Planning, Inference and Pragmatics in Sequential Language Games

Fereshte Khani
Stanford University
fereshte@stanford.edu

Noah D. Goodman
Stanford University
ngoodman@stanford.edu

Percy Liang
Stanford University
pliang@cs.stanford.edu

Abstract

We study sequential language games in which two players, each with private information, communicate to achieve a common goal. In such games, a successful player must (i) infer the partner's private information from the partner's messages, (ii) generate messages that are most likely to help with the goal, and (iii) reason pragmatically about the partner's strategy. We propose a model that captures all three characteristics and demonstrate their importance in capturing human behavior on a new goal-oriented dataset we collected using crowdsourcing.

1 Introduction

Human communication is extraordinarily rich. People routinely choose what to say based on their goals (planning), figure out the state of the world based on what others say (inference), all while taking into account that others are strategizing agents too (pragmatics). All three aspects have been studied in both the linguistics and AI communities. For planning, Markov Decision Processes and their extensions can be used to compute utility-maximizing actions via forward-looking recurrences (e.g., Vogel et al. (2013a)). For inference, model-theoretic semantics (Montague, 1973) provides a mechanism for utterances to constrain possible worlds, and this has been implemented recently in semantic parsing (Matuszek et al., 2012; Krishnamurthy and Kollar, 2013). Finally, for pragmatics, the cooperative principle of Grice (1975) can be realized by models in which a speaker simulates a listener—e.g., Franke (2009) and Frank and Goodman (2012).

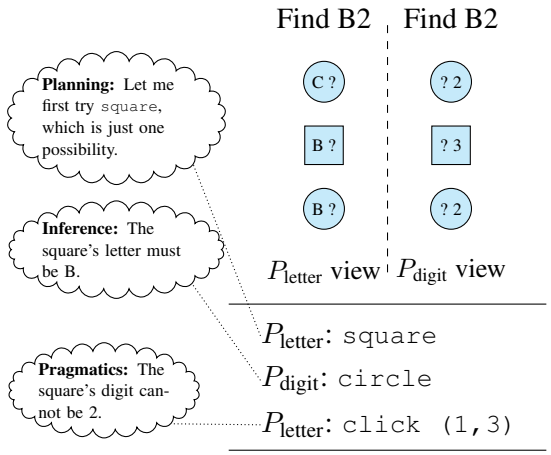


Figure 1: A game of InfoJigsaw played by two human players. One of the players (P_{letter}) only sees the letters, while the other one (P_{digit}) only sees the digits. Their goal is to identify the goal object, B2, by exchanging a few words. The clouds show the hypothesized role of planning, inference, and pragmatics in the players' choice of utterances. In this game, the bottom object is the goal (position (1, 3)).

There have been a few previous efforts in the language games literature to combine the three aspects. Hawkins et al. (2015) proposed a model of communication between a questioner and an answerer based on only one round of question answering. Vogel et al. (2013b) proposed a model of two agents playing a restricted version of the game from the Cards Corpus (Potts, 2012), where the agents only communicate once.[1] In this work, we seek to capture all three aspects in a single, unified framework which allows

[1] Specifically, two agents must both co-locate with a specific card. The agent which finds the card sooner shares the card location information with the other agent.

Transactions of the Association for Computational Linguistics, vol. 6, pp. 543–555, 2018. Action Editor: Mark Steedman.
Submission batch: 1/2018; Revision batch: 5/2018; 5/2018; Published 8/2018.

for multiple rounds of communication.

Specifically, we study human communication in a *sequential language game* in which two players, each with private knowledge, try to achieve a common goal by talking. We created a particular sequential language game called InfoJigsaw (Figure 1). In InfoJigsaw, there is a set of objects with public properties (shape, color, position) and private properties (digit, letter). One player (P_{letter}) can only see the letters, while the other player (P_{digit}) can only see the digits. The two players wish to identify the goal object, which is uniquely defined by a letter and digit. To do this, the players take turns talking; to encourage strategic language, we allow at most two English words at a time. At any point, a player can end the game by choosing an object.

Even in this relatively constrained game, we can see the three aspects of communication at work. As Figure 1 shows, in the first turn, since P_{letter} knows that the game is multi-turn, she simply says `square`; if the other player does not click on the square, she can try the bottom circle in the next turn (**planning**). In the second turn, P_{digit} infers from `square` that the square's letter is probably B (**inference**). As the digit on the square is not a 2, she says `circle`. Finally, P_{letter} infers that digits of circles are 2, and in addition she infers from `circle` that the digit on the square is not a 2 as otherwise, P_{digit} would have clicked on it (**pragmatics**). Therefore, she correctly clicks on (1,3).

In this paper, we propose a model that captures planning, inference, and pragmatics for sequential language games, which we call PIP. Planning recurrences look forward, inference recurrences look back, and pragmatics recurrences look to simpler interlocutors' model. The principal challenge is to integrate all three types in a coherent way; we present a "two-dimensional" system of recurrences to capture this. Our recurrences bottom out in very simple, literal semantics, (e.g., context-independent meaning of `circle`), and we rely on the structure of recurrences to endow words with their rich context-dependent meaning. As a result, our model is very parsimonious and only has four (hyper)parameters.

