
- [36] S. Chernova, N. DePalma, E. Morant, and C. Breazeal, "Crowdsourcing human-robot interaction: Application from virtual to physical worlds," in *RO-MAN, 2011 IEEE*, 2011, pp. 21-26.
- [37] B. Shneiderman, "The limits of speech recognition," *Communications of the ACM*, vol. 43, pp. 63-65, 2000.
- [38] M. Forsberg, "Why is speech recognition difficult," *Chalmers University of Technology*, 2003.
- [39] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, pp. 259-284, 1998.
- [40] M. F. Porter, "An algorithm for suffix stripping," *Program: electronic library and information systems*, vol. 14, pp. 130-137, 1980.

- [41] F. Wild, C. Stahl, G. Stermsek, and G. Neumann, "Parameters driving effectiveness of automated essay scoring with LSA," presented at the Proceedings of the 9th CAA Conference, Loughborough, UK, 2005.
- [42] L. Guéguen, "Segmentation by Maximal Predictive Partitioning According to Composition Biases," in *Computational Biology*, vol. 2066, O. Gascuel and M.-F. Sagot, Eds., ed: Springer Berlin Heidelberg, 2001, pp. 32-44.
- [43] C. Shi, T. Kanda, M. Shimada, F. Yamaoka, H. Ishiguro, and N. Hagita, "Easy development of communicative behaviors in social robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 5302-5309.
- [44] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita, "How close?: model of proximity control for information-presenting robots," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, Amsterdam, The Netherlands, 2008, pp. 137-144.
- [45] E. T. Hall, *The Hidden Dimension*. London, UK: The Bodley Head Ltd, 1966.
- [46] T. Kitade, S. Satake, T. Kanda, and M. Imai, "Understanding suitable locations for waiting," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, 2013, pp. 57-64.
- [47] T. Kanda, D. F. Glas, M. Shiomi, and N. Hagita, "Abstracting People's Trajectories for Social Robots to Proactively Approach Customers," *Robotics, IEEE Transactions on*, vol. 25, pp. 1382-1396, 2009.
- [48] H. Admoni and B. Scassellati, "Data-driven model of nonverbal behavior for socially assistive human-robot interactions," in *Proceedings of the 16th International Conference on Multimodal Interaction*, 2014, pp. 196-199.
- [49] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, pp. 509-517, 1975.
- [50] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, pp. 23-33, 1997.
- [51] Y. Hato, S. Satake, T. Kanda, M. Imai, and N. Hagita, "Pointing to space: modeling of deictic interaction referring to regions," in *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, 2010, pp. 301-308.
- [52] P. Liu, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "It's not polite to point: generating socially-appropriate deictic behaviors towards people," in *8th ACM/IEEE International Conference on Human-Robot Interaction*, 2013, pp. 267-274.
- [53] J. Peltason and B. Wrede, "Modeling human-robot interaction based on generic interaction patterns," *AAAI Report on Dialog with Robots*, 2010.
- [54] P. H. Kahn, N. G. Freier, T. Kanda, H. Ishiguro, J. H. Ruckert, R. L. Severson, and S. K. Kane, "Design patterns for sociality in human-robot interaction," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, Amsterdam, The Netherlands, 2008, pp. 97-104.
- [55] A. L. Thomaz and C. Chao, "Turn-taking based on information flow for fluent human-robot interaction," *AI Magazine*, vol. 32, pp. 53-63, 2011.
- [56] C. Chao and A. L. Thomaz, "Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets," *Journal of Human-Robot Interaction*, vol. 1, pp. 1-16, 2011.
- [57] T. Tadaki, S. Matsumoto, H. Ohba, M. Toda, K. Komatani, T. Ogata, and H. G. Okuno, "Dynamic communication of humanoid robot with multiple people based on interaction distance," in *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, 2004, pp. 71-76.
- [58] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita, "Footing in human-robot conversations: how robots might shape participant roles using gaze cues," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, La Jolla, California, USA, 2009, pp. 61-68.

NOTES ON CONTRIBUTORS



Phoebe Liu received her S.B. degree in Electronic Engineering from Simon Fraser University, Canada in 2011, and received her M. Eng. in engineering science in 2013 from Osaka University, Osaka, Japan. She is an internship researcher at Hiroshi Ishiguro Laboratories at the Advanced Telecommunications Research Institute

International (ATR) in Kyoto, Japan, where she has been since 2011. She has been a Ph. D. candidate in the Graduate School of Engineering Science at Osaka University since 2013. Her research interests include social human-robot interaction and machine learning.



Dylan F. Glas received his Ph.D. in Robotics from Osaka University in 2013. He received his M.Eng in Aerospace Engineering from MIT in 2000 and S.B. degrees in Aerospace Engineering and in Earth, Atmospheric, and Planetary Sciences from MIT in 1997. From 1998-2000 he worked in the Tangible Media Group at the MIT Media Lab (1998-2000). He is currently at ATR in Kyoto, Japan, where he was group leader of the Department of Cloud Intelligence and is currently a Senior Researcher in Hiroshi Ishiguro Laboratories, under the JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project. He is also a Guest Associate Professor at the Intelligent Robotics Laboratory at Osaka University. His research interests include social human-robot interaction, cloud network robot systems, ubiquitous sensing, teleoperation for social robots, and machine learning.



Takayuki Kanda received his B. Eng, M. Eng, and Ph. D. degrees in computer science from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2003, respectively. From 2000 to 2003, he was an Intern Researcher at ATR Media Information Science Laboratories, and he is currently a Senior Researcher at ATR Intelligent Robotics and Communication Laboratories, Kyoto, Japan. His current research interests include intelligent robotics and human-robot interaction.



Hiroshi Ishiguro received a D.Eng. in systems engineering from the Osaka University, Japan in 1991. He is currently professor of Department of Systems Innovation in the Graduate School of Engineering Science at Osaka University (2009-) and distinguished professor of Osaka University (2013-). He is also group leader (2002-) of Hiroshi Ishiguro Laboratories at the Advanced Telecommunications Research Institute and the ATR fellow. He was previously research associate (1992-1994) in the Graduate School of Engineering Science at Osaka University and associate professor (1998-2000) in the Department of Social Informatics at Kyoto University. He was also visiting scholar (1998-1999) at the University of California, San Diego, USA. He was associate professor (2000-2001) and professor (2001-2002) in the Department of Computer and Communication Sciences at Wakayama University. He then moved to Department of Adaptive Machine Systems in the Graduate School of Engineering at Osaka University as a professor (2002-2009). His research interests include distributed sensor systems, interactive robotics, and android science.