Dialogue Policies for Learning Board Games through Multimodal Communication

Maryam Zare, Ali Ayub, Aishan Liu, Sweekar Sudhakara, Alan Wagner, and **Rebecca Passonneau**

Pennsylvania State University, University Park

{muz50, aja5755, azl53, sks6492, azw78, rjp49}@psu.edu

Abstract

This paper presents MDP policy learning for agents to learn strategic behavior–how to play board games–during multimodal dialogues. Policies are trained offline in simulation, with dialogues carried out in a formal language. The agent has a temporary belief state for the dialogue, and a persistent knowledge store represented as an extensive-form game tree. How well the agent learns a new game from a dialogue with a simulated partner is evaluated by how well it plays the game, given its dialogue-final knowledge state. During policy training, we control for the simulated dialogue partner's level of informativeness in responding to questions. The agent learns best when its trained policy matches the current dialogue partner's informativeness. We also present a novel data collection for training natural language modules. Human subjects who engaged in dialogues with a baseline system rated the system's language skills as above average. Further, results confirm that human dialogue partners also vary in their informativeness.

1 Introduction

Agents that can learn by communicating with human have many potential benefits for human-agent interaction in real world situations, including making it easier for ordinary people to integrate agents into their daily activities. Agents that can communicate to learn games could help us understand how to design agents that can communicate to learn how to make strategic decisions, meaning to pursue a goal when the state of the world changes. Games are a useful testbed, given our reliance on extensive-form game trees, which supports generalization across games. Games model a space of interactions from very simple two-player settings (e.g., tic-tac-toe) to highly complex multi-party interactions (e.g., bridge). Our agent learns Markov Decision Process (MDP) dialogue policies to learn *in-a-row* board games by asking questions of dia-
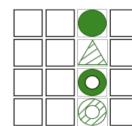
(a) D_3 Board

Question

- **MRL:** *ContextSwitch(D_3), Confirm(Color)*
  NL: Let's step back to the third example for a moment. Does this count because all four pieces are green?

Answer

- **MRL:** *Unknown()*
  NL: I'm not sure about that!

Figure 1: A turn exchange from our dialogue data collection. The original dialogues were between an agent learning Quarto and a simulated dialogue partner, using a meaning representation language (MRL). For 960 dialogues, experts translated each MRL into English. (Best viewed in color.)

logue partners, with policy differences that derive from differences in game complexity, and differences in dialogue partners.

Our MDP policies are trained offline through simulation. Agent dialogues are carried out in a general meaning representation language (MRL) we developed for communicating about games. The agent can request a visual demonstration, or can formulate context-specific verbal questions, including *"yes/no"* questions, as illustrated in Figure 1, and open-ended *"wh-"* questions. Because different humans can have different knowledge, or different dispositions for how much information to give when answering questions, we investigated the impact of policy learning that is sensitive to the informativeness of the dialogue partner. We show that an agent acquires better game knowledge from dialogues when its trained policy matches the dialogue partner. We also add elementary natural language capability, and show that human dialogue partners vary in their informativeness.

Learning through communication (Chai et al.,

2018) is related to learning from demonstration (LfD) (Mulling et al., 2013; Rana et al., 2017), where the goal is for agents to learn through immediate and direct experience rather than through offline processing of large datasets. Previous work on learning through communication has focused on joint grounding of perception and language in task learning (Liu et al., 2016), complex concept grounding (Matuszek, 2018), or collaborative action (Galescu et al., 2018; Perera et al., 2018b), rather than dialogue management. Our work investigates reinforcement learning of dialogue policies, which makes it easy to produce and compare many policies. We exploit the ability to control the behavior of simulated dialogue partners to investigate policy training when dialogue partners vary in informativeness. We develop a policy with hierarchical structure based on a global policy for context-switching, and a local policy for formulating specific questions given a context.

We present two kinds of experiments. First, we compare the MDP policies for different games and different levels of informativeness of simulated dialogue partners. Results show how policies differ across games, and for different dialogue partners. For example, the agent asks more *"wh-"* questions when the dialogue partner is more forthcoming, and more *"yes/no"* questions when the dialogue partner is withholding. Second, we conduct an experiment with human dialogue partners to show that the agent can have successful dialogues with people, and that people vary in informativeness.

To add natural language capability, we developed a novel data collection method and used it to collect a dataset of 960 dialogues (12,885 turn exchanges) for Quarto, one of three *in-a-row* games that our agent can learn. The <MRL, English> pairs are then used to train NLU/NLG modules. The MRL has communicative action types that are functions from contexts to specific questions about that context. Thus the MRL combines utterance meaning with action type (similar to dialogue act type). Figure 1 illustrates one turn exchange in a graphical user interface developed for the data collection. Trained annotators presented with dialogues in MRL translated the MRL to colloquial English. We present initial results where we trained baseline natural language understanding and generation modules from this dataset to show that the agent can learn games in dialogues with people.

No other work we know of addresses the general problem of agents learning through communication with respect to strategic knowledge, meaning knowledge about how to act when the state of the world can change through other agents' actions or natural events. Our first main contribution is development of MDP dialogue policies for learning games through communication, based on our characterization of the learning goal in relative rather than absolute terms: to learn more and better about how to play a game. Specifically, policy training addresses the tradeoff between quality of knowledge acquired from the dialogue partner and length of the dialogue, so that the agent learns how to formulate advantageous questions. Our second main contribution is experimental evidence of the benefits of dialogue policies that are customized to the informativeness of the dialogue partner. Sensitivity to the informativeness of the dialogue partner is particularly important when the role of the dialogue partner is to provide knowledge, given that different dialogue partners can have different levels of expertise, and different communication skills.

