Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning

Baolin Peng[#], Xiujun Li[*], Jianfeng Gao[*], Jingjing Liu[*], Kam-Fai Wong[#], Shang-Yu Su[$]

[*]Microsoft Research
[#]The Chinese University of Hong Kong
[$]National Taiwan University

ACL 2018

Outline

- Task-completion dialogue as optimal decision making
- Reinforcement learning using real or simulated experience
- Deep Dyna-Q
- Evaluation methodology
- Simulated user evaluation
- Human-in-the-loop evaluation
- Conclusion

An Example Dialogue with Movie-Bot

Turn 0 usr: can i get 2 tickets for race
Turn 1 sys: What date would you like to watch it?
Turn 2 us
Turn 3 sy
Turn 4 us
Turn 5 sy
Turn 6 us
Turn 7 sy
Turn 8 us
Turn 9 sy you theater in seattle at 10:00 pm.
Turn 10 usr: thanks

Actual dialogues can be more complex:

- Speech/Natural language understanding errors
  - Input may be spoken language form
  - Need to reason under uncertainty
- Constraint violation
  - Revise information collected earlier
- …

Source code available at https://github/com/MiuLab/TC-Bot

Task-oriented, slot-filling, Dialogues

- **Domain**: movie, restaurant, flight, …
- **Slot**: information to be filled in before completing a task
  - For Movie-Bot: movie-name, theater, number-of-tickets, price, …
- **Intent** (dialogue act):
  - Inspired by speech act theory (communication as action)

    request, confirm, inform, thank-you, …
  - Some may take parameters:

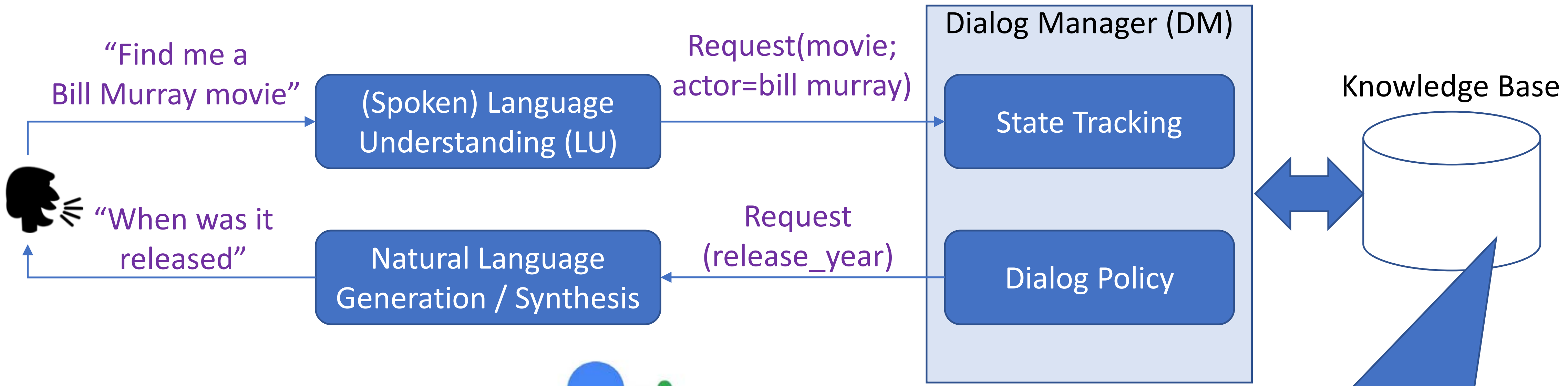
    thank-you(), request(price), inform(price=$10)

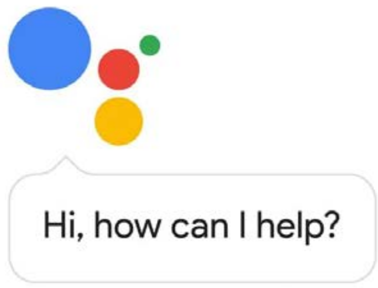
"Is Kungfu Panda the movie you are looking for?"

⇩

confirm(moviename="kungfu panda")

A Multi-turn Task-oriented Dialogue Architecture





Entity-Centric Knowledge Base

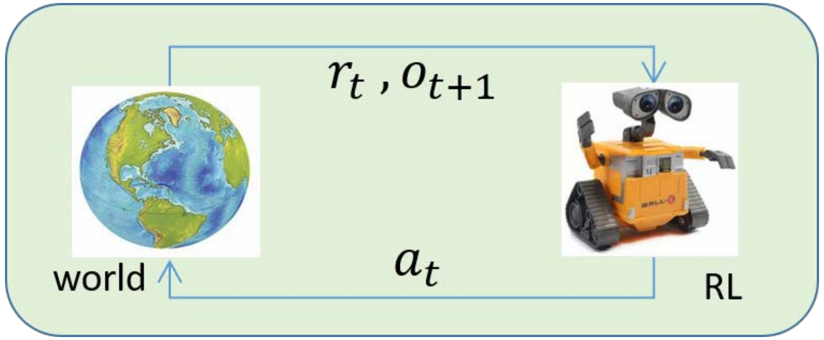
Movie	Actor	Release Year
Groundhog Day	Bill Murray	1993
Australia	Nicole Kidman	X
Mad Max: Fury Road	X	2015

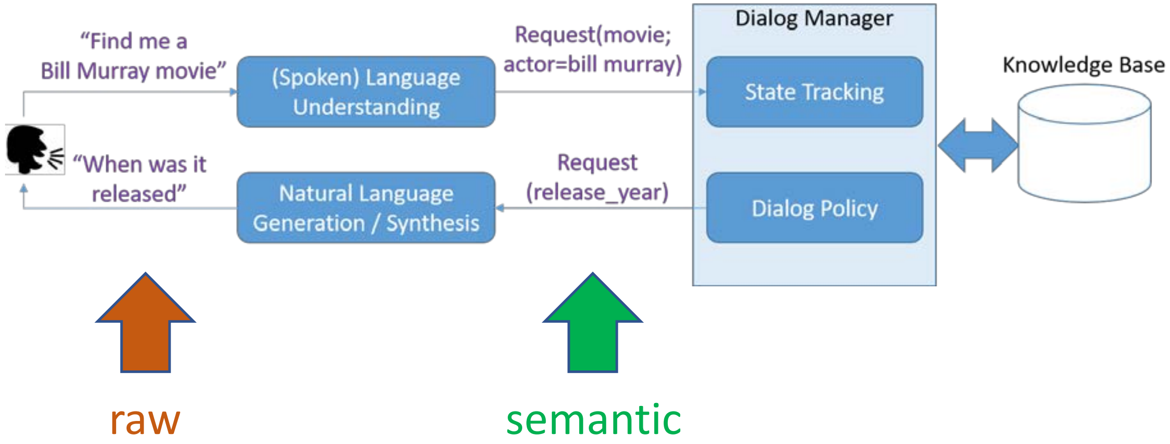
A unified view: dialogue as optimal decision making

