Task Based Dialog For Service Mobile Robot

Vittorio Perera, Manuela Veloso
vdperera@cs.cmu.edu, mmv@cs.cmu.edu
School of Computer Science
Carnegie Mellon University

CoBot is a service mobile robot that has been continuously deployed for extended periods of time in a multi-floor office-style building (Biswas and Veloso 2013). While moving in the building CoBot, is able to perform multiple tasks for its users; the robot is able to autonomously navigate to any of the rooms in the building, to deliver objects and messages and to escort visitors to their destination. While apparently very different, all the tasks CoBot is able to perform require the robot to move from one location to another.

Often, only being able to move, is not enough to accomplish the task required; if CoBot needs to deliver an object, given that it does not have arms, it cannot pick it up by itself, similarly when it needs to travel across floors it cannot push the elevator button. In order to overcome its limitation CoBot ask for help to humans, either the user or bypasser, achieving symbiotic autonomy (Rosenthal, Biswas, and Veloso 2010).

In our effort to make CoBot increasingly available to users in the building, we recently enabled it to understand spoken commands (Kollar et al. 2013) (previously you could request tasks only using a website); our research on spoken interaction pushed forward with the development of KnoWDiaL (Perera et al.), an approach for Learning and using task-relevant Knowledge from human-robot Dialog and access to the Web. We frame the problem of understanding the commands received, as finding a mapping from the audio input to a semantic frame describing the task the robot should execute. A frame is described by an action, invoking the frame itself, and a set of parameters, i.e. the frame elements. Currently CoBot is able to understand spoken commands only for two of its tasks, going to a specific location and delivering objects; Fig.1 shows the semantic frames for each of them.

When Cobot receives a command, as in Fig. 2, it records the audio input and then processes it using an Automated Speech Recognizer to get a set of speech transcriptions. Next, KnoWDiaL parses each of the speech transcriptions. To do so we trained a model, using a Conditional Random Field, that assign to each word one of the following labels: *action, toLocation, fromLocation, toPerson, fromPerson, object*. Each of the labels used corresponds either to the action of a semantic frame, or to one of the frame element; for this reason, we call this step semantic parsing. For each speech transcription, we then greedily chunk together words with the same label.

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Frame:	*GoTo*
- Parameters:	destination
Frame:	*DeliverObject*
- Parameters:	object, source, destination

Figure 1: Semantic frames of the two task CoBot is able to execute from spoken commands.

Figure 2: A user giving a spoken command to CoBot

Once the speech transcriptions have been parsed KnoWDiaL still needs *ground* them. The grounding process is divided in two steps, first to identify which frame is invoked and second to find a grounding for each of the parameters of the frame invoked. KnoWDiaL resorts to a Knowledge Base to ground the command received. To identify the frame invoked by the command, KnoWDiaL queries the KB for all the predicates of type *actionGroundsTo* matching the chunks labeled as *action*. The KB also stores a confidence for each of its predicates and, using Bayesian inference, we come up with the most likely frame invoked by the command received.

Once the frame has been identified KnoWDiaL needs to ground all its parameters. Currently it is possible to give

spoken commands to Cobot to execute two different tasks, namely moving to a specific location (*GoTo*) and delivering an object (*DeliverObject*). In order to ground the frame parameters, KnoWDiaL queries the Knowledge Base for predicates matching the labels returned by the parser. For instance, if the action identified is *GoTo* and the parser returns a chunk labeled as *toLocation*, the predicate queried will be *locationGroundsTo*; similarly if the chunk returned by the parser is labeled as *toPerson*, the predicate queried is *personGroundsTo*. Since all of the tasks CoBot can perform involve moving from one location to another, the grounding for frame parameters are, in general, x, y coordinates on its map, represented by four digit numbers associated with rooms in the building. Once all the parameters have been identified and correctly grounded, Cobot can execute the command received.

