Language Understanding for Text-based Games Using Deep Reinforcement Learning

Karthik Narasimhan, Tejas Kulkarni, Regina Barzilay

MIT
Text-based games

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

>> go east

(State 2: Ruined gatehouse)
The old gatehouse is near collapse. Part of its northern wall has already fallen down ... East of the gatehouse leads out to a small open area surrounded by the remains of the castle. ...

MUDs: predecessors to modern graphical games
Why are they challenging?

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

Location: Bridge 1
Wind level: 3
Time: 8pm

Branavan et al., 2011

No symbolic representation available
Can a computer understand language enough in order to play these games?

Understanding $\approx$ Actionable intelligence
Can a computer understand language enough in order to play these games?

Inspiration: Playing graphical games directly from raw pixels (DeepMind)
Our Approach

Reinforcement Learning utilizing in-game feedback to:

✦ Learn control policies for gameplay.

✦ Learn good representations for text description of game state.
Traditional RL framework

\[ s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \ldots \xrightarrow{} s_t \]

\[ Q(s, a) \]

Location: Bridge 1
Wind level: 3
Time: 8pm

Q-value is the agent’s notion of discounted future reward
Text-based games

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ...
Text-based games: BOW representation

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ...

Bag of words?
Can we do better?
Recurrent NN to map text to vector representation

Model

Input text → T → v → Q → Q values for all commands
Model

Recurrent NN to map text to vector representation

NN for control policy

Q values for all commands
LSTM-DQN

Action-Object Scorer

Representation Generator

Q(s, a)

Q(s, o)

Linear

ReLU

Linear

Mean Pooling

LSTM

LSTM

LSTM

LSTM

w_1

w_2

w_3

w_n

\phi_A

\phi_R
Algorithm (1)

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

Obtain Q-values
Algorithm (2)

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

Take action using $\epsilon$-greedy
Algorithm (3)

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

(State 2: Ruined gatehouse)
The old gatehouse is near collapse. Part of its northern wall has already fallen down ... East of the gatehouse leads out ...

+ reward
Algorithm (4)

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be

(State 2: Ruined gatehouse)
The old gatehouse is near collapse. Part of its northern wall has already fallen down ... East of the gatehouse leads out ...

\[ \sim \] \text{Sample transitions for updates}

\vdots

\vdots

\text{Store transition in experience memory}
Parameter update

(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

(State 2: Ruined gatehouse)
The old gatehouse is near collapse. Part of its northern wall has already fallen down ... East of the gatehouse leads out ...

\begin{align*}
\nabla_{\theta_i} L_i(\theta_i) &= \mathbb{E}_{\hat{s}, \hat{a}} \left[ 2(y_i - Q(\hat{s}, \hat{a}; \theta_i)) \nabla_{\theta_i} Q(\hat{s}, \hat{a}; \theta_i) \right] \\
\text{where } y_i &= \mathbb{E}_{\hat{s}, \hat{a}} \left[ r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) \right] \mid \hat{s}, \hat{a}
\end{align*}
Game Environment

**Evennia**: a highly extensible python framework for MUD games

Two worlds:

- small game to demonstrate task and analyze learnt representations.
- a pre-existing Fantasy world.
Home World

- Number of different quests: 16
- Vocabulary: 84 words
- Words per description (avg.): 10.5
- Multiple descriptions per room/object.
This room has two sofas, chairs and a chandelier. You are not sleepy now but you are hungry now.

> go east
This area has plants, grass and rabbits. You are not sleepy now but you are hungry now.

> go south
You have arrived in the kitchen. You can find food and drinks here. You are not sleepy now but you are hungry now.

> eat apple

Reward: +1
(State 1: The old bridge)
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

- Number of rooms: > 56
- Vocabulary: 1340 words
- Avg. no. of words/description: 65.21
- Max descriptions per room: 100

- Considerably more complex
- Varying descriptions per state created by game developers
Evaluation

Two metrics:

✦ Quest completion

✦ Cumulative reward per episode
  • Positive rewards for quest fulfillment
  • Negative rewards for bad actions

Epoch: Training for \( n \) episodes followed by evaluation on \( n \) episodes
Baselines

- Randomly select actions
- Bag of words: unigrams and bigrams

Input text → T → \[
\begin{bmatrix}
0 \\
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\] → Q → Q values
Agent Performance (Home)

Random agent performs poorly
Agent Performance (Home)

LSTM-DQN has delayed performance jump
Agent Performance (Fantasy)

Good representation is essential for successful gameplay.
Visualizing Learnt Representations

t-SNE visualization of vectors learnt by agent on Home world
Visualizing Learnt Representations

t-SNE visualization of vectors learnt by agent on Home world
Nearby states: Similar representations

<table>
<thead>
<tr>
<th>Description</th>
<th>Nearest neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are halfways out on the unstable bridge. From the castle you hear a</td>
<td>The bridge slopes precariously where it extends westwards towards the lowest</td>
</tr>
<tr>
<td>distant howling sound, like that of a large dog or other beast.</td>
<td>point - the center point of the hang bridge. You clasp the ropes firmly as the</td>
</tr>
<tr>
<td></td>
<td>bridge sways and creaks under you.</td>
</tr>
<tr>
<td>The ruins opens up to the sky in a small open area, lined by columns.</td>
<td>The old gatehouse is near collapse. east the gatehouse leads out to a small open</td>
</tr>
<tr>
<td>... To the west is the gatehouse and entrance to the castle, whereas</td>
<td>area surrounded by the remains of the castle. There is also a standing archway</td>
</tr>
<tr>
<td>southwards the columns make way for a wide open courtyard.</td>
<td>offering passage to a path along the old southern inner wall.</td>
</tr>
</tbody>
</table>
Transfer Learning (Home)

Play on world with same vocabulary but different physical configuration
Conclusions

› Addressed the task of end-to-end learning of control policies for textual games.

› Learning good representations for text is essential for gameplay.

Code and game framework are available at: http://people.csail.mit.edu/karthikn/mud-play/