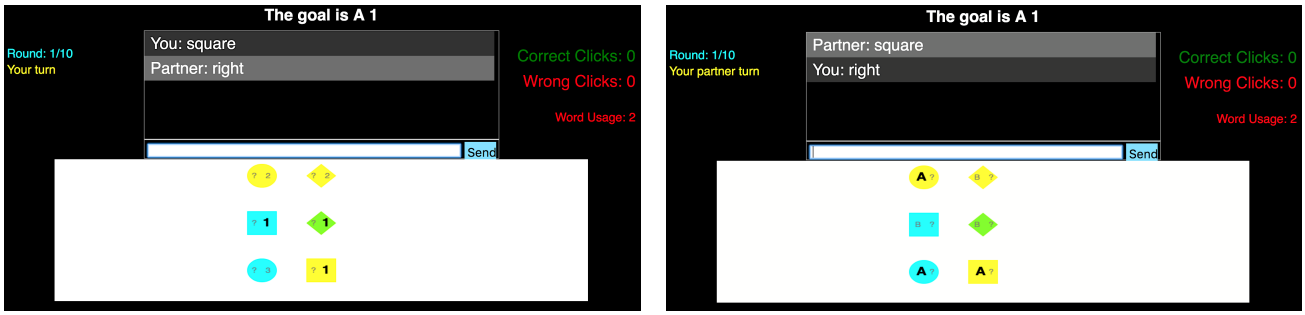
As our interest is in modeling human communication in sequential language games, we evaluate PIP on its ability to predict how humans play InfoJigsaw.[2] We paired up workers on Amazon Mechanical Turk to play InfoJigsaw, and collected a total of 1680 games. Our findings are as follows: (i) PIP obtains higher log-likelihood than a baseline that chooses actions to convey maximum information in each round; (ii) PIP obtains higher log-likelihood than ablations that remove the pragmatic or the planning components, supporting their importance in communication; (iii) PIP is better than an ablation with a truncated inference component that forgets the distant past only for longer games, but worse for shorter games. The overall conclusion is that by combining a very simple, context-independent literal semantics with an explicit model of planning, inference, and pragmatics, PIP obtains rich context-dependent meanings that correlate with human behavior.

2 Sequential Language Games

In a sequential language game, there are two players who have a shared world state w. In addition, each player $j \in \{+1, -1\}$ has a private state s_j. At each time step $t = 1, 2, \ldots$, the active player $j(t) = 2(t \bmod 2) - 1$ (which alternates) chooses an action (including speaking) a_t based on its policy $\pi_{j(t)}(a_t \mid w, s_{j(t)}, a_{1:t-1})$. Importantly that player $j(t)$ can see her own private state $s_{j(t)}$, but not the partner's $s_{-j(t)}$. At the end of the game (defined by a terminating action), both players receive utility $U(w, s_{+1}, s_{-1}, a_{1:t}) \in \mathbb{R}$. The utility consists of a penalty if players did not reach the goal and a reward if they reached the goal along with a penalty for each action. Because the players have a common utility function that depends on private information, they must communicate the part of their private information that is relevant for maximizing utility. In order to simplify notation, we use j to represent $j(t)$ in the rest of the paper.

InfoJigsaw. In InfoJigsaw (see Figure 1 for an example), two players try to identify a goal object, but each only has partial information about its identity. Thus, in order to solve the task, they must communicate, piecing their information together like a jigsaw

[2] One could in principle solve for an optimal communication strategy for InfoJigsaw, but this would likely result in a solution far from human communication.



(a) P_{digit} view

(b) P_{letter} view

Figure 2: Chat interface that Amazon Mechanical Turk (AMT) workers use to play InfoJigsaw (for readability, objects with the goal digit/letter are bolded).

puzzle. Figure 2 shows the interface that humans use to play the game.

More formally, the shared world state w includes the public properties of a set of objects: position on a $m \times n$ grid, color (blue, yellow, green), and shape (square, diamond, circle). In addition, w contains the letter and digit of the goal object (e.g., B2). The private state of player P_{digit} is a digit (e.g., 1,2,3) for each object, and the private state of player P_{letter} is a letter (e.g., A,B,C) for each object. These states are s_{+1}, s_{-1} depending on which player goes first.

On each turn t, a player $j(t)$'s action a_t can be either (i) a message containing one or two English words [3] (e.g., `circle`), or (ii) a click on an object, specified by its position (e.g., (1,3)). A click action terminates the game. If the clicked object is the goal, a green square will appear around it which is visible to both players; if the clicked object is not the goal, a red square appears instead. To discourage random guessing, we prevent players from clicking in the first time step. Players do not see an explicit utility (U); however, they are instructed to think strategically to choose messages that lead to clicking on the correct object while using a minimum number of messages. Players can see the number of correct clicks, wrong clicks, and number of the words they have sent to each other so far at the top right of the screen.

We would like to study how context-dependent meaning arises out of the interplay between a context-independent literal semantics with context-sensitive planning, inference, and pragmatics. The simplicity of the InfoJigsaw game ensures that this interplay is not obscured by other challenges.

[3] If the words are not inside the English dictionary, the sender receives an error and the message is rejected. This prevents players from circumventing the game rules by connecting multiple words without spaces.

