Data-Driven Imitation Learning for a Shopkeeper Robot with Periodically Changing Product Information

MALCOLM DOERING, DRAŽEN BRŠČIĆ, and TAKAYUKI KANDA, Kyoto University

Data-driven imitation learning enables service robots to learn social interaction behaviors, but these systems cannot adapt after training to changes in the environment, such as changing products in a store. To solve this, a novel learning system that uses neural attention and approximate string matching to copy information from a product information database to its output is proposed. A camera shop interaction dataset was simulated for training/testing. The proposed system was found to outperform a baseline and a previous state of the art in an offline, human-judged evaluation.

CCS Concepts: • **Computing methodologies** → **Learning from demonstrations**; • **Human-centered computing** → **HCI theory, concepts and models**; • **Computer systems organization** → Robotics; • **Information systems** → Question answering;

Additional Key Words and Phrases: Human-robot interaction, imitation learning, database question answering, knowledge base question answering, retail robot, service robot, social robot

ACM Reference format:
Malcolm Doering, Dražen Brščić, and Takayuki Kanda. 2021. Data-Driven Imitation Learning for a Shopkeeper Robot with Periodically Changing Product Information. *Trans. Hum.-Robot Interact.* 10, 4, Article 31 (July 2021), 20 pages.
https://doi.org/10.1145/3451883

1 INTRODUCTION

The data-driven imitation learning approach has been been increasingly explored as a method for designing end-to-end text-based dialog systems [1, 44] and for multi-modal, mobile robot interaction behaviors [6, 7, 10, 29]. The advantage of this approach is that robot interaction behaviors, which consist of speech, locomotion, object manipulation, and so forth, can be learned for domains with highly repeatable, formulaic interactions from data alone, with minimal input from human interaction designers or manual data annotation. Furthermore, by learning from natural interaction examples collected with noisy sensors, the system can become more responsive to natural variations of human speech and behavior (forgoing the necessity for interaction designers to anticipate the many ways humans might behave), and it can become more robust to sensor noise (e.g., speech recognition errors).

This work was funded by JST CREST Grant Num. JPMJCR17A2, Japan.
M. Doering, D. Brščić, and T. Kanda are also with Advanced Telecommunications Research Institute International (ATR).
Authors' address: M. Doering, D. Brščić, and T. Kanda, Kanda Laboratory, Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan; emails: {doering, drazen, kanda}@i.kyoto-u.ac.jp.

2573-9522/2021/07-ART31 $15.00
https://doi.org/10.1145/3451883

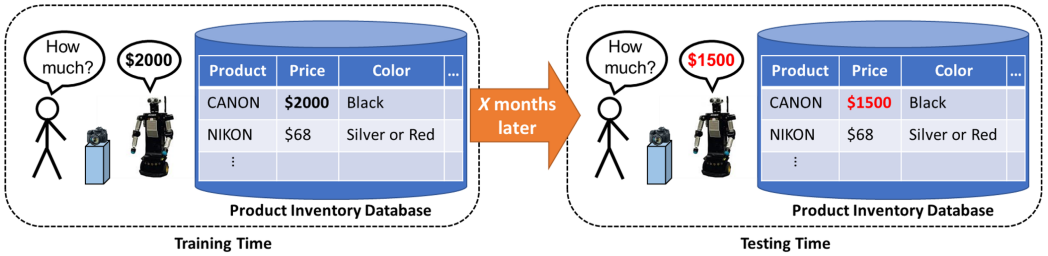

Fig. 1. In service scenarios (e.g., a camera shop), the information that should be provided by a robot may change after training. The robot should automatically adapt its actions whenever the product database information is updated, without retraining.

However, one limitation of the previous approaches is that after training, they cannot adapt to dynamic, periodically changing environments in which social robots work. For example, in a camera shop scenario, a robot might be trained on a dataset of interactions collected over a duration of time where the products in the store and the attributes of those products (e.g., prices) are fixed. To adapt the robot's behaviors to deal with changing products, a new interaction dataset with the new products would have to be collected, and the robot would have to be retrained. But since data collection and retraining take time and would require the robot to repeatedly be taken out of service, this is an infeasible solution.

Thus, this work focuses on the problem of learning and adapting the behaviors of a camera shopkeeper robot to keep its actions consistent with the products sold there over time (Figure 1). For this purpose, we propose a learning system that uses database-addressing neural attention and approximate string matching mechanisms. Approximate string matching finds mentions of product information in shopkeeper speech, and the neural network, given a representation of the state of the interaction, retrieves the most relevant contents from a product information database and outputs an appropriate shopkeeper action. Furthermore, it learns to do this automatically based on interaction examples, without manual data annotation.

The proposed system was evaluated by training and testing it on simulated examples of customer-shopkeeper interactions and snapshots of the product database from the times the interactions were collected. It was compared to a baseline without the database-addressing and string search mechanisms and to a state-of-the-art neural architecture for question answering with a knowledge base [19]. Finally, the proposed system was found to outperform the two other systems in an offline, human evaluation.

2 RELATED WORKS

2.1 Data-Driven Imitation Learning of Social Interaction Behaviors

Data-driven systems for designing behaviors for virtual humans were explored early on: The "Restaurant Game"was a virtual restaurant in which two human players, playing the roles of customer and waiter, could navigate around the restaurant and interact via text and predefined actions [37, 38]. "Plan networks" were automatically extracted from the interaction data, enabling the automation of virtual characters. The "Mars Escape"game used a similar approach and went a step further by embodying the learned behaviors in a real robot [2, 7, 8]. In contrast, data collected in the real, physical world, which is incorporated into our interaction simulation, has the additional challenge of sensor noise, which is absent in virtual worlds.

Liu et al. [29, 30] introduced the concept of training a camera shopkeeper robot from examples of natural human-human interaction without any input from a human designer. The approach

consists of clustering the raw interaction data to find discrete common actions, then training a neural network classifier using the action cluster IDs as labels.

Much research has been done in extending the data-driven imitation learning approach to human-robot interaction. Nanavati et al. [35] focused on applying the data-driven imitation learning framework to one-to-many interaction (i.e., one shopkeeper, many customers). Doering et al. [10] focused on the problem of modeling the hidden structure of interaction, which enabled resolution of ambiguous customer speech. Doering et al. [11] showed how gated recurrent neural networks could be used to learn a memory model of customer behavior. In addition, Doering et al. [12] presented a system that could explore different robot behaviors and learn online, which resulted in more varied, interesting robot behaviors than previous approaches and enabled some customization to customers' individual differences. To make these previous approaches function properly in a situation where a store's products have changed after training, further data collection and system retraining would be necessary. In contrast, the current work introduces an approach that automatically adapts the robot's behavior to account for changing products in the store.

Others have also explored learning of human-robot interaction behaviors. Kawahara [22] presents an approach for human-robot dialog but does not use a data-driven imitation learning approach. Where a data-driven approach is used, it is not end-to-end and requires annotated data. In contrast, our approach is end-to-end (from customer input to robot output) and does not require expensive data annotation. Patompak et al. [40] presented a reinforcement learning method to learn social proxemics maps of groups of people for socially appropriate robot navigation. In contrast, our approach uses imitation learning to learn speech behaviors and location targets.

2.2 Dialog and Question Answering

Recent work in end-to-end dialog systems is closely related to learning social robot interaction behaviors. These methods work by either retrieving or generating a response utterance given a dialog history. Retrieval-based methods select a response from among a set of candidate responses obtained from the training data using score functions based, for example, on neural networks or term frequency–inverse document frequency [32]. However, generation-based methods generate responses from scratch, word-by-word, using recurrent neural networks [44, 46]. Both approaches have certain advantages and disadvantages, so some approaches attempt to combine them to get the best of both [45, 55]. In contrast to our approach, these systems (except that of Yang et al. [55]) decide the output based only on the dialog history, without any external information source.