Unfortunately the Knowledge Base does not always contain the information needed to identify the frame corresponding to the commands received or to ground all of its parameters. CoBot then explicitly asks for the element it's missing; for instance, if the command received refers to the *GoTo* task but it was not possible to ground the destination, the robot will ask "Where should I go? Please give me a four digit room number."

Once the user answers all the question needed to ground the command, the robot has all the information to execute it. Before starting to move, the robot asks one more confirmation question and then starts executing the command received. Asking a confirmation question lets the user make sure the command was properly understood but also gives KnoWDiaL the opportunity to updated its Knowledge Base with new facts learned by asking explicit questions about the missing parameters.

The approach described above is effective in enabling CoBot to understand spoken commands but it also opens up many questions that we are currently addressing. Hereafter is a list, together with a short description, of ongoing research on CoBot and its interaction with users.

- **Confirmation request:** We clearly would like a robot to go to the correct location corresponding to the user request. However, given that the robot is processing speech with inevitable noise, and that the robot is using a knowledge base learned from interactions with humans, who may have or not been correct, it remains a question to decide when the robot should ask for confirmation about its eventual grounding choices.
- **Task explanation:** The outcome of the interaction with the human is the specification of a task that the robot then plans to accomplish. As such planning will involve the consideration of different routes, possibly times and coordination with other tasks, it remains a question to decide how much of such planning process should the robot inform the user about. This question is also particularly relevant if the request was complex with conjunctions, disjunctions, and conditionals, which the autonomous robot then plans according to its own task optimization criteria, which may or may not need to be explained or communicated to the human.
- **Type of interaction:** A symbiotic autonomous service robot interacts with different types of humans, namely four types: (i) the task requester, (ii) the task recipient, (iii) the potential helpers (press elevator buttons, put coffee in basket), and (iv) the random bystanders who are not related to the task. The human-robot interaction needs then to be well aware of the different type of humans, and a question remains on how to adapt and adjust the interaction with each human.
- **Delivering messages:** The basic speech interface lets the robot understand commands relating to two tasks. We would like to extend this interface to all the tasks CoBot is currently able to perform. In particular we are interested in having our robot delivering messages in a natural way. For instance if Vittorio asks CoBot "Please tell Manuela I will be late for our meeting", once the robot gets to Manuela's office, it should report the message as "Vittorio said he will be late for your meeting" rather than simply repeating the message received.
- **Asking unrelated questions:** Currently CoBot asks questions only when it is not able to ground part of the command received. We can imagine the robot also asking questions not directly related to the command received, which leads to two different problems: (i) what information is useful to a robot and should it ask about? (ii) when is it ok to ask questions that do not relate to the command received?
- **Social interactions:** As of now, we assume all the speech directed to the robot is asking it to perform a task. Unfortunately users, especially inexperienced ones, have very high expectations for robots, and will ask them all sorts of things. We should be able to reliably distinguish what refers to a command and what does not; moreover, if the speech received does not refer to a command, we need to design appropriate ways to handle it.

In conclusion, we have enabled our service mobile robot, CoBot, to understand spoken commands. The approach used proved to be effective, but it also opened new research paths that we hope will contribute to increase our understanding on how to handle a interaction between the user and a service mobile robot.

References

Biswas, J., and Veloso, M. M. 2013. Localization and navigation of the cobots over long-term deployments. *The International Journal of Robotics Research* 32(14):1679–1694.

Kollar, T.; Perera, V.; Nardi, D.; and Veloso, M. 2013. Learning environmental knowledge from task-based human-robot dialog. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 4304–4309. IEEE.

Perera, V.; Soetens, R.; Kollar, T.; Samadi, M.; Sun, Y.; Nardi, D.; Van De Molengraft, R.; and Veloso, M. Learning task knowledge from dialog and web access. *In preparation for submission*.

Rosenthal, S.; Biswas, J.; and Veloso, M. 2010. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 915–922. International Foundation for Autonomous Agents and Multiagent Systems.