smartTalk: A Learning-based Framework for Natural Human-Robot Interaction

Cameron Fabbri
Information Directorate
Air Force Research Laboratory
Rome, NY, USA.
cameron.fabbri@us.af.mil

Junaed Sattar
Computer Science and Engineering
University of Minnesota
Minneapolis, MN, USA
junaed@cs.umn.edu

***Abstract*—This paper presents a learning-based framework named smartTalk for natural-language human-robot interaction (HRI). The primary goal of this framework is to enable non-expert users to control and program a mobile robot using natural language commands. smartTalk is modality-agnostic, and is capable of integrating with both speech and non-speech (*e.g.*, gesture-based) communication. Initially, robots using this mechanism are equipped with a limited vocabulary of primitive commands and functionality; however, through extended use and interaction, the robots are able to learn new commands and adapt to user's behaviors and habits. This makes the proposed framework highly desirable for long-term deployment in a variety of HRI tasks. We present the design of this framework and experimental data on a number of realistic scenarios to evaluate its performance. A qualitative experiment on a robotic platform is also presented.**

I. Introduction

Effective interaction between humans and robots in any environment, and in any means of communication, relies heavily on the unambiguous exchange of information. In the case of human-to-human communication, spoken dialog is the predominant method of interaction, though gesture-based communication is fairly common, particularly in cases where speech-based communication is not possible (*e.g.*, being underwater, or having some form of communication handicap). For widespread deployment of robots in the society, an error-free, robust and natural method of human-robot interaction is necessary. It is not only crucial for robots to be able to respond and react to this form of conversation, but also have the ability to learn new instructions and extend their abilities by composing new functionalities from existing ones. However, mapping a natural-language command into an action by a robot "on-the-fly" remains a challenging task. An alternative approach would be to map novel instructions from a human user to an intent in a given task context. In the context of this paper, *intent* is the objective the user wants the robot to achieve. Applications in speech recognition, natural language processing, and machine learning have made it possible for a computer to determine the intent of a command issued by a human, while methods from human-robot interaction (HRI) have made it possible for robots to provide useful feedback to these commands given their environment. Our goal in this paper is to present a framework for natural language HRI which makes it possible for (a) non-expert users to interact with a robot, (b) a robot to infer a user's intent from commands previously unknown to the robot, and (c) a robot to extend its abilities by mapping the inferred intent into existing capabilities.

Fig. 1: A Husky AGV robot during a robot field trial. The exploration of large, unstructured outdoors environments is a likely scenario for adaptable natural language communication in human-robot teams.

Current HRI research has been increasingly incorporating natural language processing (*e.g.*, [1]) as speech recognition has become more accurate as of late, though issues with natural language processing pose certain challenges and complications. Associating *natural* (spoken) dialog with an action performed by a robot is often a complex process involving a multitude of factors, and consequently, robot-human communication often relies on more realiable, engineered, and *artificial* approaches. However, in a real-world environment, this natural method of communication is not only desired, but can be deemed necessary. This disconnect between intuitive communication and robot functionality poses a divide between robots performing their functions in a laboratory setting and in a real-world environment. To address this issue, we have built a framework called smartTalk, which provides robotics researchers the ability to link the intent of spoken dialog with robot functionality, as well as give useful user feedback.

A. Motivation

The proposed work enables people and robots to communicate through natural language, which may or may not be communicated aurally. In fact, even being geographically dispersed and without having a line-of-sight contact, the SMARTTALK framework can be used as a *back-end* for wireless, text- or gesture-based interaction system. Motivation for our work stems from the need to perform large-area searches for lost campers or hikers in the wilderness (*e.g.*, in mountainous regions or national parks), which is a demanding and high-risk task even for experienced rescue professionals and first responders, particularly in cold and harsh winter seasons. Additional applications exist in healthcare and caregiving tasks which, with the deployment of robotic caregivers, smart wheelchairs [2] and autonomous monitoring, are rapidly becoming more automated. Both of these scenarios require robotic systems to have natural language interaction capabilities to be deployed effectively.