- Dialogue as a Markov Decision Process (MDP)
  - Given state s, select action a according to (hierarchical) policy π
  - Receive reward r, observe new state a'
  - Continue the cycle until the episode terminates.
- Goal of dialogue learning: find optimal π to maximize expected rewards

Dialogue	State (s)	Action (a)	Reward (r)
Info Bots (Q&A bot over KB, Web etc.)	Understanding of user Intent (belief state)	Clarification questions, Answers	Relevance of answer # of turns
Task Completion Bots (Movies, Restaurants, …)	**Understanding of user goal (belief state)**	**Dialog act + slot_value**	**Task success rate # of turns**
Social Bot (XiaoIce)	Conversation history	Response	Engagement

Task-completion dialogue as RL



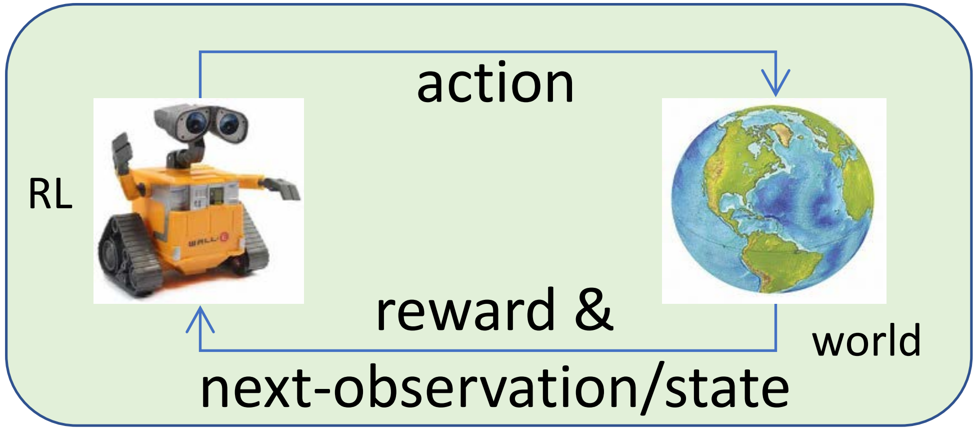


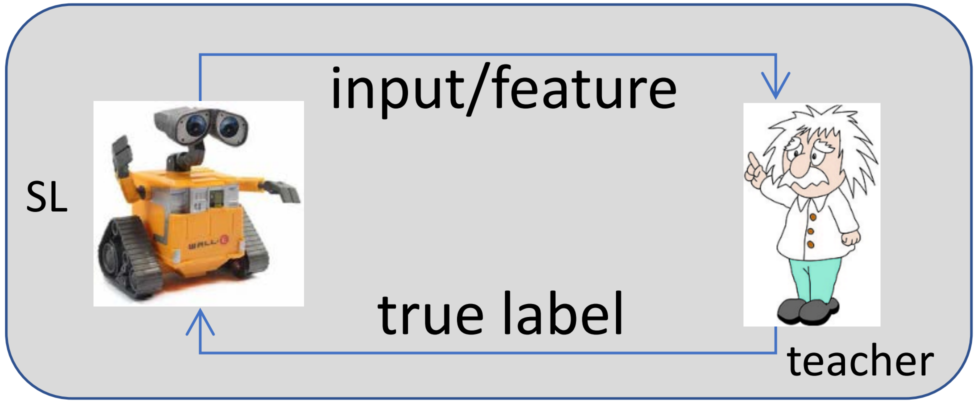
- Observation and action
  - Raw representation (utterances in natural language form)
  - Semantic representation (intent-slot-value form)
- Reward
  - +10 upon successful termination
  - -10 upon unsuccessful termination
  - -1 per turn
  - …

Pioneered by [Levin+ 00]

Other early examples: [Singh+ 02; Pietquin+ 04; Williams&Young 07; etc.]

RL vs. SL (supervised learning)





Differences from supervised learning

- **Learn by trial-and-error ("experimenting")**
  - ➢ **Need efficient exploration**
- Optimize long-term reward ($r_1 + \gamma r_2 + \cdots$)
  - ➢ Need temporal credit assignment

