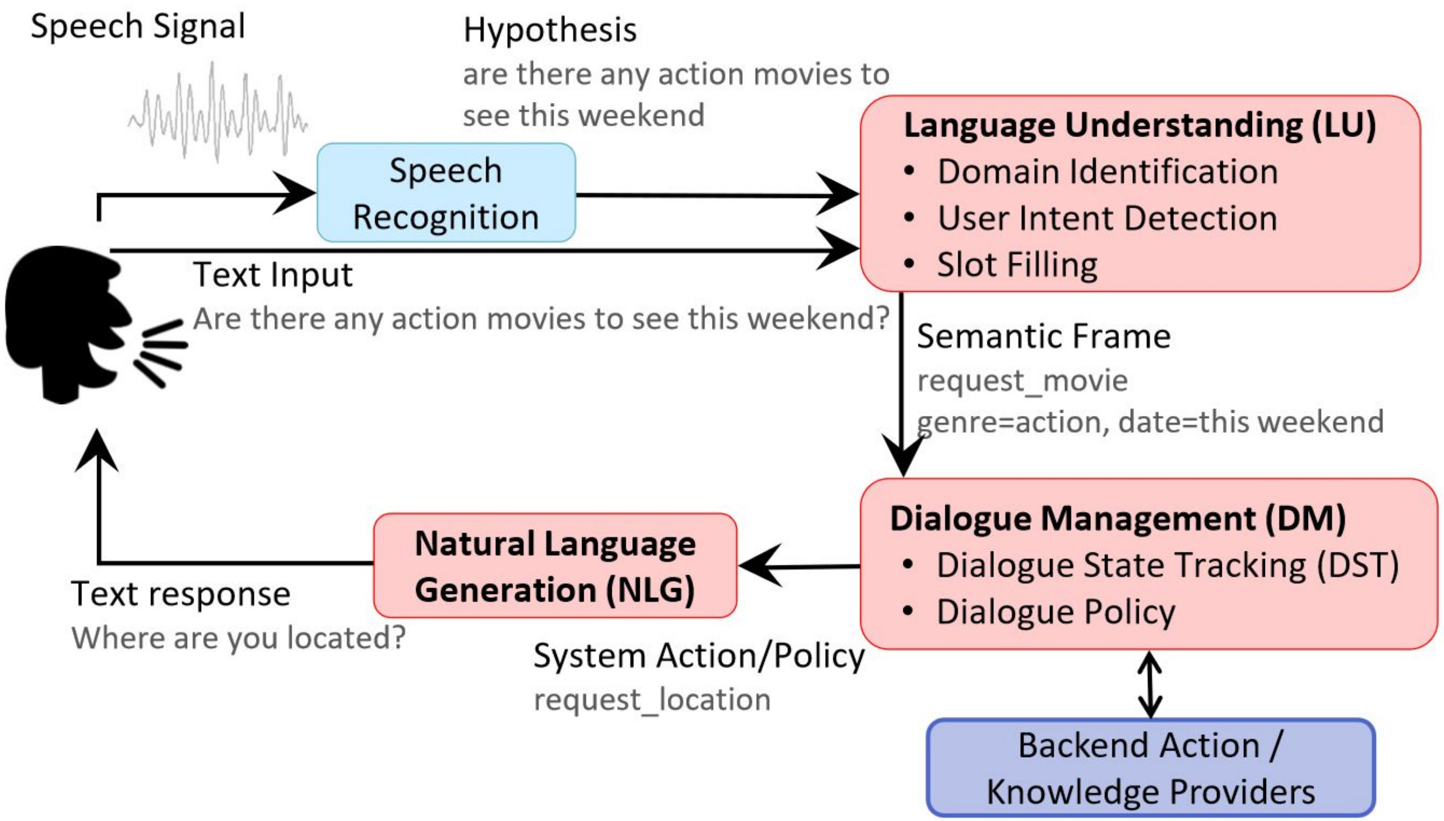
End to End Task Completion Neural Dialogue Systems

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao and Asli Celikyilmax

Microsoft Research

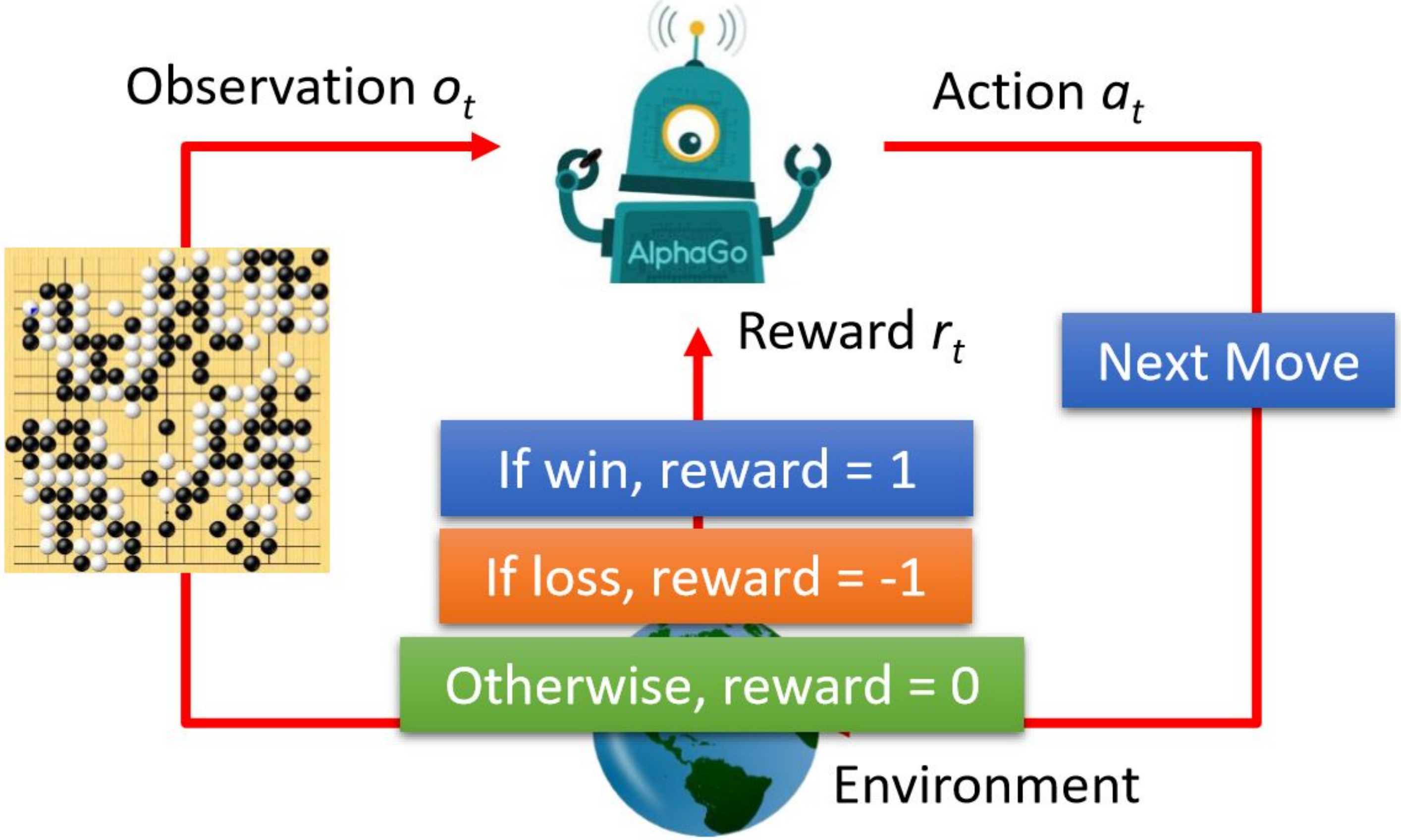
Aiyanyo Imatitikua D.



