Learning Rewards from Linguistic Feedback

Theodore R. Sumers,[1] **Mark K. Ho,**[2] **Robert D. Hawkins,**[2]
Karthik Narasimhan,[1] **Thomas L. Griffiths**[1,2]
[1]Department of Computer Science, Princeton University, Princeton, NJ
[2]Department of Psychology, Princeton University, Princeton, NJ
{sumers, mho, rdhawkins, karthikn, tomg}@princeton.edu

Abstract

We explore unconstrained natural language feedback as a learning signal for artificial agents. Humans use rich and varied language to teach, yet most prior work on interactive learning from language assumes a particular form of input (e.g., commands). We propose a general framework which does not make this assumption, using *aspect-based sentiment analysis* to decompose feedback into sentiment about the features of a Markov decision process. We then perform an analogue of inverse reinforcement learning, regressing the sentiment on the features to infer the teacher's latent reward function. To evaluate our approach, we first collect a corpus of teaching behavior in a cooperative task where both teacher and learner are human. We implement three artificial learners: sentiment-based "literal" and "pragmatic" models, and an inference network trained end-to-end to predict latent rewards. We then repeat our initial experiment and pair them with human teachers. All three successfully learn from interactive human feedback. The sentiment models outperform the inference network, with the "pragmatic" model approaching human performance. Our work thus provides insight into the information structure of naturalistic linguistic feedback as well as methods to leverage it for reinforcement learning.

1 Introduction

For autonomous agents to be widely usable, they must be responsive to human users' natural modes of communication. For instance, imagine designing a household cleaning robot. Some behaviors can be pre-programmed (e.g. how to use an outlet to recharge itself), while others must be learned (e.g. if a user wants it to charge in the living room or the kitchen). But how should the robot infer what a person wants?

Here, we focus on *unconstrained linguistic feedback* as a learning signal for autonomous agents. Humans use natural language flexibly and intuitively to express their desires via commands, counterfactuals, encouragement, explicit preferences, or other forms of feedback. For example, if a human encounters the robot charging in the living room as desired, they may provide feedback such as "Great job." If they find it charging in the kitchen, the human may respond with "You should have gone to the living room" or "I don't like seeing you in the kitchen." Our approach of learning rewards from such open-ended language differs from previous methods for interactive learning that use non-linguistic demonstrations (Abbeel and Ng 2004; Argall et al. 2009; Ho et al. 2016), rewards/punishments (Knox and Stone 2009; MacGlashan et al. 2017; Christiano et al. 2017), or language commands (Tellex et al. 2011; Wang, Liang, and Manning 2016; Tellex et al. 2020). The agent's learning challenge is to interpret naturalistic feedback in the context of its behavior and environment to infer the teacher's preferences.

We formalize this inference as linear regression over features of a Markov decision process (MDP). We first decompose linguistic feedback into a scalar sentiment and a target subset of the MDP's features (a form of aspect-based sentiment analysis (Hu and Liu 2004; Liu 2020)). We then regress the sentiment against the features to infer the teacher's latent reward function. This enables learning rewards from arbitrary language. To extract target features, we draw on educational research (Shute 2008) to first map utterances to elements of the MDP. For example, "Good job" maps to prior behavior, whereas "You should have gone to the living room" maps to an action. We then extract relevant MDP features: intuitively, positive sentiment about an action implies positive rewards on its features. We implement two versions of this model: a "literal" learner using only the explicit sentiment and a "pragmatic" learner with additional inductive biases (Grice 1975). These models allow us to study how humans express themselves; however, they rely on domain-specific lexical groundings. We thus develop a parallel approach: training an inference network end-to-end to predict latent rewards from human interactions. In our live evaluation, all three models learn from human feedback. Our sentiment-based models achieve statistically-significant improvements over the inference network; the "pragmatic" model approaches human performance. We outline related work in Section 2, then introduce our sentiment model in Section 3. Section 4 describes our task and experiment and Section 5 details our model implementations. Finally, Section 6 discusses results and Section 7 concludes.[1]

2 Background and Related Work

The work presented here complements existing methods that enable artificial agents to learn from and interact with hu-

[1]Code and data: `github.com/tsumers/rewards`.

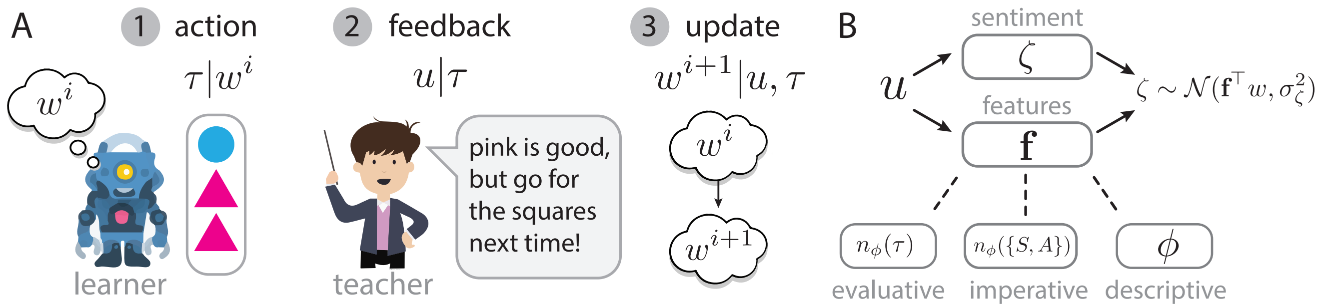

Figure 1: **A**: Episodes involve three stages. **B**: We use aspect-based sentiment analysis to factor utterances into *sentiment* and *features*, then infer latent weights w (solid lines). This allows us to integrate multiple forms of feedback (dashed lines).

mans. For example, a large literature studies how agents can learn latent preferences from non-linguistic human feedback. Algorithms such as TAMER (Knox and Stone 2009) and COACH (MacGlashan et al. 2017) transform human-generated rewards and punishments into quantities that reinforcement learning (RL) algorithms can reason with. Preference elicitation, which provides a user with binary choices between trajectories, is a similarly intuitive training method (Christiano et al. 2017). Finally, demonstration-based approaches use a set of expert trajectories to learn a policy—as in imitation learning (Ross and Bagnell 2010)—or infer an underlying reward function—as in inverse reinforcement learning (IRL) (Abbeel and Ng 2004). This idea has been extended to settings in which agents are provided with intentionally informative demonstrations (Ho et al. 2016) or themselves act informatively (Dragan, Lee, and Srinivasa 2013), such as the more general "cooperative IRL" setting (Hadfield-Menell et al. 2016).