Liu et al. [29] presented a retrieval-based method designed to work better than previous retrieval-based methods on data collected by noisy sensors in environments where robots are to be deployed. Well-formed, error-free utterances were selected for robot output from among the training utterances by speech clustering and typical utterance selection. Our approach extends this method for the situation where the system may want to alter the typical utterance by adding some information from a product information database. Thus, it could maintain robustness to noise while also including information from an external knowledge base.

There are also data-driven methods that are designed to use external information sources, such as end-to-end neural dialog systems and question answering systems. These include generation-based [13, 19, 26, 54, 57], retrieval-based [1, 47, 56], and combined approaches [55]. Furthermore, some of them are designed to access external data stored in an unstructured text document [54], arrays of sentences [1, 13, 26, 47, 55], knowledge graphs [56, 57], and databases [19].

Most closely related to our approach is that of He et al. [19], which presents a generation-based approach for answering questions based on information in a database. It has the common pitfalls of generation-based approaches, such as outputs that are overly general and ungrammatical [45, 55]. In contrast, our approach is based on the retrieval method first presented by Liu et al. [29],

which overcomes these pitfalls. Furthermore, one of the downsides of retrieval-based approaches is that they can only output candidate responses that appear in the training data, so they cannot generalize beyond previously seen outputs. Our approach overcomes this by creating templates that can be dynamically filled with information from the database at runtime.

To the best of our knowledge, our proposed approach is the only one that uses retrieval-based methods in combination with a structured external knowledge base and will work on out-of-vocabulary database contents without requiring retraining. This makes it uniquely suited to solve the problem of data-driven learning of interactive shopkeeper behaviors when there are periodically changing products.

2.3 Continual Learning

Continual learning, also known as incremental learning [15, 28], is the problem of learning from training data that is obtained over time in a dynamically changing environment. This is in contrast to traditional machine learning, where all training data is available at once and the data distribution is assumed to be static. Continual learning has been used for robotics and human-robot interaction in applications such as learning task representations [58], perception models [53], motion primitives [33], gestures [4], and body postures [52]. Moreover, reinforcement learning in combination with intrinsic motivation has been used to allow robots to explore their sensorimotor [21, 39, 43] and social [42] spaces in dynamically changing environments.

The primary question that our work focuses on is how a data-driven imitation learning system can adapt to periodically changing products in a store setting. Continual learning may be a possible approach to solving this problem; however, it is distinct from our proposed approach. In continual learning, a learning model will continuously train on newly incoming data, sometimes requiring labeling input from an oracle (e.g., a human supervisor), depending on the type of learning model [28]. In contrast, our approach does all of the training at once, on unlabeled examples of interactions and database snapshots from many points in time. This has the advantage of not requiring manual data labeling and forgoes the other requirements of continual learning, such as memory, data storage, and learning-dedicated CPU cycles during runtime. Another important difference is that continual learning aims at *remembering* previously learned concepts while learning new ones, whereas our proposed system is designed to *overwrite* previous product information with new information.

2.4 Symbol Grounding in Human-Robot Interaction

Symbol grounding, or language grounding, in the context of human-robot interaction is mainly concerned with linking linguistic instructions or descriptions of the physical environment to a robot's actions and perceptions [17, 34]. At a high level, the key to the symbol grounding problem lies in how object, spatial relation, and attribute classifiers, and primitive robot actions, can be linked to the semantic representations of sentences. Many approaches exist, including attributed relational graph matching [5], defining robot actions in terms of goal states or action sequences [5, 16], probabilistic graphical models such as conditional random fields and hierarchical adaptive distributed correspondence graphs [41, 48, 49], and active and interactive learning to learn new words and classifiers [25, 50, 51].

Symbol grounding in the context of human-robot interaction is similar to our work because it leverages externally provided information contained not in a database but in a data structure that describes a representation of the environment model at the time of the interaction. However, these related works are mainly focused on the problem of linguistic communication between the human and robot about the physical world, whereas data-driven imitation learning is intended for more social interaction and information exchange, where the contents of the speech are not necessarily about the physical world. Moreover, were a symbol grounding approach to be applied

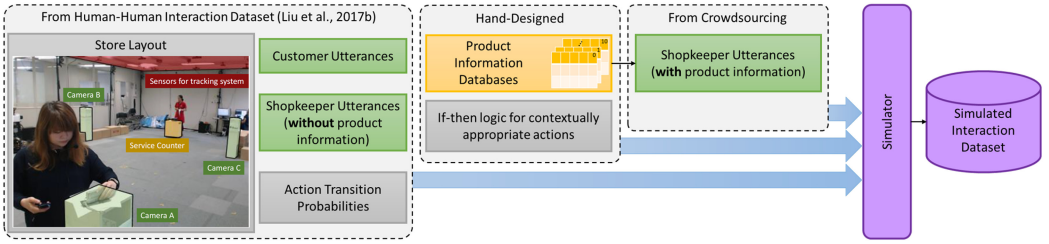

Fig. 2. Overview of the simulation procedure.

to the problem of learning to talk about periodically changing products, this would raise the question of how to ground language to language (the customer or shopkeeper's speech to the contents of the database), which is not the target of the symbol grounding methods but is solved by our proposed system.

3 INTERACTION DATASET

To train and evaluate a system that can deal with periodically changing products, we first need a dataset of interactions in a store that is taken over a prolonged time so that it includes such changing products. In the future, it will be possible to collect social interaction data in stores to train social service robots. For example, passive sensors, such as person-tracking 3D range sensors [3] and voice-recording microphone arrays, could record customer-employee dialogues about products in the store. However, such a dataset does not yet exist, and collecting one, even in a laboratory setting, would be time and cost prohibitive.

Therefore, to demonstrate the proposed system as a proof-of-concept, we simulated customer-shopkeeper interactions in a camera shop scenario, where the cameras on display periodically change. This was accomplished by creating an interaction simulator that combined elements of previously collected human-human interactions, hand-designed new products, and new utterances collected via crowdsourcing (Figure 2).

The previous human-human interactions (collected in a physical laboratory setup) were used as the basis for the simulated interactions [31]. Thus, each simulated interaction contains three cameras, one shopkeeper, and one customer. Figure 2 shows the store layout. Furthermore, abstract actions (Table 1), action transition probabilities (Section 3.1.2), and utterances (Section 3.1.4) from that human-human dataset were used for simulation.

To simulate periodically changing products, new products were hand designed and used to populate snapshots of a product information database (Section 3.1.1). Moreover, utterances about these new products were crowdsourced, as they were not present in the human-human dataset (Section 3.1.5).

3.1 Simulation Procedure

The simulator generated sequences of alternating shopkeeper and customer actions (Table 1), where action transitions were determined by transition probabilities and if-then style logic for contextually appropriate actions. Finally, customer and shopkeeper utterances from the human-human dataset and from crowdsourcing were injected into the generated action sequences.

3.1.1 Product Information Databases. Each product information database consisted of C rows of products (cameras) and A columns of attributes. In total, 33 new cameras and 11 product information databases were created, each containing 3 cameras and 15 attributes. Table 2 shows an example database. For each simulated interaction, one database was selected to represent the products in the store at that time.