Task-Oriented Dialogue System

Reinforcement Learning….

- Reinforcement Learning(RL) is learning what to do so as to maximize a numerical reward signal.
- It 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



Observation o_t
AlphaGo
Action a_t
Reward r_t
Next Move
If win, reward = 1
If loss, reward = -1
Otherwise, reward = 0
Environment

Cons to RL approach

1

Inflexible question types

- *Agent: Would you like to watch in Seattle?*

2

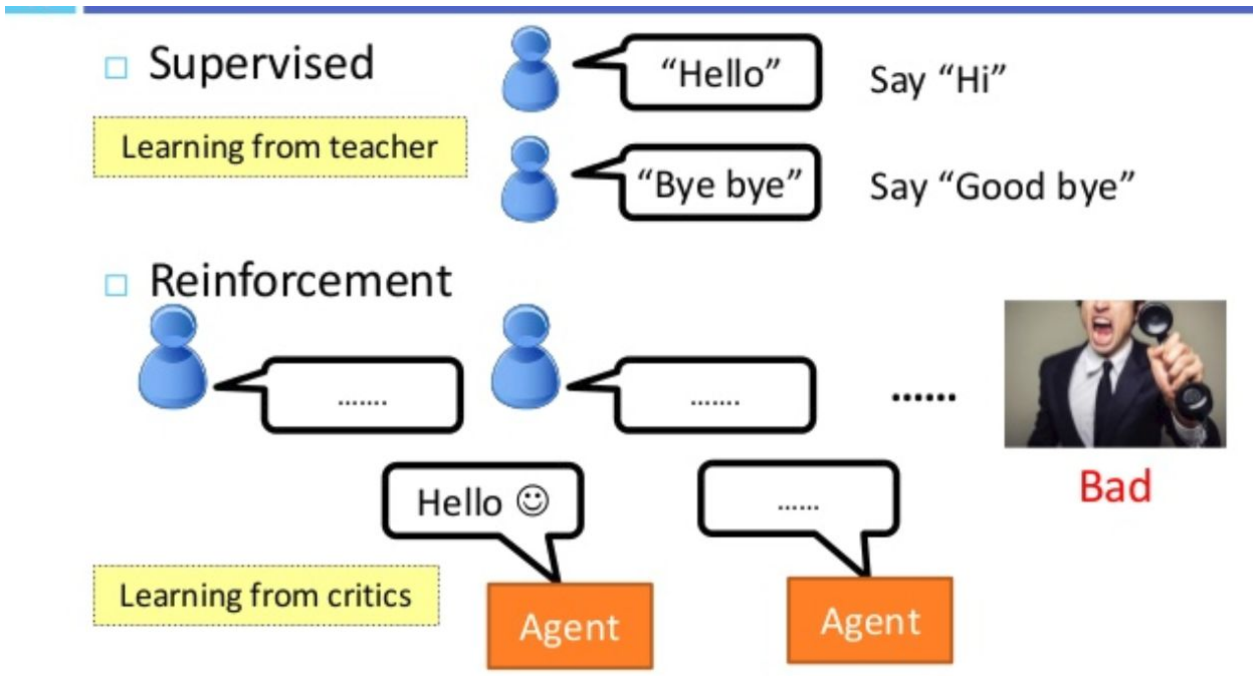
Poor robustness

- The user answers are too simple to be misunderstood so cant deal with noise in real user utterances.

