Using Natural Language for Reward Shaping in Reinforcement Learning

Prasoon Goyal, Scott Niekum and Raymond J. Mooney

The University of Texas at Austin
Motivation
Motivation

- In sparse reward settings, random exploration has very high sample complexity.
Motivation

- In sparse reward settings, random exploration has very high sample complexity.
- Reward shaping: Intermediate rewards to guide the agent towards the goal.
Motivation

- In sparse reward settings, random exploration has very high sample complexity.
- Reward shaping: Intermediate rewards to guide the agent towards the goal.
- Designing intermediate rewards by hand is challenging.
Motivation

Can we use natural language to provide intermediate rewards to the agent?
Motivation

Can we use natural language to provide intermediate rewards to the agent?

Jump over the skull while going to the left.
Problem Statement

- Standard MDP formalism, plus a natural language command describing the task.

Jump over the skull while going to the left

Observations + Reward → Agent

Agent → Action

Action → Environment
Approach Overview

- Standard MDP formalism, plus a natural language command describing the task.

- Use agent’s past actions and the command to generate rewards.
Approach Overview

- Standard MDP formalism, plus a natural language command describing the task.
- Use agent’s past actions and the command to generate rewards.

For example,

<table>
<thead>
<tr>
<th>Past actions</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLLJLLLL</td>
<td>High</td>
</tr>
<tr>
<td>RRRUULLL</td>
<td>Low</td>
</tr>
</tbody>
</table>

[L: Left, R: Right, U: Up, J: Jump]
Approach Overview

- Standard MDP formalism, plus a natural language command describing the task.
- Use agent’s past actions and the command to generate rewards.

For example,

<table>
<thead>
<tr>
<th>Past actions</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>4441444</td>
<td>High</td>
</tr>
<tr>
<td>3332244</td>
<td>Low</td>
</tr>
</tbody>
</table>

[4: Left, 3: Right, 2: Up, 1: Jump]
Problem: Given a sequence of actions (e.g. 4441444) and a command (e.g. “Jump over the skull while going to the left”), are they related?
LanguageE-Action Reward Network (LEARN)

**Problem**: Given a sequence of actions (e.g. 4441444) and a command (e.g. “Jump over the skull while going to the left”), are they related?

- Using the sequence of actions, generate an action-frequency vector:

  \[
  \begin{align*}
  \epsilon & \Rightarrow [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\
  4 & \Rightarrow [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0] \\
  42 & \Rightarrow [0 \ 0 \ 0.5 \ 0 \ 0.5 \ 0 \ 0 \ 0 \ 0] \\
  422 & \Rightarrow [0 \ 0 \ 0.7 \ 0 \ 0.3 \ 0 \ 0 \ 0 \ 0]
  \end{align*}
  \]
Problem: Given a sequence of actions (e.g. 4441444) and a command (e.g. “Jump over the skull while going to the left”), are they related?

- Using the sequence of actions, generate an action-frequency vector:
  \[ \begin{align*}
  \epsilon & \Rightarrow [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\
  4 & \Rightarrow [0, 0, 0, 0, 1, 0, 0, 0, 0, 0] \\
  42 & \Rightarrow [0, 0, 0.5, 0, 0.5, 0, 0, 0, 0, 0] \\
  422 & \Rightarrow [0, 0, 0.7, 0, 0.3, 0, 0, 0, 0, 0]
  \end{align*} \]
- Train a neural network that takes in the action-frequency vector and the command to predict whether they are related or not.
LanguaGE-Action Reward Network (LEARN)

Neural Network Architecture

- Action-frequency vector
- Linear
- Linear
- Encoded action-frequency vector ($D_1$-dimensional)
- Concat
- Linear + Softmax

Language command

Jump over the skull while going to the left.

- InferSent / GloVe+RNN / RNNOnly
- Encoded command ($D_2$-dimensional)
- Linear
- Linear
- (D_3-dimensional)
- Probabilities (RELATED / UNRELATED)
LanguagE-Action Reward Network (LEARN)

Neural Network Architecture

- Action-frequency vector passed through 3 linear layers.
LanguaGE-Action Reward Network (LEARN)

Neural Network Architecture

- Action-frequency vector passed through 3 linear layers.
- Three language encoders:
  - InferSent
  - GloVe+RNN
  - RNNOnly
LanguaGE>Action Reward Network (LEARN)

Neural Network Architecture

- Action-frequency vector passed through 3 linear layers.
- Three language encoders:
  - InferSent
  - GloVe+RNN
  - RNNOnly
- Concatenate encoded action-frequency vector and encoded language.
LanguagE-Action Reward Network (LEARN)

