Language grounding for cross-domain policy transfer and spatial reasoning

Karthik Narasimhan
OpenAI
Grounding semantics in control applications

**Input**

*Text:* instructions or knowledge

*Environment:* execute actions and observe

**Output**

*Control Policy:* sequence of actions to interact optimally

1. **Ghosts** chase and try to kill you
2. Collect all the **pellets**
3. ...
Grounding semantics in control applications

1. Use language to improve performance in control applications

- **Score:** 7
- **Score:** 107

1. **Ghosts** chase and try to kill you
2. Collect all the pellets
3. ...
Grounding semantics in control applications

1. Use language to improve performance in control applications

   *Ghosts* chase and try to kill you
   2. Collect all the pellets
   3. ...

   **Score:** 7

   **Score:** 107

2. Use feedback from control application to understand language

   *Reward* +1

   *Alleviate dependence on large scale annotation*
Reinforcement Learning

- Delayed feedback

- Large number of possible action sequences

⇒ How to perform credit assignment for individual actions

⇒ Need for effective exploration

Improved language understanding translates to improved task performance
Deep Transfer in Reinforcement Learning by Language Grounding

Karthik Narasimhan, Regina Barzilay, Tommi Jaakkola
MIT
Deep reinforcement learning for games

**Standard approach:** deep Q-learning by acting in the environment

**Steps to convergence:** ~ a few million
Traditional RL framework

Markov decision process

State \( s \) = Observed Environment
Action \( a \) = Move/Shoot/Use sword

State 1 \quad Action \quad State 2

Policy
\[ \pi : s \rightarrow a \]

Action value function
\[ Q(s, a) \]

Reward
+1
Estimating a policy

Learn from sampled experiences

Episode 1

Episode 2

Episode 3

\[ \pi(a|s) \]
• Learn transitions between states
• Identify good vs bad states
More games

- Each new game requires re-learning from scratch
More games

- Each new game requires re-learning from scratch
- Policy transfer: challenging

(Taylor and Stone, 2009; Parisotto et al., 2015, Rusu et al., 2016; Rajendran et al., 2017, …)
Why is transfer hard?

- Different state spaces and actions
- Need to explore new environment to learn mapping
- Incorrect mapping leads to negative transfer
• How do we re-use learnt information?
• Need some anchor
s1 is similar to u1
s2 is similar to u3
...

Environment 1

Environment 2
Scorpions chase you and kill you on touch.

Spiders are chasers and can be destroyed by an explosion.

- Text descriptions associated with objects/entities
- No mapping between objects in different environments
Transfer through language

- Language as domain-invariant and accessible medium
- *Traditional approaches*: direct policy transfer
- *This work*: transfer ‘model’ of the environment using text descriptions
Model-based reinforcement learning

Transition distribution \( T(s' \mid s, a) \)
Model-based reinforcement learning

Transition distribution $T(s' | s, a)$ and reward function $R(s, a)$
Value Iteration

\[ Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s' | s, a) V(s') \]

\[ V(s) = \max_a Q(s, a) \]

Accurately estimating \( T \) and \( R \) is challenging.
Text-conditioned transition distribution $T(s' | s, a, z)$

State $s$

Scorpions chase you and kill you on touch

Action $a$

Text $z$

$s'_1$

$s'_n$
Bootstrap learning through text

Scorpions chase you and kill you on touch

\[
T_1(s'|s, a, z_1) \quad R_1(s, a, z_1)
\]

Transfer

Spiders are chasers and can be destroyed

\[
\hat{T}_2(u'|u, a, z_2) \quad \hat{R}_2(u, a, z_2)
\]

- Appropriate representation to incorporate language
- Partial text descriptions
Incorporating descriptions

State $S$  \quad Description  \quad RNN  \quad $U_{zi}$  \quad $U_{Oi}$  \quad $\phi(s)$
Differentiable value iteration

\[ Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s'|s, a)V(s') \]
Differentiable value iteration

\[ V(s) = \max_a Q(s, a) \]

Convolutional neural network (CNN)

(Value Iteration Network, Tamar et al., 2016)
**Differentiable value iteration**

\[
Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s'|s, a) V(s')
\]

\[
V(s) = \max_a Q(s, a)
\]

Convolutional neural network (CNN)

\(\phi(s)\)

Reward

\(R\)

\(V\)

\(T\)

\(Q\)

\(V\)

max

k step recurrence

(Value Iteration Network, Tamar et al., 2016)
Parameter Learning

**Objective:** Minimize loss function

\[ \mathcal{L}(\Theta_i) = \mathbb{E}_{s, \hat{a}, \hat{s}' \mid s, a, Z} \left[ (r + \gamma \max_{a'} Q(\hat{s}', a', Z; \Theta_{i-1}) - Q(\hat{s}, \hat{a}, Z; \Theta_i))^2 \right] \]
Model-aware policy

Scorpions chase you

Value estimation

\[ \pi(a|s) \]

Policy
Model-aware transfer

\[ \pi_1(a|s) \]
\[ \pi_2(a|s) \]
\[ \ldots \]
\[ \pi_n(a|s) \]

Policies

Value estimation

Scorpions chase you

Spiders are chasers
Experiments

- 2-D game environments with multiple instances (each with different layouts, different entity sets, etc.)
- Text descriptions from Amazon Mechanical Turk
- **Transfer setup**: train on multiple source tasks, and use learned parameters to initialize for target tasks
- **Baselines**: DQN (Mnih et al., 2015), text-DQN, Actor-Mimic (Parisotto et al., 2016)
- **Evaluation**: Jumpstart, average and asymptotic reward