Table 1. Actions Used in the Interaction Simulation

Action	Actor
ENTERS	Customer
GREETS	Customer, Shopkeeper
WALKS TO CAMERA	Customer
LOOKING FOR A CAMERA	Customer
EXAMINES CAMERA	Customer
LEAVES	Customer
QUESTION ABOUT FEATURE	Customer
SILENT OR BACKCHANNEL	Customer
THANK YOU	Customer
NONE	Shopkeeper
RETURNS TO COUNTER	Shopkeeper
LET ME KNOW IF YOU NEED HELP	Shopkeeper
ANSWERS QUESTION ABOUT FEATURE	Shopkeeper
INTRODUCES FEATURE	Shopkeeper
INTRODUCES CAMERA	Shopkeeper
NOT SURE	Shopkeeper

Table 2. An Example Product Information Database

camera_ID	camera_name	camera_type	preset_modes	...
CAMERA_A	Sony Alpha5100	E-mount interchangeablelens camera	9 scene modes(e.g., landscape, macro)	
CAMERA_B	Nikon CoolpixA900	Compact digitalcamera	17 exposure modes(e.g., backlighting)	
CAMERA_C	Fujifilm FinePixJX660	Point and shootcamera	22 shooting modes(e.g., baby, fireworks)	

Note: Each database contains three cameras and 15 attributes (*camera ID*, *camera name*, *camera type*, *preset modes*, etc).

Table 3. An Example Simulated Interaction

Time Step	Person	Location and Spatial Formation	Action	Utterance	Camera of conversation	Previous cameras and features of conversation
⋮						
4	C	CAMERA_A	GREETS	excuse me		
	S	CAMERA_A, Face-to-face	GREETS	how can I help you?		
5	C	CAMERA_A	LOOKING FOR A CAMERA	I want a All Around camera I try to use my smartphone but the pictures don't come out so good		
	S	Present CAMERA_A	INTRODUCES CAMERA	What we have here is the Sony Alpha 5100 camera.	CAMERA_A	
6	C	CAMERA_A	QUESTION ABOUT FEATURE	does it have dog mode	CAMERA_A	CAMERA_A
	S	Present CAMERA_A	ANSWERS QUESTION ABOUT FEATURE	It features 9 scene modes including landscape, macro and night portrait.	CAMERA_A	CAMERA_A: preset_modes
7	C	CAMERA_A	SILENT OR BACKCHANNEL	wow that's pretty impressive	CAMERA_A	"
	S	Present CAMERA_A	INTRODUCES FEATURE	The Sony Alpha 5100 goes for $350.	CAMERA_A	CAMERA_A: preset_modes, price
⋮						

3.1.2 Action Transitions. Each simulated interaction starts with the customer at the door and the shopkeeper at the service counter. The customer then enters and approaches one of the three cameras. The shopkeeper may approach a browsing customer to answer questions or introduce the cameras and their features. The customer may ask about the cameras, utter backchannels, remain

silent, or thank the shopkeeper for help. The interaction proceeds until the customer decides to leave the store.

The action transition probabilities were set manually based on the interactions in the human-human dataset [31]. In that work, the customer's and shopkeeper's actions were segmented and discretized into sequences of alternating customer and shopkeeper actions. The transition probabilities were estimated based on observations of these sequences. For example, after the shopkeeper introduces a camera, there is roughly a 50% chance that the customer will remain silent (allowing the shopkeeper to take initiative) and a 50% chance that the customer will ask a question about a feature.

3.1.3 If-Then Logic for Contextually Appropriate Actions. *A priori* if-then logic was used to generate contextually appropriate actions, since action transitions alone do not fully capture the interaction history.

For each interaction, the simulator tracks the current camera of conversation, the previous cameras of conversation, the previous features of conversation, and the customer and shopkeeper's locations. Then, contextually appropriate actions were generated via if-then logic conditioned on these stored aspects of the interaction history. For example, the shopkeeper's actions were simulated such that he remembers which cameras and features had already been talked about so he would not reintroduce them. Additionally, the customers' actions were simulated such that they did not ask about features or cameras that had already been introduced or asked about.

Spatial formations, which describe the shopkeeper and customer's proxemics, were also generated using if-then rules. At each turn, one of three possible spatial formations was set: *waiting* was set when the shopkeeper was at the service counter, *face-to-face* when greeting the customer, and *present object* when talking about a camera.

3.1.4 Utterances from the Human-Human Dataset. In the final step of simulation, customer and shopkeeper utterances were assigned to each turn. For each action (Table 1), a set of matching utterances was selected from the human-human dataset from Liu et al. [31].

The utterances in the human-human dataset [31] were recorded during role-played camera shop interactions using handheld smartphones with attached head-mounted microphones. To start and stop recording their speech, participants could touch anywhere on the smartphone's screen and an audible cue would play. This setup enabled participants to speak and operate the smartphone without distraction. The recorded utterances were automatically transcribed using the Google Speech API.

Several natural variations of phrasing (e.g., backchannels such as "okay" and "I see, and . . .") and terminology are present in the human-human dataset. Furthermore, the data contains many **automatic speech recognition (ASR)** errors. In fact, of 400 utterances from a dataset collected in the same environment and scenario, and using the same equipment and recording methods, 53% were correctly recognized, 30% had minor errors (e.g., "can it should video" instead of "can it shoot video"), and 17% were complete nonsense [29]. The presence of the human-human utterances' natural variation and speech recognition errors in the simulated dataset makes the subsequent learning problem more closely resemble the challenge of learning from real data collected with noisy sensors.

3.1.5 Utterances from Crowdsourcing. The simulated interactions must also contain shopkeeper utterances about the cameras in the databases, which were not present in the human-human dataset. Therefore, we obtained such utterances for actions *INTRODUCES CAMERA*, *INTRODUCES FEATURE*, and *ANSWERS QUESTION ABOUT FEATURE* through crowdsourcing.

Crowd workers role-playing a shopkeeper were asked what they would say given a prompt, which consisted of a context (e.g., "The customer asked for the resolution of this camera"), an intent (e.g., "You want to provide the requested information"), and information for a camera. Thus, 20,534 shopkeeper utterances were collected from 24 crowd workers. They were then injected into the generated interactions by assigning them based on action type, camera, and feature of conversation.

3.2 The Simulated Interaction Dataset

Finally, 2,200 interactions[1] (200 per database) were simulated, containing a total of 36,129 customer-shopkeeper turns (mean 16.4, standard deviation 7.9 turns per interaction). Table 3 shows an example simulated interaction. Such interactions reasonably resemble what could be collected in the real world (Section 6.1).

4 PROPOSED APPROACH

4.1 Overview

The goal of the system is to learn the behaviors of a shopkeeper from examples of human-human interaction such that a robot could perform as a shopkeeper in interaction with real humans. Figure 3 shows an overview of the complete system that would enable a robot shopkeeper to function.

A sensor network collects raw interaction data in the camera shop. It consists of an array of Microsoft Kinect sensors that track the human and robot's positions and a smartphone application that records the customer's speech. ASR is used to transcribe the customer's speech into text. The sensor data is sent to a behavior abstraction module that discretizes the raw tracking data into one of the typical stopping locations (*door*, *middle*, *camera A*, *camera B*, *camera C*, and *service counter*), which were discovered by clustering stopping locations in the training data. Existing models are applied to determine proxemics formation (*waiting*, *face-to-face*, *present object*) [31].