3

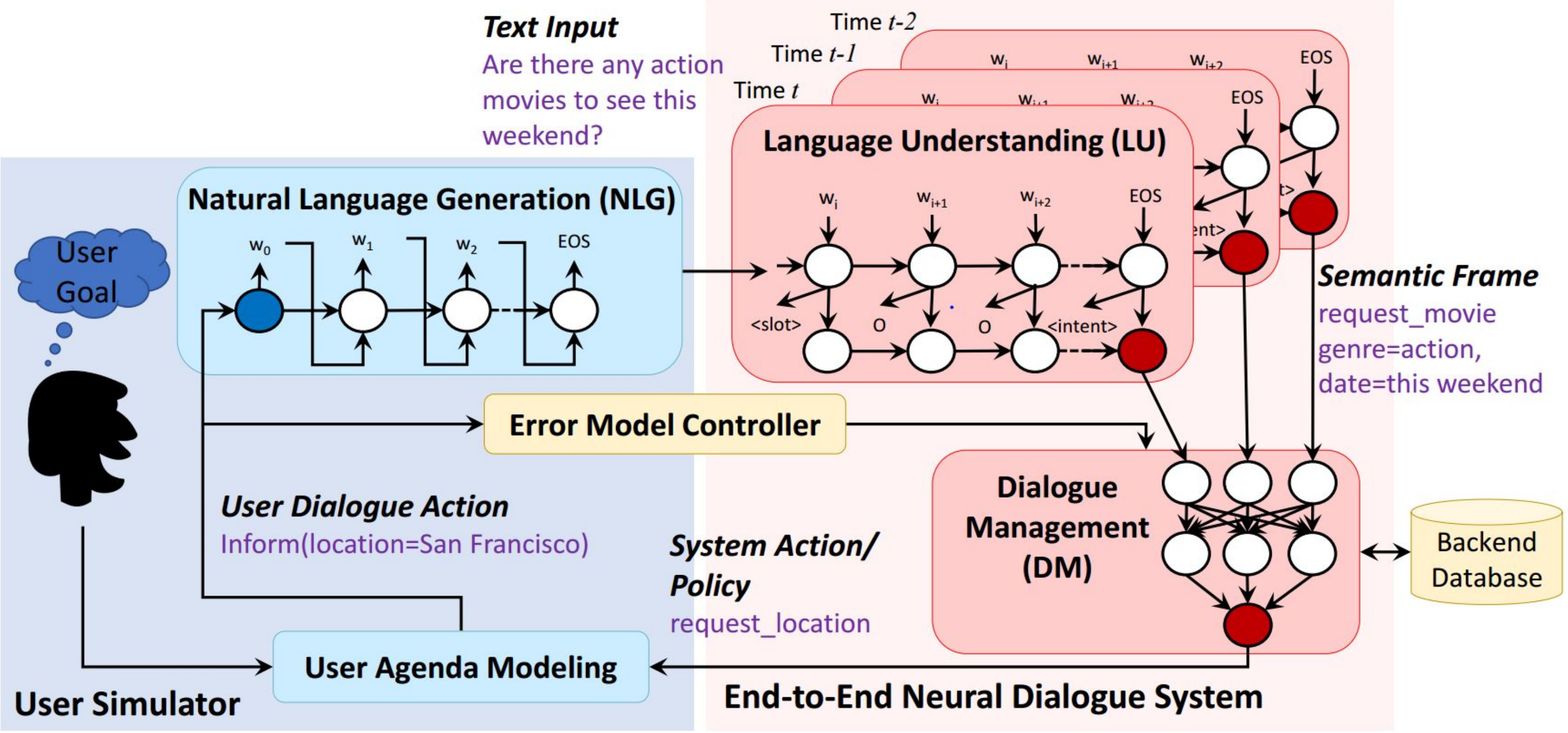
User requests during dialogues

- *User:Which theater can I book 3 tickets for 10 cloverfield lane?*



- Supervised learning for each component and reinforcement learning for end-to-end training the neural dialogue system
- The system can learn how to efficiently interact with users for task completion

Main Idea



E2E Task-Completion Bot (TC-Bot)

E2E Task Completion Neural Dialogue System Contributions:

Robustness

- Propose a neural dialogue system with greater robustness by automatically selecting actions based on uncertainty and confusion by reinforcement learning.
- Provide the first systematic analysis to investigate the impact of different types of natural language understanding errors

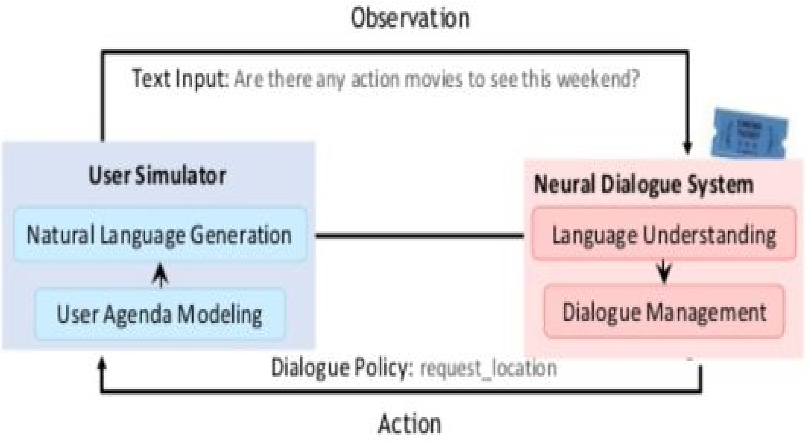
Flexibility

- This is the first neural dialogue system that allows user-initiated behaviors during conversations

Reproducibility

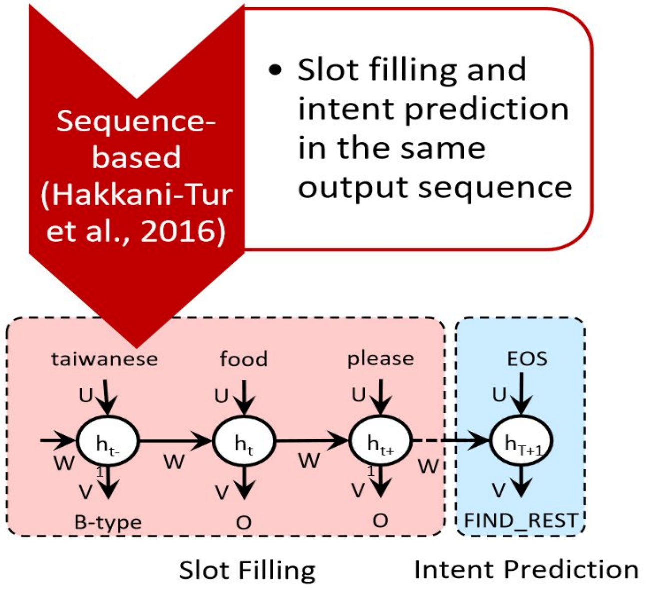
- Demonstrate how to evaluate RL dialogue agents using crowd sourced task-specific datasets and simulated users in an end-to-end fashion.

Proposed Framework



- User simulator
  - User agenda modeling component based at the dialogue act level is applied to control the conversation exchange conditioned on the generated user goal.
  - An NLG module is used to generate natural language texts corresponding to the user dialogue actions
- Neural Dialogue
  - The utterance passes through the LU and becomes a corresponding semantic frame.
  - The DM is to accumulate the semantics from each utterance, robustly track the dialogue states during the conversation and generate the next system action.