Another body of research has focused on linguistic human-agent interaction. Dialogue systems (Artzi and Zettlemoyer 2011; Li et al. 2016) learn to interpret user queries in the context of the ongoing interaction, while robots and assistants (Thomason et al. 2015; Wang et al. 2019; Thomason et al. 2020; Szlam et al. 2019) ground language in their physical surroundings (for a review of language and robotics, see Tellex et al. (2020)). A parallel line of work in machine learning uses language to improve sample efficiency: to shape rewards (Maclin and Shavlik 1994; Kuhlmann et al. 2004), often via subgoals (Kaplan, Sauer, and Sosa 2017; Williams et al. 2018; Chevalier-Boisvert et al. 2018; Goyal, Niekum, and Mooney 2019; Bahdanau et al. 2019; Zhou and Small 2020). These approaches generally interpret and execute independent declarative statements (e.g., queries, commands, or (sub)goals). Perhaps the most similar works to ours are (MacGlashan et al. 2015; Fu et al. 2019; Goyal, Niekum, and Mooney 2020), which perform IRL on linguistic input in the form of natural language commands. Our work differs in two key ways: first, we use unconstrained and unfiltered natural language; second, we seek to learn general latent preferences rather than infer command-contextual rewards. A somewhat smaller body of work investigates such open-ended language: to correct captioning models (Ling and Fidler 2017), capture environmental characteristics (Narasimhan, Barzilay, and Jaakkola 2018), or improve hindsight replay (Cideron et al. 2019). For a review of language and RL, see Luketina et al. (2019).

We aim to recover the speaker's preferences from naturalistic interactions. Thus, unlike prior approaches, we do not *solicit* a specific form of language (i.e. commands, corrections, or descriptions). We instead *elicit* naturalistic human teaching and develop inferential machinery to learn from it. This follows studies of emergent language in other domains including action coordination (Djalali et al. 2011; Djalali, Lauer, and Potts 2012; Potts 2012; Ilinykh, Zarrieß, and Schlangen 2019; Suhr et al. 2019), reference pragmatics (He et al. 2017; Udagawa and Aizawa 2019), navigation (Thomason et al. 2019), and "Wizard of Oz" experiments (Kim et al. 2009; Allison, Luger, and Hofmann 2018).

3 Learning Rewards from Language

In this section, we formalize our approach. We develop a form of aspect-based sentiment analysis (Hu and Liu 2004; Liu 2020) to decompose utterances into sentiment and MDP features, then use linear regression to infer the teacher's rewards over those features. This allows us to perform an analogy of IRL (Abbeel and Ng 2004; Jeon, Milli, and Dragan 2020) on arbitrary language. To extract MDP features, we ground utterances into the learner's context (Harnad 1990; Mooney 2008) with mappings inspired by educational research (Shute 2008; Lipnevich and Smith 2009).

3.1 Setup

We begin by defining a *learner* agent whose interactions with the environment are defined by a Markov Decision Process (MDPs) (Puterman 1994). Formally, a finite-horizon MDP $\mathcal{M} = \langle S, A, H, T, R \rangle$ is a set of states S, a set of actions A, a horizon $H \in \mathbb{N}$, a probabilistic transition function $T : S \times A \to \Delta(S)$, and a reward function $R : S \times A \to \mathbb{R}$. Given an MDP, a policy is a mapping from states to actions, $\pi : S \to A$. An *optimal policy*, π^*, is one that maximizes the future expected reward (value) from a state, $V^h(s) = \max_a R(s,a) + \sum_{s'} T(s' \mid s,a) V^{h-1}(s')$, where $V^0(s) = \max_a R(s,a)$. States and actions are characterized by features ϕ, where $\phi : S \times A \to \{0,1\}^K$ is an indicator function representing whether a feature is present for a particular action a in state s. We denote a state-action trajectory by $\tau = \langle s_0, a_0, ... s_T, a_T \rangle$. Finally, we define the feature counts over a set of state-action tuples as n_ϕ:

$$n_\phi(\{\langle s,a\rangle\}) = \sum_{s,a \in \{\langle s,a\rangle\}} \phi(s,a) \tag{1}$$

which we use to summarize arbitrary sets of state-action tuples, including trajectories.

3.2 Interactive Learning from Language

We consider a setting where the reward function is hidden from the *learner* agent but known to a *teacher* agent who is allowed to send natural-language messages u (Fig. 1A). We formulate the online learning task as Bayesian inference over possible rewards: conditioning on the teacher's language and recursively updating a belief state. Formally, we assume that the teacher's reward function is parameterized by a latent variable $w \in \mathbb{R}^K$ representing the rewards associated with features ϕ:

$$R(s,a) = w^\top \phi(s,a). \tag{2}$$

We refer to these weights as the teacher's *preferences* over features. The learner is attempting to recover the teacher's preferences from their utterances, calculating $P(w|u)$.

Learning unfolds over a series of interactive *episodes*. At the start of episode i, the learner has a belief distribution over the teacher's reward weights, $P(w^i)$, which it uses to identify its policy. First, the learner has an opportunity to *act* in the world given this policy, sampling a trajectory τ^i. Second, they receive *feedback* in the form of a natural language utterance u^i from the teacher (and optionally a reward signal from the environment). Third, the learner uses the feedback to *update* its beliefs about the reward, $P(w^{i+1}|u^i, \tau^i)$, which is then used for the next episode.

We now describe our general formal approach for inferring latent rewards from feedback. We first assume the learner extracts the *sentiment* ζ and *target features* $\mathbf{f}$ from the teacher's utterance, where $\mathbf{f} \in \mathbb{R}^K$ is a vector describing which features ϕ the utterance relates to. Extracting a sentiment and its target is known as *aspect-based sentiment analysis* (Liu 2020). Ready solutions exist to distill sentiment from language (Hutto and Gilbert 2014; Kim 2014), but extracting the target features is more challenging. We detail our approach in Section 3.3.

