
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

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

in seattle at 10:00 pm.

Turn 10 usr: thanks

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

“Find me a Bill Murray movie”

(Spoken) Language Understanding (LU)

Request(movie; actor=bill murray)

“When was it released”

Natural Language Generation / Synthesis

Request (release_year)

Dialog Manager (DM)

State Tracking

Dialog Policy

Knowledge Base

Entity-Centric Knowledge Base

<table>
<thead>
<tr>
<th>Movie</th>
<th>Actor</th>
<th>Release Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groundhog Day</td>
<td>Bill Murray</td>
<td>1993</td>
</tr>
<tr>
<td>Australia</td>
<td>Nicole Kidman</td>
<td>X</td>
</tr>
<tr>
<td>Mad Max: Fury Road</td>
<td>X</td>
<td>2015</td>
</tr>
</tbody>
</table>
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 \( \pi \)
  • Receive reward \( r \), observe new state \( a' \)
  • Continue the cycle until the episode terminates.

• Goal of dialogue learning: find optimal \( \pi \) to maximize expected rewards

<table>
<thead>
<tr>
<th>Dialogue</th>
<th>State (s)</th>
<th>Action (a)</th>
<th>Reward (r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info Bots (Q&amp;A bot over KB, Web etc.)</td>
<td>Understanding of user Intent (belief state)</td>
<td>Clarification questions, Answers</td>
<td>Relevance of answer # of turns</td>
</tr>
<tr>
<td>Task Completion Bots (Movies, Restaurants, ...)</td>
<td>Understanding of user goal (belief state)</td>
<td>Dialog act + slot_value</td>
<td>Task success rate # of turns</td>
</tr>
<tr>
<td>Social Bot (XiaoIce)</td>
<td>Conversation history</td>
<td>Response</td>
<td>Engagement</td>
</tr>
</tbody>
</table>
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
➢ ...

true label
input/feature
teacher
world
next-observation/state
action

true label
input/feature
teacher
world
next-observation/state
action

SL

RL
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?
  o Success rate: $\#\text{Successful\_Dialogues} \div \#\text{All\_Dialogues}$
  o Average turns: average number of turns in a dialogue
  o User satisfaction
  o Consistency, diversity, engaging, ...
  o Latency, backend retrieval cost, ...

• **Methodology**: how to measure those numbers?
### Evaluation methodology

<table>
<thead>
<tr>
<th></th>
<th>Lab user subjects</th>
<th>Actual users</th>
<th>Simulated users</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Truthfulness</strong></td>
<td></td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Expense</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Risk</strong></td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

#### 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

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 outperform 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