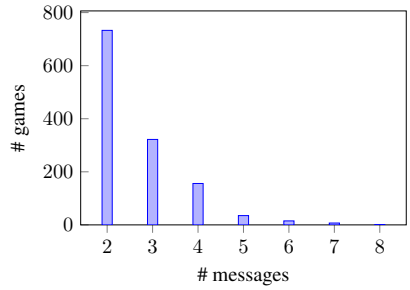
	# games	# messages	average score
All	1680	4967	7.50
Kept	1259	3358	7.48

Table 1: Statistics for all 1680 games and the 1259 games in which each message contains at least one of the 12 most frequent words or "yes", or "no".

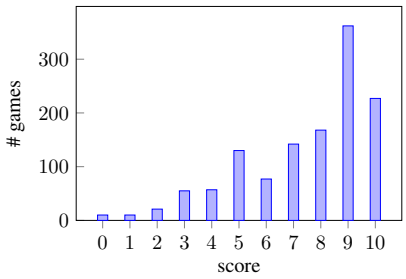
2.1 Data collection

We generated 10 InfoJigsaw scenarios as follows: For each one, we randomly choose m, n to be either 2×3 or 3×2 (which results in 64 possible private states). We randomly choose the properties of all objects and randomly designated one as the goal. We randomly choose either P_{letter} or P_{digit} to start the game first. Finally, to make the scenarios interesting, we keep a scenario if it satisfies: (i) Only the goal object (and no other objects) has the goal combination of the letter and digit; (ii) There exist at least two goal-consistent objects for each player and their sum of goal-consistent objects is at least $m \times n$; and (iii) all the goal consistent objects for each player do not share the same color, shape, or position (which means all the goal-consistent objects are not in left, right, top, bottom, or middle).

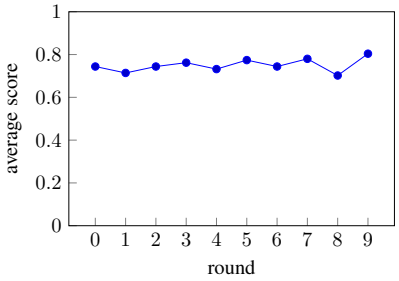
We collected a dataset of InfoJigsaw games on Amazon Mechanical Turk using the framework in Hawkins (2015) as follows: 200 pairs of players



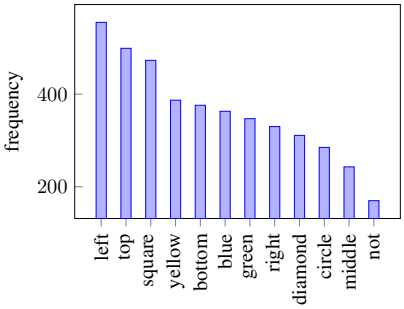
(a) Number of exchanged messages per game.



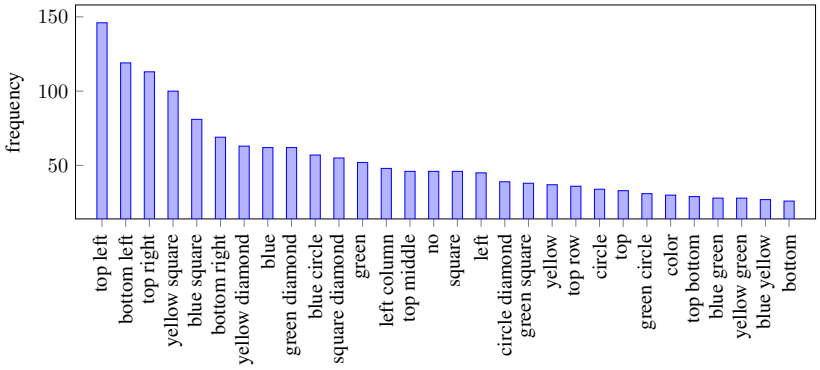
(b) Distribution of final game scores.



(c) Average score over multiple rounds of game play, which interestingly remains constant.



(d) 12 most frequent words, which make up 73% of all tokens.



(e) 30 most frequent messages, which make up 49% of all messages.

Figure 3: Statistics of the collected corpus.

each played all 10 scenarios in a random order. Out of 200 pairs, 32 pairs left the game prematurely which results in 168 pairs playing the total of 1680 games. Players performed 4967 actions (messages and clicks) total and obtained an average score (correct clicks) of 7.5 per game. The average score per scenario varied from 6.4 to 8.2. Interestingly, there is no significant difference in scores across the 10 scenarios, suggesting that players do not adapt and become more proficient with more game play (Figure 3c). Figure 3 shows the statistics of the collected corpus. Figure 4 shows one of the games, along with the distribution of messages in the first time step of all games played on this scenario.

To focus on the strategic aspects of InfoJigsaw, we filtered the dataset to reduce the words in the tail. Specifically, we keep a game if all its messages contain at least one of the 12 most frequent words (shown in Figure 3d) or "yes" or "no". For example, in Figure 4, the games containing messages such as `what color`, `mid row`, `color` are filtered because they don't contain any frequent words. Messages such as `middle`, `either middle`, `middle maybe`, `middle objects` are mapped to `middle`. 1259 of 1680 games survived. Table 1 compares the statistics between all games and the ones that were kept. Most games that were filtered out contained less frequent synonyms (e.g. `round` instead of `circle`). Some questions were filtered out too (e.g., `what color`). Filtered games are 1.15 times longer on average.

