Intelligent Conversational Bot

Policy Optimization (1)

Apr 11th, 2017

YUN-NUNG (VIVIAN) CHEN

WWW.CSIE.NTU.EDU.TW/~YVCHEN/S105-ICB

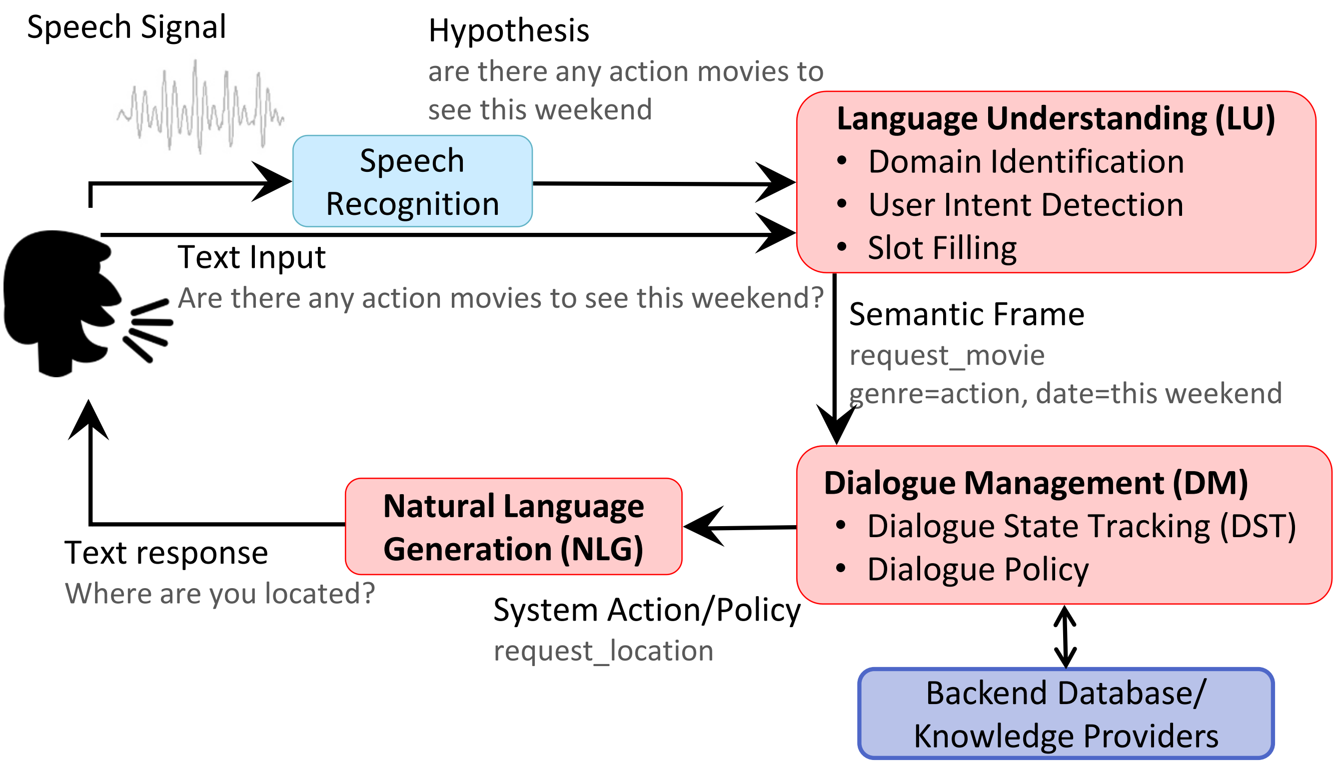
國立臺灣大學
National Taiwan University