We then formalize learning as Bayesian linear regression:

$$\zeta \sim \mathcal{N}(\mathbf{f}^\top w, \sigma_\zeta^2) \tag{3}$$

We use a Gaussian prior: $w^i \sim \mathcal{N}(\mu_i, \Sigma_i)$. After each episode, we perform Bayesian updates (Murphy 2007) to obtain a posterior: $P(w^{i+1}|u^i, \tau^i) = \mathcal{N}(\mu_{i+1}, \Sigma_{i+1})$. Thus, similar to IRL methods (Ramachandran and Amir 2007), the regression sets the teacher's latent preferences $w \in \mathbb{R}^K$ to "explain" the sentiment. Intuitively, if the teacher says "Good job," a learner could infer the teacher has positive weights on the features obtained by its prior trajectory. In the next section, we formalize this mapping to features.

3.3 Extracting MDP Features from Language

The main challenge of aspect-based sentiment analysis is extracting target features from arbitrary language. To accomplish this, we draw on educational research (Lipnevich and Smith 2009; Shute 2008), which studies the characteristic forms of feedback given by human teachers. We first identify correspondences between these forms and prior work in RL. We then show each form targets a distinct element of the MDP (e.g., a prior trajectory). Mapping language to these elements allows us to extract target features.

Evaluative Feedback. Perhaps the simplest feedback an agent can receive is a scalar value in response to their actions (e.g. environmental rewards, praise, criticism). The RL literature has previously elicited such feedback (+1/-1) from human teachers (Thomaz and Breazeal 2008; Knox and Stone 2009; MacGlashan et al. 2017). In our setting, we consider how linguistic utterances can be interpreted as evaluative feedback. For example, "Good job" clearly targets the learner's behavior, τ^i. We thus set the target features to the feature counts obtained by that trajectory: $\mathbf{f} = n_\phi(\tau^i)$.[2]

Imperative Feedback. Another form of feedback tells the learner what the correct action was. This is the general form of supervised learning. In RL, it includes labeling sets of actions as good or bad (Judah et al. 2010; Christiano et al. 2017), learning from demonstrations (Ross and Bagnell 2010; Abbeel and Ng 2004; Ho et al. 2016), and corrections to dialogue agents (Li et al. 2016; Chen et al. 2017). In our setting, imperative feedback specifies a counterfactual behavior: something the learner should (or should not) have done, e.g. "You should have gone to the living room." Imperative feedback is thus a retrospective version of a command. Extracting features takes two steps: we first ground the language to a set of actions, then aggregate their feature counts. Formally, we define a state-action grounding function $G(u, S, A)$ which returns a set of state-action tuples from the full set: $G : u, S, A \mapsto \tilde{S}, \tilde{A}$, where $\tilde{S} \subseteq S, \tilde{A} \subseteq A$. We take the feature counts of these tuples as our target: $\mathbf{f} = n_\phi(G(u, S, A))$.

Descriptive Feedback. Finally, descriptive feedback provides explicit information about how the learner should modify their behavior. Descriptive feedback is the most variable form of human teaching, encompassing explanations and problem-solving strategies. It is generally found to be the most effective (Shute 2008; Lipnevich and Smith 2009; van der Kleij, Feskens, and Eggen 2015; Hattie and Timperley 2007). Supervised and RL approaches have used descriptive language to improve sample efficiency (Srivastava, Labutov, and Mitchell 2017; Hancock et al. 2018; Ling and Fidler 2017) or communicate general task-relevant information (Narasimhan, Barzilay, and Jaakkola 2018). In IRL, descriptive feedback explains the underlying structure of the teacher's preferences and thus relates directly to features ϕ.[3] If the human says 'I don't like seeing you in the kitchen," the robot should infer negative rewards for states and actions where it and the human are both in the kitchen. Formally, we define an indicator function over features desig-

[2]As an example, an alternative teaching theory could use an inverse choice model (McFadden 1974; Train 2003). This would posit a teacher giving feedback on the learner's *implied, latent* preferences, rather than their *explicit, observed* actions.

[3]In problem settings beyond IRL, such feedback may relate to the transition function $T : S \times A$.

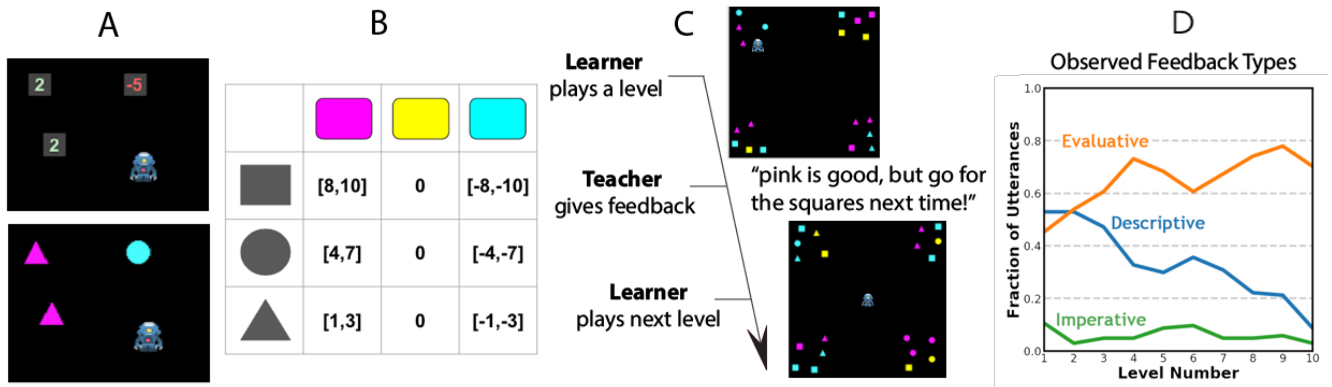

Figure 2: **A**: The learner collected objects with different rewards (top) masked with shapes and colors (bottom). The teacher could see both views. **B**: Example reward function used to mask objects (here, objects worth 1-3 reward are rendered as pink triangles; the teacher thus "prefers" pink squares, worth 8-10). **C**: Pairs played 10 episodes, each on a new level. **D**: Feedback shifted from descriptive to evaluative as learners improved. Learners scored poorly on level 6, reversing this trend.

nating whether or not that feature is referenced in the utterance: $I : u, \phi \rightarrow \{0, 1\}^K$. We then set $\mathbf{f} = I(u, \phi)$.