3 Literal Semantics

In order to understand the principles behind how humans perform planning, inference, and pragmatics, we aim to develop a parsimonious, interpretable model with few parameters rather than a highly expressive, data-driven model. Therefore, following the tradition of Rational Speech Acts (RSA) (Frank and Goodman, 2012; Goodman and Frank, 2016), we will define in this section a mapping from each word to its *literal semantics*, and rely on the PIP re-

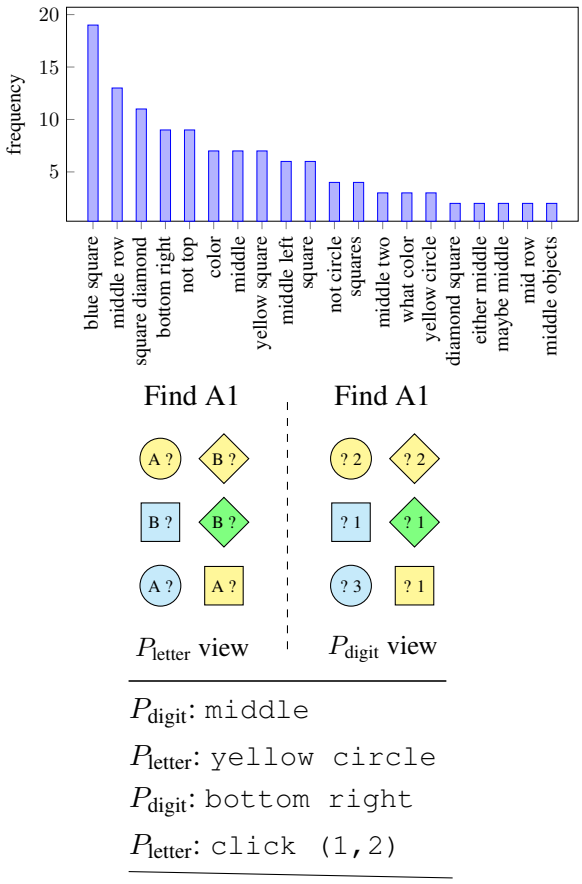
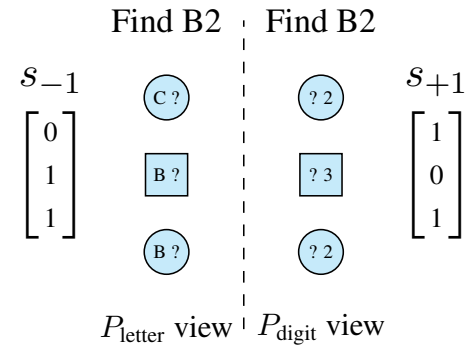


Figure 4: Bottom: one of the games played by Turkers. Top: the distribution of utterances on the first message. Players choose to explain their private state in different ways. Some use more general messages (e.g., `square diamond`), while some use more specific ones (e.g., `blue square`). Top diagram shows the first 20 most frequent messages on the first round (72% of all the messages).

currences (which we will describe in Section 4) to provide context-dependence. One could also learn the literal semantics by backpropagating through these recurrences, which has been done for simpler RSA models (Monroe and Potts, 2015); or learn the literal semantics from data and then put RSA on top (Andreas et al., 2016); we leave this to future work.

Suppose a player utters a single word `circle`. There are multiple possible context-dependent interpretations:

- Are any circles goal-consistent?
- All the circles are goal-consistent.
- Some circles but no other objects are goal-consistent.
- Most of the circles are goal-consistent.
- At least one circle is goal-consistent.



$$[\![\texttt{square}]\!] = \left\{ s : s \wedge \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

$$[\![\texttt{top bottom}]\!] = \left\{ s : s \wedge \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \vee \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

$$[\![\texttt{top blue}]\!] = \left\{ s : s \wedge \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \wedge \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right) \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

Figure 5: Private state of the players and meaning of two action sequences.

We will show that most of these interpretations can arise from a simple fixed semantics: roughly "some circles are goal consistent". We will now define a simple *literal semantics* of message actions such as `circle`, which forms the base case of PIP. Recall that the shared world state w contains the goal (e.g., B2) and, assuming P_{letter} goes first, the private state s_{-1} (s_{+1}) of player P_{letter} (P_{digit}) contains the letter (digit) of each object. For notational simplicity, let us define s_{-1} (s_{+1}) to be a matrix corresponding to the spatial locations of the objects, where an entry is 1 if the corresponding object has the goal letter (digit) and 0 otherwise. Thus s_j also represents the set of *goal-consistent* objects given the private knowledge of that player. Figure 5 shows the private states of the players.

We define two types of message actions: *informative* (e.g., `blue`, `top`) and *verifying* (e.g., `yes`, `no`). Informative messages have immediate meaning, while verifying messages depend on the previous utterance.

Informative messages. Informative messages describe constraints on the speaker's private state (which the partner does not know). For a message a,

define $[\![a]\!]$ to be the set of consistent private states. For example, $[\![\texttt{bottom}]\!]$ is all private states where there are goal-consistent objects in the bottom row.