2 Related Work

Recent work on deep reinforcement learning has made great progress in developing systems capable of learning Atari games and other games such as Chess, poker, and even Go (Silver and Hassabis, 2017; Silver and Sutskever, 2016; Dobrovsky and Hofmann, 2016). Although the agent does learn how to play the game with considerable accuracy, the process requires large amounts of data, time, and accurate perception. In contrast to this prior work, we seek an approach where an agent learns as much as it can by engaging in short, situated dialogues with human partners.

Most previous work that addresses agent learning through interaction with people, including games, involves agents learning by observing the world (learning from demonstration, or LfD). There can be some verbal input, but without significant knowledge of language or communication strategies. Virtual agents have learned games like Connect Four and Tic-Tac-Toe from demonstration videos, mapping observations to a fragment of first-order logic (Kaiser, 2012), or from sketches combined with natural language (Hinrichs and Forbus, 2013). The SOAR cognitive architecture has been applied to learning Tic-Tac-Toe and Tower of Hanoi (Kirk and Laird, 2014). In LfD, agents can also learn actions, such as how to hit a ping

pong ball (Mulling et al., 2013) or open a drawer (Rana et al., 2017). Active learning has been used for agents to ask clarification questions of a human who gives a fetching request (Whitney et al., 2017), to use pre-defined queries while learning task sequences (Racca and Kyrki, 2018), or to pose a specific question to learn a particular skill (Cakmak and Thomaz, 2012).

Previous work on learning through communication has addressed concept grounding or task learning, rather than learning how to act when the state changes due to other agents' actions. In (Matuszek, 2018), machine-learned classifiers ground words and phrases provided by a human in an agent's perception of the world. Language can also be grounded more directly in perception, by machine learning the relevant perceptual categories from data, rather than pre-specifying them in a formal semantics (Pillai et al., 2019). In (Liu et al., 2016), an agent learns cloth folding through rich verbal communication, based on AND-OR graphs. It can understand utterances with context dependencies common to human language but challenging for machines (e.g., descriptions of objects that evolve over several utterances). Language interaction via semantic parsing combined with deep reasoning is used in agents that explain their actions (Kasenberg et al., 2019b,a), using existing NLP tools for parsing into a logical form (Steedman and Baldridge, 2011), and a rule-based, broad-coverage toolkit for generating English from structured input (Gatt and Reiter, 2009). Other work that relies on rich, situated reasoning through multi-modal communication is based on an architecture for collaborative problem-solving (Galescu et al., 2018), with plan-based dialogue management (Perera et al., 2018a). These works either do not have distinct dialogue management modules, or rely on manually-engineered dialogue management rather than machine-learning. Our work presents machine-learned MDP policies using a method that generalizes across different games, and across differences in dialogue partners' informativeness.

3 Game-learning Dialogues: Overview

Three games our agent learns through communication, in order of complexity, are Connect Four, Gobblet, and Quarto. In all three, players take turns placing pieces on a grid game board. The first player with four pieces in a row wins. There are different sets of possible actions per game due to
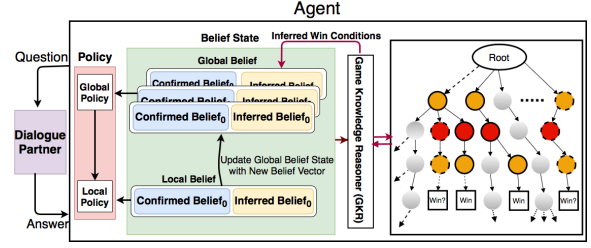

Figure 2: System Architecture.

different board sizes, numbers of game pieces, and properties that distinguish game pieces.

This paper focuses mostly on Quarto. Quarto has a 4×4 board and 16 game pieces, distinguished into two colors, two heights, two shapes, and whether they are solid or hollow. At each turn n of the game, there are $(4^2 - n) \times (4^2 - n)$ possible moves. In each turn, the opponent identifies a piece for the current player to place on the board. Four in a row wins if there is a property shared by all four pieces.

To engage in a game-learning dialogue, a Markov Decision Process (MDP) policy π chooses the agent's dialogue actions, meaning an action a_t at time t depends on the current state s_t, which is fully observable. Reinforcement learning finds an optimal policy π to choose communicative actions that will maximize the expected total reward over time, $R_t = \mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t \ r_t]$. Here we give a brief sketch of the hierarchical policy π, dialogue actions a_t, states s_t, and reward r_t.

The multi-modal dialogues are structured as sequences of sub-dialogues, where each sub-dialogue starts with a visual demonstration of a game board showing a new way to win. The use of demonstrations of win conditions is based on observations from our previous work of how people start asking questions to learn a new game (Ayub and Wagner, 2018). As indicated below, each win condition corresponds to a path to a win state in an extensive-form game tree, where the opponent's game actions are left unspecified. A global policy π_g chooses whether to continue the current subdialogue context, or initiate a new one, while a local policy π_l generates questions to prompt for additional win conditions based on the current demonstration, or additional information about what makes it a win. For example, the agent can ask whether the current configuration of pieces counts as a win due to the color of the pieces (see Figure 1). The use of game trees for knowledge representation is presented in section 4. We developed a meaning representation language (MRL) to represent specific communica-

tive actions a_t that are grounded in the actions and action properties of game trees (see section 5).

Game trees are a well-studied abstraction for representing game knowledge, and for executing play based on tree search. Game trees represent game states as nodes, actions as edges, with payoffs at relevant nodes (Kuhn, 1953). Each visual demonstration of a win condition presented to the agent updates the agent's belief state s_t, as described in section 6. The belief state is also updated after a simulated or human dialogue partner (DP) responds to a question. In turn, the belief state is used to update the agent's knowledge, represented as a game tree. For example, each visual demonstration of a win condition is interpreted as a path in a game tree from the game start to a finish in which the agent wins, and where the other player's actions are unspecified. The agent receives a greater reward r_t when the questions it asks lead to more and better game knowledge, and receives a small penalty on each next turn to encourage efficiency. Dialogues vary in length, depending on the game and the informativeness of the DP, but most dialogues are around a dozen turn exchanges. The reward function and policy training are presented in section 7. An excerpt of a Quarto dialogue from our data collection appears in appendix A.