The *input sequence* stores all actions that have occurred from the time a customer enters the shop. When a customer action is detected, or when a shopkeeper action is output, it is appended to the sequence. The input sequence is vectorized (Section 4.2.2) and fed into a trained neural network to get the next shopkeeper action. The robot moves to the output location, spatial formation, and state target and synthesizes the output utterance with text-to-speech.

The focus of this work is for the system to update its output speech based on the changing contents of a product information database. This is accomplished by searching for mentions of database contents among the shopkeeper utterances during training time (Section 4.2.3) and replacing such mentions in the output utterance during runtime (Section 4.3.3) based on database indices, which are output from the trained neural network (Section 4.3.2).

In this work, we trained and evaluated the learning system on simulated interactions, so the sensor network, abstraction of typical behavior patterns, and text-to-speech was not used. However, similar versions of the complete system have been demonstrated to work in live interaction with real humans in previous work [29, 30].

4.2 Preprocessing

Before training the neural network, customer and shopkeeper actions are vectorized, the shopkeeper utterances from the training data are searched for mentions of product information and marked, and the shopkeeper speech is clustered.

[1]The simulated dataset and neural network code is available at http://www.robot.soc.i.kyoto-u.ac.jp/en/research/dataset-simulated-customer-shopkeeper-interactions-with-periodically-changing-products/.

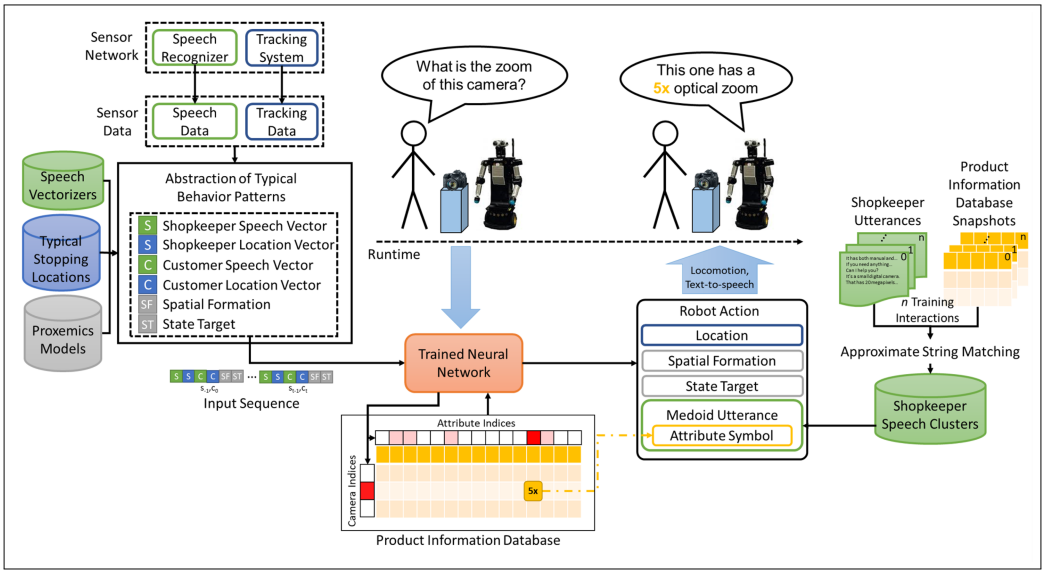

Fig. 3. System overview. Before runtime, speech features, typical stopping locations, and shopkeeper speech clusters are learned from the training data, and the neural network is trained. The shopkeeper utterances are searched for mentions of the database contents and marked with *attribute symbols*. During runtime, speech and tracking data are collected from a sensor network in the real world, and the raw data is abstracted using the models of typical behavior patterns, vectorized, and input to the trained neural network. The neural network outputs the robot's next action. If the output utterance contains an *attribute symbol*, it is replaced with the relevant information from the database. In this work, we focus on a simulated scenario, so the sensor network and physical robot are not used.

4.2.1 Typical Stopping Locations and Proxemics Formations. Typical stopping locations and proxemics formations represent the customer and shopkeeper's positions and orientations, abstracting away from raw sensor data. In previous work, the customer and shopkeeper's x, y positions were detected using a 3D person-tracking sensor network, trajectory segmentation was applied to separate moving and stopped segments of each participants' motion trajectory, and clustering was applied to the stopped segments to identify the typical stopping locations in the store [31]. Six locations were discovered: *door*, *middle*, *service counter*, *camera A*, *camera B*, and *camera C*. Furthermore, existing human-robot interaction models of proxemics formations were applied to the raw position data to determine whether the shopkeeper was *waiting*, *face-to-face*, or *presenting object*, and in the last case, whether it was presenting *camera A*, *camera B*, or *camera C*.

In this work, the customer and shopkeeper's locations and orientations were simulated at the level of typical stopping locations and proxemics formations, so extracting them from the raw data was unnecessary.

4.2.2 Input Action Vectorization. To input the customer and shopkeeper's previous actions into the neural network, they are vectorized. Customer and shopkeeper speech was vectorized using two separate binary bag-of-words models. In other words, the speech vectors were n-hot vectors with the vector values being 0 or 1 depending on whether the word appears in the utterance. Utterances were tokenized and stemmed. The customer's and shopkeeper's vocabularies contained 613 and 1,593 words, respectively.

Locations and proxemics were also vectorized. The locations vectors were of length 6 (the possible locations being *door*, *middle*, *service counter*, *camera A*, *camera B*, or *camera C*). For proxemics, one vector of length 4 was used to represent the spatial formation (*none*, *waiting*, *face-to-face*, or *present object*) and one of length 4 to represent the target of *present object* (*none*, *camera A*, *camera B*, or *camera C*).

To determine the participants' locations at runtime, their x, y positions would be matched to one of the typical stopping location clusters (4.2.1). However, since this work uses a simulated dataset, this step was unnecessary.

Thus, the input vector at each turn was of length 2,226. The first shopkeeper action vector s_{-1}, a place holder for before the customer enters the store, was set to all 0s.

4.2.3 Searching Shopkeeper Utterances for Product Information. The shopkeeper utterances were searched for mentions of product information using approximate string matching [36] so that later these parts can be dynamically updated whenever the contents of the database change.

The searching procedure consisted of preprocessing, approximate string matching, and candidate match filtering. First, the shopkeeper utterances and database contents were preprocessed (lowercase, remove punctuation, etc.) and tokenized, yielding sequences of word tokens. Then, an approximate string match algorithm[2] was applied to find subsequences among the shopkeeper utterance token sequences that approximately matched any of the database contents token sequences. The Levenshtein distance metric was used and the maximum distance set to $\frac{19}{30}$ based on empirical results. Approximate string matching yielded a set of candidate matches that were then filtered to remove overlapping matches and matches with non-matching numbers (e.g., prices).

The system must know which part of the output shopkeeper utterance to replace with the up-to-date database contents. Therefore, each mention of product information found by the search procedure was marked with an *attribute symbol*.

4.2.4 Shopkeeper Speech Clustering. The neural network outputs the shopkeeper's next action given the previous sequence of actions as input. Part of the next action is the shopkeeper's speech, which the system must learn from the human-human interaction data. However, this is challenging because data collected in the real world (where this method is intended to be applied) is messy, with many speech recognition errors and disfluencies. We solve this problem by applying a retrieval-based approach combined with speech clustering to choose the most well-formed utterances from among clusters of similar utterances (as in Liu et al. [30]).

Before the shopkeeper utterances can be clustered, they must be vectorized. We used an n-gram vector (n = 1,2,3) concatenated to a keyword vector, number vector, and attribute symbol vector to represent each utterance. n-Hot vectors were used, where the vector values were 0 or 1 depending on whether the term appears in the utterance. Keywords were extracted with the IBM Watson API.[3] To emphasize their importance, the keyword and number vectors were multiplied by a weight of 3 and the attribute symbol vector by a weight of 9.