Neural Dialog System

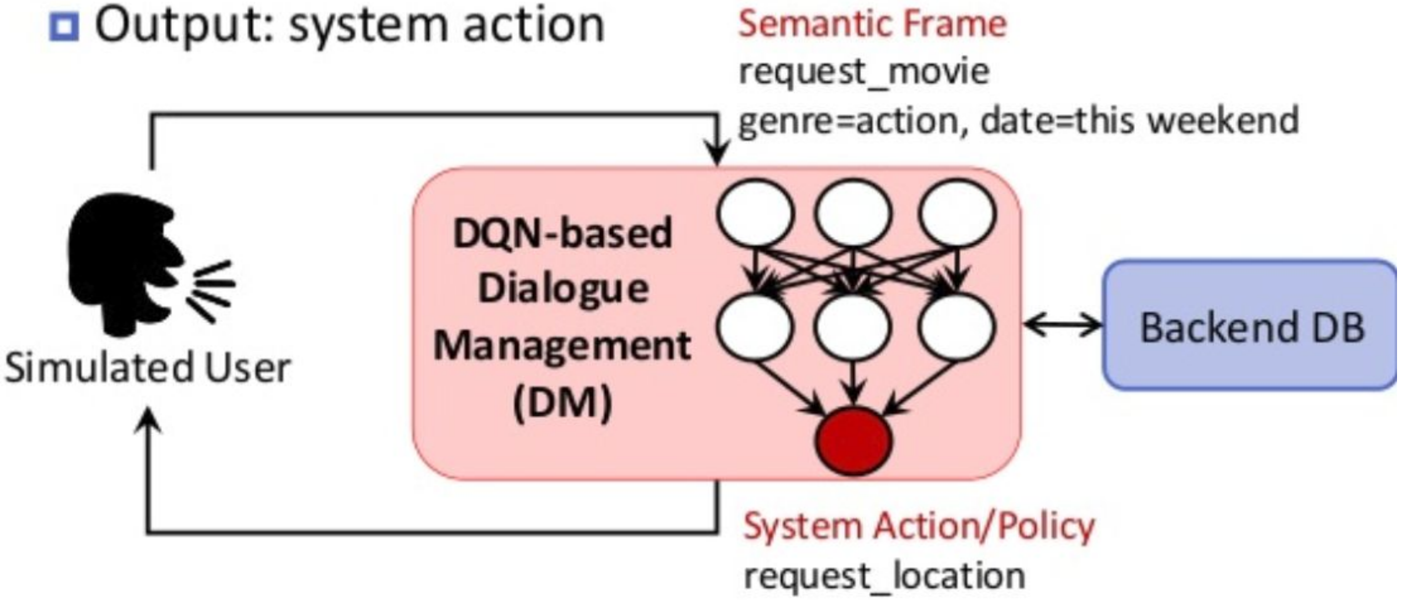


- Language Understanding
  - This is mainly viewed as an utterance classification task.
  - The LU component is implemented with a single LSTM, which performs intent prediction and slot filling simultaneously
  - The weights of the LSTM model are trained using backpropagation to maximize the conditional likelihood of the training set labels
  - The predicted tag set is a concentrated set of IOB-format slot tags and intent tags
    - thus this model can be trained using all available dialogue actions and utterance pairs in our labeled dataset in a supervised manner

Neural Dialog System

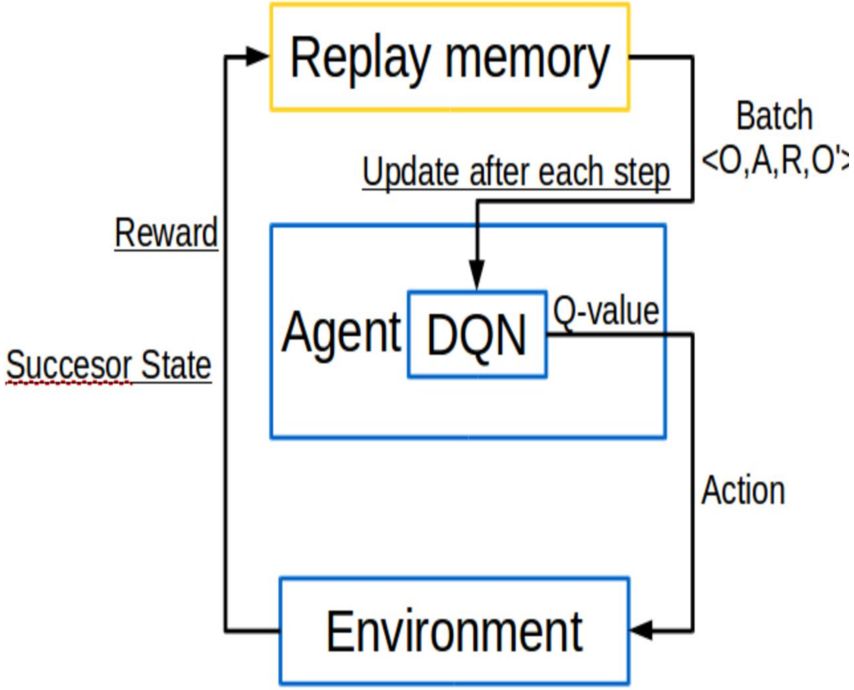
- Dialog Management
  - Dialog state tracking
    - A symbolic query is formed
    - The state tracker will be updated based on the available results
    - The state tracker will prepare the state representation for policy learning
  - Policy learning
    - Conditioned on the state representation, the policy is to generate the next available system action

- Deep RL for training DM
  - Input: current semantic frame observation, database returned results
  - Output: system action



Deep Q-Networks for Policy

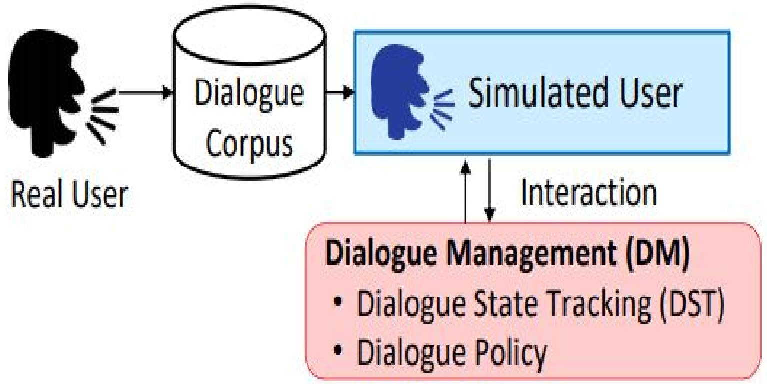
E2E Reinforcement Learning



- Two important DQN tricks
  - Target network usage
  - Experience replay strategy
- Buffer update strategy
  - Accumulate all experience tuples from the simulation and flush the pool till the current RL agent reaches a success rate threshold
    - A threshold which is equal to the performance of a rule-based agent
  - Use the experience tuples from the current RL agent to refill the buffer
- If the current DQN agent is better than the target network, the experience replay buffer will be flushed.