4   Game Trees as Knowledge

Game theory has been used to represent, reason about, and implement games (Goeree and Holt, 1999; Berlekamp et al., 1982; Ling et al., 2018). Our innovation is to use the game tree abstraction as a vehicle for 1) storing the agent's persistent knowledge about a game, 2) reasoning about that knowledge for dialogue, and 3) providing a measurement of the quality of the game knowledge that the agent acquires in the dialogue.

We developed a game knowledge reasoner (GKR) shown in Figure 2 as an interface between the agent's belief state during a dialogue, and its long-term knowledge store. The GKR assesses the strategic value of new win conditions that a DP has confirmed, and draws inferences about new ways to win that are added to the agent's belief state as unconfirmed beliefs, as discussed further below.

After a dialogue, the agent's final game tree, can be used to engage in play. In an extensive form game tree, each next depth in the tree represents action choices of alternate players. The well-known minimax algorithm (Osborne and Ru-

binstein, 1994) computes a player's optimal action from a given node at depth d_i, on the assumption that at depth d_{i+1} the opponent always chooses its best action. The challenge of learning a new extensive form game is thereby reduced to learning enough of a game tree to engage in play. The quality of what the agent learned is reflected in how often it can win.

At the start of a dialogue, an empty extensive form game tree is initialized, and incrementally extended based on answers to the agent's questions. Game-specific constraints specify how the game tree can grow, e.g. how many actions are available at each node. We use mapping functions from abstract actions in a game tree to physical actions, based on pre-defined information about the game-board and pieces.

The GKR computes a *strategic value* for a new win condition at a given dialog state as a function of the number of overlapping actions with existing win paths in the tree. Given a game tree with N win paths $\{W_1, W_2, ..., W_n\}$ of length m ($W_i = \{a_{i1}, a_{i2}, ..., a_{im}\}$), the Strategic Value (SV) for a new win path $W_j = \{a_{j1}, a_{j2}, ..., a_{jm}\}, j > n$ is a conditional summation:

$$SV(W_j) = \sum_{i=1}^{n} \sum_{k=1}^{m} 1[a_{jk} \in W_i] \qquad (1)$$

At a given depth in the game tree, sibling nodes represent the actions available to the corresponding player. In an incomplete game tree, some of these siblings are part of a set of win paths and some of them are not. If some of the actions at a given depth lead to win conditions, the agent infers that siblings of these actions might lead to similar win conditions. The GKR thus infers unseen board configurations based on the current game tree, and passes them to the dialogue manager as hypothesized win conditions. Formally, given a known win path $W_i = \{a_{i1}, a_{i2}, ..., a_{im}\}$ and a sibling list of an action a_{il} of the win condition W_i ($sibling(a_{il}) = \{a'_1, a'_2, ..., a'_k\}$) the GKR infers a maximum of k new win branches, for k remaining actions in the game, based on a sibling distance metric $SiblingDistance(a_{il}, a'_j) = d$, $j \in \{1, ..., k\}$:

$$W_j = \{a_{i1} + d, a_{i2} + d, ..., a'_j, ..., a_{im} + d\} \quad (2)$$

For Connect Four and Gobblet we use a depth two game tree to make inferences about possible win conditions. For Quarto, we don't set a depth limit. We also use the board positions of inferred

Communicative Actions of agent	
Action Type	**Meaning**
Conf(ChangeDisk)	Is D still a win after ChangeDisks?
Conf(ShiftBoard)	Is D still a win after ShiftBoard?
Conf(Property)	Is Property what makes D a win?
Req(ShiftBoard)	What ShiftBoard operations on D are also a win?
RequestOth()	Can the other player undo D?
WinC(i)	Resume discussion of the i-th D.
NewWinC()	Request an unknown D.
Communicative Actions of Interlocutor	
Action Type	**Meaning**
Inform()	Provide some/all of the requested information
Affirm()	Positive answer to a yes/no question.
Negate()	Negative answer to a yes/no question.
Unknown()	Non-answer to a question.

Table 1: Communicative Action Types, apart from Conventional, for starting or ending dialogues.

win condition W_j to find any known win condition W_i at the same board positions, so as to infer that any feature f shared by all actions in W_j is the game piece feature that contributes to this win condition. The GKR returns this information to the dialog manager. In sum, if the agent sees a new win condition in a row where it has previously seen a win condition, and the color is what distinguishes this new win, it infers that the color is a win feature.

5 Meaning Representation

The communicative action generator takes as input the current context and the communicative action type selected by the dialogue policy, and generates a specific communicative action for the agent in an MRL we describe here. The meaning representation language is described in detail in our previous work (Zare et al., 2019). Here we explain the communicative action types of the agent and dialogue partner. The Action Types at the top of Table 1 show that the agent can ask *yes/no* questions (*Confirm, ConfirmOtherPlayer*), ask *wh-* questions (*Request*), resume a previous context (*WinC()*), or prompt the DP for a new demonstration (*NewWinC()*). These Action Types can be viewed as functions that return a complete MRL as a value. If no argument is shown, the current board D_i is the implicit argument. *Confirm* and *Request* can be used to ask questions about actions that can be taken on the current board (*ChangeDisks, ShiftBoard*) or about properties of the game pieces (*Property*).