To cluster the utterances into clusters of lexically similar utterances, inter-utterance Euclidean distances were computed and the Dynamic Tree Cut hierarchical clustering algorithm was applied [27]. The clustering procedure yielded 143 speech clusters. The neural network (Section 4.3) is trained to output a speech cluster ID, which dictates the system's next utterance.

To choose the most well formed from among all of the shopkeeper utterances, the medoid utterance was found for each cluster. The medoid utterance is the utterance most similar to all other utterances in the speech cluster. Since complete utterances with few ASR errors tended to share

[2] https://github.com/taleinat/fuzzysearch.

[3] https://www.ibm.com/watson/services/natural-language-understanding/.

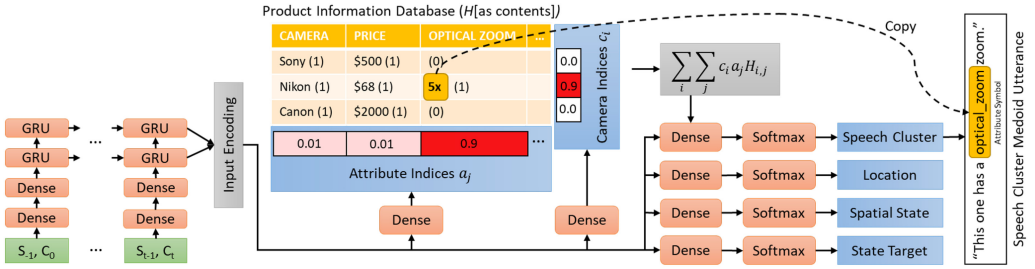

Fig. 4. The proposed neural network architecture showing inputs (green), outputs (blue), neural network layers (orange), internal vectors/scalars (gray), and components related to the product information database (yellow).

the most similarities with other utterances in the same cluster, medoid utterances tended to be well formed and easy to understand.

4.2.5 Target Output Action Vectorization. To train the neural network, the shopkeeper actions, which the network outputs, must be represented as vectors. The outputs include the next spatial state and state target, shopkeeper location, shopkeeper speech cluster, database attribute index, and database camera index.

Spatial state and state target were represented with one-hot vectors, each of length 4; shopkeeper location was represented by a one-hot vector of length 6; and shopkeeper speech cluster was represented by a one-hot vector of length 143 (i.e., speech cluster ID). Attribute indices and camera indices were represented by binary vectors of length 15 and 3, respectively. If the output shopkeeper utterance contained an *attribute symbol* (Section 4.2.3), the indices corresponding to the matching database contents were set to 1, and all other indices were set to 0.

4.3 Neural Network Architecture

The proposed network architecture (Figure 4) takes the customer and shopkeeper's previous actions during an interaction as input and outputs the shopkeeper's next action.[4]

4.3.1 Input Interaction Context Encoding. The sequence of customer and shopkeeper action vectors $s_{-1}, c_0 \ldots s_{t-1}, c_t$ (representing the actions that have occurred during an interaction) are condensed into a single *input encoding* vector representing the interaction context. To do this, they are first processed through two dense layers with leaky ReLU activation functions and then through a two-layer gated recurrent unit many-to-one sequence model [9]. The final outputs of both layers are concatenated to get the *input encoding*.

4.3.2 Relevant Product Selection. When the customer asks a question or the shopkeeper proactively introduces some information, the neural network must find the relevant information in the database. This is accomplished by using neural attention *camera* and *attribute* indices to address the database.

To get the indices, the *input encoding* is passed through two separate dense layers with sigmoid activation functions. The first outputs the *camera indices* (3 dimensions). The second outputs the *attribute indices* (15 dimensions). The sigmoid activations allow for selection of multiple relevant cameras and attributes.

[4]See footnote 1 for the link to the neural network code.

Sometimes the database does not contain the information requested by the customer. In these cases, the system must output an appropriate utterance, such as "Sorry, I don't have that information." To enable this behavior, a scalar, representing whether the database contains any information for the relevant camera(s) and attribute(s), is computed:

$$\sum_i \sum_j c_i a_j H_{i,j}, \tag{1}$$

where c_i is the camera index activation for row i, a_j is the attribute index activation for column j, and $H_{i,j}$ represents whether database cell i, j contains any information (0 or 1). This scalar is then passed as an input to the dense layer for computing the output speech cluster.

In cases where the database does contain relevant information, the *camera indices* are used to locate it (Section 4.3.3).

4.3.3 Output Shopkeeper Action Decoding. The neural network outputs the next shopkeeper action, which would then be executed by a robot. The action consists of the speech, location, spatial state, and state target.

To output the components of the shopkeeper's action, the *input encoding* vector is passed through four separate dense layers with leaky ReLU activation functions. To get the final output, each ReLU activation is passed through a softmax and the maximum dictates the next spatial state, state target, shopkeeper location, and speech cluster ID.

To get the output speech, the output speech cluster ID with maximum activation is mapped to that cluster's medoid utterance (Section 4.3.3). If that utterance contains any attribute symbols, the argmax of *camera indices* (Section 4.3.2) is used to find the value in the database and it is copied over the attribute symbol—for example, "This one has a 5x zoom" (Figure 4). In this way, the system can automatically update the shopkeeper's utterances when the contents of the product information database change.

To run the robot in live interaction with a human, the outputs of the neural network would be executed by the robot—that is, commands would be sent to the robot to move to a destination or speak an utterance [29]. The destination is determined by the *location*, *spatial formation*, and *state target* outputs. If the spatial formation is *waiting*, the robot will be sent to center of the stopping location cluster corresponding to the neural network's *location* output. When the spatial formation is *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. For *face-to-face*, the target location will not be a fixed location but rather a point in front of the customer. For *present object*, the target location will be the object of interest, determined by the *state target* output. Finally, the system's speech output would be executed by running it through the robot's text-to-speech system. In this work, however, an offline evaluation was conducted based only on the outputs of the neural network, so no actual action was executed by the robot.

5 EVALUATION

To determine the effectiveness of the proposed system at adapting its behaviors to the changing contents of the product information database, we conducted an offline evaluation comparing it to a baseline and a previous state of the art.

5.1 Experimental Conditions

COREQA is a previous state-of-the-art database question answering system that uses a modified recurrent neural network to generate answers [19]. In addition to generating output words as a

typical recurrent neural network does, it also computes a *copy score* and *retrieve score*, which allows outputting words from the input or a database, respectively.

He et al. [19] demonstrated that COREQA achieved an F1 score of 90.6 (87.4 precision, 94.0 recall) on a simulated question-answer dataset and a database filled with birthday information (year, month, day, and gender). The model was also evaluated on question-answer pairs obtained from a QA website, on which it achieved 56.6% accuracy. For these reasons, we believe this is a suitable model for comparison to the proposed system.

Our implementation of COREQA uses the input encoder from Section 4.2.2, and only uses the *retrieve score*, since this study focuses on retrieving information from a database. Database "facts"consisted of tuples of *camera ID*, *attribute*, and *value* (e.g., ["CAMERA B," "camera type," "compact digital camera"]), where each was encoded with a bidirectional gated recurrent unit network [9] of size 100. The retrieve-mode score, used to select which database fact is most relevant given the current state, was computed using a single dense layer of size 100. The vocabulary size was 1,702, and the output sequence length was 83.

