Situated language understanding in a cognitive robotics platform

Constantine Lignos
SUBTLE MURI Year 5 Review
10/12/2012
Cognitive Robotics (Troy Kelley, ARL)

- **Goal:** Have robots learn, think, react the way a human would
  - **Motivation:**
    - Apply cognitive psychological principles
    - Allow for naturalistic learning by the robot
    - Make human-robot interaction robust & natural

- **Apply lessons learned from cognitive psychology to solve robotics problems**

- **Use existing cognitive architectures**
  - (ACT-R and Soar)
  - Used to simulate human operators for weapons systems
Goal: Natural language to production system

Commander's instructions → Parsed sentences → Semantic representation → Goal vocabulary → Goal execution
The parameter settings for the DFT framework are especially important and one problem has been determining the initial estimates for these preferences. Some parameters are attached to learned knowledge and act as biases on options based on experience. We are hoping to use some kind of relational semantic network (like ConcepNet [12]) to seed DFT with the initial preferences and biases. Other parameters that can be manipulated determine lower-level capabilities. This is analogous to altering the 'personality' of the robot. A robot with a certain high threshold parameter would make slow but accurate decisions. A robot with a fast discounting rate will be focused on the present and would make fast, impulsive decisions.

DFT can naturally incorporate another concept that is observed in human decision makers as well as used in many planning algorithms to limit searches through state space: temporal discounting. The basic idea of temporal discounting is that the magnitude of payoffs is reduced as a function of the distance between the current time and the time of receiving the payoff. People often prefer receiving $100 now rather than a promise of $200 a year from now. Although such discounting is used regularly even in robotic planning algorithms, we encountered an unforeseen problem when testing in an actual dynamic environment. In a situation where the robot has to pick between two options with negative payoffs (the lesser of two evils), it can get stuck in an indecisive loop. Imagine a hallway with a negative choice at each end and the robot has to choose one. As it approaches the first option, the negative effects of the second choice are reduced since it is far off (in spatial and temporal distance) and the robot will switch its course of action. However, as it approaches the second option, its danger seems more eminent while the distant first option looks less intimidating. With a robot that can regularly re-plan its actions, temporal discounting will result in these vacillations. To stop the indecisive behavior, we included a bias for continuing with actions along a previously selected plan. This is supported by sunk cost effects observed in people. A sunk cost effect occurs when someone is willing to pursue a less desirable plan of action because they have already contributed resources toward that goal.

We have experimented with DFT to use it for path planning and navigation where several paths must be chosen. For example, we have used DFT to select whether or not to take a long path without enemies or a shorter path with enemies. We did find the application of DFT useful for the selection of paths which contained different preferences. We are continuing to incorporate DFT into SS-RICS and may use the theory as an attention selection mechanism; however, work in this area is still continuing (this will be reported in a future paper).
Integration

• Integration between MURI work and SS-RICS over three visits to ARL

• Step 1: natural language proof of concept
  • Used existing NL assets to create simple “keyword spotting” system to trigger goals

• Step 2: integration of Penn NLP pipeline and semantics
  • Ported assets to be callable from SS-RICS platform, enabling more sophisticated semantics

• Step 3: robust, easily maintained interface to SLURP
  • Access over standard channels (ROS) allows continuous integration with minimal effort
• NLP Pipeline and Semantics run as independent services
• ROS interface between NL services and SS-RICS process
  • Can be used by any ROS system!
Converting commands to productions

• Goal: Go to X:
  • Antecedent: destination is set ("CurrentDestination Arg *=*;1")
  • Consequents:
    • Execute: Say “Going to the $CurrentDestination”
    • Execute: Move to $CurrentDestination
    • Execute: Say “Commander, I am finished moving to the $CurrentDestination”
    • Unset $CurrentDestination
    • Quit
Converting commands to productions

- Upon getting a go command:
  - Action: go, Location: hallway
  - Set a fact in the mind: $\text{CurrentDestination} = \text{hallway}$
  - Trigger the GotoX production

- Goal is to create a set of standard productions that can enable a core set of scenarios
  - Tell me if you see a...
  - Find a...
  - If you observe X, do Y...

- By building up more complex production chains, we can enable more sophisticated behavior
Demo
Summary

• No fancy behaviors yet, but the fundamental connection between MURI work and ARL is there

• Integration effort has pushed us to focus on clean, modular interfaces

• SS-RICS integration provides example of our ability to support multiple architectures with core SLURP components