User Simulation

- A user simulator is required to automatically and naturally interact with the dialogue system.
- It first generates a user goal
  - Inform_slots for slot-value pairs that serve as constraints from the user
  - Request_slots for slots whose value the user has no information about but wants to get values from the agent during the conversation.

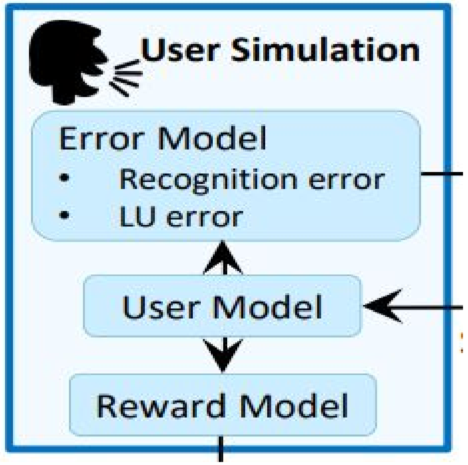


User Agenda Modeling

- The user simulator maintains a compact, stack-like representation called user agenda
  - Where the user state s_u is factored into an agenda A and a goal G.
  - The goal consists of constraints C and request R.
  - At each time-step t, the user simulator generates the next user action $a_{u,t}$ based on the current state $s_{u,t}$ and the last agent action $a_{m,t-1}$ and then updates the current status $s'_{u,t}$.

Natural Language Generation (NLG)

- The NLG module generates natural language texts
- To control the quality of user simulation given limited labeled data, a hybrid model is employed
  - Template based NLG
  - Model based NLG
    - Trained on the labeled dataset with a sequence-to-sequence model.
    - It takes dialogue acts as input, and generates sentence sketch with slot placeholders via an LSTM decoder. Then a post-processing scan is performed to replace the slot placeholders with their actual values



Intent-Level Error

- Group types
  - Group 1: greetings, thanks, closing etc
  - Group 2: inform(moviename = ‘Titantic’, starttime=‘7pm’)
  - Group 3: request(starttime; moviename = ‘Titanic’)
- Error types
  - L0: random noisy intent from *within group error* or *between group error*
  - L1: *within group error*; real intent is **request_theater** but predicted intent is **request_moviename**
  - L2: *between group error*; real intent is **request_moviename** but predicted intent is **inform_moviename**

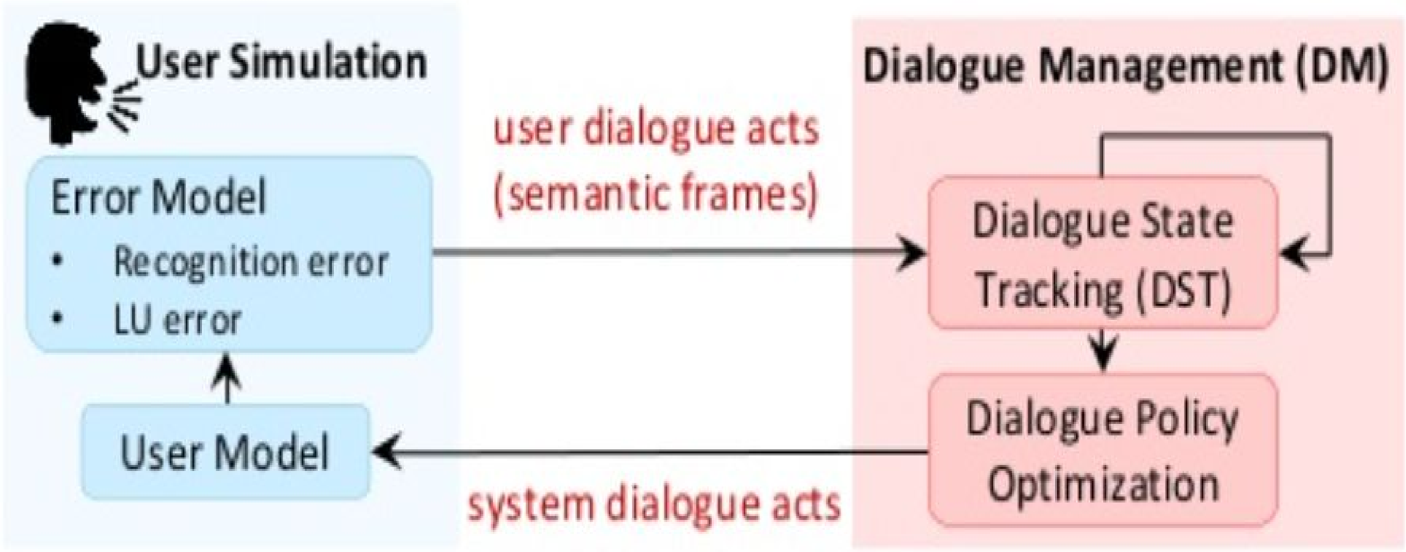
Slot-Level Error

- S0: Randomly set to the 3 types
- S1: Slot deletion; a scenario where the slot is not recognized by LU
- S2: Incorrect slot value; a scenario where the slot name is correctly recognized but the slot value is wrong
- S3: Incorrect slot; a scenario where both the slot and its value are incorrectly recognized