Slides credit from Gašić

Review

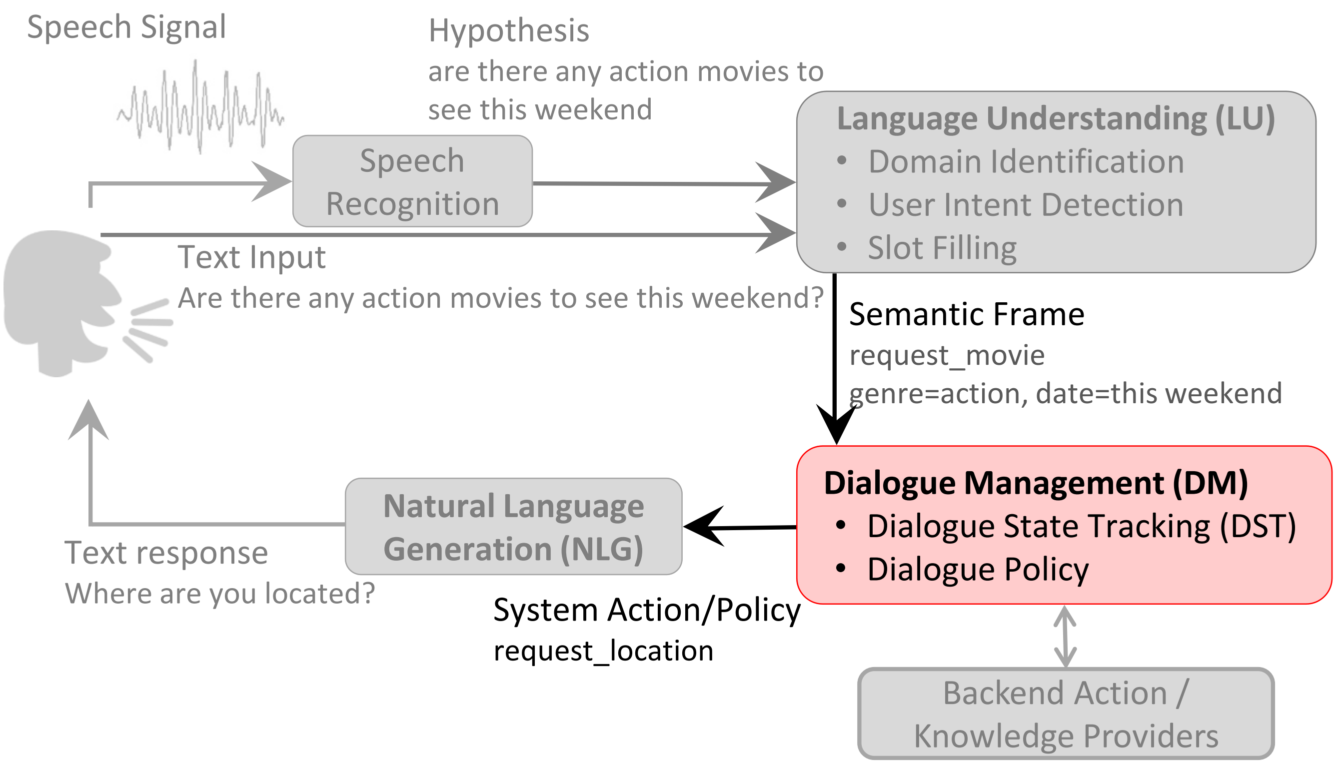
Task-Oriented Dialogue System (Young, 2000)

http://rsta.royalsocietypublishing.org/content/358/1769/1389.short



Task-Oriented Dialogue System (Young, 2000)

