TOWARDS TEACHING MACHINES WITH LANGUAGE: INTERACTIVE LEARNING FROM ONLY LANGUAGE DESCRIPTIONS OF ACTIVITIES

Khanh Nguyen, Dipendra Misra, Robert Schapire, Miro Dudík, Patrick Shafto

ABSTRACT

We present a novel interactive learning protocol that enables training request-fulfilling agents by only verbally describing their activities. Our protocol gives rise to a new family of interactive learning algorithms that offer complementary advantages against traditional algorithms like imitation learning (IL) and reinforcement learning (RL). We develop an algorithm that practically implements this protocol and employ it to train agents in two challenging request-fulfilling problems using purely language-description feedback. Empirical results demonstrate the strengths of our algorithm: compared to RL baselines, it is more sample-efficient; compared to IL baselines, it achieves competitive success rates while not requiring feedback providers to have agent-specific expertise. We also provide theoretical guarantees of the algorithm under certain assumptions on the teacher and the environment.
Algorithm 1 ILIAD protocol. Details of line 4 and line 6 are left to specific implementations.

1: Initialize agent policy \( \pi_\theta : S \times D \rightarrow \Delta(A) \)
2: for \( n = 1, 2, \ldots, N \) do
3: World samples \( q = (R, d^*, s_1) \sim \mathbb{P}_* (\cdot) \) (reward function \( R \) is not revealed to agent)
4: Agent generates execution \( \hat{e} \) given \( \pi_\theta, d^* \), and \( s_1 \)
5: Teacher generates description \( \hat{d} \sim \mathbb{P}_T (\cdot | \hat{e}) \)
6: Agent uses \( (d^*, \hat{e}, \hat{d}) \) to update \( \pi_\theta \)
return \( \pi_\theta \)

Algorithm 2 ADEL: our implementation of the ILIAD protocol. \( \mathbb{P}_T (d | e) \) is the teacher model.

1: Input: approximate marginal \( \mathbb{P}_{\pi_e} (e | s_1) \), mixing rate \( \lambda \in [0, 1] \), annealing rate \( \beta \in (0, 1) \)
2: Initialize \( \pi_\theta : S \times D \rightarrow \Delta(A) \) and \( B = \emptyset \)
3: for \( n = 1, 2, \ldots, N \) do
4: World samples task \( q = (R, d^*, s_1) \sim \mathbb{P}_* (\cdot) \)
5: Agent generates \( \hat{e} \sim \lambda \mathbb{P}_{\pi_e} (\cdot | s_1) + (1 - \lambda) \mathbb{P}_{\pi_\theta} (\cdot | s_1, d^*) \)
6: Teacher generates description \( \hat{d} \sim \mathbb{P}_T (\cdot | \hat{e}) \). Add datapoint: \( B \leftarrow B \cup (\hat{e}, \hat{d}) \)
7: Update agent policy: \( \theta \leftarrow \max_{\theta'} \sum_{(e, \hat{d}) \in B} \sum_{(s, a_s) \in \hat{e}} \log \pi_{\pi_e} (a_s | s, \hat{d}) \) where \( a_s \) is the action taken by the agent in state \( s \). Anneal the mixing rate: \( \lambda \leftarrow \lambda \cdot \beta \)
return \( \pi_\theta \)

with a language description of the agent’s execution \( \hat{d} \sim \mathbb{P}_T (\cdot | \hat{e}) \). We assume that the descriptions are specified in the same language that is used to specify the requests. Hence, by grounding the descriptions to the corresponding executions, the agent can acquire knowledge about the description language and thus can gradually improve its request-fulfilling capability. Crucially, the agent receives no other feedback such as ground-truth demonstration (Mei et al. 2016), scalar reward (Hermann et al. 2017), or constraint (Miryoosefi et al. 2019). At test time, the teacher is not present and the agent must execute requests autonomously. The objective of the agent is to find a policy \( \pi \) with maximum value, where we define the policy value \( V(\pi) \) as:

\[
V(\pi) = \mathbb{E}_{q \sim \mathbb{P}_*(\cdot), \hat{e} \sim \mathbb{P}_e (\cdot | s_1, d^*)} \left[ \sum_{i=1}^H R(s_i, \hat{a}_i) \right]
\]