Neural Network Architecture

- Action-frequency vector passed through 3 linear layers.
- Three language encoders:
  - InferSent
  - GloVe+RNN
  - RNNOnly
- Concatenate encoded action-frequency vector and encoded language.
- Pass through linear layers followed by softmax layer.
LanguaGE-Action Reward Network (LEARN)

Data Collection

- Used Amazon Mechanical Turk to collect language descriptions for trajectories.
LanguagE-Action Reward Network (LEARN)

Data Collection

- Used Amazon Mechanical Turk to collect language descriptions for trajectories.
- Minimal postprocessing to remove low quality data.
LanguagE-Action Reward Network (LEARN)

Data Collection

- Used Amazon Mechanical Turk to collect language descriptions for trajectories.
- Minimal postprocessing to remove low quality data.
- Used random pairs to generate negative examples.
Putting it all together...

Jump over the skull while going to the left
Putting it all together...

- Using the agent’s past actions, generate an action-frequency vector.
Putting it all together...

- Using the agent’s past actions, generate an action-frequency vector.

- LEARN: scores the relatedness between the action-frequency vector and the language command.
Putting it all together...

- Using the agent’s past actions, generate an action-frequency vector.
- LEARN: scores the relatedness between the action-frequency vector and the language command.
- Use the relatedness scores as intermediate rewards, such that the optimal policy does not change.
Experiments

- 15 tasks
Experiments

- Amazon Mechanical Turk to collect 3 descriptions for each task.

- JUMP TO TAKE BONUS WALK RIGHT AND LEFT THE CLIMB DOWNWARDS IN LADDER
- Jump Pick Up The Coin And Down To Step The Ladder
- jump up to get the item and go to the right
Experiments

- Different rooms used for training LEARN and RL policy learning.
Experiments

- Different rooms used for training LEARN and RL policy learning.
Results

- Compared RL training using PPO algorithm with and without language-based reward.
Results

- Compared RL training using PPO algorithm with and without language-based reward.

- ExtOnly: Reward of 1 for reaching the goal, reward of 0 in all other cases.
Results

- Compared RL training using PPO algorithm with and without language-based reward.
- ExtOnly: Reward of 1 for reaching the goal, reward of 0 in all other cases.
- Ext+Lang: Extrinsic reward plus language-based intermediate rewards.
Analysis
Analysis

- For a given RL run, we have a fixed natural language description.
Analysis

For a given RL run, we have a fixed natural language description.

At every timestep, we get an action-frequency vector, and the corresponding prediction from LEARN.
For a given RL run, we have a fixed natural language description.

At every timestep, we get an action-frequency vector, and the corresponding prediction from LEARN.

Compute Spearman correlation coefficient between each component (action) and the prediction.
Analysis

go to the left and go under skulls and then down the ladder

go to the left and then go down the ladder

move to the left and go under the skulls
Related Work
Related Work

Language to Reward
[Williams et al. 2017, Arumugam et al. 2017]
Related Work

Language to Reward
[Williams et al. 2017, Arumugam et al. 2017]

Language to Subgoals
[Kaplan et al. 2017]
Related Work

Language to Reward
[Williams et al. 2017, Arumugam et al. 2017]

Language to Subgoals
[Kaplan et al. 2017]

Adversarial Reward Induction
[Bahdanau et al. 2018]
Summary

- Proposed a framework to incorporate natural language to aid RL exploration.
Summary

- Proposed a framework to incorporate natural language to aid RL exploration.

- Two-phase approach:
  1. Supervised training of the LEARN module.
  2. Policy learning using *any* RL algorithm with language-based rewards from LEARN.
Summary

• Proposed a framework to incorporate natural language to aid RL exploration.

• Two-phase approach:
  1. Supervised training of the LEARN module.
  2. Policy learning using *any* RL algorithm with language-based rewards from LEARN.

• Analysis shows that the framework discovers mapping between language and actions.
Summary

- Proposed a framework to incorporate natural language to aid RL exploration.

- Two-phase approach:
  1. Supervised training of the LEARN module.
  2. Policy learning using any RL algorithm with language-based rewards from LEARN.

- Analysis shows that the framework discovers mapping between language and actions.

- Extensions:
  - Temporal information
  - Continuous action space
  - State information
  - Multi-step instructions
Summary

• Proposed a framework to incorporate natural language to aid RL exploration.

• Two-phase approach:
  1. Supervised training of the LEARN module.
  2. Policy learning using any RL algorithm with language-based rewards from LEARN.

• Analysis shows that the framework discovers mapping between language and actions.

• Extensions:
  ○ Temporal information
  ○ Continuous action space
  ○ State information
  ○ Multi-step instructions

Code and Data available at [www.cs.utexas.edu/~pgoyal](http://www.cs.utexas.edu/~pgoyal)