Prior RL algorithms generally operate on one of these forms. Interactions are constrained, as the algorithm *solicits* feedback of a particular type. Our framework unifies them, allowing us to learn from a wide range of naturalistic human feedback. Concretely, we define a *grounding* function $f_G : u \mapsto \textit{form}, \textit{form} \in \{\text{Imperative, Evaluative, Descriptive}\}$, then extract $\mathbf{f}$ accordingly:

$$\mathbf{f} \in \mathbb{R}^K = \begin{cases} n_\phi(\tau^i) & \text{if } f_G(u) = \text{Evaluative} \\ n_\phi(G(u, S, A)) & \text{if } f_G(u) = \text{Imperative} \\ I(u, \phi) & \text{if } f_G(u) = \text{Descriptive} \end{cases} \tag{4}$$

This procedure is illustrated in Fig. 1. Table 1 shows examples of this decomposition for various forms. In Section 4, we elicit and analyze naturalistic human-human teaching, observing these three forms. In Section 5, we describe a pair of agents implementing this model. Finally, we train an end-to-end neural network and probe its representations, showing that it learns to interpret different forms of feedback in a similar manner.

4 Human-Human Instruction Dataset

To study human linguistic feedback and generate a dataset to evaluate our method, we designed a two-player collaborative game (Fig. 2). One player (the learner) used a robot to collect a variety of colored shapes. Each yielded a reward, which the learner could not see. The second player (the teacher) watched the learner and could see the objects' rewards. Teacher-learner pairs engaged in 10 interactive episodes. We describe the experiment below.

4.1 Experiment and Gameplay

We recruited 208 participants from Amazon Mechanical Turk using psiTurk (Gureckis et al. 2016). Participants were paid \$1.50 and received a bonus up to \$1.00 based on the learner's score. The full experiment consisted of instructions and a practice level, followed by 10 levels of gameplay. Each level contained a different set of 20 objects. We generated 110 such levels, using 10 for the experiment and 100 for model evaluation (Section 6). Collecting each object yields a reward between -10 and 10. Objects were distributed to each of the four corners (Fig. 2C). In each *episode*, the learner had 8 seconds to *act* (move around and collect objects), then the teacher had unlimited time to provide *feedback* (send chat messages). Both players were shown the score and running bonus during the feedback phase. This leaked information about the reward function to the learner, but we found it was important to encourage active participation. The primary disadvantage is that the human baseline for human-model comparisons benefits from additional information not seen by our models.

4.2 Task MDP and Rewards

Human control was continuous: the learner used the arrow keys to steer the robot. However, the only rewarding actions were collecting objects and there was no discounting. As a result, the learner's score was the sum of the rewards of the collected objects. Due to object layout and short time horizon, learners only had time to reach one corner. Each corner had 5 objects, so there were 124 possible object combinations per level.[4] We refer to these combinations as trajectories τ, and formalize the task as choosing one. Concretely, the learner samples its beliefs $w \sim p(w^i)$, then chooses the optimal trajectory: $\pi := \text{argmax}_\tau V^\tau = w^\top n_\phi(\tau)$. To induce teacher preferences, we assigned each teacher a reward function which masked objects with shapes and colors. Thus the distribution of actions and rewards on each level were the same for all players, but the objects were displayed differently depending on the assigned reward function. Our reward functions combined two independent perceptual dimensions, with color (pink, blue, or yellow) encoding sign

[4]Choice of corner, then up to 5 objects: $4 * (\binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5}) = 124$.

Utterance	Feedback Form	Grounding (f_G)	Features (**f**)	Sentiment (ζ)
"Keep it up excellent"	Evaluative	$n_\phi(\tau^i)$	Behavior-dependent	+17
"Not a good move"	Evaluative	$n_\phi(\tau^i)$	Behavior-dependent	-10
"Top left would have been better"	Imperative	$n_\phi(G(u, S, A))$	Environment-dependent	+17
"The light-blue squares are high valued"	Descriptive	$I(u, \phi)$	$\phi_{\text{BlueSquare}}$	+13
"I think Yellow is bad"	Descriptive	$I(u, \phi)$	ϕ_{Yellow}	-16

Table 1: Example feedback from our experiment with feature / sentiment decomposition.

and shape (circles, squares, or triangles) encoding magnitude (Fig. 2B). We permuted the shapes and colors to generate 36 different functions.

4.3 Human-Human Results and Language

Our 104 human pairs played 10 games each, yielding 1040 total messages (see Table 1 for examples; note empty messages were allowed). We use our feedback classifier (see Section 5.1) to explore the prevalence of various forms of feedback. We observe that humans use all three in a curriculum structure known as "scaffolding" (Shute 2008): teachers initially use descriptive feedback to correct specific behavior, then switch to evaluative as the learners' score improves (Fig 2D). Teachers could send unlimited messages and thus sometimes used multiple forms. Most episodes contained *evaluative* (63%) or *descriptive* (34%) feedback; only 6% used *imperative*. The infrequency of imperative feedback is reasonable given our task: specifying the optimal trajectory via language is more challenging than describing desirable features. Not all pairs fared well: some learners did not listen, leading teachers to express frustration; some teachers did not understand the task or sent irrelevant messages. We do not filter these out, as they represent naturalistic human language productions under this setting.

5 Agent Models

We now describe our three models. The first (Section 5.1) directly implements our sentiment-based framework. The second (Section 5.2) extends it with pragmatic biases based on Gricean maxims (Grice 1975). Finally, we train a neural net end-to-end from experiment episodes (Section 5.3).

5.1 "Literal" Model

Our literal model uses a supervised classifier to implement f_{G} and a small lexicon to extract target features.

Utterance Segmentation and Sentiment. Teachers often sent multiple messages per episode, each potentially containing multiple forms of feedback. To process them, we first split each message on punctuation (!.,;), then treated each split from each message as a separate utterance. To extract sentiment, we used VADER (Hutto and Gilbert 2014), which is optimized for social media. VADER provides an output $\zeta \in [-1, 1]$, which we scaled by 30 (set via grid search).

Utterance Features. To implement f_{G}, we labeled 685 utterances from pilot experiments and trained a logistic regression on TF-IDF unigrams and bigrams, achieving a weighted-F1 of .86. For evaluative feedback, as described in Eq. 4, we simply used the feature counts from the learner's trajectory $\mathbf{f} = n_\phi(\tau^i)$. Imperative feedback requires a task-specific action-grounding function $G(u, S, A)$. While action grounding in complex domains is an open research area (Tellex et al. 2020), in our experiment all imperative language referenced a cluster of objects (e.g. "Top left would have been better"). We thus used regular expressions to identify references to corners and aggregated features over actions in that corner. For descriptive feedback, we defined a similar indicator function $I(u, \phi)$ identifying features in the utterance. We used relatively nameable shapes and colors, so teachers used predictable language to refer to object features ("pink", "magenta", "purple", "violet"...). Again, we used regular expressions to match these synonyms. Finally, we normalized $\|\mathbf{f}\|_1 = 1$ so all forms carry equal weight.

