Problem Setting

We investigate how to train Imitation Learning agents in instruction following environments with ambiguous instructions. Our experiments are with the BabyAI Platform, in which an agent must perform a specific task in a gridworld.

Classifier Network Architecture

Fine-tuned GPT-2: The classifier is a fine-tuned GPT-2 model from the HuggingFace library. The logits of the predicted distribution over next tokens is concatenated with a flattened state representation, and fed into fully connected layers to produce the classifier output.

Figure 1: Example BabyAI task. The agent (red triangle) must follow the given instruction (“go to a yellow box”).

LSTM: The LSTM is an RNN variant that can take into account long term dependencies in sequential data. We convert word vectors into embeddings, and feed them through LSTM layer and linear layer to output whether the instruction is ambiguous.

IL Agent Network Architecture

The instruction is fed through a GRU, and the state a CNN. The outputs are combined using a fusion mechanism (feature-wise linear modulations) layers, which is then fed into an LSTM that outputs actions and state values.

Ambiguous Instructions

To test our method, we designed a method to automatically make BabyAI instructions ambiguous. BabyAI instructions have an internal tree-based representation:

```
Instruction (2)
├── Go to a blue door, then turn left
│   ├── Go to a blue door (1)
│       ├── Try to open the door
│       │   ├── Open the door
│       │       ├── Pick up the key
│       │       │   └── Insert the key
│       └── Go to the blue door
```

We make instructions ambiguous by recursively making the subtrees of each node ambiguous. When we arrive at the leaves (which represent objects), we randomly drop certain descriptors of the object, such as its color, type, or location.

Classifier Experiments and Results

Dataset:
- 4000 instructions-state pairs (2000 in training, 800 in validation, and 800 in test)
- Randomly selected to be turned ambiguous, labelled by assessing whether the instruction is ambiguous with respect to the environment.
- Concatenate state and tokenized instruction, convert the vector to word index vectors

Performance & Discussion:
- Similar accuracy
- GPT-2 converges faster (pretrained)
- GPT-2 performance not significantly better than a concatenated state and instruction as input, and GPT-2 has no experience in comprehending the state, which is necessary in judging whether the instruction is ambiguous.

RL Results

In addition to the classifier and IL experiments, we trained RL agents on the BabyAI environments. RL agents learned a policy of interacting with as many objects as possible to complete the task. Adding a penalty was insufficient to discourage this behavior.

Evaluation of the effect of the ambiguity classifier:
- If classifier detects ambiguity, clarification is requested, and the ambiguous instruction is replaced with the unambiguous instruction.
- Effect of ambiguity classifier evaluated on three environments with 0.5 ambiguity rate.
- Each agent evaluated for 3,000 episodes.

Results and Discussion:
- Using the ambiguity classifier results in slight increases in success rates.
- Marginal benefits in easy environments.
- Larger benefits in difficult environments.

Imitation Learning Results

Imitation Learning agent is trained on a dataset collected by the BabyAI expert B.O.T.

Table 2: Evaluation Success Rates

<table>
<thead>
<tr>
<th>Method</th>
<th>Goldilocal</th>
<th>Picklocal</th>
<th>Picklocal</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Ambiguous (baseline)</td>
<td>88.0%</td>
<td>51.4%</td>
<td>42.1%</td>
<td>60.5%</td>
</tr>
<tr>
<td>No Classifier</td>
<td>88.7%</td>
<td>53.2%</td>
<td>51.6%</td>
<td>64.8%</td>
</tr>
<tr>
<td>LSTM Classifier</td>
<td>90.0%</td>
<td>55.0%</td>
<td>51.6%</td>
<td>65.5%</td>
</tr>
<tr>
<td>GPT-2 Classifier</td>
<td>90.0%</td>
<td>55.0%</td>
<td>51.6%</td>
<td>65.5%</td>
</tr>
</tbody>
</table>

Selected References