To formulate the learning problem in ILIAD, we define a joint distribution over tasks and executions:

\[
\mathbb{P}^*(e, R, s_1, d) = \mathbb{P}^*(e | s_1, d) \mathbb{P}_e (R, d, s_1)
\]

where \( \pi^* \) be the optimal policy that maximizes \( V(\pi) \). We then implement the ILIAD protocol by reducing it to a density-estimation problem: given that we can effectively draw samples from the marginal \( \mathbb{P}^*(s_1, d) \) and an approximately consistent teacher \( \mathbb{P}_T (d | e) \approx \mathbb{P}^*(d | e) \), how do we learn a policy \( \pi_\theta \) such that \( \mathbb{P}_{\pi_\theta} (e | s_1, d) \) is close to \( \mathbb{P}^*(e | s_1, d) \)? Here, the marginal \( \mathbb{P}^*(s_1, d) \), the consistent teacher \( \mathbb{P}^*(d | e) \), and the (ground-truth) execution distribution \( \mathbb{P}^*(e | s_1, d) = \mathbb{P}_{\pi^*} (e | s_1, d) \) are obtained from the joint distribution defined in Equation 2.

We develop an algorithm named ADEL: Activity-Description Explorative Learner (Algorithm 2) that offers practical solutions to this problem. ADEL implements a semi-supervised sampling scheme that efficiently explores the execution space. Specifically, in the algorithm, the agent generates executions from a mixture distribution that combines a request-agnostic execution distribution \( \mathbb{P}_{\pi_e} (e | s_1) \), which can be learned from unlabeled executions, and a request-guided execution distribution \( \mathbb{P}_{\pi_\theta} (\cdot | s_1, d^*) \) (line 5). The agent then employs behavior cloning (Pomerleau 1991) to ground descriptions to executions (line 7). We theoretically prove convergence for a variant of ADEL in the contextual bandit setting (Langford & Zhang 2008).

Our paper does not argue for the primacy of ILIAD over other protocols like RL or IL. In fact, an important point we raise is that there are multiple, possibly competing metrics for comparing learning protocols. We focus on the trade-off between the learning effort of the agent and the teacher in each protocol (Table 1). In all protocols, the agent and the teacher establish a communication channel that
Table 1: Trade-offs between the learning effort of the agent and the teacher in learning protocols. Each protocol employs a different medium for the teacher to convey feedback. If a medium is not natural to the teacher (e.g., IL-style demonstration), it must learn to encode feedback intent using that medium \textit{(teacher communication-learning effort)}. Similarly, if a medium is not natural to the agent (e.g., human language), it needs to learn to interpret feedback \textit{(agent communication-learning effort)}. The agent also learns tasks from information decoded from feedback \textit{(agent task-learning effort)}. The qualitative claims on the “agent learning effort” column summarize our empirical findings on the learning efficiency (measured by sample complexity) of these protocols.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Feedback medium</th>
<th>Teacher (communication learning)</th>
<th>Agent (communication &amp; task learning)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IL</td>
<td>Demonstration</td>
<td>Highest</td>
<td>Lowest</td>
</tr>
<tr>
<td>RL</td>
<td>Scalar reward</td>
<td>None</td>
<td>Highest</td>
</tr>
<tr>
<td>ILLiAD</td>
<td>Language description</td>
<td>None</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 2: Main results. We report means and standard deviations of success rates (%) over five runs with different random seeds. RL-Binary and RL-Cont refer to the RL settings with binary and continuous rewards, respectively. Sample complexity is the number of training episodes (or number of teacher responses) required to reach a validation success rate of at least $c$.

<table>
<thead>
<tr>
<th>Learning setting</th>
<th>Algorithm</th>
<th>Val success rate (%) $\uparrow$</th>
<th>Test success rate (%) $\uparrow$</th>
<th># Demonstrations</th>
<th># Rewards</th>
<th># Descriptions</th>
<th>Sample complexity $\downarrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision-language navigation</td>
<td>IL DAgger</td>
<td>35.6 ± 1.35</td>
<td>32.0 ± 1.63</td>
<td>45K ± 26K</td>
<td>-</td>
<td>-</td>
<td>(c = 30.0%)</td>
</tr>
<tr>
<td>RL-Binary</td>
<td>REINFORCE</td>
<td>22.4 ± 1.15</td>
<td>20.5 ± 0.58</td>
<td>-</td>
<td>+∞</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RL-Cont</td>
<td>REINFORCE</td>
<td>11.1 ± 2.19</td>
<td>11.3 ± 1.25</td>
<td>-</td>
<td>+∞</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ILLiAD</td>
<td>ADEL</td>
<td>32.2 ± 0.97</td>
<td>31.9 ± 0.76</td>
<td>-</td>
<td>-</td>
<td>406K ± 31K</td>
<td></td>
</tr>
<tr>
<td>Word modification</td>
<td>IL DAgger</td>
<td>92.5 ± 0.53</td>
<td>93.0 ± 0.37</td>
<td>118K ± 16K</td>
<td>-</td>
<td>-</td>
<td>(c = 85.0%)</td>
</tr>
<tr>
<td>RL-Binary</td>
<td>REINFORCE</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>-</td>
<td>+∞</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RL-Cont</td>
<td>REINFORCE</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>-</td>
<td>+∞</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ILLiAD</td>
<td>ADEL</td>
<td>88.1 ± 1.60</td>
<td>89.0 ± 1.30</td>
<td>-</td>
<td>-</td>
<td>573K ± 116K</td>
<td></td>
</tr>
</tbody>
</table>