Experiments

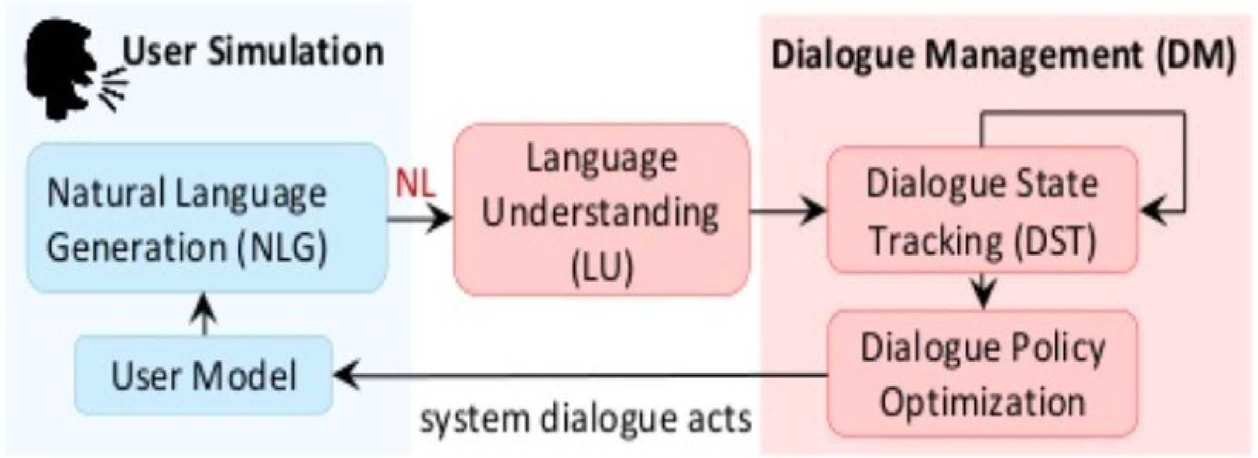
- Goal: Booking movie tickets
- Criteria:
  - whether a movie is booked
  - Whether the movie satisfies the user constraints
- Dataset: From Amazon Mechanical Turk
  - 280 labeled dialogues
  - 11 dialogue acts and 29 slots

Annotations	
Intent	request, inform, deny, confirm_question, confirm_answer, greeting, closing, not_sure, multiple_choice, thanks, welcome
Slot	actor, actress, city, closing, critic_rating, date, description, distanceconstraints, greeting, implicit_value, movie_series, moviename, mpaa_rating, numberofpeople, numberofkids, taskcomplete, other, price, seating, starttime, state, theater, theater_chain, video_format, zip, result, ticket, mc_list



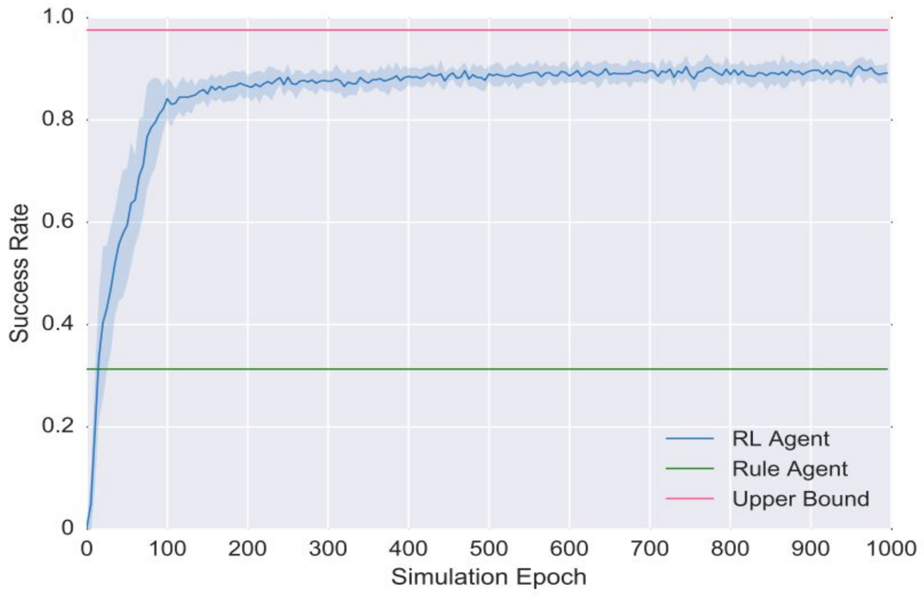
- Dialog Manager receives frame level information
  - No error model: perfect recognizer and language understanding
  - Error model: simulate the possible errors

Frame-Level Interaction

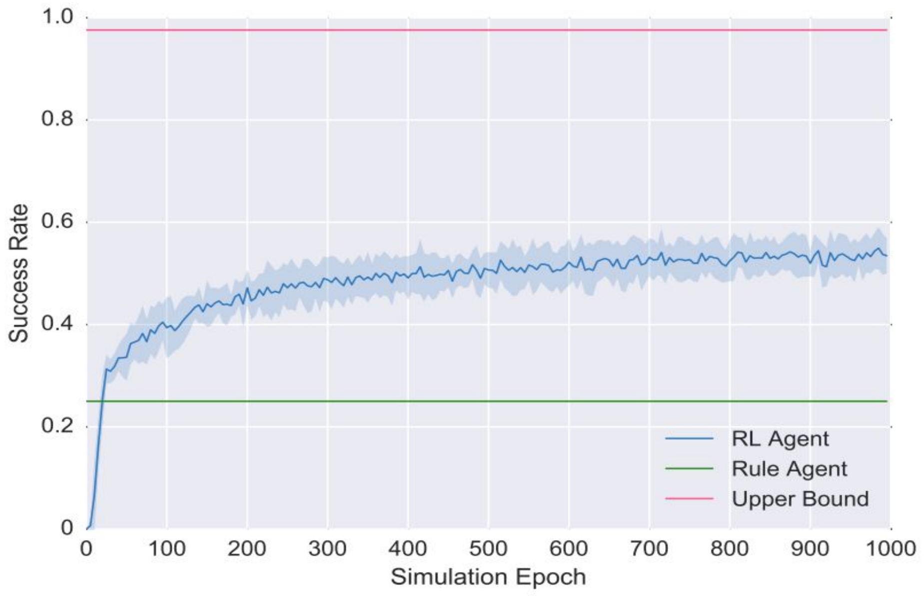


- User simulator sends natural language
  - No recognition error
  - Errors from Natural Language Generator or Language Understanding module