The turn exchange in Figure 1 references a demonstrated win condition D_3. It shows the MRL

for a *yes/no-* question asking about the contribution of color of the pieces in D_3. Given an informative DP, a *yes/no-* question elicits a yes or no answer to an agent's question. Here, however, the DP did not provide an answer. The kinds of answers that the agent currently understands are shown at the bottom of Table 1. A *wh-* question elicits an *Inform()* act, and a *yes/no* question elicits a positive (*Affirm()*) or negative (*Negate()*) answer, or *Unknown()*. Here we assume dialogue partners will be truthful, but may not always know the answers to questions, and may provide incomplete answers.

6 Belief State

The global belief space is a set of belief vectors B that represent beliefs acquired during a dialogue (see Figure 2). Each new demonstration D_i instantiates a new local belief vector B_i to represent confirmed information observed in D_i or acquired from responses to questions about D_i. Inferences the GKR makes about possible win conditions are also represented. A game board is represented as a vector representing each board position (e.g., 0 to 15 for Quarto), with a belief value in [0,1] for each vector position. Confirmed beliefs (B_C) and inferred beliefs (B_I) about ways to reconfigure a win condition are similar vectors with an additional position None. Formally, the game belief vector B is defined as concatenated vectors that each pertain to an observed property of game pieces (e.g., color) or a type of physical rearrangement of a configuration of pieces (e.g., rotate):

$$B_C = b_{Color_c} \oplus ... \oplus b_{Size_c} \oplus b_{Rotate_c}$$
$$\oplus b_{Translate_c} \oplus b_{OtherPlayer_c} \oplus b_{Board_c}$$
$$B_I = b_{Translate_i} \oplus b_{Color_i} \oplus ... \oplus b_{Quantity_i}$$
$$B = B_C \oplus B_I \tag{3}$$

Figure 1 illustrates a board demonstration D_3 for Quarto with a vertical sequence of four game pieces starting in position 2. The board D_3 is the implicit argument in the question. B_I is updated at the end of each turn with inferences derived by the GKR. For updating B_C, we rely on the baseline belief tracking method proposed in (Wang and Lemon, 2013). Given a response to a particular question, the component belief vector $vect_t$ gets updated if the turn exchange is a question and answer about a function (e.g. translate) or a property (e.g. game piece shape). When the response from the DP is positive or contains new information, the corresponding belief vectors get updated according to

equation (4). When the DP response is negative, the relevant sub-belief vectors are updated according to equation (5).

$$P_{vect_t} = 1 - \left(1 - P_{vect_{t-1}}\right)\left(1 - P_{u_t}\right) \quad (4)$$
$$P_{vect_t} = \left(1 - P_{vect_{t-1}}\right)\left(1 - P_{u_t}\right) \quad (5)$$

Currently, the confidence score P_{u_t} over the DP utterance is always 1.0, because there is no uncertainty in the interpretation of the MRL. (In future work, we plan to train Partially Observable MDP policies to accommodate the uncertainty in natural language interactions with humans.)

7 Policy Learning and Reward

Through simulation, we can control the informativeness of the DP's responses, and thus investigate the impact of informativeness on policy learning. We train multiple policies, setting the DP informativeness to a value between 0 and 1. A 100% informative DP responds to all questions completely. For lower informativeness, we keep a list of all the possible winning conditions sorted by the number of times they have been presented by the DP in ascending order. When the agent asks a *NewWinC()* question, a DP with $x\%$ informativeness randomly chooses a win condition from the top $(100-x)\%$ of the sorted list. A $x\%$ informative DP responds with *Unknown()* to *Confirm()* queries with $100 - x\%$ probability, and provides only $x\%$ of a complete answer to *Request()* queries.

Gašić and Young (2014) achieved good results with less training for a Gaussian process approach to policy learning. The model has few hyperparameters and converges quickly to a local optimum ($< 20k$ epochs). We adopted their model and trained dialogue polices for $10k$ epochs. The policy gets updated at the end of each interaction.

The reward is designed to encourage the agent to acquire as many new win condition paths as possible, to prefer paths with higher strategic value, and to end the dialogue when the turn costs outweigh the gains in knowledge. Equation 6 shows the reward R for a turn exchange t as a function of the number of new win conditions in the DP's response to a question, the strategic value SV of the response, and a turn cost C (through tuning, we found good performance from $\alpha = 0.2$, $\beta = 3$, and $C = 2$.):

$$R = \lceil\frac{\#WaystoWin}{\beta}\rceil \times \alpha + SV - C \quad (6)$$

Quarto Policies with Six Dialogue Partner Levels						
Commun Act	100%	80%	60%	50%	40%	20%
NewWinC()	0.46	0.44	0.39	0.40	0.33	0.31
WinC()	0.45	0.49	0.52	0.51	0.56	0.60
Req(ShiftBoard)	0.17	0.22	0.23	0.21	0.16	0.20
Conf(ShiftBoard)	0.33	0.15	0.06	0.04	0.12	0.09
Conf(ChangeDisk)	0.01	0.01	0.03	0.04	0.09	0.15
Conf(Property)	0.47	0.62	0.65	0.68	0.58	0.47
RequestOth()	0.02	0.00	0.03	0.03	0.05	0.09
Dialogue Length	10.6	10.3	10.3	9.8	10.2	10.1

Table 2: Dialogue length and action type frequencies.

We progress here through five questions to investigate how considerations of DP informativeness can affect learning through communication.

Our first question is *how dialogue policy learning differs across levels of DP informativeness.* Figure 3 shows a sensitivity analysis of the training process over $10k$ epochs, using change in total reward, for six informativeness levels ranging from 100% to 20%. The informativeness conditions clearly differ, with lower reward for lower informativeness. We achieved similar results for Connect Four and Gobblet with much faster convergence for Connect Four, the simplest game.

