The Creation of a Dialogue Management System

Anna Flynn

Completed as part of the Distributed Research Experience for Undergraduates Program.

Robotics Interaction Lab, University of Southern California. 2014

Introduction: What is a Dialogue Management System?

Human-computer interaction is an area of computer science in which tremendous advancements have been made in the past decade. Computers are more user friendly than ever and users have a myriad of options that were not possible before in how they interact with computers in their day to day lives. A subset of HCI is the interaction between humans and humanoid robots, ie a dialogue exchange between a human user and a computer that is meant to imitate and represent a human as closely as possible. These interactions are becoming more interesting as robots become more sophisticated in terms of speech recognition and processing, creating more natural interactions and expanding the opportunities in which human robot interaction can be used. It should be noted that these dialogues between humans and robots have different degrees of complexity. For example some information can be conveyed in a single statement, some programs may need to present the user with different options that it then explores, and some more advanced programs may have tens of thousands of different scenarios that can be played out.[1] This paper will focus on nonlinear dialogue involving multiple possibilities for both actions a robot takes and states its environment is in. In particular, it will focus on dialogue management systems: the planning tools used to keep track of and map these complex dialogue sequences. These systems are necessary to maintain a database of possible interactions and expected actions and responses of robots in their communications with humans. There currently exist a few available software applications that can be used to map connections between states robots are in and the actions (spoken dialogue or other) that they take in response to those states. Some systems use actual inputted speech to then model possible responses, which also allows for the testing of the speech recognition tool being used and of the probabilistic model for the robot's responses. However, many of these such systems also offer a view of only one step into the future, which may not be helpful if the user would like to model or plan an interaction well in advance.[2] The goal of this project is to create a system that allows users to have a concrete visualization of possible dialogue sequences that reflects a robot's capabilities.

Having a visualization like this will aid researchers who may not have the time nor access to run repeated tests on a robot to see where a specific dialogue will ‘end up’ or how the robot will react to different environmental changes.

Socially Assistive and Humanoid Robot Interaction with Human Users

The ways in which robots interact with humans can vary greatly in complexity, as stated earlier. Depending on the level of difficulty and the amount of options as well as probabilistic events associated with an interaction, different techniques must be applied for a robot to produce a useful dialogue. A simple interaction may be one in which a user has a finite number of choices that the robot gives it and knows how to handle. In the scope of socially assistive and humanoid robots, this could include helping an elderly person recall information: ie the human asks for a telephone number associated with a name and the robot retrieves and produces it for the user. A much more complex situation would be something like a disaster relief effort, where robots must respond to and maneuver a constantly changing and dangerous environment in real time to, for example, rescue humans trapped in a fallen building. Though these situations are very different, there exist commonalities between the challenges in each. The first hurdle faced in making robots that can partake in effective interactions is speech recognition and, more generally, environment sensing.[3] A socially assistive robot may be needed to interact with children with mental disabilities that do not follow ‘expected’ human patterns of speech and behavior, it may need to work with adults that have physical impairments that do not allow them to perform day to day typical human tasks, among some examples. All of these cases need to be accounted for depending on the goal behavior of the robot. The resulting possible dialogues in these cases will also be affected: for example the robot may need to have an ambiguous or default case for all states, for when speech is intelligible. Another important function a robot like this may have is the ability to adapt to its user’s preferences and needs. The robot may need to increase or decrease its volume, move closer or father from the user, use more gestures rather than spoken dialogue etc… To make an adaptable robot like this is very difficult and has to incorporate many different kinds of probabilistic reasoning (ie is the subject backing up because the robot is too close and it wants space or is the subject backing up in a way that it wants the robot to follow it?).[4] Having adaptable robots like this also increases the possible dialogue sequences and mappings. In fact, the adaptability component of the robot may need to be modeled on its own, as a sort of subroutine of the overarching dialogue taking place. To the resulting dialogue management system, these will all be ‘black box’ things. Different research teams will implement speech recognition and adaptability in their own ways and the tool will simply allow them to model whatever dialogue is meant to be produced by their programs in a general form that can be understood without knowing the details of the platform it is implemented on.

Existing Dialogue Management Systems: Chatmapper and Twine

Before beginning work on the system, two other similar programs with similar functionalities were examined to determine the needs of a system such as this: Twine and Chatmapper. Twine is a very simplistic system that was developed by an individual programmer, Chris Klimas, and available free online for use. It is commonly marketed as a story telling tool with a ‘choose you own adventure’ theme. It allows for basic nonlinear dialogue creation. The interface consists only of squares and lines, squares encompass text including a title and lines are used to form directed connections between squares. Using this simple tool, visual mappings can be created to represent basic interactions between humans and robots. Twine does not have a built in feature to permit a user to easily distinguish between robot actions and human/environment states, which can lead to confusion in trying to use it as a dialogue management tool, especially in cases with a large amount of possible states and actions. Allover, Twine has a very easy to use interface and because it is available online it can be used cross-platform, but it does not have a robust set of features to allow for more detailed dialogue management and planning. This seems to be fine with Twine’s developer, as it is marketed more as a tool for entertainment than for use in applicable research. Chatmapper is another application that is available for download from Urban Brain Studios Inc. It has many options in terms of dialogue mapping. Users can create state or actions blocks, color code them, and customize the text within them. The blocks can also be linked to a specific player (Robot 1, Robot 2, User 1, Child, NAO bot, etc…) and are linked by directed arrows placed by the user. The system essentially has everything needed for basic dialogue management. It does not, however, include a way for dialogue to be imported and automatically modeled. The user must enter every state and action individually using the provided interface. For a user creating their dialogue mapping entirely from scratch, this is not a problem, but for existing dialogues that have been generated other ways, a user may want Chatmapper to simply provide a visualization of the inputted data automatically. Chatmapper also does not include a way to introduce or account for probabilities in robot actions.[5] The goal of this project was to create a new software application that combined the usability of Twine and the features of Chatmapper, along with several other features.