<table>
<thead>
<tr>
<th>Condition</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>F&amp;E-1 → F&amp;E-2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>F&amp;E-1 → Freeway</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Bomberman → Boulderchase</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Source and target game instances for transfer
Average reward

F&E-1 to Freeway

Reward

No transfer

0.15

0.0

0.2

0.4

0.6

0.8
Average reward

F&E-1 to Freeway

No transfer: 0.15
DQN: 0.06
Actor Mimic: 0.08
Average reward

F&E-1 to Freeway

<table>
<thead>
<tr>
<th>Method</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>No transfer</td>
<td>0.15</td>
</tr>
<tr>
<td>DQN</td>
<td>0.06</td>
</tr>
<tr>
<td>Actor Mimic</td>
<td>0.08</td>
</tr>
<tr>
<td>text-DQN</td>
<td>0.38</td>
</tr>
<tr>
<td>text-VIN</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Transfer results (F&E-1 to Freeway)
Agent: (3,4)

Entity: (2,6)

Seen ‘friendly’ entity

Unseen entity

Unseen entity + ‘friendly’ text

Unseen entity + ‘enemy’ text
Representation Learning for Grounded Spatial Reasoning

Michael Janner, Karthik Narasimhan, Regina Barzilay
MIT
Understanding spatial references

Pick up the red block on top of a green one

Human robot interaction

Autonomous navigation
A spatial reasoning task

- Interactive navigation world
- Goal specified in natural language
- Rewards for reaching goals
- No domain knowledge

Reach the cell above the westernmost rock
A spatial reasoning task

Reach the cell above the westernmost rock
A spatial reasoning task

Interpreting spatial references is highly contextual!

Goal: Generalize effectively over different layouts and spatial references

Reach the cell above the westernmost rock
Types of spatial references

1. Refer to specific entity

“Go to the circle”
Types of spatial references

1. Refer to specific entity

2. Location using a single referent entity

“Reach the cell above the circle”
Types of spatial references

1. Refer to specific entity

2. Location using a single referent entity

3. Location using multiple referent entities

“Move to the goal one square to the right of triangle and two squares to the bottom of star”
Types of spatial references

1. Refer to specific entity
2. Location using a single referent entity
3. Location using multiple referent entities

1. (Local) Go two spaces above the heart.
2. (Global) Reach the northernmost house.
Challenges

- Interpretation of spatial references is highly context-dependent.

- Rich, flexible ways of verbalizing spatial references

- Only source of supervision is reward-based feedback
Markov Decision Process

\[ \langle S, A, X, T, R \rangle \]

\( S \) : State configurations

\( A \) : Actions

\( X \) : Goal specifications in language

\( T \) : Transition distribution

\( R \) : Reward function
Value Iteration

\[ Q(s, a, x) = R(s, x) + \gamma \sum_{s' \in S} T(s'|s, a, x)V(s', x) \]

\[ V(s, x) = \max_a Q(s, a, x) \]

**Goal-conditioned action policy:** \[ \pi(s, x) = \arg \max_a Q(s, a, x) \]
Model requirements

- Joint representation of observations \((s)\) and instructions \((x)\)

- Flexible representation of goals, encoding both local structure and global spatial attributes.

- Must be compositional, to generalize over linguistic variety in instructions.
Universal Value Function Approximators

(Schaul et al., 2015)
Our Model

<object embeddings>
\( \phi(s) \)
Our Model

reach the rock nearest the heart

Global gradient coefficients

Convolutional filter

object embeddings

\phi(s)

\text{LSTM}

h(t)
Our Model

\[ \mathcal{L}(\Theta) = \mathbb{E}_{s \sim \mathcal{D}} \left[ \hat{V}(s, x; \Theta) - \left( R(s, x) + \gamma \max_a \sum_{s'} T(s'|s, a) \hat{V}(s', x; \Theta^-) \right) \right]^2 \]
Experimental setup

- Puddle world with randomized layout and randomly placed unique (6) and non-unique (4) objects
- Text instructions for randomized goals collected from Amazon Mechanical Turk (max length 43)

<table>
<thead>
<tr>
<th>Split</th>
<th>Local</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>1566</td>
<td>1071</td>
</tr>
<tr>
<td>Test</td>
<td>399</td>
<td>272</td>
</tr>
</tbody>
</table>

![Heatmap showing instruction edit distance and map distance]
Baselines

- UVFA (text): UVFA model (Schaul et al., 2015) adapted to use text for goal specifications

- CNN+LSTM: Separately process image and text and then feed concatenated representations to MLP (Misra et al., 2017)
Results: Policy Quality

<table>
<thead>
<tr>
<th>Model</th>
<th>Local</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>UVFA (text)</td>
<td>0.56</td>
<td>0.62</td>
</tr>
<tr>
<td>CNN+LSTM</td>
<td>0.80</td>
<td>0.82</td>
</tr>
<tr>
<td>Our model</td>
<td>0.87</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Results: Distance to goal

Local
- UVFA (text): 4.71
- CNN+LSTM: 5.73
- Our model: 2.18

Global
- UVFA (text): 6.28
- CNN+LSTM: 6.13
- Our model: 3.35
Sample efficiency
Value maps

Reach the northernmost house.
Multiple levels of indirection

reach cell one to the right and two down from horse located to the right of the heart

Redundant information

locate the cell that is filled w/a rock to the left of the teal spade and above the purple house which is above and to the left of the blue circle
Conclusions

- Language provides a compact medium for encoding knowledge for policy transfer.

- Model-aware methods more suitable for leveraging language for transfer.

- Spatial reasoning is highly contextual and a challenging grounding task.

- Joint reasoning over text and environment is crucial for effective generalization over unseen worlds and linguistic variety.