B. Related Work

A large body of literature addresses natural language communication in general and in applications of HRI in particular, though few combine learning approaches with interaction. Learning-by-demonstration is a popular method for teaching robots new capabilities that attempts to learn low-level motor primitives from human demonstrations to be repeated independently by robots in future tasks [3]. Visual cues are often used alongside (or in lieu of, for example in the underwater domain) spoken dialog by several researchers for communication between robots in a network comprised of heterogeneous robots, for example by Dunbabin et al. [4]. The robotics literature has extensive examples of gesture-based robot control, particularly for direct Human-Robot Interaction (HRI). Both explicit as well as implicit communication frameworks between human operators and robotic systems have been considered (*e.g.*, [5]–[7]). Pateras et al. uses fuzzy logic to reduce uncertainty to reduce high-level task descriptions into robot sensor-specific commands in a spoken-dialog HRI model [8]. Montemerlo et al. have investigated risk functions for a safer navigation of a robotic nurse in the care of the elderly [9]. Researchers have looked into POMDP formulations of human-robot dialog models [10], [11] and planning cost models for efficient human–robot interaction tasks [12], [13]. Work also exists in enhanced natural language dialog for human-robot communication [14], real-time task planning [15], HRI for multiplayer games [16], and the generation of directed questions to reduce uncertainty in natural language human-robot communication [17].

II. TECHNICAL APPROACH

In order to deploy robots to perform a set of tasks, a natural method of communication must be established. The three main aspects of natural communication we wish to address are:

1) Communication using spoken dialog,
2) Associating a command with a specific set of robot actions, and
3) Bi-directional communication enabling feedback to the user.

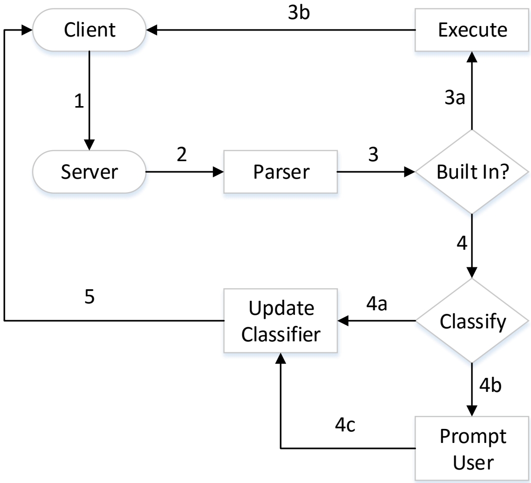

Fig. 2: System flow diagram in the SMARTTALK framework.

SMARTTALK uses a client-server model for communication between an end-user and robot. The *client* is the entity interacting with the robot; *e.g.*, a smart phone being used for human-robot communication for geologically distributed deployments, or a human user during direct interaction with a robotic wheelchair are both considered to be clients. Figure 2 shows an example task flow of the system assuming a user with a smart phone acting as the client, which can be broken down into the following steps:

1) The user running a client application speaks into the device, which uses voice recognition to parse speech into text and sends it as a command to the server.
2) The server runs the command through the parser
3) The parser checks if the command is an existing one. If it is, then
   a) the robot will execute its corresponding action and
   b) send acknowledgment feedback to user.
4) If the command is not an existing one, then the current classification model will be used to determine its "class" (see Section II-B below).
   a) If the classifier determines that the likelihood of a command being given passes a certain threshold, then the command is successfully classified, and the model is updated with the command and the associated label.
   b) If the command did not pass the threshold, the user is prompted to provide more information so that a label can be applied to the command.
   c) In either case above, the classifier shall be updated with new information.
5) Provide task feedback to the user.

By applying a client-server model for communication, a user with any device connected to a network with voice recognition capabilities has the ability to communicate with the robot. The following subsections provide descriptions of the various components of the SMARTTALK framework.

A. Natural Language Input

For a robot to perform any type of analysis on natural language input from a user, particularly in speech form, it must first be translated into text. Our implementation currently uses Google's speech recognition capability, which adapts the learned speech model to individual users, resulting in highly accurate speech recognition. For disjoint locations between the user and the robot, a smart phone acts as the client and is only responsible for translating the user's speech to text, as well as sending and receiving messages. When using SMARTTALK locally without a medium such as a smart phone, speech recognition occurs on the server.