Baseline is the same as the proposed system but without the mechanisms for addressing and copying from the database. Specifically, the parts that of the proposed system that were not used for the baseline were searching shopkeeper utterances for product information (Section 4.2.3), relevant product selection (Section 4.3.2), and copying of database contents into the typical utterance during action decoding (Section 4.3.3). For the baseline, a separate shopkeeper speech clustering (Section 4.2.4) was conducted, although without the *attribute symbols*. It yielded 553 clusters. The neural network architecture is the same as the proposed system but without the camera index and attribute index layers. By comparing the proposed system to this baseline, the effectiveness of the database mechanisms at adapting the system's behaviors can be determined.

Proposed is the full system described in Section 4.

5.2 Training the Systems

All three action predictors were trained using the same procedure. An 11-fold cross validation was conducted, in which interactions for nine of the simulated datasets were used for training (1,800 interactions containing an average of 29,520 customer input/shopkeeper output turns), interactions from one database were used for validation (200 interactions containing an average of 3,280 turns), and interactions from one database were used for testing. Finally, after training, the first 3 interactions from each of the testing sets (33 interactions containing 458 turns) were used for human evaluation.

The number of units in hidden layers was set to 100. All dense layers used He normal initialization [18]. All recurrent layers used Glorot normal initialization [14]. Cross entropy was used to compute the loss function. Moreover, since the attribute index targets are sparse, their losses were weighted inversely proportional to their number of occurrences in the training data based on King and Zeng [23]. Similarly, the speech cluster targets, whose occurrence in the training data has a long tail distribution, were also weighted inversely proportional to their number of occurrences. The Adam optimizer [24] was used to update the parameter values for 300 epochs with a learning rate of 0.0001.

5.3 Evaluation Procedure

One human evaluator, an internally recruited fluent English speaker (F, age 36) blind to the experimental conditions, evaluated the 458 action predictions made on the simulated interactions (three interactions per database). The evaluator assigned each a a binary label of either correct or incorrect. To receive a correct label, the system's action must be socially appropriate (e.g., appropriately greeting the customer, being polite, answering questions), and for cases where the shopkeeper

Table 4. Percentage of Output Actions That Were Judged to Be Correct by a Human Evaluator

Condition	Shopkeeper Utterances		
	Overall	Provides Information	Other
COREQA	37% *	3% *	70% *
Baseline	53% *	10% *	**94%**
Proposed	**90%**	**86%**	**94%**

Note: *Provides Information* is the subset of actions where the shopkeeper should provide product information. *Other* is the subset of all other actions. An asterisk (*) marks results that significantly differed ($p < .001$) with *Proposed*.

Table 5. Example Outputs from the Proposed System and the Most Common Reasons for Errors

Customer Speech	Output Shopkeeper Speech	Reason for Error
I'm looking for a professional camera.	This is a e mount interchangeable lens camera. ✓	N/A
Sounds good.	It is priced at $350. ✓	N/A
Is this a full manual camera? <...later...> Interesting ok.	The camera has both manual and automatic settings. <...later...> The settings on this one are manual and automatic settings. ✗	Repeated utterance
Ok thanks. <...later...> Excuse me.	[F]eel free to ask any other questions, I'll be here. <...later...> Good afternoon, how can I help today? ✗	Inappropriate greeting
Does it have like preset modes?	It has . ✗	Copied from empty DB cell
Does this camera have preset modes?	It's a 4k point and shoot camera with limited manual settings. ✗	Provided wrong attribute

provided information (from the product information database), it must be the correct information. In the case that the evaluator was unable to determine if the predicted action was correct or not, the instance was not included in the evaluation (0.5% of instances). Evaluations of 11% of the data (50 instances) by a second evaluator, an internally recruited fluent English speaker (M, age 31) blind to the experimental conditions, showed a substantial degree of agreement, with a kappa coefficient of 0.671, so the evaluations were judged to be reliable.

To better understand how well each system was at providing information from the database, we divided the overall set of output actions into two subsets: actions that should *provide information* about a product and *other* actions.

5.4 Results

The evaluation and statistical analysis results are shown in Table 4. Table 5 shows example speech outputs from the proposed system.

Overall, the proposed system performed the best with 90% correct actions. This was significantly better than the baseline with 53% ($\chi^2(1, N = 912) = 151.84$, $p < .001$) and COREQA with 37% ($\chi^2(1, N = 909) = 271.95$, $p < .001$).

On the *provides information* subset of actions, the proposed system performed the best with 86% correct. This was significantly better than the baseline with 10% ($\chi^2(1, N = 446) = 259.62$, $p < .001$) and COREQA with 3% ($\chi^2(1, N = 446) = 310.55$, $p < .001$).

On the *other* subset of actions, the proposed system had 94% correct. This was significantly better than COREQA with 70% correct ($\chi^2(1, N = 463) = 42.08$, $p < .001$), although not significantly different from the baseline with 94% correct ($\chi^2(1, N = 466) = 0.152$, $p = .697$).

5.5 Analysis of Errors

Although the proposed system performed better *overall* and on the *provides information* subset than the other systems, there were some errors (46 instances). The most common reasons for errors were repeated utterances (26% of errors), inappropriate greetings (22% of errors), copying from an empty database cell (15% of errors), and providing information about the wrong attribute (13% of errors) (Table 5).

The high proportion of repeated utterances (26% of errors) (e.g., reintroducing the same feature multiple times) suggests that the interaction history is not being sufficiently taken into account by the neural network. This could perhaps be fixed by fine tuning the hyper-parameters of the encoding dense and GRU layers, such as the number of units per layer and number of layers.

Inappropriate greetings (22% of errors) occur, for example, when the shopkeeper greets a customer for the second time as if meeting for the first time (row 5 in Table 5) or when greeting in a way that is not specific to the customer's greeting (e.g., C: "Sorry [to] ask you another question about this camera."S: "Good afternoon."). This is an artifact of the interaction simulation method; the shopkeeper's utterances were chosen from a pool for S_GREETS regardless of whether greeting for the first or second time and the specifics of the customer's utterance. As a result, the system learned to imitate such behaviors.

The instances of the system copying from an empty database cell (15% of errors) were caused by the proposed system incorrectly outputting speech templates (determined by speech cluster ID) for providing information (e.g., "It has <preset_modes>") instead of one stating that the database contained no information about the requested attribute (e.g., "I'm sorry, I don't have that information"). Further analysis showed that for some of these instances, the camera or attribute index outputs were not focused on a single camera or attribute, causing an incorrect calculation of whether or not the database contains relevant contents (1). This problem could perhaps be fixed by using Gumbel softmax [20] or some other mechanism to focus the index outputs to be categorical.

The instances of providing information about the wrong attribute (13% of errors) were caused by the proposed system incorrectly outputting speech templates for providing information about the wrong attribute. For these instances, since the customer speech often clearly indicated which attribute was requested (e.g., row 7 in Table 5), we hypothesize that the errors are caused by incorrectly attending to irrelevant features of the interaction history and not sufficiently to important keywords in the customer speech (e.g., "preset modes").

In contrast to the proposed system, the baseline system's most frequent reason for errors was providing incorrect information (77% of errors), which demonstrates the effectiveness of the proposed database retrieval mechanism. COREQA's most frequent reason for errors was word repetition (45% of errors), which demonstrates the effectiveness of the proposed system's retrieving medoid utterances of speech clusters, instead of COREQA's word-by-word sequence generation.