Formally, for each word x that specifies some object property (e.g., `blue`, `top`), define v_x to be an $n \times m$ matrix where an entry is 1 if the corresponding object has the property x, and 0 otherwise. Then, define the literal semantics of a single-word message x to be $[\![x]\!] \stackrel{\text{def}}{=} \{s : s \wedge v_x \neq 0\}$, where $\wedge$ denotes element-wise *and* and 0 denotes the zero matrix. That is, single-property messages can be glossed as "some goal-consistent object has property x".

For a two-word message xy, we define the literal semantics depending on the relationship between x and y. If x and y are mutually exclusive, then we interpret xy as x *or* y (e.g., `square circle`); otherwise, we interpret xy as x *and* y (e.g., `blue top`). Formally, $[\![xy]\!] \stackrel{\text{def}}{=} \{s : s \wedge (v_x \wedge v_y) \neq 0\}$ if $v_x \wedge v_y \neq 0$ and $\{s : s \wedge (v_x \vee v_y) \neq 0\}$ otherwise. See Figure 5 for some examples.

Action sequences. We now define the literal semantics of an entire action sequence $[\![a_{1:t}]\!]_j$ with respect to player j, which is the set of possible partner private states s_{-j}. Intuitively, we want to simply intersect the set of consistent private states of the informative messages, but we need to also handle verifying messages (`yes` and `no`), which are context-dependent. Formally, we say that private state $s_{-j} \in [\![a_{1:t}]\!]_j$ if the following holds: for all informative messages a_i uttered by $-j$, $s_{-j} \in [\![a_i]\!]$; and for all verifying messages a_i uttered by $-j$ if $a_i = \texttt{yes}$ then, $s_{-j} \in [\![a_{i-1}]\!]$; and if $a_i = \texttt{no}$ then, $s_{-j} \notin [\![a_{i-1}]\!]$.

4 The Planning-Inference-Pragmatics (PIP) Model

Why does P_{digit} in Figure 1 choose `circle` rather than `top` or `click(1,2)`? Intuitively, when a player chooses an action, she should take into account her previous actions, her partner's actions, and the effect of her actions on future turns. She should do all these while reasoning pragmatically that her partner is also a strategic player.

At a high-level, PIP defines a system of recurrences revolving around three concepts, depicted in Figure 6: player j's beliefs over the partner's private state $p_j^k(s_{-j} \mid s_j, a_{1:t})$, her expected utility of the game $V_j^k(s_{+1}, s_{-1}, a_{1:t})$, and her policy $\pi_j^k(a_t \mid s_j, a_{1:t-1})$. Here, t indexes the current time and k indexes the depth of pragmatic recursion, which will be explained later in Section 4.3. To simplify the notation, we have dropped w (shared world state) from the notation, since everything conditions on it.

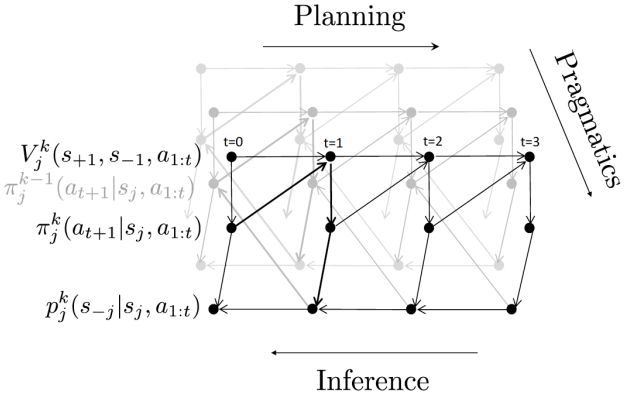


Figure 6: PIP is defined via a system of recurrences that simultaneously captures planning, inference, and pragmatics. The arrows show the dependencies between beliefs p, expected utilities V, and policy π.

4.1 Inference

From player j's point of view, the purpose of inference is to compute a distribution over the partner's private state s_{-j} given all actions thus far $a_{1:t}$. We first consider a "level 0" player, which simply assigns a uniform distribution over all states consistent with the literal semantics of $a_{1:t}$, which we defined in Section 3:

$$p_j^0(s_{-j} \mid s_j, a_{1:t}) \propto \begin{cases} 1 & s_{-j} \in [\![a_{1:t}]\!]_j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

For example, Figure 7, shows the P_{letter}'s belief about P_{digit}'s private state after observing `circle`. Remember we show the private state of the players as a matrix where an entry is 1 if the corresponding object has the goal letter (digit) and 0 otherwise.

A player's own private state s_j can also constrain her beliefs about her partner's private state s_{-j}. For example, in InfoJigsaw, the active player knows there is a goal, and so we set $p_j^k(s_{-j} \mid s_j, a_{1:t}) = 0$ if $s_{-j} \wedge s_j = 0$.

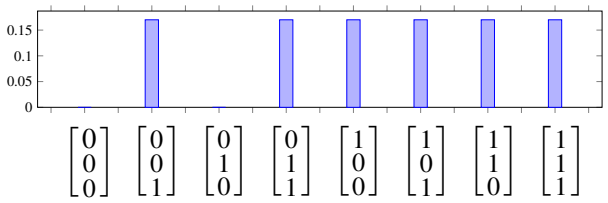


Figure 7: P_{letter}'s probability distribution over P_{digit}'s private state after P_{digit} says `circle` in the game shown in Figure 5.