Integrating Partially Observable Markov Decision Processes and Probability

A key component of nonlinear dialogues is a robot's ability to make decisions under uncertainty. With a prompt, a computer must 'decide' which action it should take, ideally choosing the 'best' or most relevant one. Its decision can be broken down into two components: what it believes the user said or did (or what it believes the environment state is, to put it in more general terms) and what action is best to take in that state (best here could be determined based on which action is the most common, which action the user has responded to positively before in a feedback based model, or based on some other decided upon criteria). A partially observable Markov decision process, POMDP, is a framework that allows for modeling these decisions. The POMDP's belief state in this case is the probability distribution over all the possible actions. A more simplistic Markov decision process, MDP, would use a single belief state to represent the entire dialogue history, which is not as useful. A POMDP provides a way to represent the uncertainty encountered in each belief state, ie at each action/state in the system.[6] As well as this, MDPs do not handle speech that is distorted in one way or another well. In a spoken dialogue management system, this is a key feature. In the case of this system, which does not use spoken dialogue as input, it is still important to be able to model these uncertainties for situations in which the robot will have trouble with speech recognition. The POMDP model allows this and gives the computer a quantitative way to choose which action is 'best'. Ideally, each state/action represented in the system will have an associated probability with it, which the user can choose, given the previous state or action taken. The node with the highest probability of being chosen (based on frequency, etc…) can then be kept track of in a meaningful way. The system will then use a POMDP as an underlying model to make decisions and update probabilities when it is in use.

The Resulting Technology

The dialogue management system that was created can be used to model data from a .YAML file or to create a model from scratch. When a file is input, the system will store the information in the underlying data structure, with each robot action and human/environment state stored in its own node. The program will create a graph picture where each of these nodes is labeled with its title and the information pertaining to it: eg. Title: “Action: Greeting” Info: “Robot says Hello/Hi/How are you?”. All nodes are connected to at least one other node of a different type with directional arrows. When a user wants to create their structure from scratch, they can manually add nodes, by dragging and dropping shapes, of different colors and types (action/state) and fill in their data, including title and relevant information. They can also manually make connections between these nodes. All of this information is then stored in the underlying data structure and updated as the user makes changes. Whether the user inputs data by hand or from a file, everything is editable, moveable, and modifiable. The overall user interface is simplistic and easy to use, and the resulting visualization is easily understandable. It allows the user to personalize their data in a way that is meaningful for them, while also keeping a more general set of it in the underlying data structure and then in the .YAML file that is created on save.

Future Research

The system was, unfortunately, not completed to include all possible features. Future tools to be added would be: planning functionality, player functionality, and the factoring in manipulatable probability statistics. The planner tool would allow a user to plan out a path they would like to see the robot take. This would allow the user to create their own dialogue scenarios in an ideal environment (where results are not confused with voice recognition problems, environmental constraints, etc…). The second tool, the player, would be useful to researchers that wanted to map from a specific starting environment state to a specific end state (a goal state). The player would take a start and an end node as input, then highlight all possible paths between them. This would allow researchers to see what actions the robot’s user should take to get the robot to end up doing the desired final action. This would be especially useful in large datasets, where it is not immediately clear from looking at the visualized dialogue map which paths will lead to a specific node. The other useful functionality that will be implemented would be a way to introduce probability not only as an additional feature for each node to have, but also to be used when calculating paths for the player and planner. As robots become more advanced, their actions and responses start having different probabilities, to give them more human-like reactions and create more interesting dialogues. In order for this system to keep up with how probability is being used in robotics, it must incorporate it at some point. Overall, the program is ready for use as a modeling and visualization tool. This was the main goal of the program and of the project in general. It is currently being used by researchers in the lab, but has room to grow even more.

Relevant literature:

1. Takebayashi, Yoichi, Hiroyuki Tsuboi, Yoichi Sadamoto, Yasuki Yamashita, Yoshifumi Nagata, Shigenobu Seto, Hideaki Shinchi, and Hideki Hashimoto. “Speech Dialogue System for Facilitating Improved Human-computer Interaction”. Kabushiki Kaisha Toshiba, Toshiba Software Engineering Corp., assignee. Patent US 5577165 A. 19 Nov. 1996.
2. Rudnicky, Alexander, and Wei Xu. "An agenda-based dialog management architecture for spoken language systems." *IEEE Automatic Speech Recognition and Understanding Workshop*. 1999. http://www.cs.cmu.edu/~xw/asru99-agenda.pdf
3. Allen, James F., Donna K. Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. “Toward Conversational Human-Computer Interaction”. *AI Magazine.* 22.4 (2001). AAAI Press / The MIT Press.
4. Torrey, Cristen, Aaron Powers, Matthew Marge, Susan R. Fussell, and Sara Kiesler. "Effects of Adaptive Robot Dialogue on Information Exchange and Social Relations." *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction* (2006): 126-33.
5. Hennigan, Brian. “Branching Dialogue System Development: Harnessing the Power and Complexity of Verbal Communication in Digital Games and Simulation”. Diss. University of Texas at Dallas, 2012.
6. Henderson, James, and Oliver Lemon. "Mixture model POMDPs for efficient handling of uncertainty in dialogue management." *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008.

7. Roy, Nicholas, Joelle Pineau, and Sebastian Thrun. "Spoken dialogue management using probabilistic reasoning." *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2000.