http://rsta.royalsocietypublishing.org/content/358/1769/1389.short



5

Dialogue Management

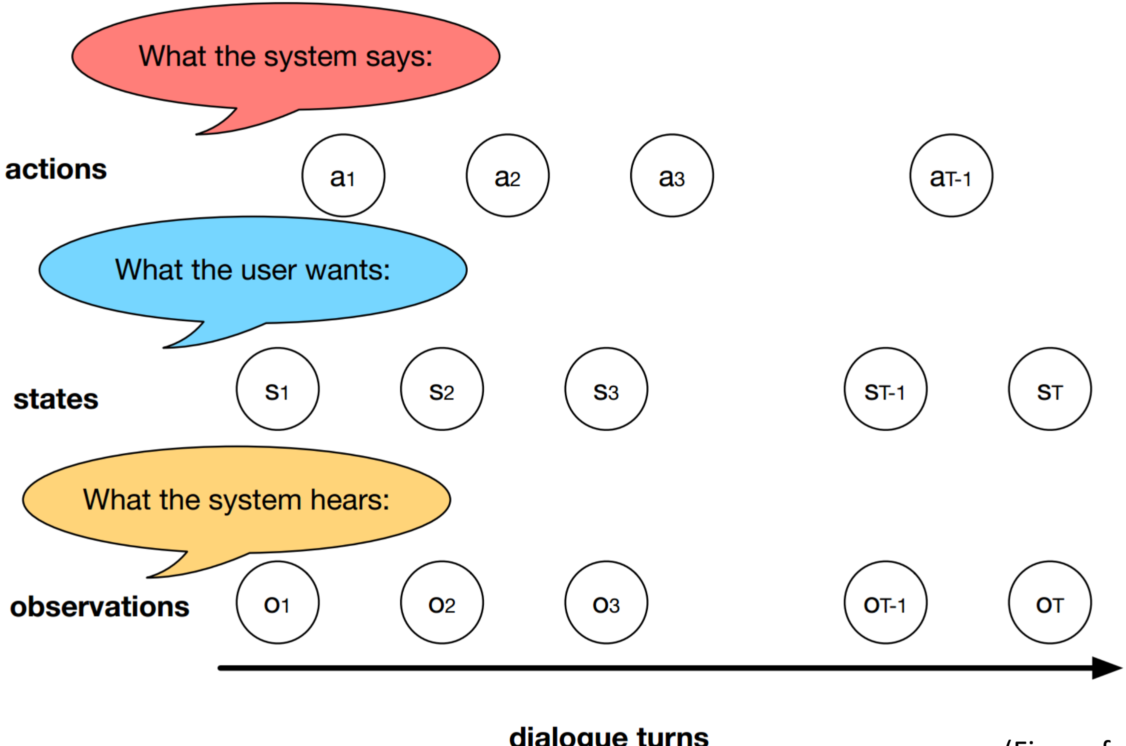
Example Dialogue

Utterance	Dialogue act
Hello, how may I help you?	greeting ()
I’m looking for a Thai restaurant.	request (restaurant; foodtype=Thai)
What part of town do you have in mind?	request (area)
Something in the centre.	inform (area=centre)
Bangkok city is a nice place, it is in the centre of town and it serves Thai food.	inform (restaurant=Bangkok city, area=centre of town, foodtype=Thai)
What’s the address?	request (address)
Bangkok city is a nice place, their address is 24 Green street.	inform (address=24 Green street)
Thank you, bye.	bye ()

Example Dialogue

Utterance	Dialogue act
Hello, how may I help you?	greeting ()
I'm looking for a Thai restaurant.	request (restaurant; foodtype=Thai)
What part of town do you have in mind?	request (area)
Something in the centre.	inform (area=centre)
Bangkok city is a nice place, it is in the centre of town and it serves Thai food.	inform (restaurant=Bangkok city, area=centre of town, foodtype=Thai)
What's the address?	request (address)
Bangkok city is a nice place, their address is 24 Green street.	inform (address=24 Green street)
Thank you, bye.	bye ()

Elements of Dialogue Management



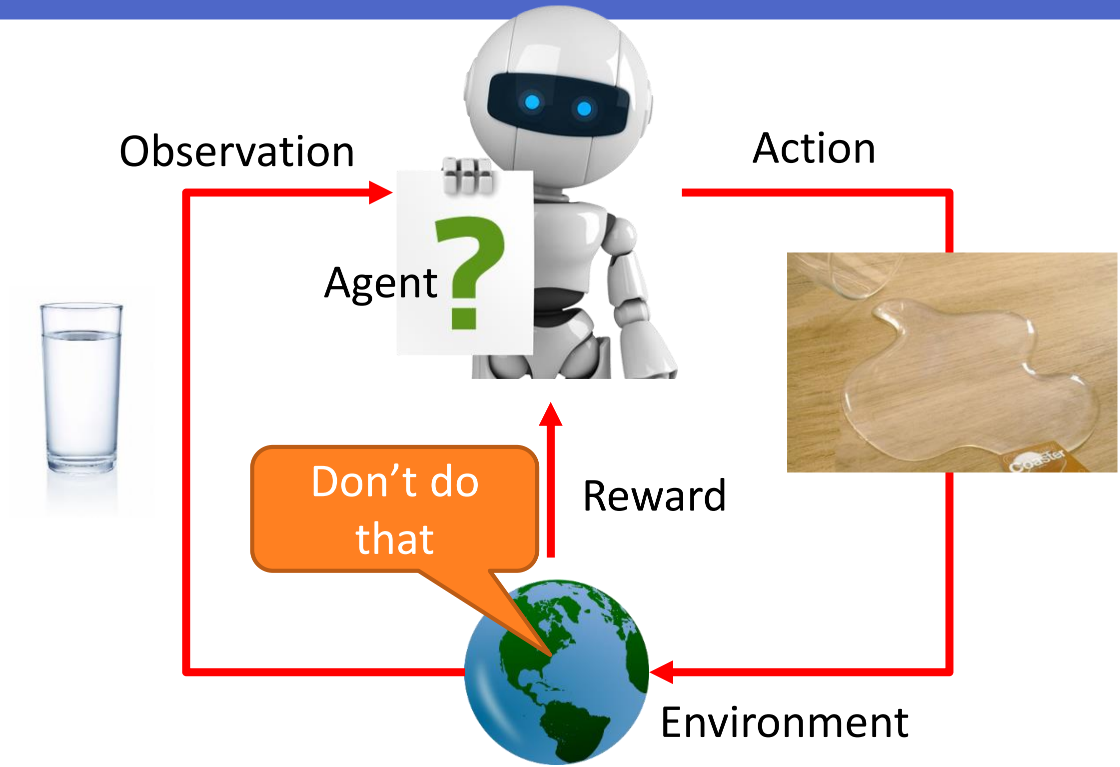
(Figure from Gašić)