5.6 Robustness to ASR Errors

The proposed system is robust to ASR errors in two ways. First, it can provide correct shopkeeper responses to customer utterances that contain ASR errors. In fact, the proposed system responded correctly to 82% (51) of the customer utterances in the testing set that contained ASR errors. (The testing set contained 304 customer utterances, and 20% (62) of them contained ASR errors.) The proposed system is able to respond correctly to errorful customer utterances because it is trained on examples with similar errors. This is an advantage of a data-driven approach to learning interactive robot behaviors.

Second, the proposed system is able to filter out shopkeeper utterances that contain ASR errors when it learns to imitate the shopkeeper. This is accomplished by shopkeeper speech clustering and typical utterance selection (Section 4.2.4). During shopkeeper speech clustering, lexically similar

utterances are clustered together. Some of these utterances may contain ASR errors. However, for the system's outputs, one typical utterance is selected from each speech cluster (the medoid utterance), which is usually a grammatically well-formed utterance without ASR errors. In this way, the system learns to imitate shopkeeper speech without ASR errors.

In the current work, shopkeeper utterances with ASR errors are few because most (approximately 73%) of the simulated shopkeeper utterances came from crowdsourcing on the web. However, previous work has demonstrated that shopkeeper speech clustering results in significantly greater "correctness of wording" when ASR errors are present in the shopkeeper training utterances [29].

5.7 Action Correctness Depends on the Proportion of Information Providing Actions and the Dynamic Nature of the Database

The overall accuracy reported in Table 4 is influenced by the relative sizes of the provides information and other action subsets. In fact, the exact relationship is described by the following equation:

$$correct(P)\frac{|P|}{|P \cup O|} + correct(O)\frac{|O|}{|P \cup O|} = correct(P \cup O), \tag{2}$$

where P is the *provides information* subset of actions, O is the *other* subset of actions, $P \cup O$ is the *overall* set of actions, and $correct()$ is a function that returns the rate of correctness (as displayed in Table 4). Thus, the degree of improvement of the proposed system over the baseline system is proportional to the size of the *provides information* subset: when this subset is much greater than the *other* subset, the proposed system's correctness rate will be much greater than the baseline's correctness rate, and as the size of the *provides information* subset decreases, the performance of the two systems will become equal.

Furthermore, the degree of improvement of the proposed system's performance over the baseline system's performance depends on the dynamic nature of the product information database. When the information in the database does not change from training time to testing time, the proposed system's and baseline systems' performance is expected to be equal. When it changes a lot, the proposed system is expected to perform much better than the baseline.

6 DISCUSSION

6.1 Reality of the Dataset

This work is a proof-of-concept for how a learning system can be trained automatically on interaction data to automate social interaction behaviors, toward the goal that a robot may one day be deployed into real-world service scenarios. Since real-world data is not yet available, we demonstrated the proposed system using simulated data, containing examples of behaviors and interaction patterns that may be found in real human-human interaction. Granted, real-world interactions would contain greater variation of speech and non-verbal behavior than the simulated data, and therefore would be more difficult to learn from. However, by collecting greater amounts of real-world data using passive sensor networks, it will be possible to overcome this challenge and learn the repeatable, formulaic behaviors that make up the core of many service interactions.

Therefore, we believe that the simulated interactions are sufficiently similar to real-world interaction for the purpose of demonstrating this proof-of-concept. In the future, it would be interesting to apply data-driven imitation learning methods to real-world interaction data.

6.2 Generalizability

Ideally, the proposed system will work for any scenario characterized by highly repeatable, formulaic interactions. The proposed system worked for the simulated camera shop scenario, which has

these characteristics, suggesting it will work for other scenarios with these characteristics. However, it also makes some assumptions that may not hold true generally. Here we suggest how the system could be made more generalizable.

In the simulated camera shop scenario, it is assumed that there are only three products in the store, that they are all of the same type (cameras) with the same database table format, and that their location in the database (row) corresponds to their location in the store. Our evaluation did not explore these issues, but they could potentially be solved by using different database addressing mechanisms. In other words, using a (*product_type*, *product_ID*, *attribute*, *value*) tuple encoding similar to He et al. [19] would allow for an arbitrary number of products of any type. Furthermore, using multi-hop inference as in Sukhbaatar et al. [47] would enable location-based lookup to mitigate changing locations.

We also found that mentions of product information in the shopkeeper utterances (which were crowdsourced) often roughly matched the information in the product information database. This enabled the approximate string match algorithm to find mentions at the word level. However, in a real scenario, where shopkeepers are providing information from memory, these mentions would likely occur with alternate wordings. Therefore, more research needs to be done on algorithms that search for mentions based on *semantics* instead of surface word form.

6.3 Contribution and Implications

This work advances the state of the art in data-driven imitation learning for social robot behaviors by enabling a behavior learning system to automatically update shopkeeper (robot) speech about periodically changing products based on the contents of a product information database, without requiring retraining on newly collected data. To the best of our knowledge, the proposed approach is the first one to solve this problem, on which it achieved an action correctness rate of 90%. Moreover, comparing this to the baseline system's action correctness rate of 53% demonstrates the importance of the proposed mechanisms for addressing and copying from the database.

Lately, there has been much work on integrating information from the web, databases, and knowledge bases for question answering and dialog systems trained via machine learning (Section 2.2); however, none of these are suited for the problem that the proposed approach aims to solve. Namely, the proposed approach overcomes the shortcomings of generation based models (e.g., repeated words and susceptibility to ASR errors, as demonstrated and discussed in Section 5 by using a novel retrieval-based approach (i.e., outputting medoid utterances of speech clusters and copying contents directly from the database to the output) (Section 4.2.4 and Section 4.3.3). Furthermore, in contrast the previous retrieval-based approaches that use information from external sources, the proposed approach is able work for out-of-vocabulary database contents.

Finally, the proposed system's results are promising, but more research is required before such system can be robustly deployed to real-world stores. Specifically, the limitations presented earlier must be addressed, and further testing and analysis will be necessary to deal with challenges of learning from real-world data. In the future, we hope to extend the proposed approach to enable the robot to adapt to additional changing aspects of its environment that can be stored in a knowledge base, such as the store layout and the size, shape, location, and type of products present, and to test it in the real world.

7 CONCLUSION

In this work, a novel shopkeeper behavior learning system that updates its speech to account for periodically changing products was presented. This was accomplished by using approximate string matching to find mentions of product information in shopkeeper speech at training time, and replacing those mentions with information selected from a product information database by

neural attention mechanisms at test time. In a cross validation using a simulated camera shop interaction dataset, a human evaluator judged the proposed system to be better at outputting correct shopkeeper actions than a baseline and a previous state-of-the-art database question answering system.

REFERENCES

[1] Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*. https://openreview.net/forum?id=S1Bb3D5gg.

[2] Cynthia Breazeal, Nick DePalma, Jeff Orkin, Sonia Chernova, and Malte Jung. 2013. Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment. *Journal of Human-Robot Interaction* 2, 1 (2013), 82–111.

[3] Drazen Brscic, Takayuki Kanda, Tetsushi Ikeda, and Takahiro Miyashita. 2013. Person tracking in large public spaces using 3-D range sensors. *IEEE Transactions on Human-Machine Systems* 43, 6 (2013), 522–534.

[4] Sylvain Calinon and Aude Billard. 2007. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 255–262.

[5] Joyce Y. Chai, Rui Fang, Changsong Liu, and Lanbo She. 2016. Collaborative language grounding toward situated human-robot dialogue. *AI Magazine* 37, 4 (2016), 32–45.

[6] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. 2019. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA'19)*. IEEE, Los Alamitos, CA, 6015–6022.