Similarities to supervised learning

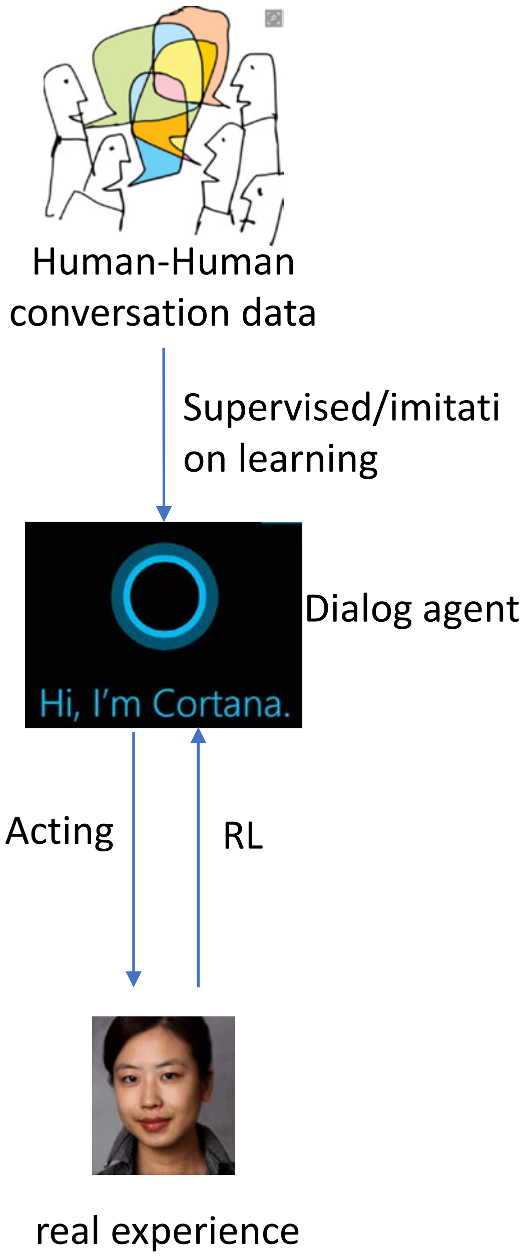
➢ Generalization and representation

➢ Hierarchical problem solving

➢ …

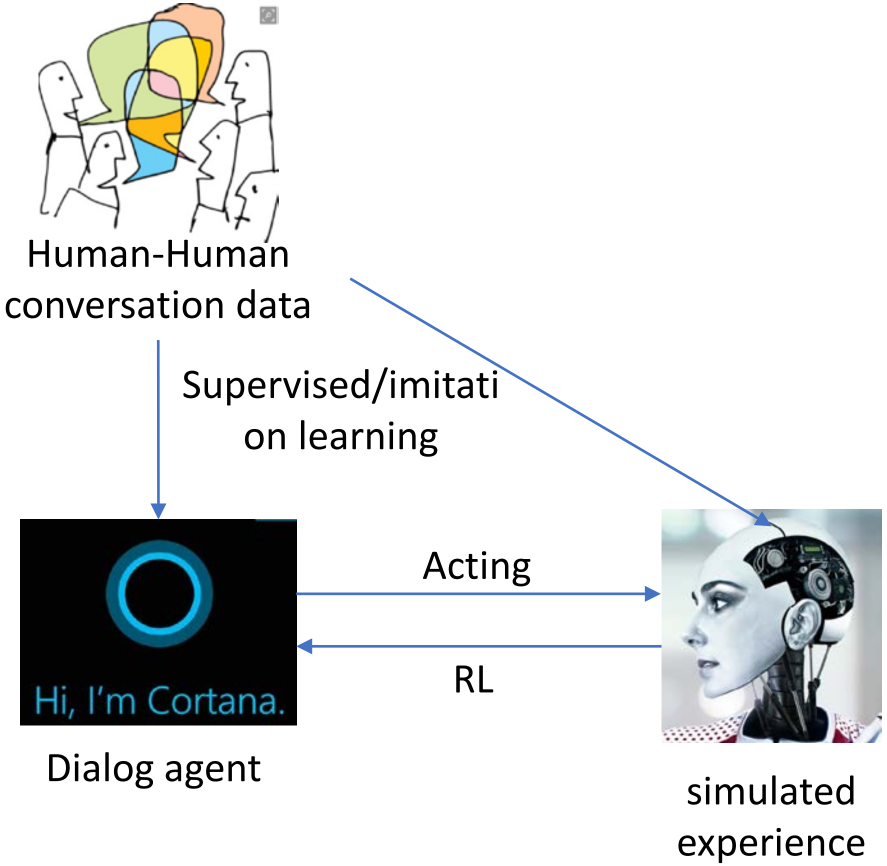
Learning w/ real users

- Expensive: need large amounts of real experience except for very simple tasks
- Risky: bad experiences (during exploration) drive users away



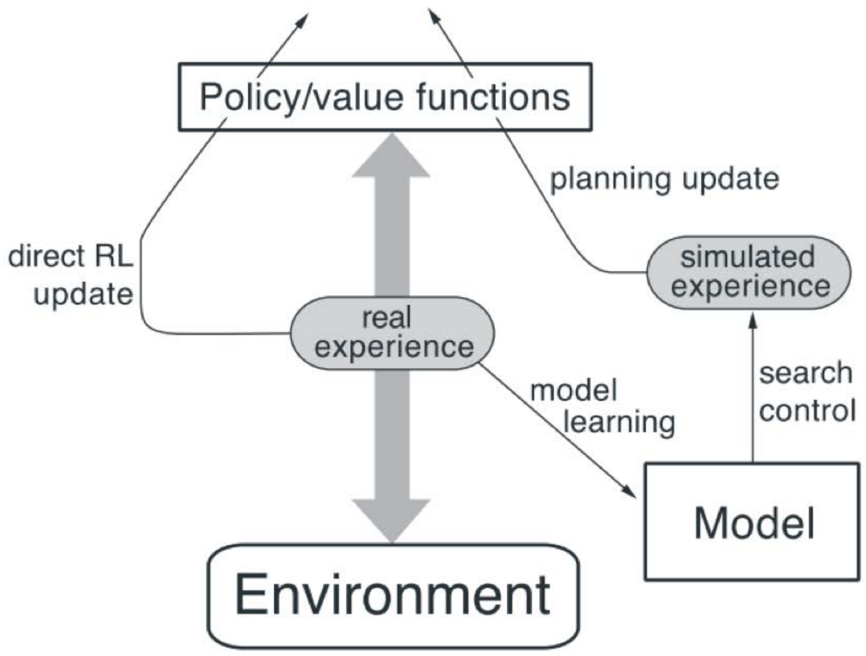
Learning w/ user simulators

- Inexpensive: generate large amounts of simulated experience for free
- Overfitting: discrepancy btw real users and simulators



Dyna-Q: integrating planning and learning
[Sutton+ 90]

- combining model-free and model-based RL
- tabular methods and linear function approximation
- **direct reinforcement learning**
- **(world) model learning**
- **planning/search control**



