ConvoBot: A Conversational Bot via Deep Q-Learning and Query Simulation

Vidush Mukund vmukund@stanford.edu
Brian Wilcox wilcoxee@stanford.edu

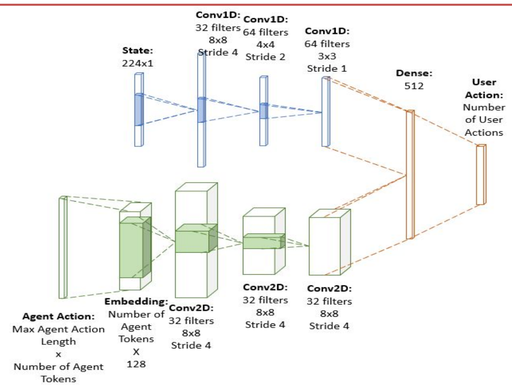
Motivation

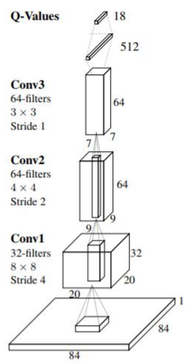
Generally, conversational chatbots are difficult to train as they require well-structured dialogue formats and significant overhead to make sure that knowledge is learned by agents that do not directly know the truth queries. A non-goal oriented approach is practically infeasible as it would require universal knowledge and the memory buffer for immediately learned knowledge would not be sufficient for any kind of complex query. This requires the scope of the problem to be reduced to a specific goal. Additionally, identifying proper behavior by a chatbot agent is difficult as it requires extensive data labeling and generation for both the end-user and the chatbot agent itself.

With this project, there were two major goals at hand. The first goal of this project is to expand upon past work in using generative chatbots to help 'teach' a reinforcement learning agent how to engage in a dialogue environment. The second goal is to take a NLP-based approach to the user agent. In our approach, the user agent is modeled by the SA2A architecture. for conversational response generation and is trained on a corpus of conversation dialogue. This user agent is then used to train a DQN agent as a chatbot via the end-to-end dialogue system.

Models

State-Action2Action Model for User Simulation Module.





DQN Architecture used for the Agent Module [2].

Data and Features

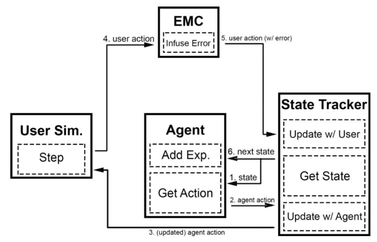
The pipeline was trained on a database of movie theater tickets represented as a dictionary where the keys are the indices of the tickets and the values are also dictionaries which contain the movie information.

```
0L: {'city': 'hamilton', 'theater': 'manville 12 plex', 'zip': '08835', 'critic_rating': 'good', 'genre': 'comedy', 'state': 'nj', 'starttime': '10:30am', 'date': 'tomorrow', 'moviename': 'zootopia'}
897L: {'city': 'seattle', 'theater': 'pacific place 11', 'moviename': 'how to be single', 'zip': '98101', 'critic_rating': 'top', 'date': 'tonight', 'state': 'washington', 'other': 'date', 'starttime': '9',
'theater_chain': 'amc', 'genre': 'romance'}
721L: {'city': 'bellevue', 'theater': 'regal meridian 16', 'zip': '98101', 'state': 'washington', 'mpaa_rating': 'pg', 'starttime': 'matinee', 'date': '7th', 'moviename': 'kung fu panda 3'}
536L: {'city': 'seattle', 'theater': 'regal meridian 16', 'zip': '98133', 'moviename': 'risen race spotlight', 'date': 'tomorrow', 'state': 'wa', 'other': 'large number of movies', 'starttime': '7pm',
'theater_chain': 'amc loews oak tree 6', 'genre': 'comedy'}
61L: {'city': 'johnstown', 'theater': 'cinemas', 'video_format': 'standard/2D version', 'state': 'pennsylvania', 'starttime': 'earliest showing', 'date': 'tomorrow afternoon', 'moviename': 'zootopia'}
```

Movie Database Examples

Methods

Our Training Loop [1].