Reinforcement Learning

- RL is a general purpose framework for **decision making**
  - RL is for an *agent* with the capacity to *act*
  - Each *action* influences the agent's future *state*
  - Success is measured by a scalar *reward* signal
  - Goal: *select actions to maximize future reward*

Big three: action, state, reward

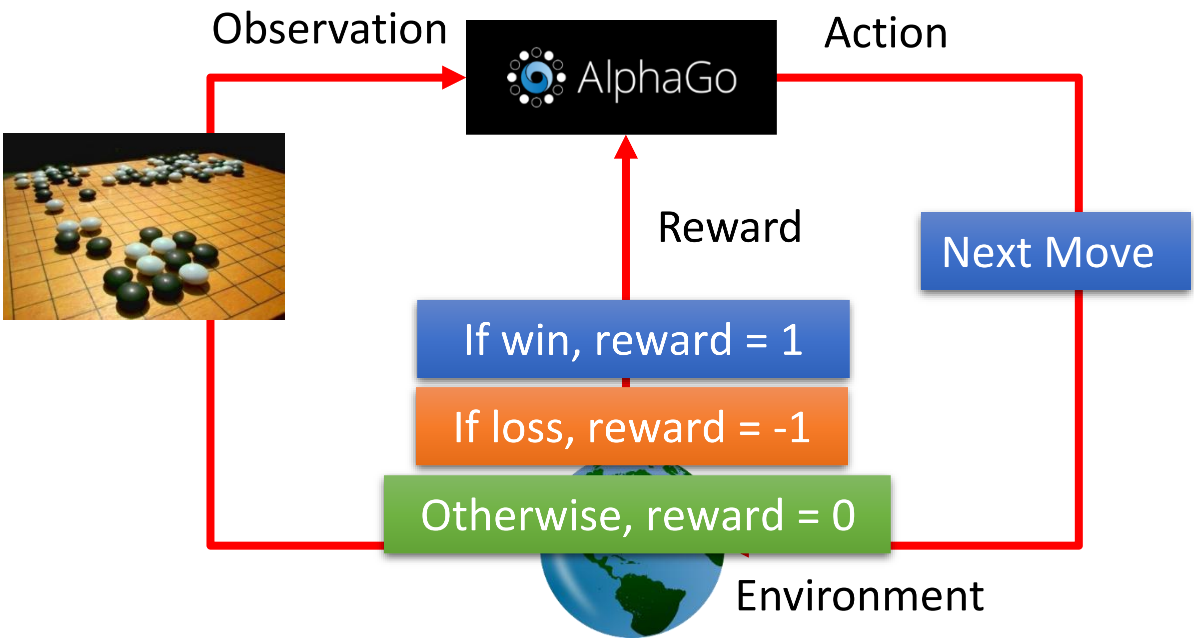
Reinforcement Learning



Reinforcement Learning

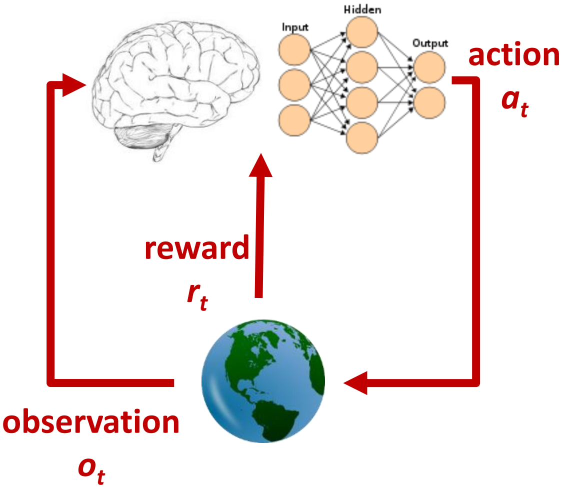
Agent learns to take actions to maximize expected reward.

Scenario of Reinforcement Learning



Agent learns to take actions to maximize expected reward.

Agent and Environment



- At time step t
  - The agent
    - Executes action a_t