Deep Dyna-Q (DDQ): Integrating Planning for Dialogue Policy Learning

DDQ

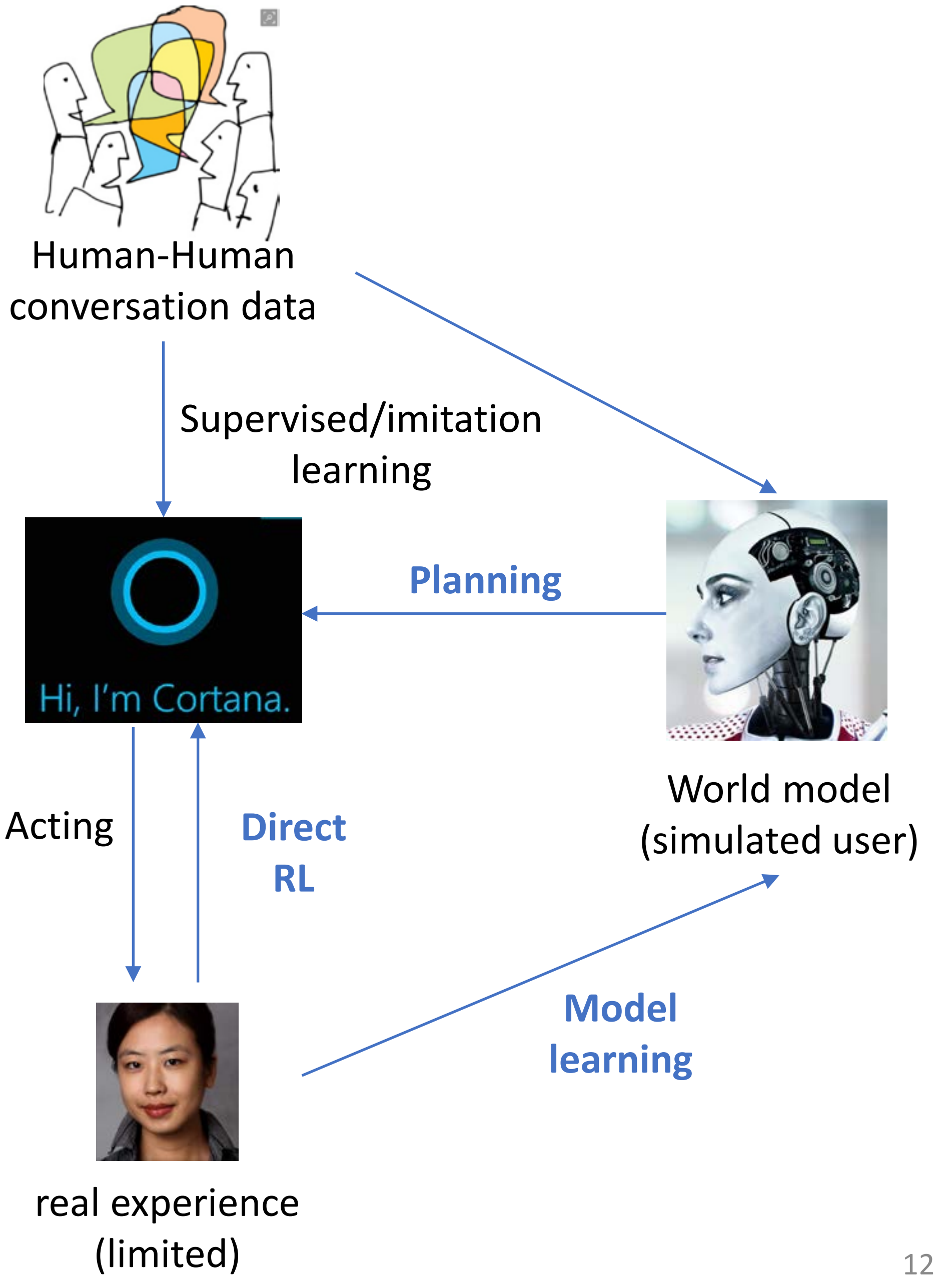
- Based on Dyna-Q
- Policy as DNN, trained using DQN
- Apply to dialogue: simulated user as world model

Dialogued agent trained using

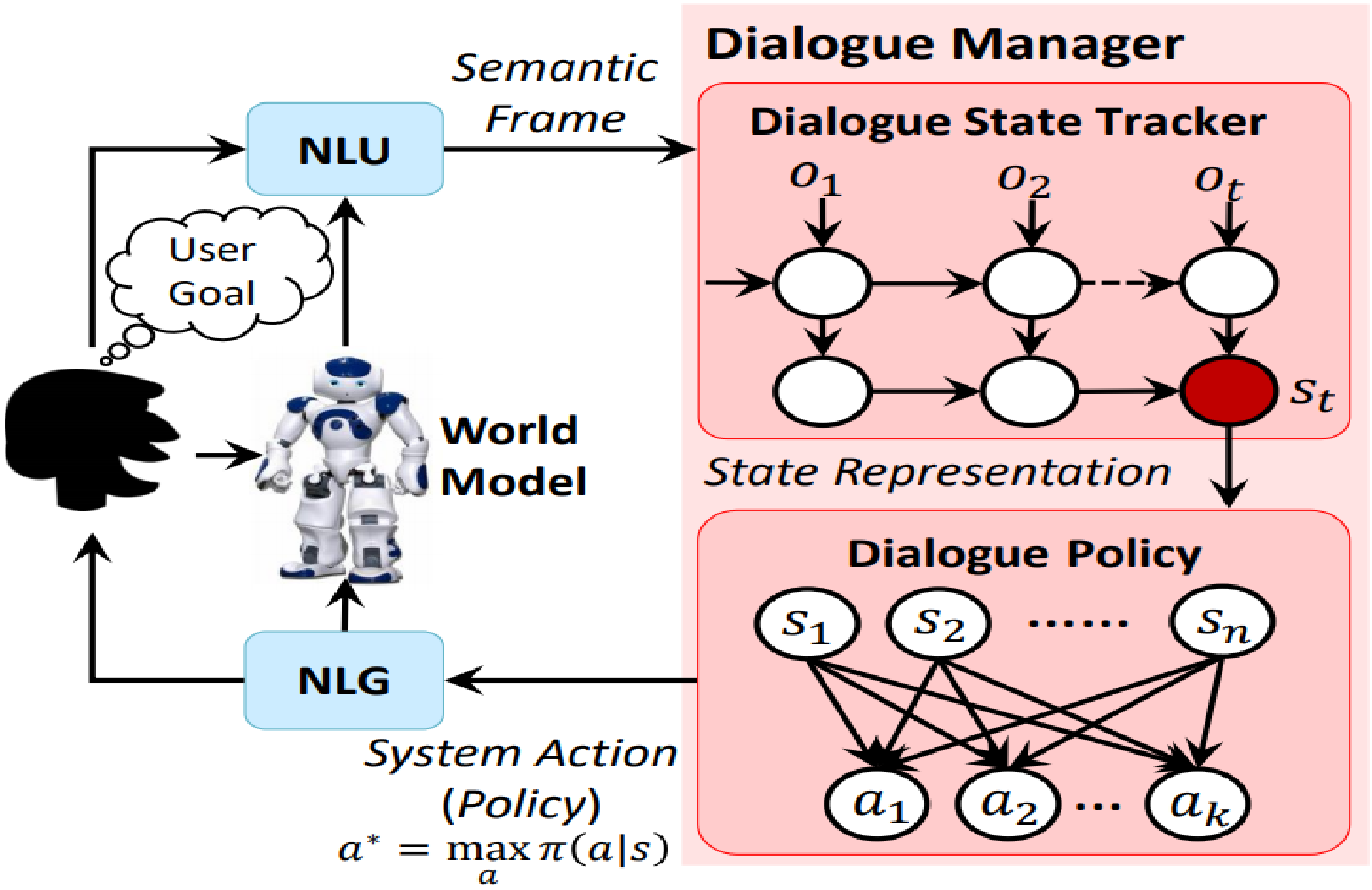
- Limited real user experience
- Large amounts of simulated experience

Limited real experience is used to improve

- Dialog agent
- World model (simulated user)

