#### MiuLab / TC-Bot

#### Join GitHub today

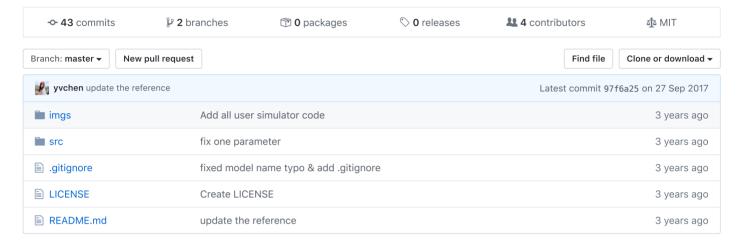
Dismiss

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

Sign up

User Simulation for Task-Completion Dialogues

#user-simulator #nlg #dialogue-agents #nlu #end-to-end



**■ README.md** 

# **End-to-End Task-Completion Neural Dialogue Systems**

An implementation of the

End-to-End Task-Completion Neural Dialogue Systems and A User Simulator for Task-Completion Dialogues.

This document describes how to run the simulation and different dialogue agents (rule-based, command line, reinforcement learning). More instructions to plug in your customized agents or user simulators are in the Recipe section of the paper.

#### Content

- Data
- Parameter
- Running Dialogue Agents
- Evaluation
- Reference

#### Data

all the data is under this folder: ./src/deep\_dialog/data

Movie Knowledge Bases

```
movie_kb.1k.p --- 94% success rate (for user_goals_first_turn_template_subsets.v1.p )
movie_kb.v2.p --- 36% success rate (for user_goals_first_turn_template_subsets.v1.p )
```

· User Goals

user\_goals\_first\_turn\_template.v2.p --- user goals extracted from the first user turn user\_goals\_first\_turn\_template.part.movie.v1.p --- a subset of user goals [Please use this one, the upper bound success rate on movie\_kb.1k.json is 0.9765.]

https://github.com/MiuLab/TC-Bot

- NLG Rule Template
   dia\_act\_nl\_pairs.v6.json --- some predefined NLG rule templates for both User simulator and Agent.
- Dialog Act Intent dia\_acts.txt
- Dialog Act Slot slot\_set.txt

#### **Parameter**

#### **Basic setting**

```
--agt : the agent id
--usr : the user (simulator) id
--max_turn : maximum turns
--episodes : how many dialogues to run
--slot_err_prob : slot level err probability
--slot_err_mode : which kind of slot err mode
--intent_err_prob : intent level err probability
```

## Data setting

```
--movie_kb_path : the movie kb path for agent side
--goal_file_path : the user goal file path for user simulator side
```

#### Model setting

```
--dqn_hidden_size: hidden size for RL (DQN) agent
--batch_size: batch size for DQN training
--simulation_epoch_size: how many dialogue to be simulated in one epoch
--warm_start: use rule policy to fill the experience replay buffer at the beginning
--warm_start_epochs: how many dialogues to run in the warm start
```

# Display setting

--run\_mode: 0 for display mode (NL); 1 for debug mode (Dia\_Act); 2 for debug mode (Dia\_Act and NL); >3 for no display (i.e. training)

```
    --act_level: 0 for user simulator is Dia_Act level; 1 for user simulator is NL level
    --auto_suggest: 0 for no auto_suggest; 1 for auto_suggest
    --cmd_input_mode: 0 for NL input; 1 for Dia_Act input. (this parameter is for AgentCmd only)
```

#### **Others**

```
--write_model_dir: the directory to write the models
--trained_model_path: the path of the trained RL agent model; load the trained model for prediction purpose.
```

——learning\_phase: train/test/all, default is all. You can split the user goal set into train and test set, or do not split (all); We introduce some randomness at the first sampled user action, even for the same user goal, the generated dialogue might be different.

# **Running Dialogue Agents**

#### **Rule Agent**

# **Cmd Agent**

**NL** Input

Dia Act Input

#### **End2End RL Agent**

Train End2End RL Agent without NLU and NLG (with simulated noise in NLU)

Train End2End RL Agent with NLU and NLG

Test RL Agent with N dialogues:

# **Evaluation**

To evaluate the performance of agents, three metrics are available: success rate, average reward, average turns. Here we show the learning curve with success rate.

- Plotting Learning Curve python draw\_learning\_curve.py --result\_file
   ./deep\_dialog/checkpoints/rl\_agent/noe2e/agt\_9\_performance\_records.json
- 2. Pull out the numbers and draw the curves in Excel

## Reference

Main papers to be cited

```
@inproceedings{li2017end,
    title={End-to-End Task-Completion Neural Dialogue Systems},
    author={Li, Xuijun and Chen, Yun-Nung and Li, Lihong and Gao, Jianfeng and Celikyilmaz, Asli},
    booktitle={Proceedings of The 8th International Joint Conference on Natural Language Processing},
    year={2017}
}

@article{li2016user,
    title={A User Simulator for Task-Completion Dialogues},
    author={Li, Xiujun and Lipton, Zachary C and Dhingra, Bhuwan and Li, Lihong and Gao, Jianfeng and Chen,
    Yun-Nung},
    journal={arXiv preprint arXiv:1612.05688},
    year={2016}
}
```

https://github.com/MiuLab/TC-Bot