allows the teacher to encode feedback and send it to the agent, who learns tasks based on information decoded from feedback. At one extreme, IL places the burden of establishing the communication channel entirely on the teacher. To provide a demonstration, the teacher in IL must learn to control the agent to accomplish tasks by specifying actions that lie in the agent’s action space. To compensate for this effort, the agent usually learns very efficiently with IL because it does not have to learn to interpret feedback, and the feedback directly specifies desired behavior. At another extreme, we have RL and ILLiAD, where the teacher provides feedback via agent-agnostic media (reward and language, respectively). RL eliminates the agent communication-learning effort by hard-coding the semantics of scalar rewards into the learning algorithm. But the trade-off of using such limited feedback is that the effort required by the agent to learn the task increases. State-of-the-art RL algorithms are notorious for their high sample complexity, making them expensive to use outside simulators. ILLiAD offers a compromise between RL and IL: it can be more sample-efficient than RL while not requiring the teacher to master the agent’s control interface. Overall, no protocol is superior in all metrics and the choice of protocol depends on users’ preferences.

We empirically evaluate ADEL against IL and RL baselines on two tasks: vision-language navigation (Anderson et al., 2018), and word-modification via regular expressions (Andreas et al., 2018).

\[^{1}\text{Third-person or observational IL (Stadie et al., 2017; Sun et al., 2019) allows the teacher to demonstrate tasks with their action space. However, this framework is non-interactive because the agent imitates pre-collected demonstrations and does not interact with a teacher. We consider interactive IL (Ross et al., 2011), which is shown to be more effective than non-interactive counterparts.}\]

\[^{2}\text{By design, RL algorithms understand that higher reward value implies better performance.}\]
Figure 1: Illustrations of the two request-fulfilling problems that we conduct experiments on.

(a) Vision-language navigation (NAV): a (robot) agent fulfills a navigational natural-language request in a photo-realistic simulated house. Locations in the house are connected as a graph. In each time step, the agent receives a photo of the panoramic view at its current location (due to space limit, here we only show part of a view). Given the view and the language request, the agent chooses an adjacent location to go to.

(b) Word modification via regular expressions (REGEX): an agent is given an input word and a natural-language request that asks it to modify the word. The agent outputs a regular expression that follows our specific syntax. The regular expression is executed by the Python’s `re.sub()` method to generate an output word.

Figure 2 illustrates an example of training an agent to fulfill a navigation request using ADEL. Our results (Table 2) show that ADEL significantly outperforms RL baselines in terms of both learning efficiency and effectiveness. On the other hand, ADEL’s success rate is competitive with those of the IL baselines on the navigation task and is lower by 4% on the word modification task. It takes approximately 5-8 times more training episodes than the IL baselines to reach comparable success rates, which is quite respectable considering that the algorithm has to search in an exponentially large space for the ground-truth executions whereas the IL baselines are given these executions. Therefore, ADEL can be a preferred algorithm whenever annotating ground-truth executions is not feasible or is substantially more expensive than describing executions. For example, in the word-modification task, ADEL teaches the agent without requiring a teacher with knowledge about regular expressions, who can be costly to recruit in practice. We believe the capability of non-experts to provide feedback will make ADEL and more generally the LIAD protocol a strong contender in many scenarios.
REFERENCES


David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. In Association for the Advancement of Artificial Intelligence, 2011.


Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Listen, attend, and walk: Neural mapping of instructions to action sequences. In Association for the Advancement of Artificial Intelligence (AAAI), 2016.