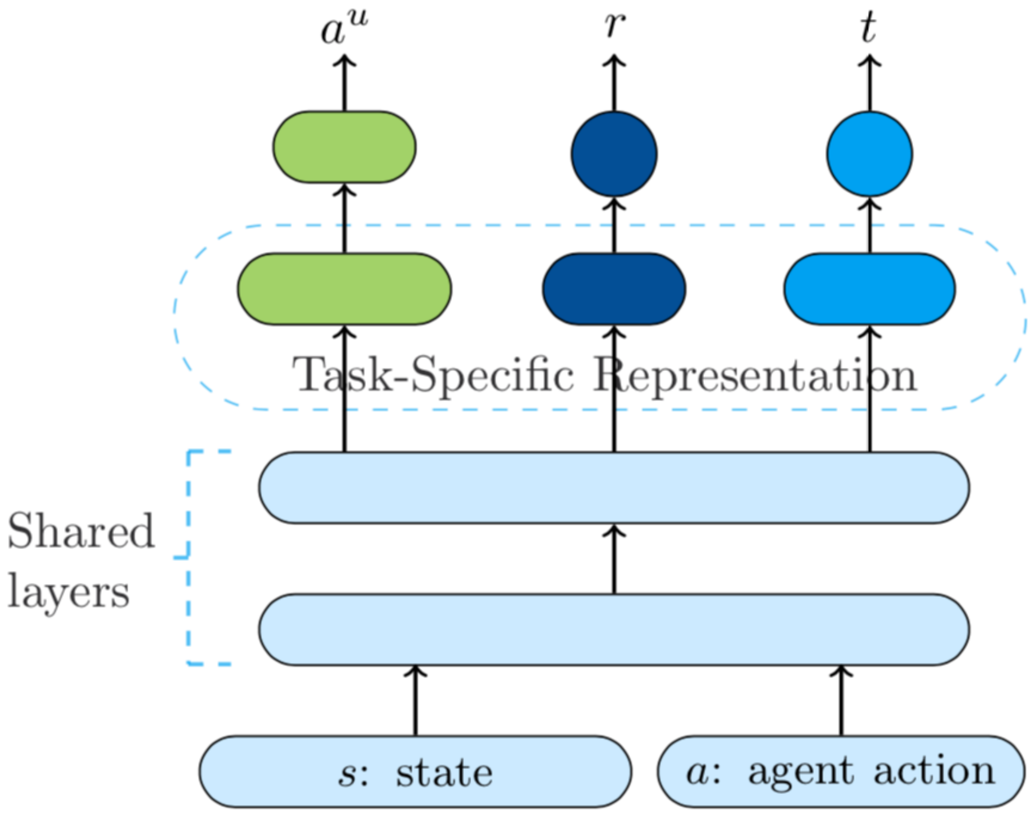


Task-completion DDQ dialogue agent



The world model architecture

- Multi task MLP
  - Reward r
  - User action a^u
  - Termination t



Dialogue System Evaluation

- **Metrics**: what numbers matter?
  - Success rate: #Successful_Dialogues / #All_Dialogues
  - Average turns: average number of turns in a dialogue
  - User satisfaction
  - Consistency, diversity, engaging, ...
  - Latency, backend retrieval cost, …

- **Methodology**: how to measure those numbers?

Evaluation methodology

	Lab user subjects	Actual users	Simulated users
Truthfulness		✓	✗
Scalability	✗	✓	✓
Flexibility	✗		✓
Expense	✗		✓
Risk	✓	✗	✓

A Hybrid Approach

User Simulation

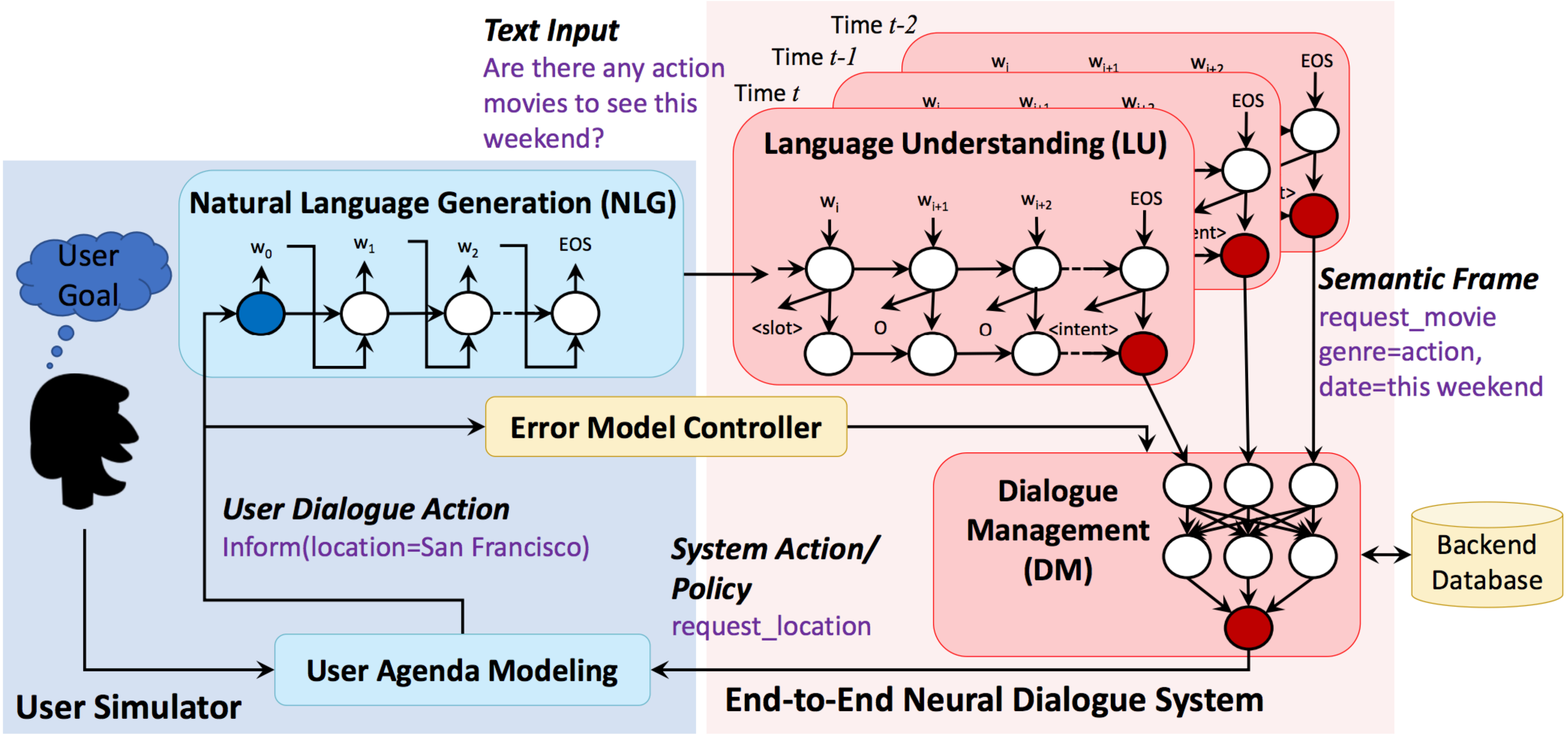
⇩

Small-scale Human Evaluation (lab, Mechanical Turk, …)