B. Classification

To understand the user's intent, SMARTTALK uses a multi-class Naive Bayes classifier [18]. This model applies a feature extractor indicating which words in the current model are contained in the given command. For example, the sentence "This is sharp" may consist of the features "contains(sharp): True" or "contains(dull): False". This classification technique assumes that every feature value is independent of every other feature value given the class variable. The method for assigning a class label $y = C_k$ for some k is shown in equation 1:

$$y = \underset{k=1 \rightarrow K}{\operatorname{argmax}} \, p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{1}$$

where x is a vector $x = (x_1, x_2, ..., x_n)$ representing n features. Commands input into the classifier are checked by an uncertainty threshold which determines whether or not the robot should proceed with the given command or prompt the user for more information. If the probability passes the threshold, then the current model is updated with the new data, the command, and the predicted label. If the probability does not pass the threshold, then the user is prompted to provide the correct classification for that command, and again the model is updated. Through this model, SMARTTALK is able to continuously learn throughout its usage.

Commands received by the server are parsed in two steps. The first step is checking whether or not the given command is built-in (*i.e.*, preexisting). Because our classification model loses accuracy when classifying very short commands (one or two words), the built-in command feature alleviates that problem by providing mapping from short phrases to robot actions. SMARTTALK has two built-in commands for learning purposes: `train` and `test`. The `train` command allows users to give more examples for previous commands, as well as give a new command to be classified. The `test` allows a user to view what class a command is classified as without actually updating the model. This is to prevent incorrect learning during early stages of use, or when using commands with very little training data. Users also have the ability to write their own built-in commands for various tasks.

C. User Feedback

With no feedback mechanism, the user is left without any knowledge of the robot's intentions, confusions, or concerns. This ambiguity can lead to unexpected results, and provide no information as to how those conclusions were reached by the robot. The three types of feedback we are most concerned with are *acknowledgment*, *risk assessment*, and *uncertainty*. Acknowledgment provides the user with the knowledge that a given command was correctly classified, and that the knowledge base has expanded. This confirmation for accepting a given task allows a human to direct their focus on a different task at hand, knowing that the robot knows what it needs to do. The assessment of risk for a given command provides a safe, and ultimately successful interaction [19]. Because our system provides intent, it is possible for a command to be successfully understood in the context of the classification model, but also carry significant risk in execution – risk which may not be possible to estimate simply from the given command. Without measuring task context or assessing the environment through sensory perception, it would not be possible to build a dynamically updating (and accurate) model of risk involved in task execution, nor would it be possible to assign priorities in the presence of significant risk. SMARTTALK provides the ability to use this calculated risk assessment during the interaction; however the framework itself does not address risk computation. This risk assessment value must pass a threshold for the robot to execute the command.

Uncertainty is communicated back to the user during cases of low likelihoods for every class given by the classifier. By placing a threshold on the classification probability of a given command, we can determine when the robot is unable to infer the intent of that command. In this case, the robot provides the intent it is most confident with, and prompts the user to confirm that is indeed what they intended. If not, then the robot again prompts the user for more information to better classify the command. This confirmation updates the classification model for understanding the intent of a new command with a previously known action. When given a completely new command for an action which does not yet exist, the robot will be able to classify this command by updating its model, but will be unable to link that new command to an action. However, given a new complex instruction, the robot, with additional information from the user, will be able to represent it by combining existing primitives. In this way, we are able to form a "macro" for this instruction. As an example scenario, consider a quad-copter with a classification model only equipped with two instructions, "tilt up", and "fly straight". Consider a new command, "loop the loop", for which the quad-copter has no function for. This command can be broken down into two primitives: "fly straight", "continuously tilt up". SMARTTALK makes it possible for the user to construct the instructions by dictating, "fly straight, continue to tilt up until you are level with the ground, then fly straight again". By

Command	Label
bring my plates to the kitchen and put them on the counter	deliver
put my clothes in the hamper	deliver
put my mug into the dishwasher please	deliver
bring my trash outside and put it in the barrel	deliver
throw my shoes in the closet	deliver
take this glass to the kitchen	deliver
get me my shoes from the closet	get
bring me the remote	get
go grab my clothes from the dryer	get
can you go get the mail please	get
can you go to the cellar and get me a bottle of soda	get
hurry up and get me my keys, weŕe late	get
clean up this mess	clean
sweep up the kitchen	clean
pick up all of my clothes off of the ground and put them away	clean
get the vacuum out of the closet and vacuum the hallway	clean
can you pick up all this junk on the ground	clean
how much battery do you have left	communicate
hey robot where are you right now	communicate
what time is it	communicate
do you know where I left my car keys	communicate
what is the temperature in here	communicate

TABLE I: Training data depicting interactions with a home-service robot. The "Label" column shows the intent for the given command.

combining primitives in this way, a robot is able to learn new functionality given individual functions which it already knows how to perform. If there exists no primitives for a new command, SmartTalk would still be able to classify it correctly; obviously the functionality needs to be mapped to some action by the user *a priori* to be useful. Nevertheless, SmartTalk provides the mechanism to detect and construct the understanding of novel instructions, and the ability to communicate this to the user.