Natural Language Level Interaction



(a) Frame-level semantics for training



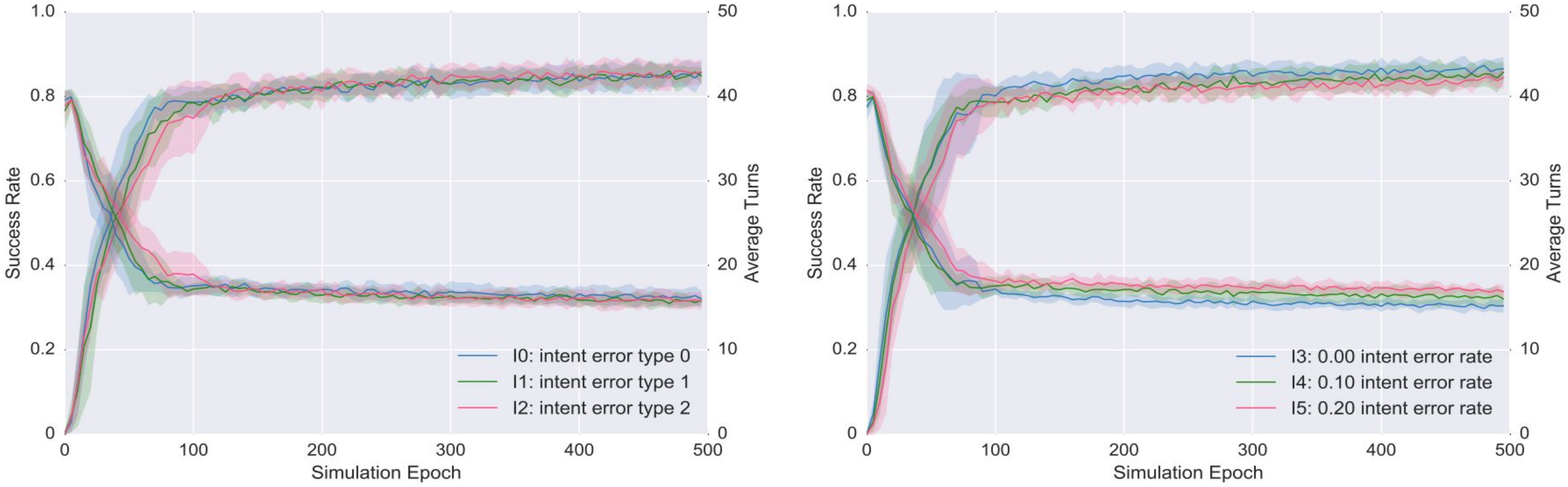
(b) Natural language for end-to-end training

- The RL Agents performs significantly better than Rule based systems
- However, adapting to noises from LU and NLG takes longer when training natural language
- Frame level semantics show greater robustness in real-world scenarios

Simulated User Evaluation

Intent Error

- Error Type
  - L0: random noisy intent
  - L1: within group error
  - L2: between group error
- Error Rate
  - L3: 0.00
  - L4: 0.10
  - L5: 0.20



(a) Intent Error Type Analysis

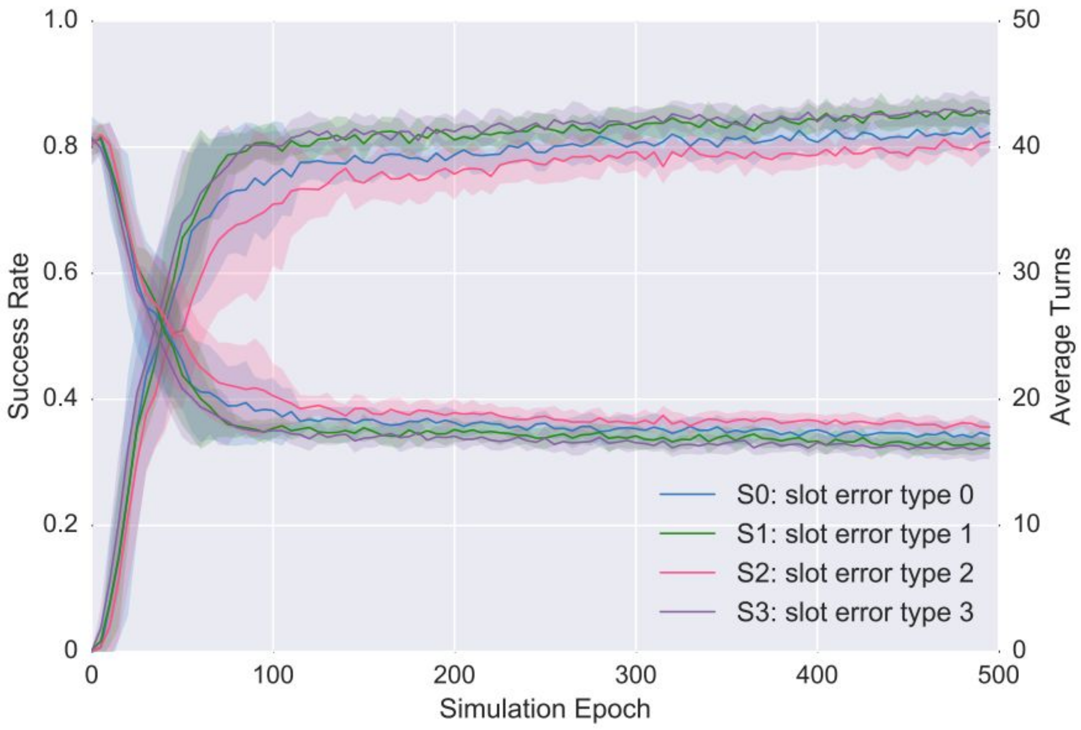
(b) Intent Error Rate Analysis

- Incorrect intents have similar impact no matter what categories they belong to
- When the intent error rate increases, the dialogue agent performs slightly worse, but the difference is subtle
- All RL agents can converge to a similar success rate in both intent error analysis