⇩

Large-scale Deployment (optionally with continuing incremental refinement)

A Simulator for E2E Neural Dialogue System [Li+ 17]



Agenda-based Simulated User [Schatzmann & Young 09]

- User state consists of (agenda, goal); goal is fixed throughout dialogue
- Agenda is maintained (stochastically) by a first-in-last-out stack

```
New episode, user goal:
{
    "request_slots": {
        "ticket": "UNK"
        "theater": "UNK"
        "starttime": "UNK"
    },
    "inform_slots": {
        "numberofpeople": "3",
        "date": "tomorrow",
        "moviename": "10 cloverfield lane"
    }
}
```

Req

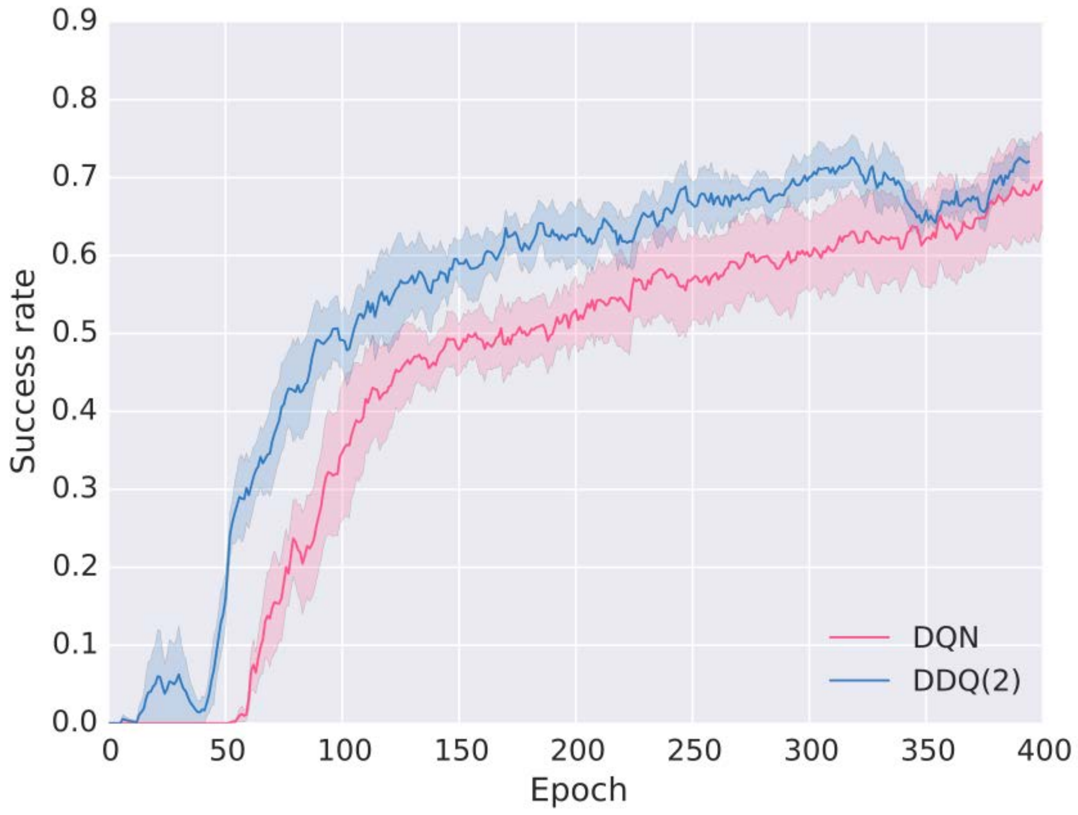
Inform

User: Which theater can I book 3 tickets for 10 cloverfield lane?
Agent: What time would you like to see it?
User: Which theater and start time are available tomorrow?
Agent: 11:45am is available.
User: Which theater is available?
Agent: regal la live stadium 14 is available.
User: Could you help me to book the tickets?
Agent: Okay - I was able to book 3 tickets for you to see 10 cloverfield lane at regal la live stadium 14 at 11:45am tomorrow.
User:Thank you.
Agent: Thank you.