Using the fully trained policies from Figure 3, we ask *how communicative actions differ during learning dialogues*. In each informativeness level, the agent engages in 100 dialogues. Table 2 reports the average frequencies of each communicative act type (except Conventional, which is always 9%, since every dialogue has an opening and a closing), and the average dialogue length in turn exchanges. *NewWinC()* and *WinC()* are equiprobable only for the 100% condition; in the other conditions, the latter is somewhat more frequent. More interestingly, the dialogue length is invariant as the agent can still learn from a low informative DP. The frequency of *Confirm(Property)* is highest for the 50% condition, the DP who is neither very informative, nor very uninformative. Similar trends were observed for Gobblet as well. However, for Connect Four, dialogues get shorter as informativeness decreases.
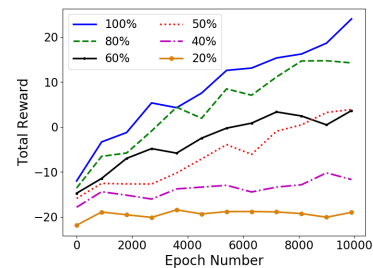

Figure 3: Total reward for six levels of DP informativeness.

	Policy-Dialogue Partner Condition				
Type	**100-100**	**100-50**	**50-50**	**50-100**	**20-20**
Row	40%	25%	40%	20%	10%
Col	50%	25%	45%	50%	0%
Diag	50%	0%	25%	20%	0%
AntiD	75%	25%	50%	0%	25%

Table 3: Final game knowledge under 5 dialogue conditions.

We next ask *how the policy affects what is learned in a given dialogue from a given DP type, and what happens if the agent's learned policy for a DP level X is used when interacting with a DP of level Y*. Table 3 shows five policy-DP (X-Y) conditions we tested. Under each condition, one dialogue from a set of ten dialogues was randomly selected where we inspected the final game tree knowledge. Quarto has four win condition locations, labeling the table rows. The most interesting result common among all three games is that if the DP is neither informative nor uninformative (50%), the agent gains the most game knowledge from using a matching policy (50-50). Note that the agent learns less from a 100% DP using the wrong policy than from a 50% DP using the right policy.

We next ask *how well can the agent play after a learning dialogue.* For Connect Four and Gobblet, we recruited 16 students to play with the agent, using the same conditions and knowledge states from Table 3. Because the slow movements of our Baxter robot (Rethink robotics) resulted in tedious 20-minute games, we used a simulated agent at a terminal. Prior to data collection, each subject played a few practice games to become familiar with the game and the interface. Each subject played 10 games, randomly ordered among the 5 conditions. We set a time limit of 2.5 minutes for each game and used a Minimax algorithm with 2 step look-ahead. We observed that the quantity differences in knowledge acquired by the agent show up directly as quality differences for Connect Four. For Gobblet the proportion of outcomes for the agent were more or less the same across the conditions involving a 50% policy and/or a 50% DP. We attributed the uniform Gobblet results to the time limit for the play and to the need for greater look-ahead, given the many action choices.

	Policy-Dialogue Partner Condition				
Result	**100-100**	**100-50**	**50-50**	**50-100**	**20-20**
Wins	0.94	0.19	0.50	0.18	0.12
Losses	0.00	0.81	0.47	0.78	0.82
Draws	0.06	0.00	0.03	0.04	0.06

Table 4: Percentage of agent wins/losses/draws.

For Quarto, we altered the experiment by removing the restriction on length of play and depth of search. We also developed a graphical user interface to display game pieces in a more realistic way. We recruited 18 students to play Quarto. The game results in Table 4 show that the agent won games more often when it had learned the game from a more informative DP, as long as it used the corresponding policy.

Our final question was *whether the agent could use the same policy to continue learning over a sequence of dialogues.* Here we looked at three conditions: where the learned policy matched the DP informativeness of 100%, 50% and 20%. In each condition, the agent had four dialogues, starting with no knowledge. The agent began each next dialogue with the knowledge it had gained from its previous dialogue. We averaged the final reward at the end of each dialogue. Results show that the agent continues to learn more and more about the game, especially from the 100% informative DP. Results for Gobblet were very similar to Quarto. However for Connect Four, there is usually little reward (knowledge) left to gain after the first or second dialogue in higher informativeness levels, so the reward plateaus after two or three dialogues.

8 Dialogue Data Collection

To add natural language capability for the agent, we developed a novel data collection method to produce a corpus consisting of <Gameboard,MRL,NL> tuples for each utterance in 960 dialogues between an agent and simulated dialogue partner. The Quarto Dialogue corpus is distinctive in that it is agent-agent situated, multi-modal dialogue where agents' utterances are in an MRL, then all dialogues translated by experts into English.

To our knowledge, this is the first corpus of its kind. Most previous dialogue corpora we know of fall into one of three other categories: human-*Wizard-of-Oz*, human-agent, or human-human. Corpora for human-*Wizard-of-Oz* are used either to in-
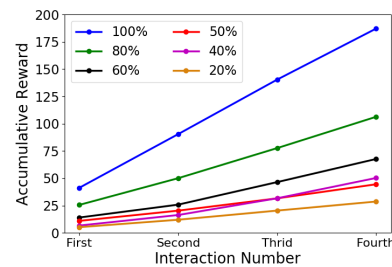

Figure 4: Consecutive dialogues reward trend.

form manually engineered dialogue management or as training data for machine learned dialogue managers. These corpora are collected for the purpose of restaurant reservation (Henderson et al., 2013), finding available vacation accommodations (Asri et al., 2017), or even open-domain information retrieval systems (Rosset and Petel, 2006). Human-agent corpora are often annotated with dialogue acts for applications such as travel booking systems (Bennett and Rudnicky, 2002). Human-human corpora are either collected under constrained settings where humans are instructed to follow a series of instructions (Brennan et al., 2013; Heeman and Allen, 1995), or are naturally occurring conversations between humans (Asher et al., 2016; Afantenos et al., 2012; Passonneau and Sachar, 2014). Distinctive characteristics of the Quarto corpus are that every utterance has an MRL and a natural language version where the MRL is a communicative act. The dialogues involve a shared multi-modal context, leading to deictic reference to the game board and with a known structure into sub-dialogues.