4.2 Planning

The purpose of planning is to compute a policy π_j^k, which specifies a distribution over player j's actions a_t given all past actions $a_{1:t-1}$. To construct the policy, we first define an expected utility V_j^k via the following forward-looking recurrence: When the game is over (e.g., in InfoJigsaw, one player clicks on an object), the expected utility of the dialogue is simply its utility as defined by the game:

$$V_j^k(s_{+1}, s_{-1}, a_{1:t}) = U(s_{+1}, s_{-1}, a_{1:t}). \quad (2)$$

Otherwise, we compute the expected utility assuming that in the next turn, player j chooses action a_{t+1} with probability governed by her policy $\pi_j^k(a_{t+1} \mid s_j, a_{1:t})$:

$$V_j^k(s_{+1}, s_{-1}, a_{1:t}) = \sum_{a_{t+1}} \pi_j^k(a_{t+1} \mid s_j, a_{1:t}) \; V_{-j}^k(s_{-1}, s_{+1}, a_{1:t+1}). \quad (3)$$

Having defined the expected utility, we now define the policy. First, let D_j^k be the gain in expected utility $V_{-j}^k(s_{+1}, s_{-1}, a_{1:t})$ over a simple baseline policy that ends the game immediately, yielding utility $U(s_{+1}, s_{-1}, a_{1:t-1})$ (which is simply a penalty for not finding the correct goal and a penalty for each action). Of course, the partner's private state s_{-j} is unknown and must be marginalized out based on player j's beliefs; let E_j^k be the expected gain. Let the probability of an action a_t be proportional to $\max(0, E_j^k)^\alpha$, where $\alpha \in [0, \infty)$ is a hyperparameter that controls the rationality of the agent (a larger α means that the player chooses utility-maximizing actions more aggressively). Formally:

$$D_j^k = V_{-j}^k(s_{+1}, s_{-1}, a_{1:t}) - U(s_{+1}, s_{-1}, a_{1:t-1}),$$
$$E_j^k = \sum_{s_{-j}} p_j^k(s_{-j} \mid s_j, a_{1:t-1}) D_j^k,$$
$$\pi_j^k(a_t \mid s_j, a_{1:t-1}) \propto \max\left(0, E_j^k\right)^\alpha. \quad (4)$$

In practice, we use a depth-limited recurrence, where the expected utility is computed assuming that the game will end in f turns and the last action is a click action (meaning that we only consider the action sequences with size $\leq f$ and a clicking action as the last action). Figure 8 shows how P_{digit} computes the expected gain (Eqn. 4) of saying `circle`.

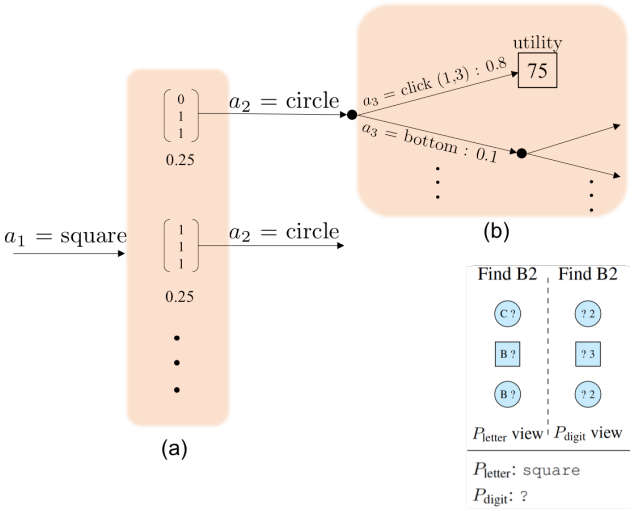


Figure 8: Planning reasoning for the game in Figure 1 (reproduced here in the bottom right). (a) In order to calculate the expected gain (E) of generating `circle`, for every state s, P_{digit} computes the probability of s being the P_{letter}'s private state. (b) She then computes the expected utility (V) if she generates `circle` assuming P_{letter}'s private state is s.

4.3 Pragmatics

The purpose of pragmatics is to take into account the partner's strategizing. We do this by constructing a level-k player that infers the partner's private state, following the tradition of Rational Speech Acts (RSA) (Frank and Goodman, 2012; Goodman and Frank, 2016). Recall that a level-0 player p_j^0 (Section 4.1) puts a uniform distribution over all the

