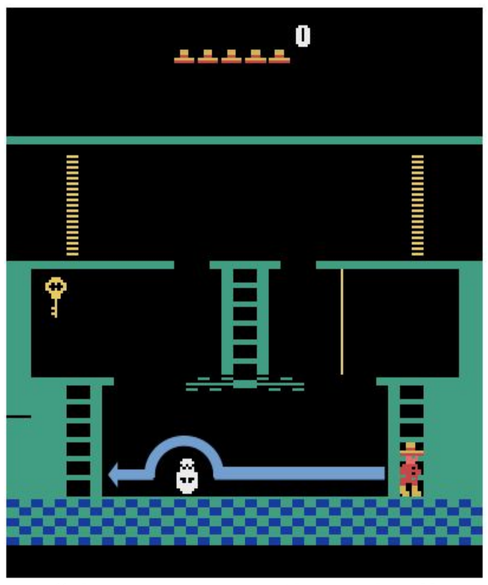
Using Natural Language for Reward Shaping in Reinforcement Learning

Prasoon Goyal, Scott Niekum, Raymond J. Mooney
The University of Texas at Austin

Introduction

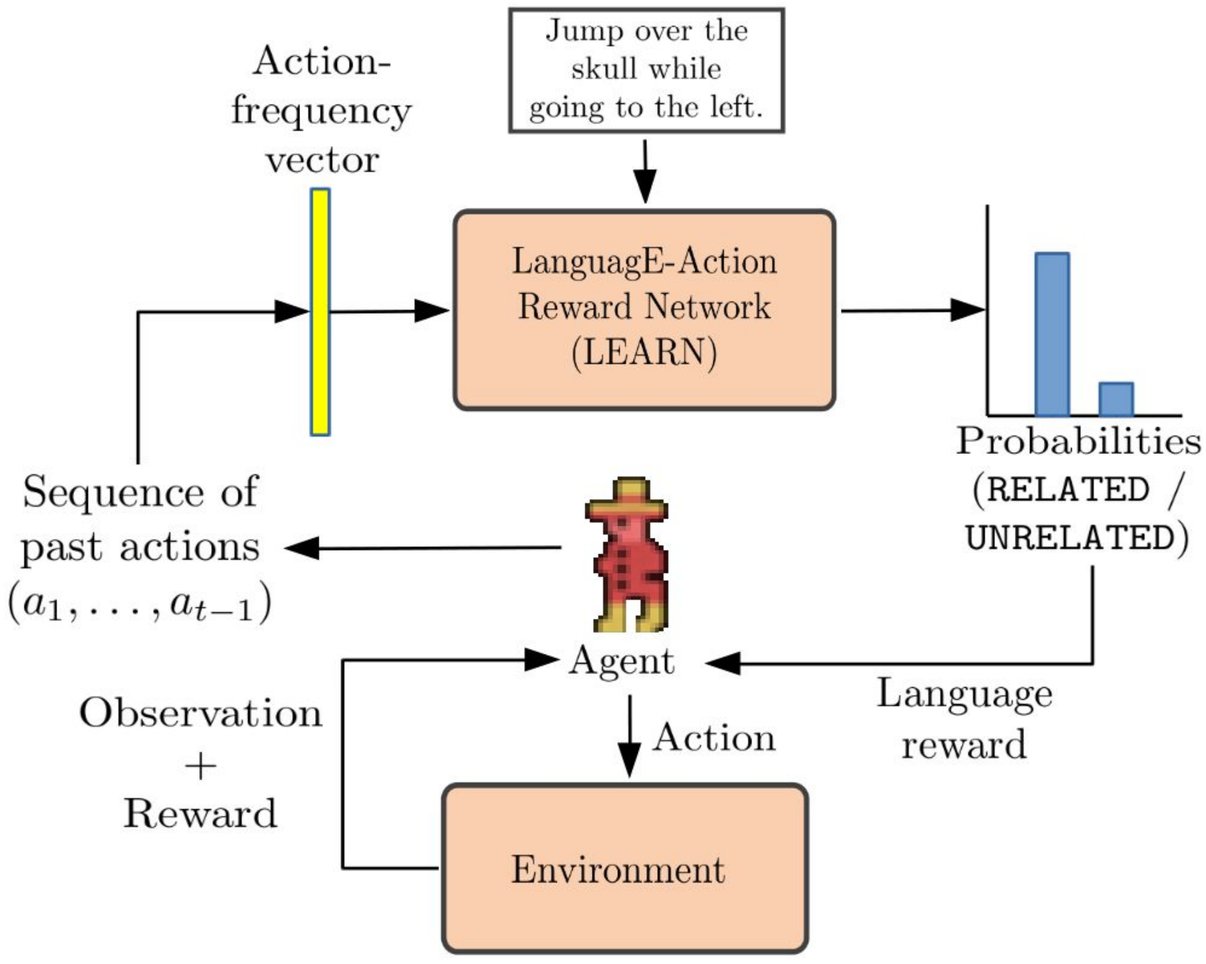
- Most successful applications of reinforcement learning (RL) involve dense environment rewards (e.g. Atari games like Breakout) and/or hand-engineered rewards (e.g. robot manipulation tasks).
- Environments with sparse
- Rewards (e.g. Montezuma's Revenge) require a lot of samples!