Training of the DQN is performed in a loop that involves:

1. Generating a user action from a simulator
2. Infusing error in the user action to model human error
3. Updating a state tracker that maintains the conversational state
4. Produce an agent action based on the conversational state
5. Update the conversational state based on the agent action.
6. Use the agent action output to update the user state and output the success/reward of the agent action.
7. Get the next user action and repeat Steps 2- until the max number of samples per update is reached
8. Update the DQN weights based on the output rewards in Step 6.

.

To train the generative User module, the SA2A approach is trained offline on data collected from a training sequence of the pipeline that uses the rule-based User module. In essence, this supervised learning approach utilizes the behavior of the rule-based User module to train the SA2A User.

Results

Model	Average Reward	Average Success
Vanilla	21.073 ± 27.499	0.8296
DQN	22.702 ± 25.992	0.8502

Table 1. DQN versus Vanilla Results

F1 Score	Recall	Precision
0.944	0.884	0.913

Table 2. SA2A Model Metrics

- Test results for the baseline show a success rate in answering queries correctly of 0.83 and an average reward of 21.07
- DQN agent performs marginally better with a success rate of of 0.85 and an average reward return of 22.7
  - tighter bounds on standard deviation
  - DQN model has far greater capacity than the simpler vanilla model.

- SA2A model, the approach has a high micro-average f1 score of 0.94 as well as high precision and recall scores
- due to the nature of acquisition via simulation, is unbalanced. It is promising to see the model perform well even with imbalances in data.

```
True User Action:{intent:inform,request_slots:{},inform_slots:{'city': 'portland'}}
Model User Action:{intent:inform,request_slots:{},inform_slots:{'city': 'los angeles'}}
```

Example 1: Wrong City

```
True User Action:{intent:inform,request_slots:{},inform_slots:{'starttime': '10:00 pm'}}
Model User Action:{intent:inform,request_slots:{},inform_slots:{'starttime': '9:30 pm'}}
```

Example 2: Wrong Start Time

```
True User Action:{intent:inform,request_slots:{},inform_slots:{'date': 'tuesday'}}
Model User Action:{intent:inform,request_slots:{},inform_slots:{'theater': 'amc lowes oak tree 6'}}
```

Example 3: Wrong answer altogether.

SA2A Error Cases

Discussion

The results shown in Table 1 suggest that the DQN model is more scalable for the chatbot agent than the vanilla implementation. This suggests that the chatbot pipeline benefits from a DQN implementation and could benefit from more complex implementations in the future.

The training of the SA2A showed strong results in terms of f1 score, precision, and recall. It would help to have a baseline to compare this to but since the default user simulator is our baseline in this case, we are comparing to perfect accuracy. Overall, our model can generalize well for this dataset but it would be worthwhile to explore this approach for other methods like action2action directly.

Conclusion/Future Work

- Shown the effectiveness of using a DQN agent for a conversation chatbot in a database query setting as well as the first steps into designing an end-to-end training pipeline with rule agnostic user simulation
- Shown the greater generality afforded by a model like the SA2A model for user simulation
- Explore variations on the DQN architecture for Agent module
- Switch to a larger dataset with more natural conversational queries and step away completely from a simplistic rule-based approach

References

[1] Max Brenner.Goal-oriented chatbot with deep rl.https://towardsdatascience.com/training-a-goal-oriented-chatbot-with-deep-reinforcemnt-learning-part-i-introduction-and-dce3af21d383 , 2018. Accessed: 2019-02-12.

[2] Matthew J. Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps.CoRR, abs/1507.06527, 2015. URL http://arxiv.org/abs/1507.06527.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[4] Mahipal Jadeja, Neelanshi Varia, and Agam Shah. Deep reinforcement learning for conversational AI. CoRR, abs/1709.05067, 2017. URL http://arxiv.org/ abs/1709.05067