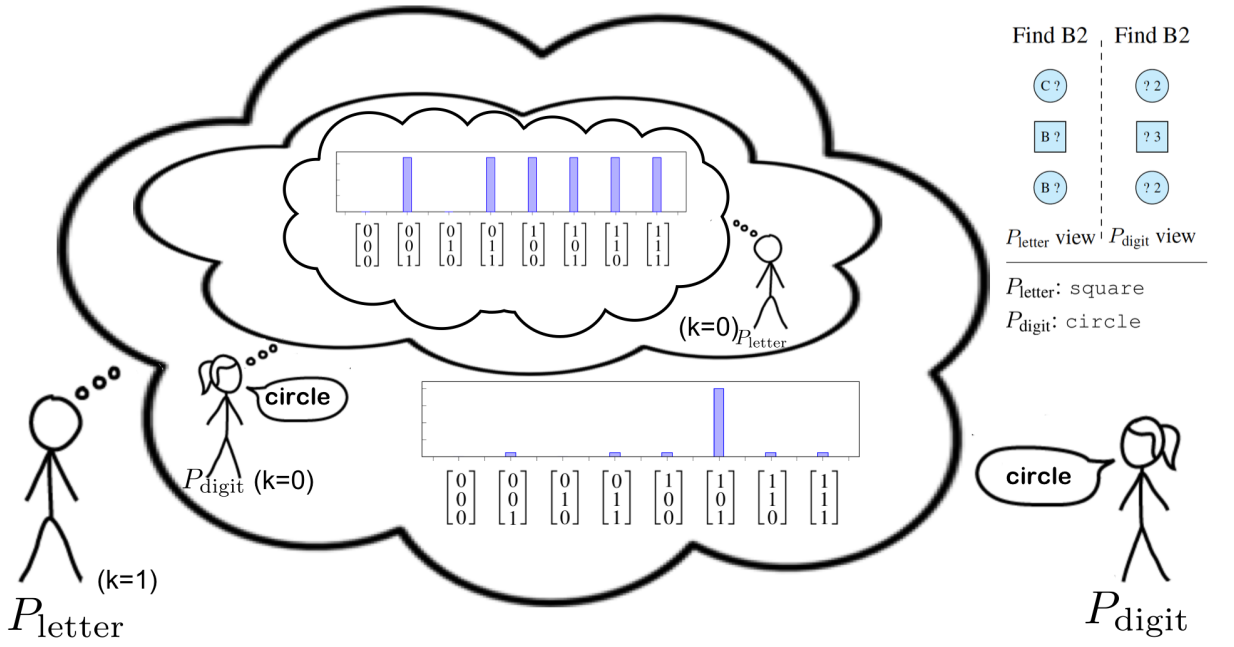


Figure 9: Pragmatic reasoning for the game in Figure 1 (reproduced here in the upper right) at time step 3. Players reason recursively about each others beliefs: the level-0 player puts a uniform distribution p_j^0 over all the states in which at least one circle is goal-consistent independent of the shared world state and previous actions. The level-1 player assigns probability over the partner's private states s_{-j} proportional to the probability that she would have performed the last action given that state s_{-j}. For example, if $\begin{bmatrix}0\\0\\1\end{bmatrix}$ were P_{digit}'s private state, then saying `bottom` would be more probable (given the shared world state); if $\begin{bmatrix}1\\1\\1\end{bmatrix}$ were P_{digit}'s state, then clicking on the square would be a better option (given the previous actions). But given that P_{digit} uttered `circle`, $\begin{bmatrix}1\\0\\1\end{bmatrix}$ is most likely, as reflected by p_j^1.

semantically valid private states of the partner. A level-k player assigns probability over the partner's private state proportional to the probability that a level-$(k-1)$ player would have performed the last action a_t:

$$p_j^k(s_{-j} \mid s_j, a_{1:t}) \propto \pi_{-j}^{k-1}(a_t \mid s_{-j}, a_{1:t-1}) \\ p_j^k(s_{-j} \mid s_j, a_{1:t-2}). \quad (5)$$

Figure 9 shows an example of the pragmatic reasoning.

4.4 A closer look at the meaning of actions

In the Section 4.2, we modeled the players as rational agents that choose actions that lead to higher gain utility. In the pragmatics section (Section 4.3), we described how a player infers the partner's private state taking into account that her partner is acting cooperatively. The phenomena that emerges from the combination of the two is the topic of this section.

We first define the belief marginals B_j of a player j to be the marginal probabilities that each object is goal-consistent under the hypothesized partner's private state $s_{-j} \in \mathbb{R}^{m \times n}$, conditioned on actions $a_{1:t}$:

$$B_j(s_j, a_{1:t}) = \sum_{s_{-j}} p_j^k(s_{-j} \mid s_j, a_{1:t}) s_{-j}. \quad (6)$$

At time $t = 0$ (before any actions), the belief marginals of both players are $m \times n$ matrices with 0.5 in all entries. The change in a belief marginal after observing an action a_t gives a sense of the effective (context-dependent) meaning of that action.

We first explain how pragmatics ($k > 0$ in (Eqn. 5)) leads to rich action meanings. When a player observes her partner's action a_t, she assumes this ac-

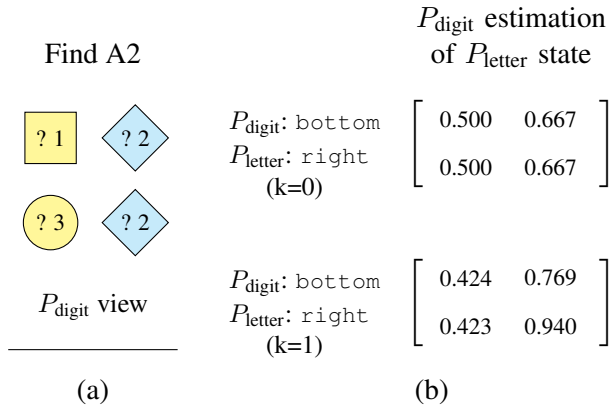


Figure 10: Belief marginals of P_{digit} (Eqn. 6) after observing sequences of actions for different pragmatic depths k. (b) Without pragmatics ($k = 0$), P_{digit} thinks both objects on the right has the same probability to be goal-consistent. With pragmatics ($k = 1$), P_{digit} thinks that the object in the bottom right is more likely to be goal-consistent.

tion was chosen because it results in a higher utility than the alternatives. In other words, she infers that her partner's private state cannot be one in which a_t does not lead to high utility. As an example, saying `circle` instead of `top circle` or `bottom circle` implies that there is more than one goal-consistent circle. The pragmatic depth k governs the extent to which this type of reasoning is applied.