Belief Updates. Because players had seen object values in practice levels ranging between -10 and 10, we initialized our belief state as $\mu_0 = 0$, $\Sigma_0 = \text{diag}(25)$. This gives an approximately 95% chance of feature weights falling into that range. For each utterance, we perform Bayesian updates to obtain posteriors $P(w^{i+1}|u^i, \tau^i) = \mathcal{N}(\mu_{i+1}, \Sigma_{i+1})$. We use $\sigma^2_\zeta = \frac{1}{2}$ for all updates, which we set via grid search.

5.2 "Pragmatic" Model

We augment the "literal" model with two biases based on pragmatic principles (Grice 1975). While pragmatics are often derived as a result of recursive reasoning (Goodman and Frank 2016), we opt for a simpler heuristic approach.

Sentiment Pragmatics. The Gricean "maxim of quantity" states that speakers bias towards parsimony. Empirically, teachers often referenced a feature or an action without an explicit sentiment. Utterances such as "top left" or "pink circles" implied positive sentiment (e.g. "pink circles [are good]"). To account for this, we defaulted to a positive bias ($\zeta = 15$) if the detected sentiment was neutral.

Reference Pragmatics. The Gricean "maxim of relation" posits that speakers provide information that is relevant to the task at hand. We assume utterances describe important features, and thus unmentioned features are *not* useful for decision making. We implemented this bias by following each Bayesian update with a second, negative update ($\zeta = -30$, set via grid search) to all features not referenced by the original update, gradually decaying weights of unmentioned features.

5.3 End-to-end Inference Network

To complement our lexicon-based sentiment models, we train a small inference network to predict the teacher's la-

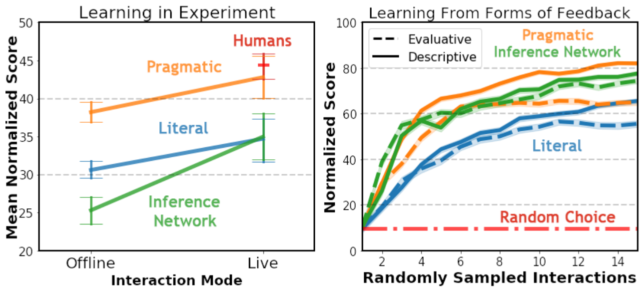

Figure 3: Left: learning within our experiment structure. We plot averaged normalized score over the 10 learning episodes; bars indicate 1 SE (68% CI). Right: learning with specific feedback types. We plot averaged normalized score on 100 test levels after each episode.

tent rewards. We use human data from our experiment to learn an end-to-end mapping from the (u, τ) tuples to the teacher's reward parameters. Conceptually, this is akin to "factory-training" a housecleaning robot, enabling it to subsequently adapt to its owners' particular preferences.

Model Architecture. We use a feed-forward architecture. We tokenize the utterance and generate a small embedding space (D=30), representing phrases as a mean bag-of-words (MBOW) across tokens. We represent the trajectory with its feature counts, $n_\phi(\tau)$. We concatenate the token embeddings with the feature counts, use a single fully-connected 128-width hidden layer with ReLU activations, then use a linear layer to map down to a 9-dimension output.

Model Training. Our dataset is skewed towards positive-scoring games, as players learned over the course of the experiment. To avoid learning a default positive bias, we first downsample positive-score games to match negative-score ones. This left a total of 388 episodes from 98 different teachers with a mean score of 1.09 (mean of all games was 8.53). We augment the data by exchanging the reward function (Fig. 2B), simulating the same episode under a different set of preferences. We take a new reward function and switch both feature counts and token synonyms, preserving the relationships between u^i, τ^i, and w. We repeat this for all 36 possible reward functions, increasing our data volume and allowing us to separate rewards from teachers. We used ten-fold CV with 8-1-1 train-validate-test splits, splitting both teachers and reward functions. Thus the network is trained on one set of rewards (i.e. latent human preferences) and teachers (i.e. linguistic expression of those preferences), then tested against unseen preferences and language. We used stochastic gradient descent with a learning rate of .005 and weight decay of 0.0001, stopping when validation set error increased. We train the network, including embeddings, end-to-end with an L2 loss on the true reward.

Multiple Episodes. Given a (u, τ) tuple, our model predicts the reward $\hat{w}$ associated with every feature. To evaluate it over multiple trials in Section 6, we use a comparable update procedure as our structured models. Concretely, we initialize univariate Gaussian priors over each feature $\mu_0 = 0$, $\sigma_0 = 25$, then run our inference network on each interaction

Model	Experiment			Interaction Sampling			
	Offline	Live	n	All	Eval	Desc	Imp
Literal	30.6	34.7	46	40.5	38.7	40.6	16.7
Pragmatic	**38.2**	**42.8**	47	**52.5**	50.4	**58.2**	**31.7**
Inference	25.3	35.0	55	47.6	**54.3**	53.2	–
Human	–	44.3	104	–	–	–	–

Table 2: Normalized scores averaged over 10 episodes of learning. "Experiment" plays the 10 experiment episodes with a single human; "Interaction Sampling" draws (u, τ) tuples from the entire corpus and plays 100 test levels after each update.

and perform a Bayesian update on each feature using our model's output as an observation with the same fixed noise. For each feature, $P(w^{i+1}|u^i, \tau^i) = \mathcal{N}(\mu_i, \sigma_i) * \mathcal{N}(\hat{w}, \frac{1}{2})$. In all offline testing, we use the network from the appropriate CV fold to ensure it is always evaluated on its test set (teachers and rewards).

6 Results and Analysis

We seek to answer several questions about our models. First, do they *work*: do they learn the humans' reward function? Second, does our sentiment approach provide an advantage over the end-to-end learned model? And finally, do the "pragmatic" augmentations improve the "literal" model? We run a second interactive experiment pairing human teachers with our models and find the answer to all three is yes (Section 6.1). We then analyze *how* our models learn by testing forms of feedback separately (Section 6.2).