III. Experiments

We test our classification model with simulated robots in different environments, using commands commonly used in each domain. For each environment, we use a small data set to train a Naive Bayes classifier. We then proceed to test that model on a holdout testing set to find the accuracy and a confusion matrix, which allows us to visualize the performance of our classifier. Each column in the confusion matrix represents the instances in a predicted class, whereas each row represents the instance in the actual class. Due to space limitations, not all testing data is provided; however, the training and testing instructions for the service robot and the hospital robot scenarios (described below in Sections III-A and III-B) are shown in Tables I, II, III, and IV respectively. SmartTalk has been implemented in Python, and is currently hosted on GitHub[1].

A. Service Robot

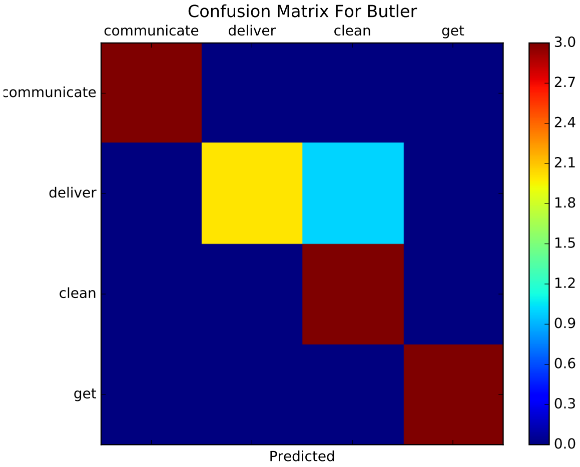

Fig. 3: Confusion Matrix in the service robot scenario.

Tailored towards the elderly or disabled, we show a simple classification model for a robot aiding users with tasks at home. Physical limitations, such as being wheelchair bound, unable to use hands, etc. could inhibit the user from communicating with the robot using gestures, text, or other type of communication. Spoken dialog provides communication given these physical limitations, while also functioning as the most common and natural form of communication for people in their homes. Our training set consisted of four labels common to commands for household activities: *deliver, get, clean*, and *communicate*. With five training examples for each label, we achieve an accuracy of 91.6%. Figure 3 displays the confusion matrix.

B. Hospital-care Robot

Hospitals are already making use of robots within operating rooms. We present a case for which a robot could aid the

[1] https://github.com/cameronfabbri/smartTalk

Command	Label
take my plate and glass back to the kitchen please	deliver
put this shirt on my bed please	deliver
can you put all of the dishes into the dishwasher	deliver
get my car keys from the drawer	get
go to the fridge and grab me a beer please	get
go get me another pair of socks from my dresser	get
sweep up all of these crumbs on the floor	clean
can you vacuum all of the bedrooms	clean
pick all of this garbage off of the ground	clean
how much battery do you have	communicate
what's the time	communicate
do you know where I put my wallet	communicate

TABLE II: Testing data for the home-service robot.

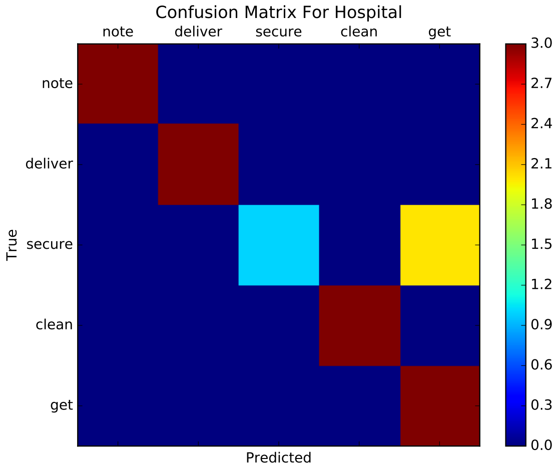

Fig. 4: Confusion Matrix for a hospital-care robot.

