Multi-policy dialogue management

Pierre Lison
Logic & natural language group, Department of Informatics
University of Oslo, Norway.

Many dialogue domains are naturally open-ended, and exhibit both partial observability and large state spaces. Most approaches to DM seek to capture the full complexity of the interaction in a single dialogue model and control policy. We present an alternative approach where the dialogue manager operates directly with a collection of small, interconnected policies.

We present an alternative approach where the dialogue manager operates directly with a collection of small, interconnected policies. This allows for easier specification of non-trivial policies that are hard to implement in a single monolithic one. Each small policy is connected to others via a control mechanism, allowing for integration of small policies into a large model. We study how to combine multiple policies into a large model for open-ended dialogue.

Key idea: We define a meta-control (Ⅷ) of the policies. We exploit the meta-control to control the execution of the policies. It becomes possible to integrate both handcrafted and learned policies into the same control.

Extra-linguistic environment

Input speech signal
(output utterance)

Output speech signal

User

Dialogue manager

Production

Recognition hypotheses

Speech understanding

Speech recognition

Scene

Extracted processes

Structured repository of policies

Information exchange

Interpreted utterance

Intended response

Approach

We decompose dialogue policies into 3 distinct functions:

1. Observation update (Update: S \xrightarrow{O} S_i): given the current state s and observation o, update state to s_i.
2. Action selection (\sigma: S \rightarrow A_i): takes the updated state s_i as input, outputs the optimal action a_i to execute (if any).
3. Action update (Act-update: A \xrightarrow{A} S): re-updates the current state s_i given the execution of the action a_i.

We additionally define two new functions for each policy:

1. Likelihood functions (\pi: S \times O \rightarrow [0,1]) to return the likelihood of the observation o if policy i is active and in state s.
2. Activation functions (\phi: S \rightarrow [0,1]) which are implemented via heuristics and can be computed at runtime.

These functions are used to decide which policy is being active. If an abstract action is found, a new process is forked. Algorithm 1 illustrates the execution process: selection of the most active process (1-5), extraction of optimal action (6), and update of activation vector (7-11).

Evaluation

We experiment with a simple, simulated dialogue domain: visual learning task between a human and a robot. The user must ask the robot questions about the objects in a scene (e.g., the robot says “green apple” and the user asks “which green object?”). The human must ask questions about the objects, and the robot answers. The human asks questions about the objects, and the robot answers.

We study how policies can be combined and how they can be learned. The evaluation shows the benefits of combining policies and learning policy parameters on multiple policies. We also present an approach to infer the current state given the history of actions and the history of observations. We introduce an approach to infer the current state given the history of actions and the history of observations.

Conclusion

We introduced a new approach to dialogue management. It allows for easier specification of non-trivial policies that are hard to implement in a single monolithic one. Each small policy is connected to others via a control mechanism, allowing for integration of small policies into a large model.

We introduced a new approach to dialogue management. It allows for easier specification of non-trivial policies that are hard to implement in a single monolithic one. Each small policy is connected to others via a control mechanism, allowing for integration of small policies into a large model. The evaluation shows the benefits of combining policies and learning policy parameters on multiple policies. We also present an approach to infer the current state given the history of actions and the history of observations. We introduce an approach to infer the current state given the history of actions and the history of observations.

Want to know more? Check out my papers at:
http://talk.ifi.uio.no/lsison

Or email me at: lsison@ifi.uio.no