To collect our corpus, we developed two graphical user interfaces (GUIs) to display a schematic representation of the current board demontration (cf. Figure 1), and to allow annotators to page through each turn exchange. One GUI was for the translation task, and a second was to collect ratings on the translations. Thirteen undergraduate students from a course in Artificial Intelligence participated as part of their course work. Students were first trained in the MRL, including comparisons with the first order logic translations of English that students had learned in class. Their instructions were to translate into colloquial English. Meetings were held where students discussed examples and asked questions. All translations were rated for correctness and naturalness on a five-point scale where 5 was the top. On average, correctness was 4.79 and naturalness was 4.72.

The 960 dialogues contain 12,885 turn exchanges. The English translations contain 229,641 word tokens, and 1,498 word types. The NLG data has 146,055 tokens and 1,102 types. The NLU data is somewhat less rich, with 83,586 tokens and 952 types. The 960 dialogues consist of 535 from a 60% informative simulator, 255 from a 100% informative simulator, and 170 from a 50% simulator. We are currently augmenting the data to synthesize new examples for Quarto, and to synthesize Connect Four and Gobblet data.

Because all turn exchanges are tied to a physical board, the corpus is rich in spatial references. The students referred to the pieces by specific attributes (e.g. *next to that green circular piece*), exact location on the board (e.g. *top corner piece*), relation with other pieces (e.g. *to the right of the square piece*), or deictic reference (e.g. *this piece here*). There are also many anaphoric references (e.g. *about that win you showed*, *the second win*).

9 Human-Agent Dialogues

The dataset described above provides training data for NLU and NLG modules to enable the agent to engage in dialogue with humans. Two other changes needed to support future human-agent dialogues are clarification sub-dialogues to handle misunderstandings or confusions, and modification of the policy training and belief updates to address uncertainty in the NLU. To preview our future challenges, we developed baseline NLU and NLG modules, and asked the 18 subjects who played Quarto with our agent to engage in text-based dialogues. Here we describe the dialogue interface, the baseline NLU and NLG modules, the dialogue outcomes, and the subjects' informativeness.

We developed a text-based GUI for subjects to engage in dialogues with an agent, similar to the GUI used for translating MRL into English. For NLG and NLU, we trained two sequence-to-sequence RNN models with two hidden layers and a Bahdanau attention layer (Bahdanau et al., 2015). The Adam optimizer was used for training (Kingma and Ba, 2014) (20 epochs for NLG, and 15 for NLU). The MDP policy for 100% informativeness was used, and belief updating remained the same.

Each subject engaged in two dialogues. Average dialogue length was 10.96 turn exchanges (min 9, max 15, std 2.15), which is similar to dialogues with the simulator. Subjects also completed a questionnaire. The questionnaire[1] asked subjects 1) whether they understood the agent's questions, 2) to list the confusing questions by turn number, 3) to rate the dialogues on a 5-point scale for the agent's command of English, and 4) to tell us how willing they would be to have another dialog with this agent. Fourteen of the subjects said they understood the agent most of the time. Inspection of the questions listed as confusing indicated they all had incomplete or incorrect NLG output. The average fluency rating was 2.93. Eleven subjects said they

---

[1]See Appendix B for the complete list of questions.

Win Type	Mean	Min	Max	SDev
Row	20%	0%	35%	7.8
Col	20%	0%	40%	8.3
Diag	15%	0%	50%	13.3
Anti-Diag	5%	0%	10%	1.2

Table 5: Average final knowledge states for the 36 dialogues

would be willing to have more dialogues, one was neutral, and six were somewhat dissatisfied.

The overall quality of the NLG was good; two thirds of the agent questions were fluent and correct. Of 197 total turn exchanges, 58 were less than perfect. One of the co-authors rated all the generated questions on a five-point scale for correctness and intelligibility, yielding an average score of 4.19 (min 1, max 5, std 1.23). The NLU quality was less good. Subjects' answers were translated to a gold-standard MRL by one of the co-authors, and compared with the NLU output; only 60% of the answers were interpreted correctly. Despite the agent's frequent failure to understand subjects' responses, the average total reward of 12.45 was comparable to the reward for an 80% informative simulator with a matching policy (cf. Figure 3). Table 5 gives the average final knowledge states for the 36 dialogues, which is in the same range as for dialogues with a 50% informative DP and matching policy (see Table 3). To assess the subjects' informativeness, we examined the 139 turn exchanges that subjects understood well, comparing the subjects' answers to 100% informative answers. Subjects' answers were 100% informative only 41% of the time.

The comparison of baseline human-agent learning dialogues with those between an agent and simulated DP shows promise for reinforcement learning of policies that are trained offline in simulation. Subjects provided less than 100% informative answers, and the agent's final knowledge states were similar to those where the agent interacted with a 50% informative simulator, using a matching policy. Even without the ability to engage in clarification sub-dialogues with a human to clear up confusions, the dialogues were all completed. The agent was completely understandable two thirds of the time. The agent learned as much about Quarto as in the 50%-50% simulator condition.

A question raised by these results is how an agent could benefit from having access to multiple dialogue policies. In robotics, a very similar question has been addressed for agents learning motor skills through simulation and deploying the learned policies in real-world environments with unknown dynamics. Approaches include learning to linearly combine a family of policies (Zhang et al., 2018), learning a classifier for environment parameters to choose the correct policy (Yu et al., 2017), or searching directly within a family of policies using the current accumulated reward (Yu et al., 2019). Similar methods could be applied to exploit a family of dialogue policies to adapt questioning strategies in different ways, depending on the observed behavior of the dialogue partner.