    - Receives observation o_t
    - Receives scalar reward r_t
  - The environment
    - Receives action a_t
    - Emits observation o_{t+1}
    - Emits scalar reward r_{t+1}
  - t increments at env. step

State

- Experience is the sequence of observations, actions, rewards

$$o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t$$

- State is the information used to determine what happens next
  - what happens depends on the history experience
    - The agent selects actions
    - The environment selects observations/rewards
- The state is the function of the history experience

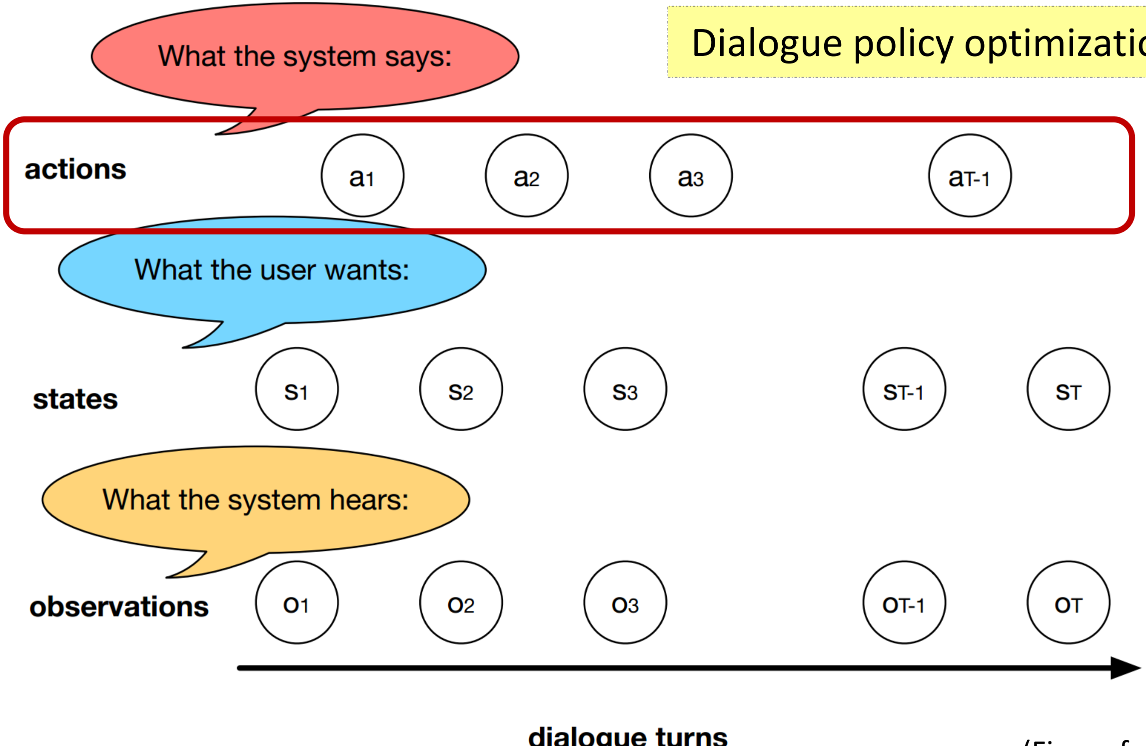
$$s_t = f(o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t)$$