6.1 Learning from Live Interactions

To evaluate our models in an online setting, we recruited 148 additional participants from Prolific, an online participant recruitment tool, and paired each with one of three model learners in our task environment. We measured the averaged normalized score across all 10 levels (i.e. the mean percentage of the highest possible score achieved). To assess the effect of interactivity, we also evaluated the same three model learners on replayed sequences of (u, τ) tuples drawn from our earlier human-human experiment. The results are shown in Fig. 3 and summarized in Table 2. We conducted a mixed-effects regression (Kuznetsova, Brockhoff, and Christensen 2017) using performance as the dependent variable, including fixed effects of time (i.e. trial 1, trial 2, etc.), interactivity (i.e. live vs. offline), and learner model (i.e. neural vs. literal vs. pragmatic), as well as an interaction between interactivity and time. We also included random intercepts and random effects for the learner model for each dyad to control for clustered variance. The categorical factor of the language model was contrast-coded to first compare the neural against the two sentiment models and then compare the two sentiment models directly against each other.

First, we found a significant main effect of time, $t(4138) = 32.77, p < .001$, indicating that performance improves over successive levels. Second, although there was no significant main effect of interactivity, $t(446) = -.08, p =$

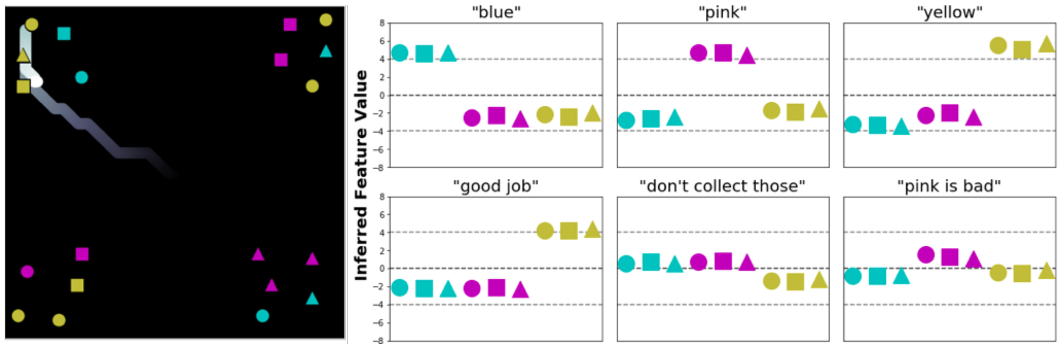

Figure 4: Left: A trajectory from our experiment. Right: Inference network output given this trajectory. Top row: the model learns to map feature-related tokens (descriptive feedback) directly to rewards, independent of the trajectory. Bottom left / center: the model maps evaluative feedback (praise and criticism) through the feature-counts from the trajectory. Bottom right: a failure mode. Due to the MBOW utterance representation, descriptive feedback with *negative* sentiment is not handled correctly.

.94, there was a significant interaction between interactivity and time, $t(4138) = 2.32, p = .02$, suggesting that the benefits of the live condition manifest over successive episodes as the teacher provides feedback conditioned on the learner's behaviors. Finally, turning to the models themselves, we find that the "family" of sentiment models collectively outperform the neural network $t(132) = 3.57, p < .001$ and the "pragmatic" sentiment model outperforms the "literal" one, $t(147) = -2.37, p = .019$. Post-hoc pairwise tests (Tukey 1953) find an estimated difference of $d = 7.8, 95\%$ CI: $[2.7, 12.9]$ between the "pragmatic" and "literal" models; $d = -3.77, 95\%$ CI: $[-11.8, 4.3]$ between the neural and "literal"; and $d = -11.5, 95\%$ CI: $[-19.4, -3.7]$ between neural and "pragmatic." This suggests the end-to-end model learns how to use most of the literal information in the data, while the inductive biases we encoded into the "pragmatic" further improve performance, comparable to human learners (with the caveat that our experiments used different platforms to recruit participants).

6.2 Learning from Different Forms of Feedback

To characterize model learning from different "forms" of feedback, we design a second evaluation independent of the experiment structure. Our "episode" sequence is as follows: we draw a (u, τ) tuple at random from the human-human experiment, *update* each model, and have it *act* on our 100 pregenerated test levels. We take its averaged normalized score on these levels. We repeat this procedure 5 times for each cross-validation fold, ensuring the learned model is always tested on its hold-out teachers and rewards. This draws feedback from a variety of teachers and tests learners on a variety of level configurations, giving a picture of overall learning trends. Normalized scores over test levels are shown in Fig. 3 and Table 2 ("Interaction Sampling"). We note that all models improve when sampling the entire corpus ("All") versus being paired with an individual teacher (Section 6.1). This suggests blending feedback across teachers helps mitigate individual idiosyncrasies. This particularly benefits the inference network, which outperforms the "literal" model. We then use our feedback classifier (Section 5.1) to selectively sample one form of feedback. This reveals that our "pragmatic" augmentations help on all forms, but most dramatically on "Descriptive" feedback. Finally, we confirm our inference network learns to contextualize feedback appropriately (Fig. 4). It maps evaluative feedback through its prior behavior and descriptive (feature-based) tokens directly to the appropriate features. Inspection of utterances reveals several failure modes on rarer speech patterns, most notably descriptive feedback with negative sentiment.

7 Conclusion

We presented two methods to recover latent rewards from naturalistic language: using aspect-based sentiment analysis and learning an end-to-end mapping from utterances to rewards. In evaluation with live humans, we find our sentiment-based models offer statistically significant performance gains over the end-to-end model. The "pragmatic" model achieves near-human performance, suggesting that humans do indeed draw inferences beyond the literal information content. Finally, we note that the relative model performance is qualitatively different across our three evaluations: the learned model underperforms the "literal" offline, performs equivalently in live experiments, and outperforms it on randomly sampled interactions. This underscores the importance of realistic (live) testing.

We see several future research directions. First, our sentiment models could be improved via theory-of-mind based pragmatics, while our end-to-end model could benefit from stronger language models (recurrent networks or pre-trained embeddings). Most promisingly, hybridizing sentiment and learned approaches (Jiang et al. 2011; Xu et al. 2019) could offer the best of both. We also see potential synergies with instruction following: treating commands as "Imperative" feedback could provide a preference-based prior for interpreting future instructions. Finally, we anticipate extending our approach to more complex MDPs in which humans teach both rewards and transition dynamics (Narasimhan, Barzilay, and Jaakkola 2018). In general, we hope the methods and insights presented here facilitate the adoption of truly natural language as an input for learning.