10 Conclusion

Our results show that agents can learn MDP policies to learn board games through multi-modal dialogues using a relative knowledge goal, namely to increase the agent's game knowledge as much as possible during a short dialogue. We also show that the agent learns different dialogue policies depending on the dialogue partner's informativeness. This work exploits the benefits of a knowledge domain that has a very abstract representation in the form of game trees, where a novel meaning representation language is grounded in the game tree abstraction. This approach can generalize to a wide range of two-person board games, and provides a foundation for communication learning about other strategic activities. In addition, an agent that can learn new games and then engage in play has potential benefits in Socially Assistive Robotics (Feil-Seifer and Mataric, 2005). Board games have been used to delay the onset of dementia (Dartigues et al., 2013), and have been shown to help children learn computational concepts (Berland and Lee, 2011).

Additionally, we have demonstrated that MDP policies trained offline in simulation can lead to fairly effective human-robot learning dialogues, based on training data for natural language modules we collected through a novel procedure. Our future work will expand the communicative actions to include clarifications, will train POMDP policies, and will borrow ideas from reinforcement learning of robotic motor skills to close the reality gap between offline training of dialogue policies and engaging in real-world dialogues with humans.

References

Stergos Afantenos, Nicholas Asher, Farah Benamara, Myriam Bras, Cécile Fabre, Lydia-Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paule Pery-Woodley, Laurent Prévot, et al. 2012. An empirical resource for discovering cognitive principles of discourse organisation: the ANNODIS corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2727–2734.

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 16)*.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*.

Ali Ayub and Alan R Wagner. 2018. Learning to win games in a few examples: Using game-theory and demonstrations to learn the win conditions of a Connect Four game. In *Social Robotics*, pages 349–358. Springer International Publishing.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *(International Conference on Learning Representations ICLR)*.

Christina Bennett and Alexander I Rudnicky. 2002. The Carnegie Mellon Communicator corpus. In *Seventh International Conference on Spoken Language Processing*.

Matthew Berland and Victor R. Lee. 2011. Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, 1(2):65–81.

E. Berlekamp, J. H. Conway, and R. Guy. 1982. Winning ways for your mathematical plays: Games in general. *Academic Press*.

Susan Brennan, Katharina Schuhmann, and Karla Batres. 2013. Entrainment on the move and in the lab: The walking around corpus. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35.

M. Cakmak and A. L. Thomaz. 2012. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE International conference on Human-Robot Interaction*.

Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. 2018. Language to action: Towards interactive task learning with physical agents. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2–9. International

Joint Conferences on Artificial Intelligence Organization.

Jean Françcois Dartigues, Alexandra Foubert-Samier, Mélanie Le Goff, Mélanie Viltard, Hélène Amieva, Jean Marc Orgogozo, Pascale Barberger-Gateau, and Catherine Helmer. 2013. Playing board games, cognitive decline and dementia: a French population-based cohort study. *BMJ open*, 3(8).

Borghoff-U. M. Dobrovsky, A. and M. Hofmann. 2016. An approach to interactive deep reinforcement learning for serious games. In *7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*.

David Feil-Seifer and Maja J. Mataric. 2005. Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 465–468.

Lucian Galescu, Choh Man Teng, James Allen, and Ian Perera. 2018. Cogent: A generic dialogue system shell based on a collaborative problem solving model. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 400–409, Melbourne, Australia. Association for Computational Linguistics.

Milica Gašić and Steve Young. 2014. Gaussian processes for POMDP-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93, Athens, Greece. Association for Computational Linguistics.

Jacob K Goeree and Charles A Holt. 1999. Stochastic game theory: For playing games, not just for doing theory. *Proceedings of the National Academy of sciences*, 96(19):10564–10567.

Peter A Heeman and James F Allen. 1995. The trains 93 dialogues. Technical report, Rochester University, Dept. of Computer Science.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2013. Dialog state tracking challenge 2 & 3 handbook. *camdial. org/mh521/dstc*.

Thomas R. Hinrichs and Kenneth D. Forbus. 2013. X goes first: Teaching simple games through multimodal interaction. In *Proceedings of the Second Conference on Advances in Cognitive Systems*, pages 31–46.

Lukasz Kaiser. 2012. Learning games from videos guided by descriptive complexity. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 963–69.

Daniel Kasenberg, Antonio Roque, Ravenna Thielstrom, Meia Chita-Tegmark, and Matthias Scheutz. 2019a. Generating justifications for norm-related agent decisions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 484–493, Tokyo, Japan. Association for Computational Linguistics.

Daniel Kasenberg, Antonio Roque, Ravenna Thielstrom, and Matthias Scheutz. 2019b. Engaging in dialogue about an agent's norms and behaviors. In *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI)*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

James R. Kirk and John E. Laird. 2014. Interactive task learning for simple games. In *Advances in Cognitive Systems*, pages 11–28.

H.W. Kuhn. 1953. Extensive form games and the problem of information. In *Contributions to the Theory of Games II*, page 193–216. Princeton University Press.

Chun Kai Ling, Fei Fang, and J. Zico Kolter. 2018. What game are we playing? end-to-end learning in normal and extensive form games. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 396–402. International Joint Conferences on Artificial Intelligence Organization.

Changsong Liu, Shaohua Yang, Sari Saba-Sadiya, Nishant Shukla, Yunzhong He, Song-Chun Zhu, and Joyce Chai. 2016. Jointly learning grounded task structures from language instruction and visual demonstration. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1492, Austin, Texas. Association for Computational Linguistics.

Cynthia Matuszek. 2018. Grounded language learning: Where robotics and nlp meet. In *IJCAI*, pages 5687–5691.