Dialogue Policy Optimization

Decision Making

Elements of Dialogue Management

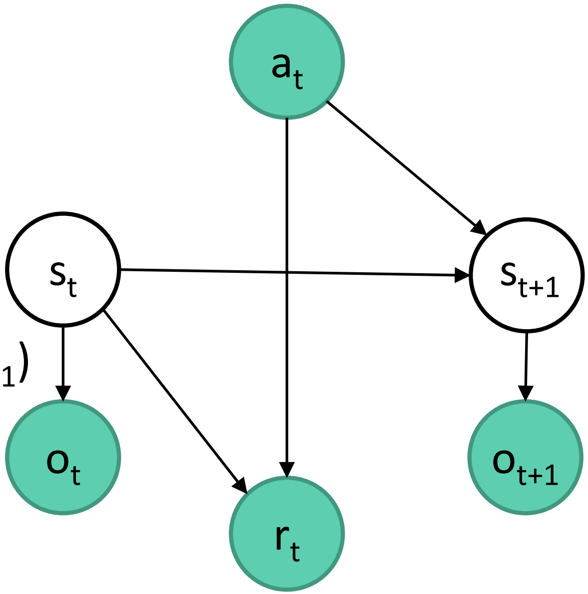
Dialogue policy optimization



(Figure from Gašić)

Partially Observable Markov Decision Process (POMDP)

- Dialogue states: s_t
- Noisy observation: o_t
- System actions: a_t
- Rewards: r_t
- Transition probability: $p(s_{t+1}|s_t, a_t)$
- Observation probability: $p(o_{t+1}|s_{t+1})$
- Distribution over states: $b(s_t)$



DM as Partially Observable Markov Decision Process (POMDP)

Data

- Noisy observation of dialogue states
- Reward – a measure of dialogue quality

Model

- Partially observable Markov decision process (POMDP)

Prediction

- Distribution over dialogue states
- Optimal system actions – Dialogue Policy Optimization

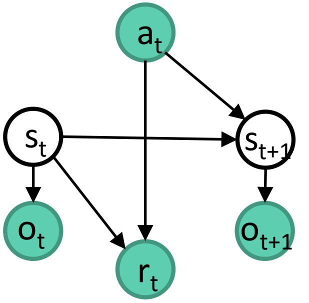
Decision Making in POMDP

- Policy: $\pi : B \rightarrow A$ belief estimation mapping
- Return: $R_t = \sum_{k=0}^{T-1} \gamma^k \cdot r_{t+k}$ accumulated reward
- Value function
  - How good the system is in a particular belief state

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{T-1} \gamma^k \cdot r_{t+k} \mid s_t = s \right\}$$

$$= r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \sum_{o'} p(o' \mid s') V^\pi(s')$$

$$V^\pi(b) = \sum_s V^\pi(s) b(s)$$

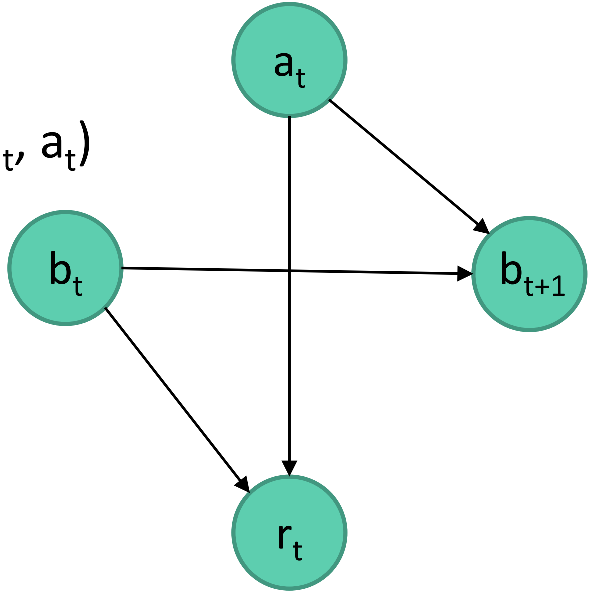


POMDP Policy Optimization

- Finding value function associated with optimal policy, i.e. the one that generates maximal return
  - Problem: tractable only for very simple cases (Kaelbling et al., 1998)
  - Alternative solution: discrete space POMDPs can be viewed as a continuous space MDP with states as belief states $b_t = b(s_t)$