Ethical Statement

Equipping artificial agents with the capacity to learn from linguistic feedback is an important step towards improving value alignment between humans and machines, one aspect of developing systems that support safe and beneficial interactions. However, one future risk is expanding the set of roles that such agents can play to those requiring significant interaction with humans – roles currently restricted to human agents. As a consequence, certain jobs may be more readily replaced by automated systems. On the other hand, being able to provide feedback to artificial agents verbally could expand the group of people who will be able to interact with those agents, potentially creating new opportunities for people with disabilities or less formal training in computer science.

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. ICML, 1. New York, NY, USA.

Allison, F.; Luger, E.; and Hofmann, K. 2018. How Players Speak to an Intelligent Game Character Using Natural Language Messages. *Transactions of the Digital Games Research Association* 4.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57(5): 469–483.

Artzi, Y.; and Zettlemoyer, L. 2011. Bootstrapping Semantic Parsers from Conversations. In *EMNLP 2011*, 421–432. ACL.

Bahdanau, D.; Hill, F.; Leike, J.; Hughes, E.; Hosseini, S. A.; Kohli, P.; and Grefenstette, E. 2019. Learning to Understand Goal Specifications by Modelling Reward. In *ICLR 2019*.

Chen, L.; Yang, R.; Chang, C.; Ye, Z.; Zhou, X.; and Yu, K. 2017. On-line Dialogue Policy Learning with Companion Teaching. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 198–204. Valencia, Spain: Association for Computational Linguistics.

Chevalier-Boisvert, M.; Bahdanau, D.; Lahlou, S.; Willems, L.; Saharia, C.; Nguyen, T. H.; and Bengio, Y. 2018. BabyAI: First Steps Towards Grounded Language Learning With a Human In the Loop. *ArXiv* abs/1810.08272.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *NeurIPS*, 4299–4307.

Cideron, G.; Seurin, M.; Strub, F.; and Pietquin, O. 2019. Self-Educated Language Agent With Hindsight Experience Replay For Instruction Following. *ArXiv* abs/1910.09451.

Djalali, A.; Clausen, D.; Lauer, S.; Schultz, K.; and Potts, C. 2011. Modeling Expert Effects and Common Ground Using Questions Under Discussion. In *Proceedings of the AAAI Workshop on Building Representations of Common Ground with Intelligent Agents*. Washington, DC: AAAI Press.

Djalali, A.; Lauer, S.; and Potts, C. 2012. Corpus Evidence for Preference-Driven Interpretation. In Aloni, M.; Kimmelman, V.; Roelofsen, F.; Sassoon, G. W.; Schulz, K.; and Westera, M., eds., *Proceedings of the 18th Amsterdam Colloquium: Revised Selected Papers*, 150–159. Berlin: Springer.

Dragan, A. D.; Lee, K. C.; and Srinivasa, S. S. 2013. Legibility and Predictability of Robot Motion. In *ACM/IEEE International Conference on Human-Robot Interaction*.

Fu, J.; Korattikara, A.; Levine, S.; and Guadarrama, S. 2019. From Language to Goals: Inverse Reinforcement Learning for Vision-Based Instruction Following. *ArXiv* abs/1902.07742.

Goodman, N. D.; and Frank, M. C. 2016. Pragmatic Language Interpretation as Probabilistic Inference. *Trends in Cognitive Sciences* 20(11): 818 – 829.

Goyal, P.; Niekum, S.; and Mooney, R. J. 2019. Using natural language for reward shaping in reinforcement learning. In *IJCAI*, 2385–2391. AAAI Press.

Goyal, P.; Niekum, S.; and Mooney, R. J. 2020. PixL2R: Guiding Reinforcement Learning Using Natural Language by Mapping Pixels to Rewards.

Grice, H. P. 1975. Logic and Conversation. In *Syntax and Semantics: Vol. 3: Speech Acts*, 41–58. New York: Academic Press.

Gureckis, T. M.; Martin, J.; McDonnell, J.; Rich, A. S.; Markant, D.; Coenen, A.; Halpern, D.; Hamrick, J. B.; and Chan, P. 2016. psiTurk: An open-source framework for conducting replicable behavioral experiments online. *Behavior Research Methods* 48(3): 829–842. ISSN 1554-3528.

Hadfield-Menell, D.; Russell, S. J.; Abbeel, P.; and Dragan, A. 2016. Cooperative Inverse Reinforcement Learning. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *NeurIPS*, 3909–3917. Curran Associates, Inc.

Hancock, B.; Varma, P.; Wang, S.; Bringmann, M.; Liang, P.; and Ré, C. 2018. Training Classifiers with Natural Language Explanations. *ACL* .

Harnad, S. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42(1-3): 335–346.

Hattie, J.; and Timperley, H. 2007. The Power of Feedback. *Review of Educational Research* 77(1): 81–112.

He, H.; Balakrishnan, A.; Eric, M.; and Liang, P. 2017. Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings. *ACL* .

Ho, M. K.; Littman, M.; MacGlashan, J.; Cushman, F.; and Austerweil, J. L. 2016. Showing versus Doing: Teaching by Demonstration. In *NeurIPS*, 3027–3035.

Hu, M.; and Liu, B. 2004. Mining and summarizing customer reviews. In *SIGKDD*, 168–177.

Hutto, C. J.; and Gilbert, E. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *ICWSM*. The AAAI Press.

Ilinykh, N.; Zarrieß, S.; and Schlangen, D. 2019. MeetUp! A Corpus of Joint Activity Dialogues in a Visual Environment. *arXiv* abs/1907.05084.

Jeon, H. J.; Milli, S.; and Dragan, A. D. 2020. Reward-rational (implicit) choice: A unifying formalism for reward learning. *ArXiv* abs/2002.04833.

Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; and Zhao, T. 2011. Target-dependent twitter sentiment classification. In *AACL*, 151–160.

Judah, K.; Roy, S.; Fern, A.; and Dietterich, T. G. 2010. Reinforcement Learning via Practice and Critique Advice. In *AAAI*, 481–486.

Kaplan, R.; Sauer, C.; and Sosa, A. 2017. Beating Atari with Natural Language Guided Reinforcement Learning. *ArXiv* abs/1704.05539.

Kim, E. S.; Leyzberg, D.; Tsui, K. M.; and Scassellati, B. 2009. How people talk when teaching a robot. In *ACM/IEEE International Conference on Human-Robot Interaction*, 23–30.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. *EMNLP 2014* .

Knox, W. B.; and Stone, P. 2009. Interactively Shaping Agents via Human Reinforcement: The TAMER Framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*, 9–16. New York, NY, USA: Association for Computing Machinery.

Kuhlmann, G.; Stone, P.; Mooney, R.; and Shavlik, J. 2004. Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. *AAAI Workshop - Technical Report* .