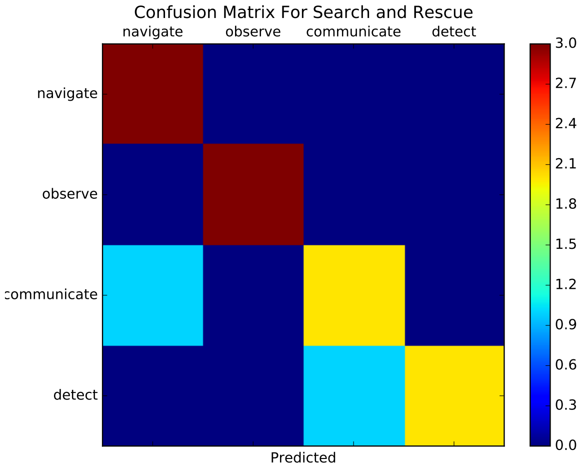

Fig. 5: Confusion Matrix for a search-and-rescue robot.

nurses on the hospital floors. In many hospitals, nurses are very short staffed, leaving only one nurse with multiple patients. For tasks that do not require in-depth knowledge or intuition a doctor or nurse possesses, a robot could be substituted to provide better care for patients. Delivering food, cleaning up, and fetching tools are some examples of these tasks. In many hospitals, there are rooms that contain high radiation that are very harmful to humans, for which the use of robots could be a possibility. During patient inspection, the use of a robot could also provide information collection and recollection, alleviating the need for manual note taking. Using five training samples for each label, we achieve an accuracy of 86.6%. Figure 4 displays the confusion matrix.

C. Search and Rescue

Many search and rescue teams encounter dangerous situations, whether it be the environment, the weather, or wildlife. We can increase the safety of humans in these situations by placing most of the risk on the robot. Humans can avoid exploring dangerous mines, caves, or various types of rubble, while still obtaining control of the robot and receiving feedback with information. Using five training examples for each label, we achieve an accuracy of 83.3%. Figure 5 displays the confusion matrix.

D. UAV

Unmanned air vehicles are becoming more popular as a way of conducting surveillance or providing assistance in a number of situations. However, UAVs control mechanisms are usually complicated and require rigorous, specialized training. With the use of spoken dialog, a less strenuous method of control can be achieved, as a simplified *wrapper* over the standard control mechanism. Using five training examples for each label, we achieve an accuracy of 80%. Figure 6 displays the confusion matrix.

E. Underwater Robot

Search and rescue, inspection, monitoring, and repairing are all tasks robots working in the ocean are capable of completing. Human moderators on the boat could be able to control these using natural language, instead of sending divers down into the water, potentially putting them in danger. Using five training examples for each label, we achieve an accuracy of 91.6%. Figure 7 displays the confusion matrix.

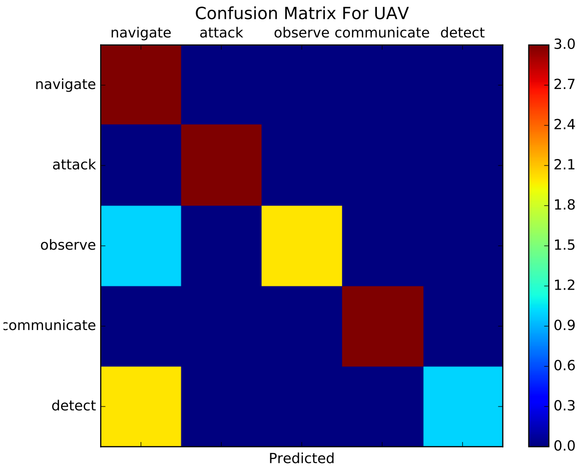

Fig. 6: Confusion Matrix for an unmanned aerial vehicle.

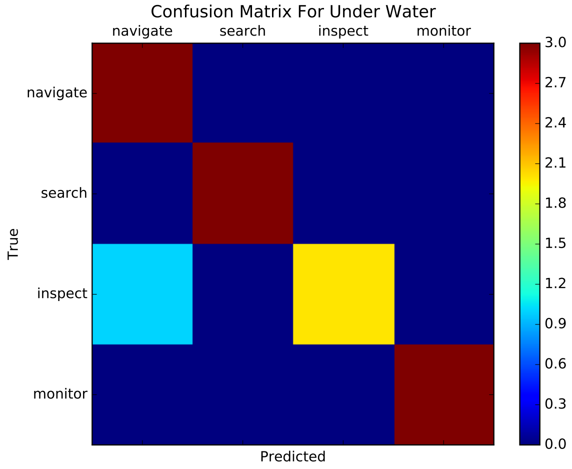

Fig. 7: Confusion Matrix for an underwater robot.