Implementation of a simplified user simulator: https://github.com/MiuLab/TC-Bot

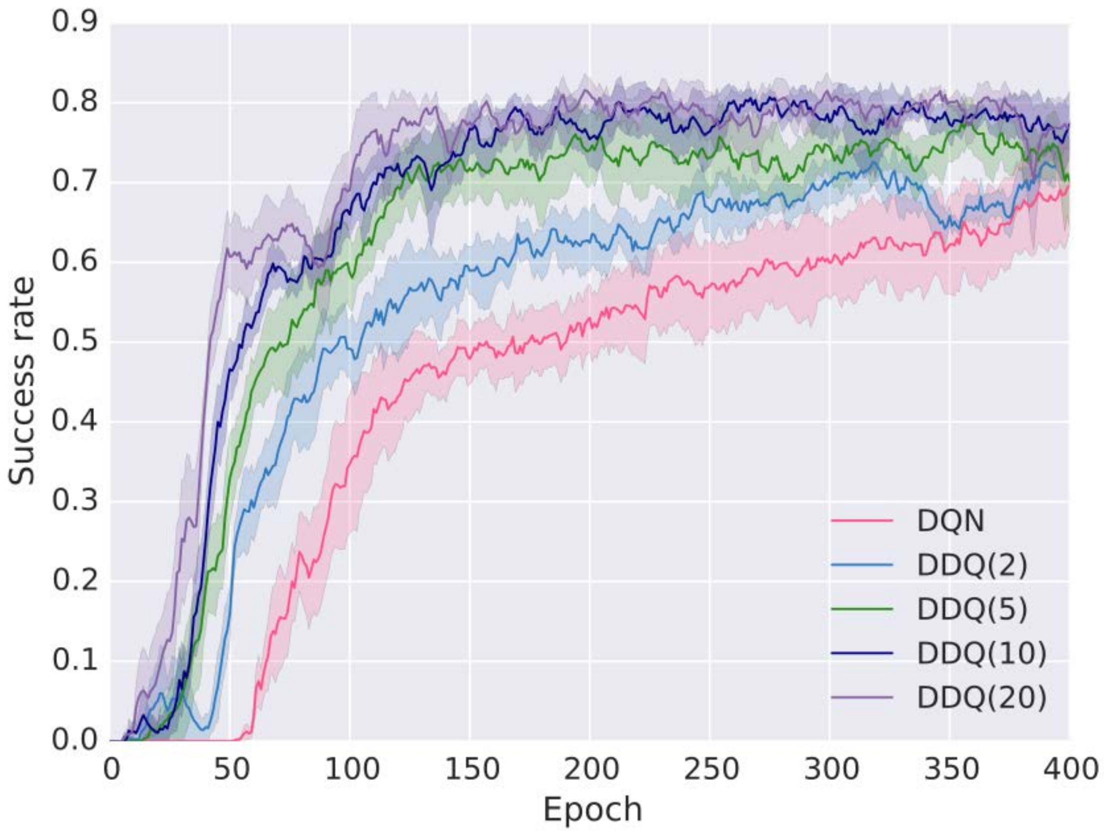
Simulated user evaluation

- DQN vs DDQ (K)
  - K: number of planning steps (generating K simulated dialogues per real dialogue)
  - $K = 2$



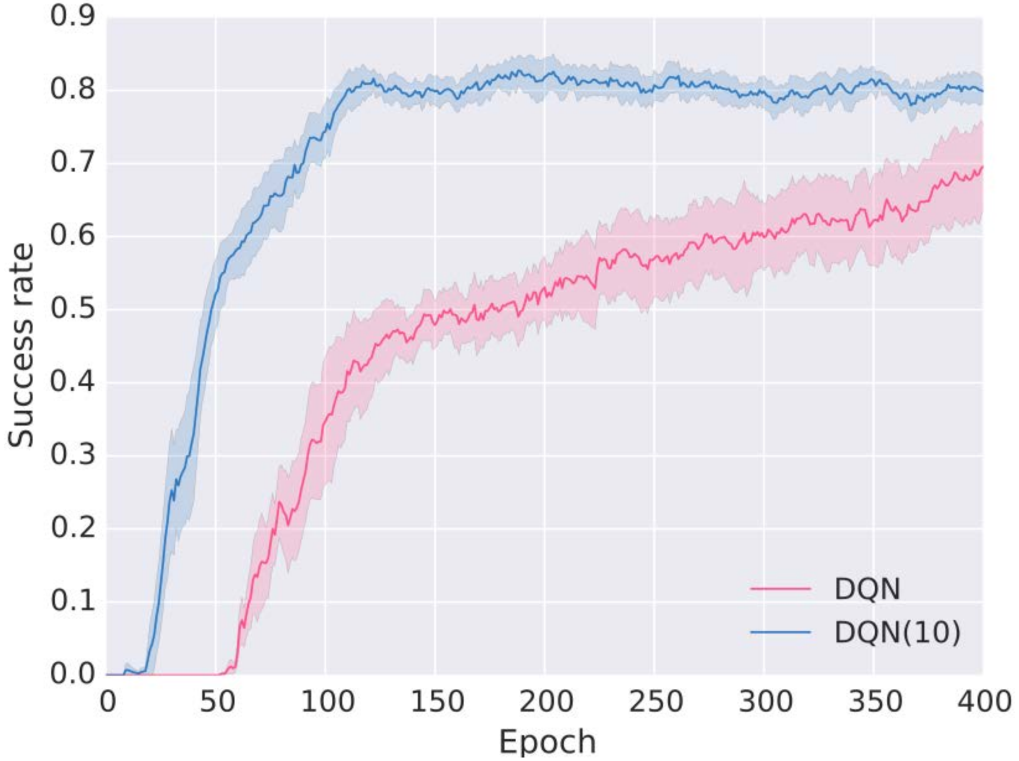
Simulated user evaluation

- DQN vs DDQ (K)
  - K: number of planning steps (generating K simulated dialogues per real dialogue)
  - $K = 2$, 5, 10, 20