Markov Decision Process (MDP)

- Belief state from tracking: $b_t = s_t$
- System actions: a_t
- Rewards: r_t
- Transition probability: $p(b_{t+1} | b_t, a_t)$



DM as Markov Decision Process (MDP)

Data

- Belief dialogue states (continuous)
- Reward – a measure of dialogue quality

Model

- Markov decision process (MDP) & reinforcement learning

Prediction

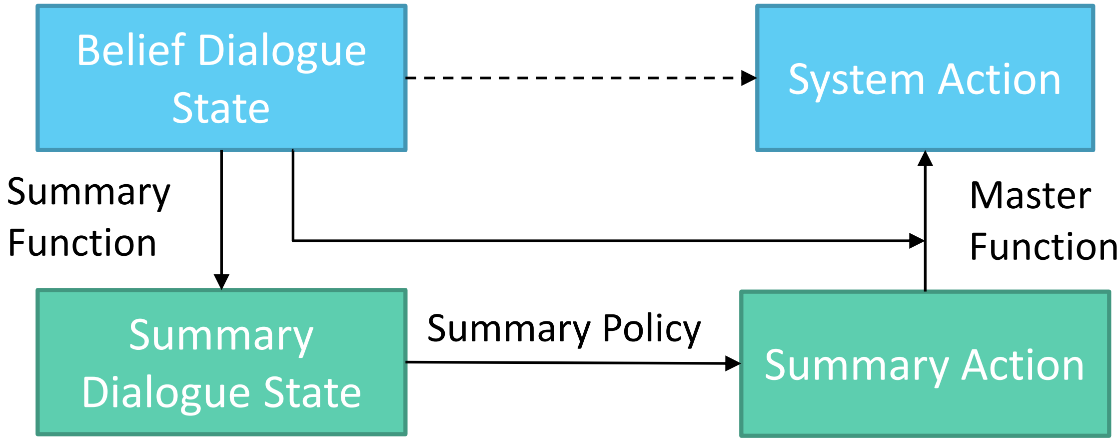
- System actions – Dialogue Policy Optimization

Policy Optimization Issue

- Optimization problem size
  - Belief dialogue state space is large and continuous
  - System action space is large
- Knowledge environment (user)
  - Transition probability is unknown (user status)
  - How to get rewards