K. Mulling, J. Kober, O. Kroemer, and J. Peters. 2013. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research (IJRR)*.

Martin J Osborne and Ariel Rubinstein. 1994. *A course in game theory*. MIT press.

Rebecca J. Passonneau and Evaneet Sachar. 2014. Loqui human-human dialogue corpus (transcriptions and annotations). Columbia University Academic Commons.

Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018a. Building and learning structures in a situated blocks world through deep language understanding. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 12–20, New Orleans. Association for Computational Linguistics.

Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018b. A situated dialogue system for learning structural concepts in blocks world. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 89–98, Melbourne, Australia. Association for Computational Linguistics.

Nisha Pillai, Cynthia Matuszek, and Francis Ferraro. 2019. Deep learning for category-free grounded language acquisition. In *Proc. of the NAACL Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics (NAACL-SpLU-RoboNLP)*, Minneapolis, MI, USA.

Mattia Racca and Ville Kyrki. 2018. Active robot learning for temporal task models. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 123–131. ACM.

M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots. 2017. Towards robust skill generalization: Unifying learning from demonstration and motion planning. In *Conference on Robot Learning(CoRL)*.

Sophie Rosset and Sandra Petel. 2006. The ritel corpus-an annotated human-machine open-domain question answering spoken dialog corpus. In *LREC*, volume 6, pages 1640–1643.

Huang-A. Maddison C. J. Guez A. Sifre L. Driessche G. v. d. Schrittwieser J. Antonoglou I. Panneershelvam V. Lanctot M. Dieleman S. Grewe D. Nham J. Kalchbrenner N. Silver, D. and I. Sutskever. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489.

Hubert-T. Schrittwieser J. Antonoglou I. Lai M. Guez A. Lanctot M. L. Sifre Kumaran D. Graepel T. Lillicrap T. Simonyan K. Silver, D. and D. Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv Reprint arXiv:1712.01815*.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In R.D. Borsley and K. Börjars, editors, *Non-Transformational Syntax*, pages 181–224. Wiley.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.

D. Whitney, E. Rosen, J. MacGlashan, L. L. Wong, and S. Tellex. 2017. Reducing errors in object-fetching interactions through social feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Wenhao Yu, C. Karen Liu, and Greg Turk. 2019. Policy transfer with strategy optimization. In *International Conference on Learning Representations (ICLR)*.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. 2017. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts.

Maryam Zare, Ali Ayub, Alan R. Wagner, and Rebecca J. Passonneau. 2019. Show me how to win: A robot that uses dialog management to learn from demonstrations. In *Fourth Games and Natural Language Processing Workshop (GAMNLP-19)*, San Luis Obispo, CA.

Chao Zhang, Yang Yu, and Zhi-Hua Zhou. 2018. Learning environmental calibration actions for policy self-evolution. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*.

A Appendix A: Excerpt of a Quarto Dialogue

Here is a sample Quarto dialog from the data collection, showing the MRL,NL pairs, and images of the relevant board contexts.

– **Q1)** *Start()*
 NL: Hi there! Can you teach me how to play Quarto?
 A1) *Start()*
 NL: Sure thing, let's get started.

– **Q2)** *RequestNewWinCondition()*
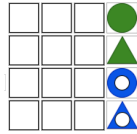 **NL:** Can you show me an example of a win?
 A2)

Figure 5: First Win Condition (D_1)

– **Q3)** *RequestNewWinCondition()*
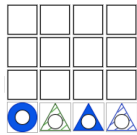 **NL:** Alright, how about another?
 A3)

Figure 6: Second Win Condition (D_2)

– **Q4)** *ContextSwitch (D_1), Confirm(translate(col0))*
 NL: In the first example you showed me, would it still count if the pieces were placed down the first column instead of down the fourth?
 A4) *Affirm()*
 NL: Yeah, it would.

– **Q5)** *Same, Request(rotate())*
 NL: Say I keep the blue triangle in the bottom-right corner, but try placing the remaining pieces along some other direction upward, to the left, or a combination of both. Could I still get a win?
 A5) *Inform([135, 180)*
 NL: You could get a win so long as you placed the remaining pieces along the anti-diagonal, or leftward along the bottom row.

– **Q6)** *Same, Confirm(shape)*
 NL: Does this count because of the shape of the pieces?
 A6) *Negate()*
 NL: No, this win is not based on the shapes.

– **Q7)** *Same, Confirm(quantity)*
 NL: Well, does this count because there are four and only four pieces?
 A7) *Affirm()*
 NL: Yes, it wouldn't count for any other number of pieces.

– **Q8)** *ContextSwitch(D_2), Confirm(removepiece((0,1)*
 NL: Okay, in the second example you showed me, what would happen if I removed the green triangle?
 A8) *Negate()*
 NL: Then you would no longer have a win.

– **Q9)** *RequestNewWinCondition()*
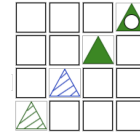 **NL:** Can you show me another example of a win?
 A9)

Figure 7: Third Win Condition (D_3)

– **Q10)** *Finish*
 NL: Great, thanks for helping me get started!
 A10) *Finish*
 NL: I'm happy I could help!

B Appendix B: Questionnaire

Here we present the complete list of questions asked from students.

1. Were you able to understand what the agent was asking you even when it was not completely fluent English.

2. If there were questions you could not understand, please list them below by turn number.

3. For questions you could not understand please try to explain to us your confusion for each of the turns you listed above. (Please use the turn number again)

4. On a scale of 1 to 5 (5 is the best) how likely you would be think that this dialog was typed in by an English speaker?

5. What aspects of this dialog did you find interesting, if any?

6. How likely you would come back and have another dialog about a game with this agent?

7. What aspects of the GUI do you think can be improved?