Action embedding

Here's how I envision we ground language and vision in robotics. Figure 4 summarizes the complete pipeline, but first we start with intuitive motivations below.

1 Natural embedding of robot actions

The state of a robot can be entirely captured by a vector. For example, the Baxter robot has two arms, each with 7 degrees of freedom, as shown in Figure 1. The state of the Baxter torso is simply a 14 dimensional vector.

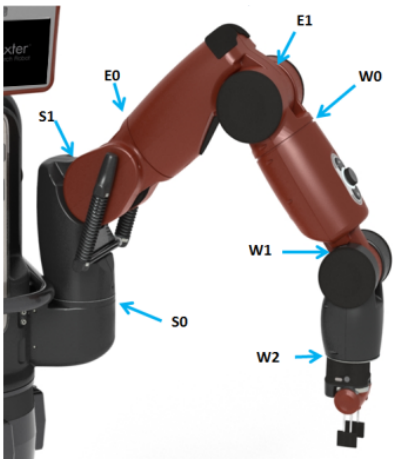

Figure 1: Each Baxter arm has 7 degrees of freedom.

This 14 dimensional space becomes a natural embedding of robot states. Points close together in this space represent similar states of a robot. The external pressure of human values may drive the robot to perform actions, as shown in Figure 2.

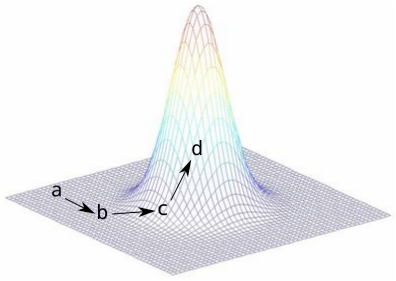

Figure 2: Baxter drifts in this 14 dimensional space.

2 Hierarchical organization of actions

Robot actions are organized using the temporal And-Or graph (Figure 3), because the actions a robot performs are compositional (`AND` nodes) and variational (`OR` nodes). Every node of the graph is thought of as a fluent-change.

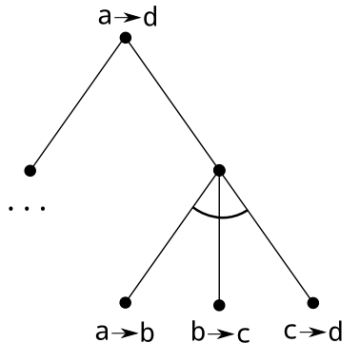

Figure 3: The temporal And-Or Graph is a sparse description of how to perform an action.

3 Grounding words to fluent-change

To perform a command communicated through natural language, the robot needs to infer corresponding fluent-changes. We propose an approach that jointly learns associations between vision, language, and action-sequences. Some examples of pair-wise associations already in literature are the following:

- $Vision \leftrightarrow Language$
  Baroni et al. studied how to associate images with words. They learned a way to jointly embed images and natural language together. Given any image, they can guess the corresponding label by zero-shot inference. Conversely, given any word, they could roughly generate a corresponding image.
- $Language \leftrightarrow Action$
  While some work exists under this category, generating action-sequences from language (or vice versa) is still a difficult problem. But, that's why it's an exciting opportunity for us to make an impact here.
- $Action \leftrightarrow Vision$
  The Atari deep reinforcement learning approach is a successful and recent example of associating vision with action-sequences. However, there's little to no language or reasoning supported.

In the proposed approach, I suggest we jointly embed visual, linguistic, and action-sequences into the same vector space. Each concept is a three-tuple $\langle V, L, A\rangle$ of visual (V), linguistic (L), and action-sequence (A) embeddings. The leaf-nodes of the AOG are these $\langle V, L, A\rangle$ concepts. That way, there's a label (L) associated with each action (A), making it easier to ground dialogue.

Figure 4 shows the overview pipeline for the project. There will be two core algorithms working in parallel: the sensory engine and the reasoning engine.

1. The sensory engine maps $\langle V, L, A \rangle$ concepts to a vector space so images, natural language, and action-sequences that describe the same concept map to the same point.
2. The reasoning engine grounds logical deduction to this $\langle V, L, A \rangle$ embedding. It manages the AOG and generates language for question-answering (QA).

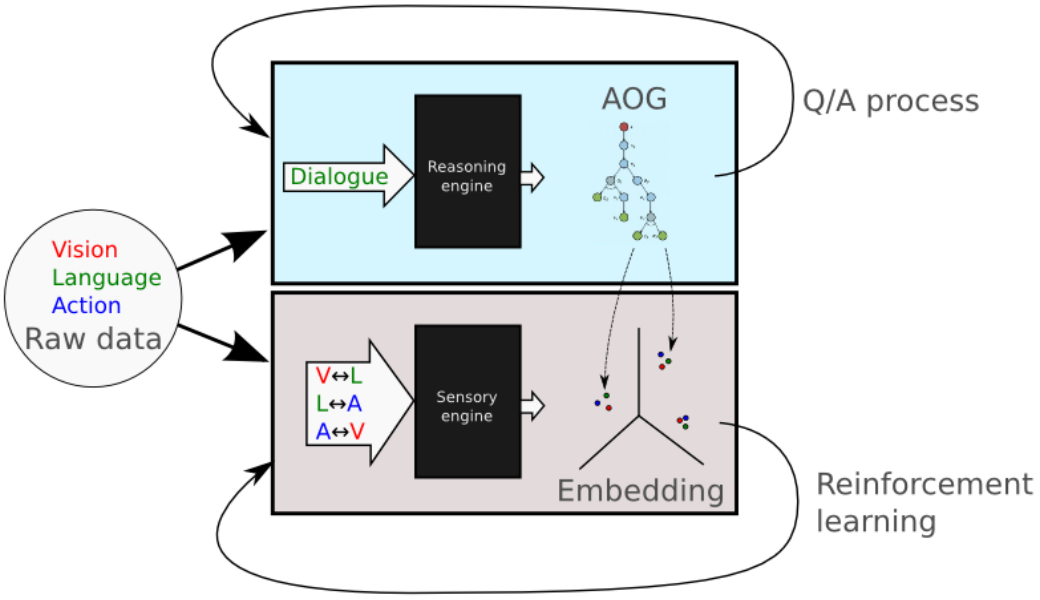

Figure 4: The reasoning engine manages the AOG while the sensory engine grounds visual, linguistic, and temporal concepts.

4 Evaluation

The core insight is that jointly embedding visual (V), ligustic (L), and action (A) information allows high-level reasoning (AOG) to be grounded. Here are possible experiments to demonstrate this framework's strengths. Positive results in any of the following might be of publishable quality.

1. Ask robot to grip novel objects (to demonstrate zero-shot inference)
2. Ask robot to perform multi-step tasks (to demonstrate AOG grounding, zero-shot inference)

   Make tea, organize desk, cook food, assemble small furniture, etc.
3. Teach robot new task from natural language (to demonstrate AOG grounding, zero-shot learning)
4. Let robot ask questions when uncertain about object (to demonstrate life-long learning)
5. Let robot ask questions when uncertain about task (to demonstrate life-long learning)
6. Ask robot to explain what it is doing (to demonstrate language generation and embedding)