[7] Sonia Chernova, Nick DePalma, Elisabeth Morant, and Cynthia Breazeal. 2011. Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *Proceedings of the 2011 IEEE RO-MAN Conference*. IEEE, Los Alamitos, CA, 21–26.

[8] Sonia Chernova, Jeff Orkin, and Cynthia Breazeal. 2010. Crowdsourcing HRI through online multiplayer games. In *Proceedings of the AAAI Fall Symposium: Dialog with Robots*. 14–19.

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078.

[10] Malcolm Doering, Dylan F. Glas, and Hiroshi Ishiguro. 2019. Modeling interaction structure for robot imitation learning of human social behavior. *IEEE Transactions on Human-Machine Systems* 49, 3 (2019), 219–231.

[11] Malcolm Doering, Takayuki Kanda, and Hiroshi Ishiguro. 2019. Neural-network-based memory for a social robot: Learning a memory model of human behavior from data. *Journal of Human-Robot Interaction* 8, 4 (Nov. 2019), Article 24, 27 pages. https://doi.org/10.1145/3338810

[12] Malcolm Doering, Phoebe Liu, Dylan F. Glas, Takayuki Kanda, Dana Kulic, and Hiroshi Ishiguro. 2019. Curiosity did not kill the robot: A curiosity-based learning system for a shopkeeper robot. *ACM Transactions on Human-Robot Interaction* 8, 3 (2019), 15.

[13] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-Tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 249–256.

[15] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, and Alan C. Schultz. 2005. Designing robots for long-term social interaction. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Los Alamitos, CA, 1338–1343.

[16] Nakul Gopalan, Dilip Arumugam, Lawson L. S. Wong, and Stefanie Tellex. 2018. Sequence-to-sequence language grounding of non-Markovian task specifications. In *Proceedings of Robotics: Science and Systems*.

[17] Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42, 1-3 (1990), 335–346.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*. 1026–1034.

[19] Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 199–208.

[20] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with Gumbel-Softmax. arXiv:1611.01144.

[21] Frederic Kaplan and Pierre-Yves Oudeyer. 2011. *From Hardware and Software to Kernels and Envelopes: A Concept Shift for Robotics, Developmental Psychology, and Brain Sciences*. Technical Report. Cambridge University Press.
[22] Tatsuya Kawahara. 2019. Spoken dialogue system for a human-like conversational robot ERICA. In *9th International Workshop on Spoken Dialogue System Technology*, Luis Fernando D'Haro, Rafael E. Banchs, and Haizhou Li (Eds.). Springer Singapore, Singapore, 65–75.
[23] Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political Analysis* 9, 2 (2001), 137–163.
[24] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.
[25] Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. 2013. Active learning for teaching a robot grounded relational symbols. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. 1451–1457.
[26] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the International Conference on Machine Learning*. 1378–1387.
[27] Peter Langfelder, Bin Zhang, and Steve Horvath. 2008. Defining clusters from a hierarchical cluster tree: The dynamic tree cut package for R. *Bioinformatics* 24, 5 (2008), 719–720.
[28] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. 2020. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion* 58 (2020), 52–68.
[29] Phoebe Liu, Dylan F. Glas, Takayuki Kanda, and Hiroshi Ishiguro. 2016. Data-driven HRI: Learning social behaviors by example from human-human interaction. *IEEE Transactions on Robotics* 32, 4 (2016), 988–1008.
[30] Phoebe Liu, Dylan F. Glas, Takayuki Kanda, and Hiroshi Ishiguro. 2017. Learning proactive behavior for interactive social robots. *Autonomous Robots* 42, 5 (2017), 1067–1085.
[31] Phoebe Liu, Dylan F. Glas, Takayuki Kanda, and Hiroshi Ishiguro. 2017. Two demonstrators are better than one—A social robot that learns to imitate people with different interaction styles. *IEEE Transactions on Cognitive and Developmental Systems* 11, 3 (2017), 319–333.
[32] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 285–294. https://doi.org/10.18653/v1/W15-4640
[33] Guilherme Maeda, Marco Ewerton, Takayuki Osa, Baptiste Busch, and Jan Peters. 2017. Active incremental learning of robot movement primitives. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL'17)*. 37–46.
[34] Cynthia Matuszek. 2018. Grounded language learning: Where robotics and NLP meet. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. 5687–5691.
[35] Amal Nanavati, Malcolm Doering, Dražen Brščić, and Takayuki Kanda. 2020. Autonomously learning one-to-many social interaction logic from human-human interaction data. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI'20)*. ACM, New York, NY, 419–427. https://doi.org/10.1145/3319502.3374798
[36] Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Computing Surveys* 33, 1 (2001), 31–88.
[37] Jeff Orkin and Deb Roy. 2007. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development* 3, 1 (2007), 39–60.
[38] Jeff Orkin and Deb Roy. 2009. Automatic learning and generation of social behavior from collective human gameplay. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 385–392.
[39] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V. Hafner. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation* 11, 2 (2007), 265–286.
[40] Pakpoom Patompak, Sungmoon Jeong, Itthisek Nilkhamhang, and Nak Young Chong. 2020. Learning proxemics for personalized human-robot social interaction. *International Journal of Social Robotics* 12, 1 (Jan. 2020), 267–280. https://doi.org/10.1007/s12369-019-00560-9
[41] Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M. Howard. 2018. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *International Journal of Robotics Research* 37, 10 (2018), 1269–1299. https://doi.org/10.1177/0278364918777627
[42] Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. 2018. Intrinsically motivated reinforcement learning for human-robot interaction in the real-world. *Neural Networks* 107 (2018), 23–33.
[43] Jurgen Schmidhuber. 2013. Maximizing fun by creating data with easily reducible subjective complexity. In *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 95–128.
[44] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. 3776–3783.
[45] Yiping Song, Rui Yan, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, and Dongyan Zhao. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*.

[46] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, New York, NY, 553–562.

[47] Sainbayar Sukhbaatar, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*. 2440–2448.

[48] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine* 32, 4 (2011), 64–76.

[49] Stefanie A. Tellex, Thomas Fleming Kollar, Steven R. Dickerson, Matthew R. Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*. 1507–1514.

[50] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J. Mooney. 2017. Opportunistic active learning for grounding natural language descriptions. In *Proceedings of the Conference on Robot Learning*. 67–76.

[51] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Nick Walker, Yuqian Jiang, Harel Yedidsion, Justin Hart, Peter Stone, and Raymond J. Mooney. 2019. Improving grounded natural language understanding through human-robot dialog. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA'19)*. IEEE, Los Alamitos, CA, 6934–6941.

[52] Nguyen Tan Viet Tuyen, Sungmoon Jeong, and Nak Young Chong. 2018. Emotional bodily expressions for culturally competent robots through long term human-robot interaction. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'18)*. IEEE, Los Alamitos, CA, 2008–2013.

[53] Sepehr Valipour, Camilo Perez, and Martin Jagersand. 2017. Incremental learning for robot perception through HRI. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'17)*. IEEE, Los Alamitos, CA, 2772–2777.

[54] Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*.

[55] Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W. Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1341–1350.

[56] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[57] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. 4623–4629. https://doi.org/10.24963/ijcai.2018/643

[58] Xuefeng Zhou, Hongmin Wu, Juan Rojas, Zhihao Xu, and Shuai Li. 2020. Incremental learning robot task representation and identification. In *Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection*. Springer Singapore, Singapore, 29–49. https://doi.org/10.1007/978-981-15-6263-1_3

Received May 2020; revised December 2020; accepted January 2021