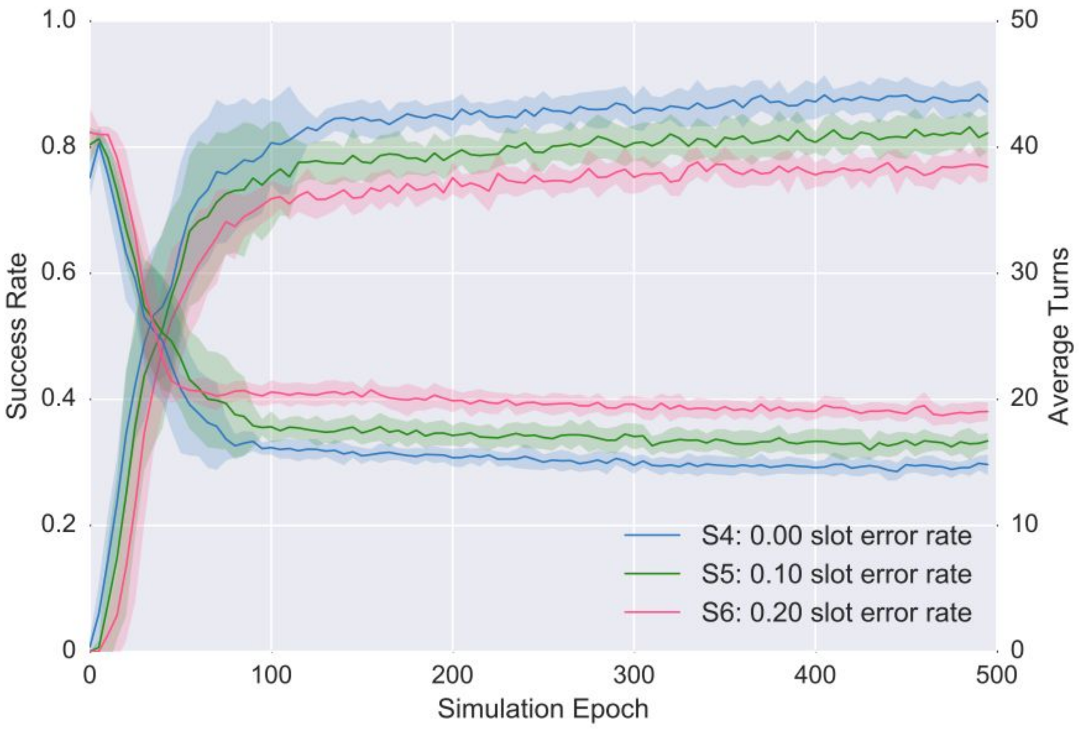
Intent Error Analysis

Slot Error

- Error Type
  - S0: Random
  - S1: Slot deletion
  - S2: Incorrect slot value
  - S3: Incorrect slot
- Error Rate
  - S3: 0.00
  - S4: 0.10
  - S5: 0.20



(c) Slot Error Type Analysis



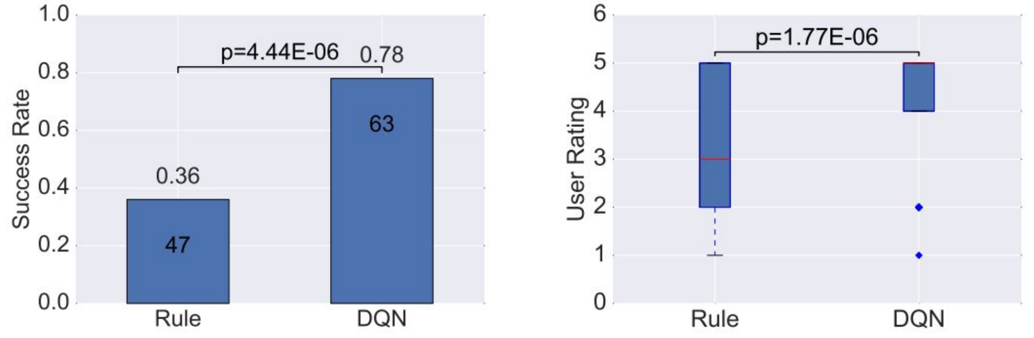
(d) Slot Error Rate Analysis

- Incorrect slot value(s2) performs worst.
- The dialogue agent performs worse as the slot error increases

Slot Error Analysis

Human Evaluation

- 110 dialogue sessions from 8 human users
- Gave a rating on a scale from 1(worst) to 5(best) based on both naturalness and coherence of the dialogue



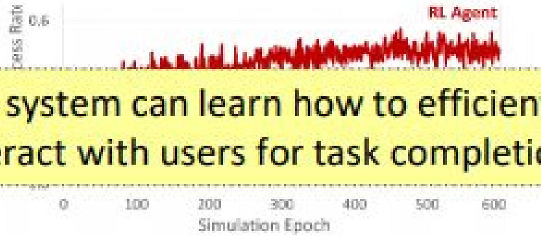
(a) Success Rate (b) User Rating Distribution

Conclusion

User goal: ***Two tickets for "the witch" tomorrow 9:30 PM at regal meridian 16, Seattle.***

RULE BASED SYSTEM
User: Can I get 2 tickets for the witch?
Agent: Which theater do you want?
User: Regal meridian 16
Agent: What time would you like to see it?
User: 9:30 pm
Agent: What date would you like to see it?
User: Tomorrow!
Agent: How many people are you?
User: Just two adults.
Agent: What is your location?
User: Seattle!
Agent: What kind of video format do you like?
User: Oh, any kind is okay.
Agent: Do you have any price constraint?
User: No, I don't care.
Agent: Great - I was able to purchase 2 tickets for you to see the witch tomorrow at regal meridian 16 theater in Seattle at 9:30 pm.
User: Thank you.
Agent: Thank you!

REINFORCEMENT LEARNING SYSTEM
User: Can I get 2 tickets for the witch?
Agent: What date would you like to see it?
User: Tomorrow
Agent: Which theater do you prefer?
User: Regal meridian 16
Agent: What is your location?
User: Seattle
Agent: What time would you like to see it?
User: 9:30 pm.
Agent: Great - I was able to purchase 2 tickets for you to see the witch tomorrow at regal meridian 16 theater in Seattle at 9:30 pm.
User: Thanks.
Agent: Thanks!



The system can learn how to efficiently interact with users for task completion