Kuznetsova, A.; Brockhoff, P. B.; and Christensen, R. 2017. lmerTest package: tests in linear mixed effects models. *Journal of statistical software* 82(13): 1–26.

Li, J.; Miller, A. H.; Chopra, S.; Ranzato, M.; and Weston, J. 2016. Dialogue Learning With Human-In-The-Loop. *ArXiv* abs/1611.09823.

Ling, H.; and Fidler, S. 2017. Teaching Machines to Describe Images with Natural Language Feedback. In *NIPS*.

Lipnevich, A.; and Smith, J. 2009. Effects of differential feedback on students' examination performance. *Journal of Experimental Psychology: Applied* 15(4): 319–333.

Liu, B. 2020. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Luketina, J.; Nardelli, N.; Farquhar, G.; Foerster, J.; Andreas, J.; Grefenstette, E.; Whiteson, S.; and Rocktäschel, T. 2019. A Survey of Reinforcement Learning Informed by Natural Language. In *IJCAI 2019*, volume 57. AAAI Press.

MacGlashan, J.; Babes-Vroman, M.; desJardins, M.; Littman, M. L.; Muresan, S.; Squire, S.; Tellex, S.; Arumugam, D.; and Yang, L. 2015. Grounding English Commands to Reward Functions. In *Robotics: Science and Systems*.

MacGlashan, J.; Ho, M. K.; Loftin, R.; Peng, B.; Wang, G.; Roberts, D. L.; Taylor, M. E.; and Littman, M. L. 2017. Interactive Learning from Policy-Dependent Human Feedback. JMLR.

Maclin, R.; and Shavlik, J. W. 1994. Incorporating advice into agents that learn from reinforcements. AAAI.

McFadden, D. 1974. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics* 105–142.

Mooney, R. J. 2008. Learning to Connect Language and Perception. In *AAAI*, 1598–1601.

Murphy, K. 2007. Conjugate Bayesian analysis of the Gaussian distribution URL https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf.

Narasimhan, K.; Barzilay, R.; and Jaakkola, T. 2018. Grounding language for transfer in deep reinforcement learning. *Journal of Artificial Intelligence Research* 63: 849–874.

Potts, C. 2012. Goal-Driven Answers in the Cards Dialogue Corpus. In Arnett, N.; and Bennett, R., eds., *Proceedings of the 30th West Coast Conference on Formal Linguistics*, 1–20. Somerville, MA: Cascadilla Press.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Ramachandran, D.; and Amir, E. 2007. Bayesian Inverse Reinforcement Learning. In *IJCAI*, volume 7, 2586–2591.

Ross, S.; and Bagnell, D. 2010. Efficient reductions for imitation learning. In *AISTATS*, 661–668.

Shute, V. J. 2008. Focus on Formative Feedback. *Review of Educational Research* 78(1): 153–189.

Srivastava, S.; Labutov, I.; and Mitchell, T. 2017. Joint Concept Learning and Semantic Parsing from Natural Language Explanations. In *EMNLP 2017*, 1527–1536. ACL.

Suhr, A.; Yan, C.; Schluger, J.; Yu, S.; Khader, H.; Mouallem, M.; Zhang, I.; and Artzi, Y. 2019. Executing Instructions in Situated Collaborative Interactions. *EMNLP 2019* .

Szlam, A.; Gray, J.; Srinet, K.; Jernite, Y.; Joulin, A.; Synnaeve, G.; Kiela, D.; Yu, H.; Chen, Z.; Goyal, S.; Guo, D.; Rothermel, D.; Zitnick, C. L.; and Weston, J. 2019. Why Build an Assistant in Minecraft? *ArXiv* abs/1907.09273.

Tellex, S.; Gopalan, N.; Kress-Gazit, H.; and Matuszek, C. 2020. Robots That Use Language. *Annual Review of Control, Robotics, and Autonomous Systems* 3(1): 25–55.

Tellex, S.; Kollar, T.; Dickerson, S.; Walter, M. R.; Banerjee, A. G.; Teller, S.; and Roy, N. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *AAAI*.

Thomason, J.; Murray, M.; Cakmak, M.; and Zettlemoyer, L. 2019. Vision-and-Dialog Navigation. *ArXiv* abs/1907.04957.

Thomason, J.; Padmakumar, A.; Sinapov, J.; Walker, N.; Jiang, Y.; Yedidsion, H.; Hart, J.; Stone, P.; and Mooney, R. J. 2020. Jointly Improving Parsing and Perception for Natural Language Commands through Human-Robot Dialog. *The Journal of Artificial Intelligence Research (JAIR)* 67: 327–374.

Thomason, J.; Zhang, S.; Mooney, R.; and Stone, P. 2015. Learning to Interpret Natural Language Commands through Human-Robot Dialog. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 1923–1929. AAAI Press.

Thomaz, A. L.; and Breazeal, C. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172(6-7): 716–737.

Train, K. 2003. *Discrete Choice Methods with Simulation*. Cambridge University Press.

Tukey, J. W. 1953. Section of mathematics and engineering: Some selected quick and easy methods of statistical analysis. *Transactions of the New York Academy of Sciences* 16(2 Series II): 88–97.

Udagawa, T.; and Aizawa, A. 2019. A Natural Language Corpus of Common Grounding under Continuous and Partially-Observable Context. *AAAI* 33: 7120–7127. ISSN 2159-5399.

van der Kleij, F.; Feskens, R.; and Eggen, T. 2015. Effects of Feedback in a Computer-Based Learning Environment on Students' Learning Outcomes: A Meta-Analysis. *Review of Educational Research* 85.

Wang, S. I.; Liang, P.; and Manning, C. D. 2016. Learning Language Games through Interaction. *ACL* .

Wang, X.; Huang, Q.; Çelikyilmaz, A.; Gao, J.; Shen, D.; Wang, Y.-F.; Wang, W. Y.; and Zhang, L. 2019. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. In *CVPR*, 6629–6638.

Williams, E. C.; Gopalan, N.; Rhee, M.; and Tellex, S. 2018. Learning to Parse Natural Language to Grounded Reward Functions with Weak Supervision. In *ICRA*, 4430–4436.

Xu, H.; Liu, B.; Shu, L.; and Philip, S. Y. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *NAACL*.

Zhou, L.; and Small, K. 2020. Inverse Reinforcement Learning with Natural Language Goals. *ArXiv* abs/2008.06924.