Command	Label
bring the medicine to the patient in room 110	deliver
give these papers to a nurse on floor 3	deliver
take these papers and bring them back to my office	deliver
bring this tray to room 366	deliver
bring these flowers to the front desk on floor 4	deliver
go to the operating room and give the doctor the scissors	deliver
go to the front desk and ask for patient number 45's papers	get
go to room number 32 and bring back the food trays	get
get me my glasses from my office	get
bring me the stethoscope from the room next door	get
find Mary and tell her to come here	get
go get the mop and bucket from the janitors closet	get
take vital signs from the patient in room 132	get
remember this	note
okay take a note	note
listen up	note
sweep the hallway please	clean
mop up the blood on the floor	clean
make all of the beds	clean
go to room 341 and mop the floor	clean
don't let anybody else in the door	secure
make sure all the doors are locked on this floor	secure
stop the people coming from the floor below	secure

TABLE III: Hospital robot training data.

IV. Robot Trials

To demonstrate the usability of the SMARTTALK framework, a qualitative interaction experiment on a physical robot were carried out by using natural language to control a robot. The robot tested was a custom built indoor robot composed of an Arduino UNO microcontroller, equipped with an ultrasonic distance sensor and a Raspberry Pi 2 Model B computer. The robot base is complete with a body and differential four-wheel drive, as seen in Figure 8. Indoor experiments were conducted using SMARTTALK running on the Raspberry Pi. Commands sent to the Raspberry Pi were parsed by SMARTTALK, allowing the execution of the robot's actions controlled through the Arduino. Flexibility on the parsing engine by use of text classification allows for a natural flow of speech as opposed to very specific and limited set of built in commands.

Field experiments on an outdoor robot (a Clearpath Robotics Husky 200 AGV, using a Google Glass for natural language interaction) have also been conducted. In those experiments, the Husky was commanded through the Google Glass to perform basic navigation tasks, analogous to those that a human user would give a robot to explore an (indoor or outdoor) environment. Experience from these tests show that voice recognition in an indoor environment resulted in a high-accuracy recognition. Connection using the Google Glass was done through a TCP socket over Wireless Ethernet. Experiments using SMARTTALK were conducted through a client computer connecting to the Raspberry Pi over a TCP socket. The functionality of the robot was to move forward, backward, turn left, turn right, and stop. Trials concluded that the robot was controllable by using a variety of commands by using our classification model. Some example commands are: "Okay robot drive forward", "Go forward", "Stop moving", "I'd like you to turn right now", and "Turn left". Throughout this interaction, SMARTTALK provided the link between the robot's functionality and natural language input, as well as the

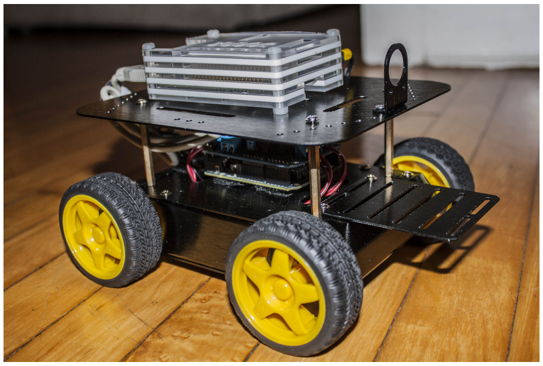

Fig. 8: Custom-built test robot equipped with a Raspberry Pi computer and Arduino microcontroller.

bring these bottles of percocet back to the lab	deliver
take these pills and bring them to the patient in room 12	deliver
give bob his lunch, he is in room 6	deliver
go get the scissors from the nurse	get
can you bring me the medicine from the lab	get
go get the doctor, quickly	get
listen to me	note
I need you to remember this	note
alright take a note	note
go to the operating room and mop up the floor	clean
clean up the OR	clean
pick up the trash in the hallway	clean
make sure the doors are locked	secure
stop that man from running away	secure
close that door and lock it	secure

TABLE IV: Hospital robot testing data.

continuation of understanding the user's *intent* by updating its classification model with each successfully identified class.

V. Conclusions

This paper presented a learning-based framework for natural language human-robot interaction called SMARTTALK. We present the design and capabilities of the framework, and present experimental evaluation of the classification system and brief on-board robot trials. Current performance demonstrate its utility by allowing non-expert users to communicate with a robot and also enabling a robot to understand novel instructions by engaging in interactive dialog with the user.