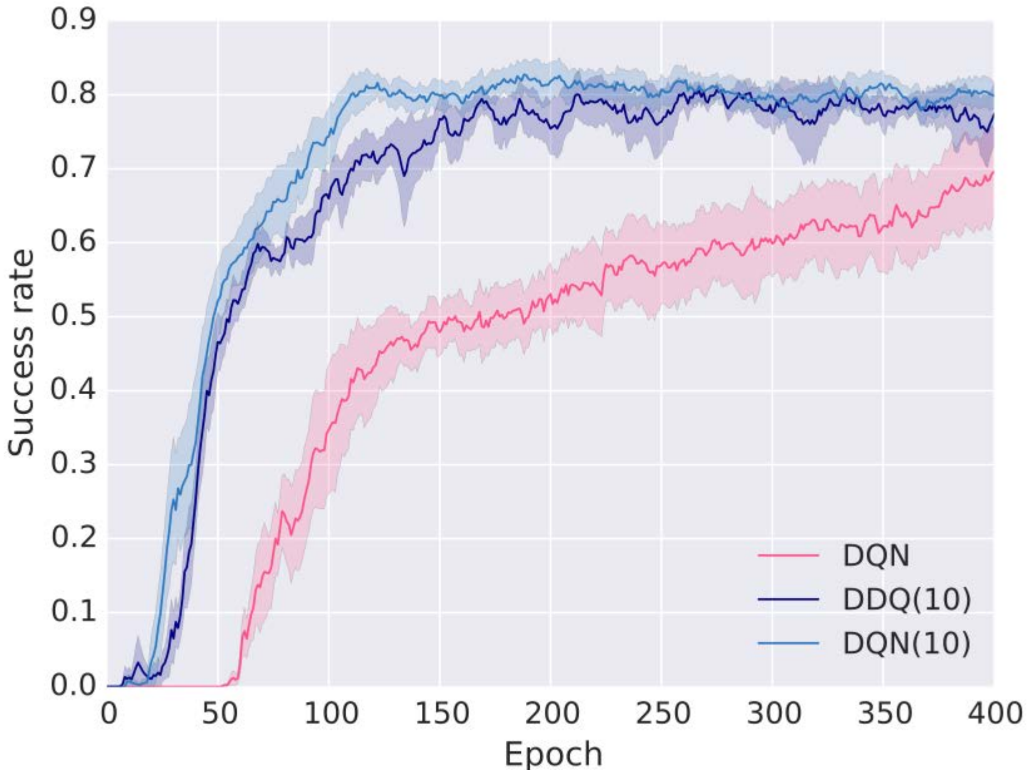
Impact of world model quality

- DQN(10):
  - perfect world model



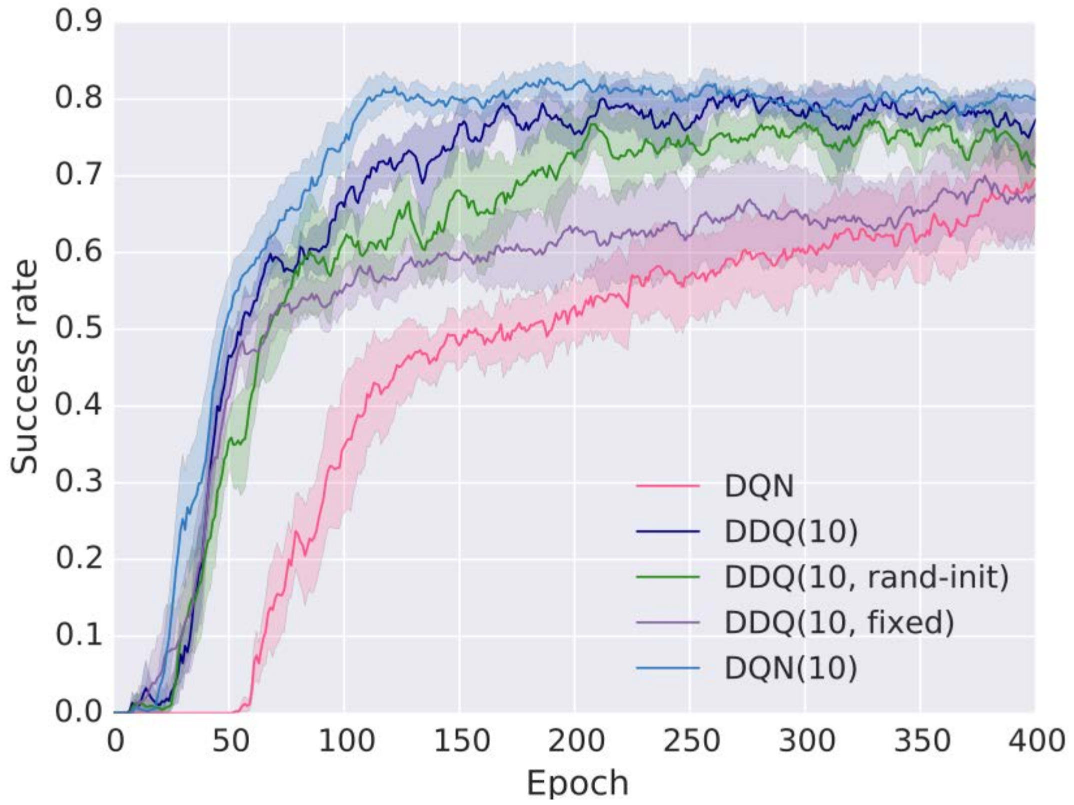
Impact of world model quality

- DQN(10)
  - perfect world model
- DDQ(10):
  - pretrained on labeled data, and updated using real dialogue on the fly



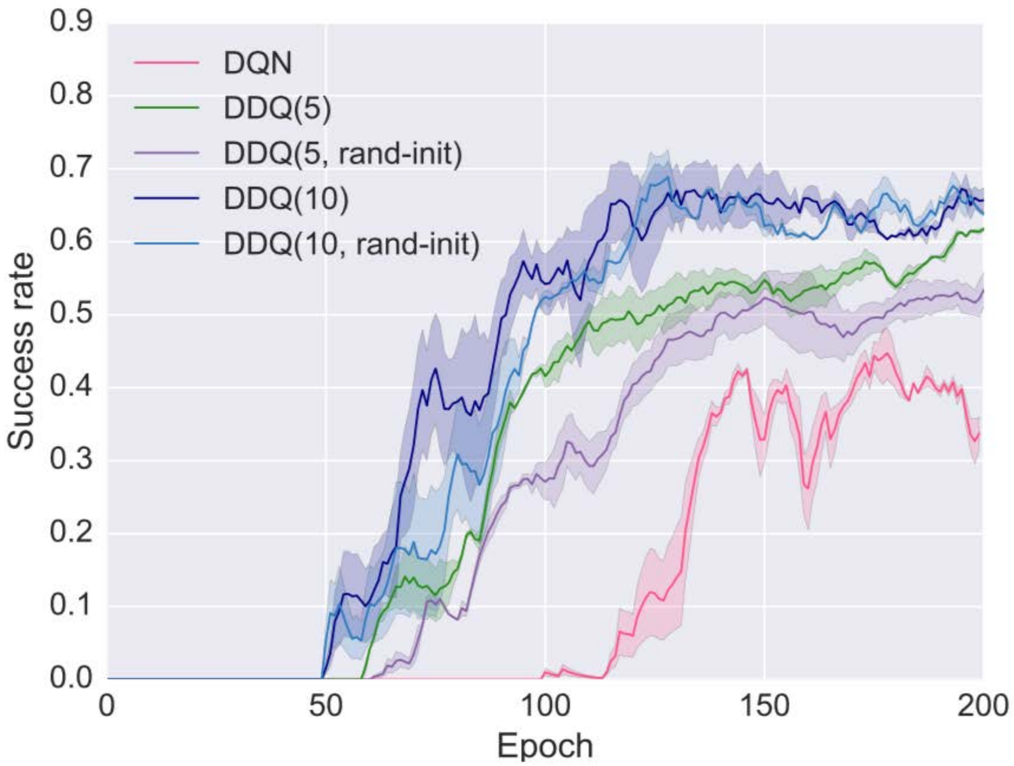
Impact of world model quality

- DQN(10)
  - perfect world model
- DDQ(10):
  - pretrained on labeled data, and updated using real dialogue on the fly
- DDQ(10, rand-init):
  - ~~pretrained on labeled data~~, and updated using real dialogue on the fly
- DDQ(10, fixed):
  - pretrained on labeled data, and ~~updated using real dialogue on the fly~~



Human-in-the-loop experiments
- learning dialogue via interacting with real users

- DDQ agents significantly outperforms the DQN agent
- A larger K leads to more aggressive planning and better results
- Pre-training world model with human conversational data improves the learning efficiency and the agent's performance



Conclusion and Future Work

- Deep Dyna-Q: integrating planning for dialogue policy learning
  - Improves learning efficiency
  - Make the best use of limited real user experiences
- Future research
  - Learning when to switch between real and simulated users
  - Exploration in planning
    - Exploration: trying actions to improve the world model
    - Exploitation: trying to behave in the optimal way given the current world model

Microsoft Dialogue Challenge at SLT-2018

- 07/16/2018: Registration is now open.
- Task: build E2E task-completion dialogue systems
- Data: labeled human conversations in 3 domains
- Experiment platform with built-in user simulators for training and evaluation
- Final evaluation in simulated setting and by human judges

- More information:

https://github.com/xiul-msr/e2e_dialog_challenge