➔ **Goal: Use natural language to guide the agent's exploration via reward shaping.**



"Jump over the skull while going to the left."

Approach



- Standard MDP formalism, plus a natural language command describing the task.
- Using the agent's trajectory so far in the current episode, generate an *action-frequency vector* -- vector of dimension |A| with component i equal to the fraction of times action i was performed.
- LEARN: scores the relatedness between the action-frequency vector and the language command.
- Use the relatedness scores as intermediate rewards
  ⇒ Can be plugged into any standard RL algorithm.

LEARN module:

- Trained offline using supervised learning, on paired (trajectory, language) data collected using Amazon Mechanical Turk.
- Task-agnostic.

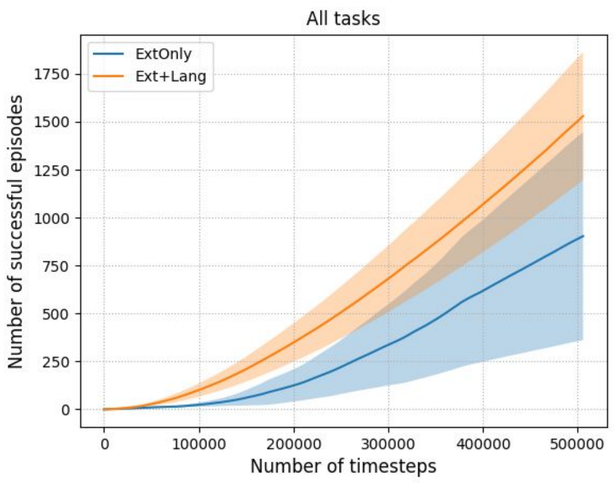
For example: If the command is "Jump over the skull while going to the left", the trained LEARN module should assign high relatedness score to trajectories with actions "jump" and "left". Therefore, using the relatedness scores as rewards encourages taking those actions more often.

Data Collection

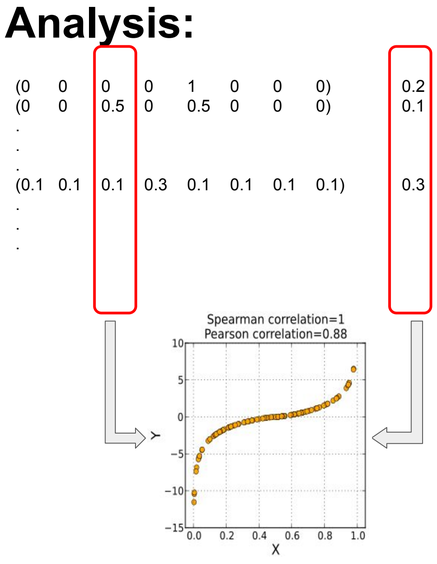
- Used 20 trajectories of human gameplay from the Atari Grand Challenge dataset.
- Amazon Mechanical Turk for collecting annotations: workers were shown short clips from the game and were asked to provide natural language descriptions.
- Minimal filtering to eliminate low quality descriptions.
- 6,780 descriptions after filtering.
- Example descriptions:
  1. wait ⇒ Uninformative
  2. using the ladder on standing ⇒ Ill-formed
  3. going slow and climb down the ladder
  4. move down the ladder and walk left
  5. go left watch the trap and move on
  6. climbling down the ladder ⇒ Spelling error
  7. ladder dwon and running this away ⇒ Spelling error
  8. stay in place on the ladder
  9. go down the ladder
  10. go right and climb up the ladder

Experiments

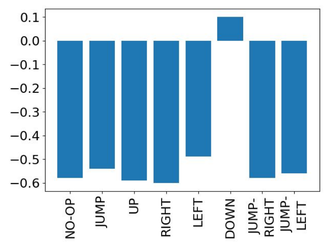
- 15 tasks : 3 descriptions per task collected using Mechanical Turk.
- Baseline: Only extrinsic reward (1 for reaching the goal, 0 otherwise).
- Using language-based rewards gives 60% relative improvement.



Analysis:



"Move on spider and down on lader"



"Go to the left and then go down the ladder"