Future work will focus on deployment of SMARTTALK on a variety of robots, including underwater robots, thus extending our work to visual, gesture-based communication. We plan to incorporate a risk-assessment framework with SMARTTALK to provide another source for learning and differentiating between potentially safe and unsafe commands preemptively. Ongoing work is investigating approaches to couple interaction with robot learning in a wider scale to minimize the disconnect between a robot's abilities and a human user's abilities to command a robot to perform tasks in arbitrary domains.

Acknowledgment

The authors are grateful to the Department of Computer Science at Clarkson University for supporting this research and providing the Husky AGV for field trials.

References

[1] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[2] P. TalebiFard, J. Sattar, and I. M. Mitchell, "A Risk Assessment Infrastructure for PWC Motion Commands without Full Sensor Coverage." in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, IL, USA, September 2014, pp. 3592–3597.

[3] M. J. Mataric, "Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics," in *Imitation in Animals and Artifacts*. Citeseer, 2000.

[4] M. Dunbabin, I. Vasilescu, P. Corke, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle," in *International Conference on Robotics and Automation, ICRA 2006*, Orlando, Florida, May 2006.

[5] D. Kortenkamp, E. Huber, and R. P. Bonasso, "Recognizing and interpreting gestures on a mobile robot," in *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2*, ser. AAAI'96. AAAI Press, 1996, pp. 915–921.

[6] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 34, no. 2, pp. 154–167, May 2004.

[7] A. Xu and G. Dudek, "Towards Efficient Collaborations with Trust-Seeking Adaptive Robots," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot InteractionHRI*, ser. HRI'15 Extended Abstracts. New York, NY, USA: ACM, 2015, pp. 221–222. [Online]. Available: http://doi.acm.org/10.1145/2701973.2702711

[8] C. Pateras, G. Dudek, and R. D. Mori, "Understanding Referring Expressions in a Person-Machine Spoken Dialogue," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1995. (ICASSP '95)*, vol. 1, May 1995, pp. 197–200.

[9] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma, "Experiences with a mobile robotic guide for the elderly," in *Proceedings of the 18th National Conference on Artificial Intelligence AAAI*, 2002, pp. 587–592.

[10] F. Doshi, J. Pineau, and N. Roy, "Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs," in *Proceedings of the 25th International Conference on Machine learning (ICML)*. ACM New York, NY, USA, 2008, pp. 256–263.

[11] F. Doshi and N. Roy, "Spoken language interaction with model uncertainty: An adaptive human-robot interaction system," *Connection Science*, vol. 20, no. 4, pp. 299–318, 2008.

[12] K. Krebsbach, D. Olawsky, and M. Gini, "An Empirical Study of Sensing and Defaulting in Planning," in *Artificial intelligence planning systems: proceedings of the first international conference, June 15-17, 1992, College Park, Maryland*. Morgan Kaufmann Publishers, 1992, p. 136.

[13] D. Kulic and E. Croft, "Safe Planning for Human-Robot Interaction," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2004.

[14] M. Scheutz, R. Cantrell, and P. Schermerhorn, "Toward humanlike task-based dialogue processing for human robot interaction," *AI Magazine*, vol. 32, no. 4, pp. 77–84, 2011.

[15] R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz, "Tell me when and why to do it!: Run-time planner model updates via natural language instruction," in *Proceedings of the 2012 Human-Robot Interaction Conference*, Boston, MA, March 2012.

[16] S. Chernova, N. DePalma, and C. Breazeal, "Crowdsourcing real world human-robot dialog and teamwork through online multiplayer games," *AI Magazine*, vol. 32, no. 4, pp. 100–111, 2011.

[17] S. Tellex, P. Thaker, R. Deits, D. Simeonov, T. Kollar, and N. Roy, "Toward information theoretic human-robot dialog," in *Proceedings of Robotics: Science and Systems VII*, Sydney, New South Wales, Australia, June 2012.

[18] T. Misu and T. Kawahara, "Bayes risk-based dialogue management for document retrieval system with speech interface," *Speech Communication*, vol. 52, no. 1, pp. 61–71, 2010.

[19] J. Sattar and J. J. Little, "Ensuring Safety in Human-Robot Dialog – a Cost-Directed Approach," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA.*, Hong Kong, China, May 2014, pp. 6660–6666. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5979633