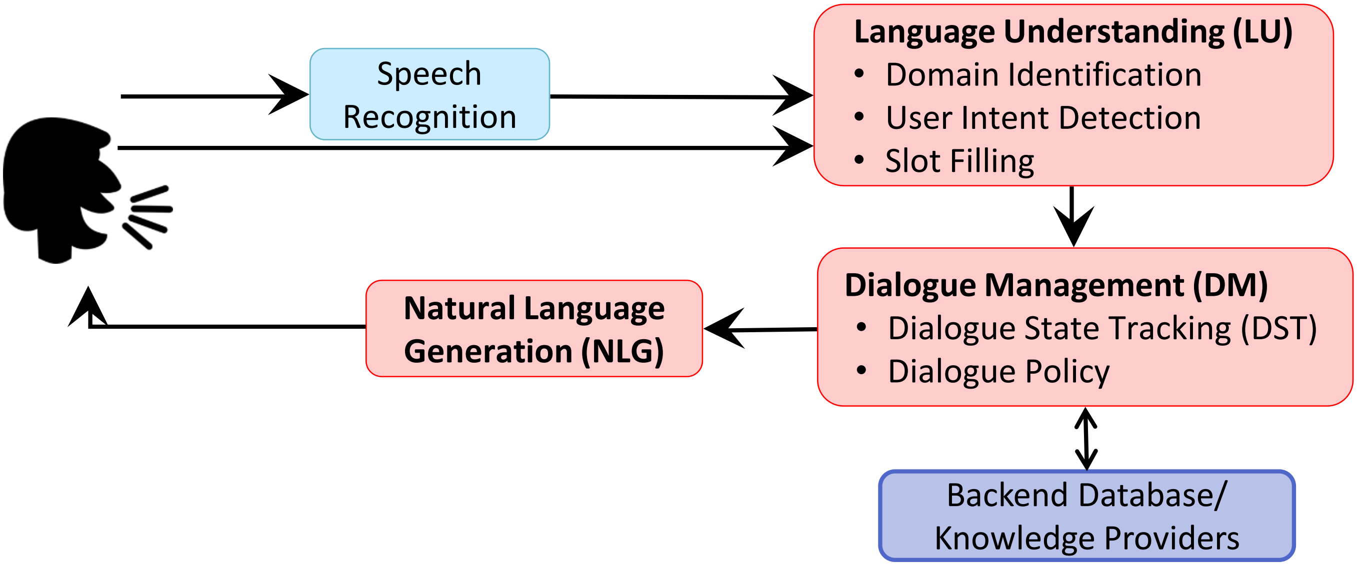
Large Belief Space and Action Space

- Solution: perform optimization in a <u>reduced summary space</u> built according to the heuristics



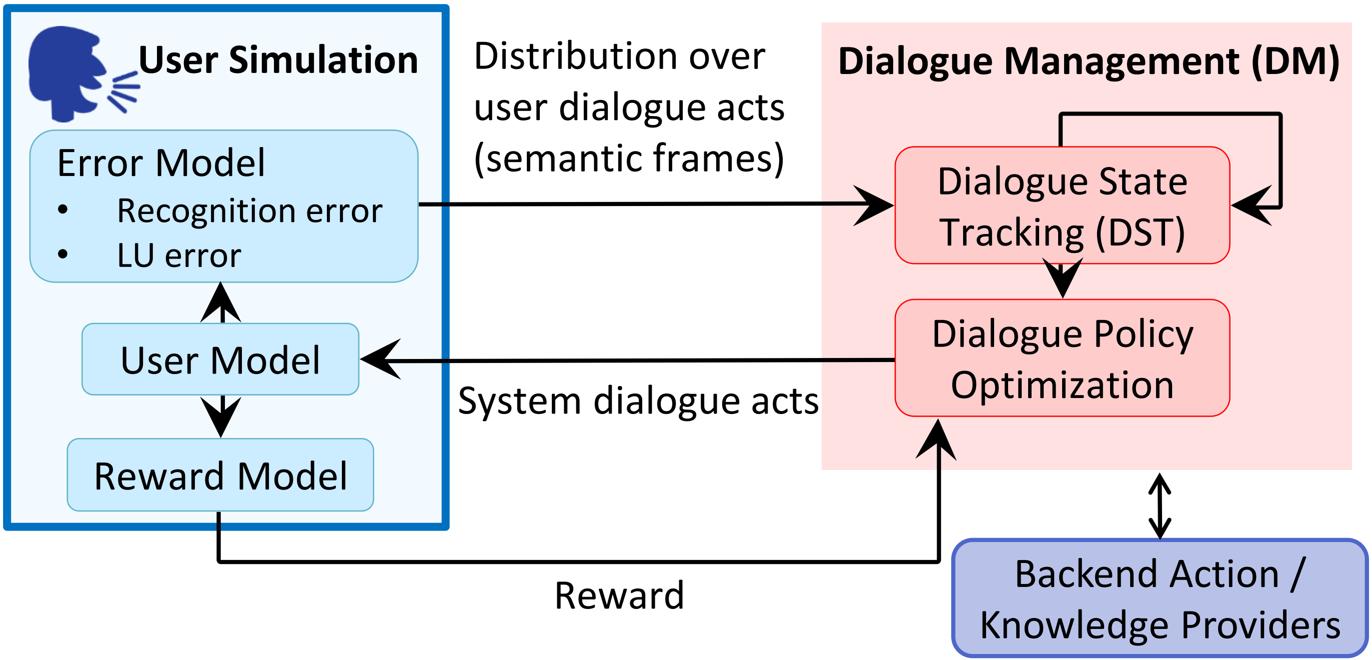
Transition Probability and Rewards

- Solution: learn from real users



Transition Probability and Rewards

- Solution: learn from a simulated user



Concluding Remarks

- **Dialogue policy optimization** can be viewed as an RL task
- POMDP can be viewed as a continuous space MDP
- Belief dialogue state space can be summarized to reduce computational complexity
- Transition probability and reward come from
  - Real user
  - Simulated user