Recall in Section 4.2, a player chooses an action conditioned on all previous actions, and the other player assumed this context-dependence. As an example, Figure 10(d) shows how `right` changes its meaning when it follows `bottom`.

5 Experiments

5.1 Setup

We *a priori* set the reward of clicking on the goal to be $+100$ and clicking on the wrong object to be -100. We set the smoothing $\alpha = 10$ and the action cost to be -50 based on the data. The larger the action cost, the fewer messages will be used before selecting an object. Formally, after k actions:

$$\text{Utility} = -50k + \begin{cases} +100 & \text{the goal object is clicked,} \\ -100 & \text{otherwise.} \end{cases} \tag{7}$$

We smoothed all polices by adding 0.01 to the probability of each action and re-normalizing. By default, we set $k = 1$ (pragmatic depth (Eqn. 4)). When computing the expected utility (Eqn. 3) of the game, we use a lookahead of $f = 2$. Inference looks back b time steps (i.e., (Eqn. 1) and (Eqn. 5) are based on $a_{t-b+1:t}$ rather than $a_{1:t}$); we set $b = \infty$ by default.

We implemented two baseline policies:

Random policy: for player j, the random policy randomly chooses one of the semantically valid (Section 3) actions with respect to s_j or clicks on a goal-consistent object. Formally, the random policy places a uniform distribution over:

$$\{a : s_j \in [\![a]\!]\} \cup \{\texttt{click}(u, v) : (s_j)_{u,v} = 1\}. \tag{8}$$

Greedy Policy: assigns higher probability to the actions that convey more information about the player's private state. We heuristically set the probability of generating an action proportional to how much it shrinks the set of semantically valid states. Formally, for the message actions:

$$\pi_j^{\text{msg}}(a_t \mid a_{1:t-1}, s_j) \propto |[\![a_{1:t-1}]\!]_{-j}| - |[\![a_{1:t}]\!]_{-j}| \tag{9}$$

For the clicking actions, we compute the belief state as explained in Section 4.4. Remember $B_{u,v}$ is the marginal probability of the object in the row u and column v being goal-consistent in the partner's private state. Formally, for clicking actions:

$$\pi_j^{\text{click}}(\texttt{click}(u, v) \mid a_{1:t}, s_j) \propto \min((s_j)_{u,v}, B_j(s_j, a_{1:t})_{u,v}). \tag{10}$$

Finally, the greedy policy chooses a click action with probability γ and a message action with probability $1 - \gamma$. So that γ increases as the player gets more confident about the position of the goal, we set γ to be the probability of the most probable position of the goal: $\gamma = \max_{u,v} \pi_j^{\text{click}}(\texttt{click}(u, v) \mid a_{1:t}, s_j)$.

5.2 Results

Figure 11 compares the two baselines with PIP on the task of predicting human behavior as measured by log-likelihood.[4] To estimate the best possible

[4]We bootstrap the data 1000 times and we show 90% confidence intervals.

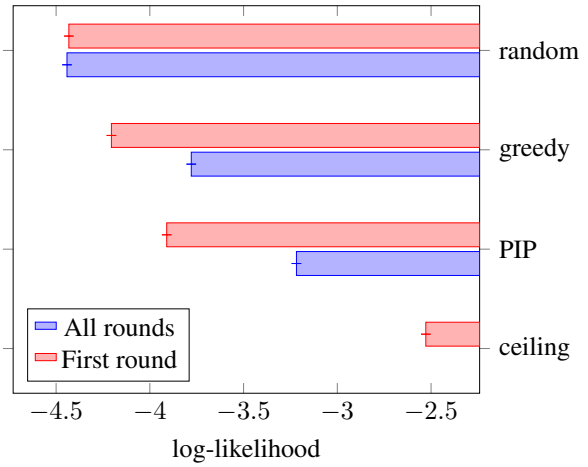


Figure 11: Average log-likelihood across messages. (a) Performance of PIP and baselines on all time steps. (b) Performance of PIP and baselines on only the first time step along with the ceiling given by the entropy of the human data. The error bars show 90% confidence intervals.

(i.e., ceiling) performance, we compute the entropy of the actions on the first time step based on approximately 100 data points per scenario. For each policy, we rank the actions by their probability in decreasing order (actions with the same probability are randomly ordered), and then compute the average ranking across actions according to the different policies; see Figure 13 for the results.

To assess the different components (planning, inference, pragmatics) of PIP, we run PIP, ablating one component at a time from the default setting of $k = 1$, $f = 2$, and $b = \infty$ (see Figure 12).

Pragmatics. Let PIP$_{\text{-prag}}$ be PIP but with a pragmatic depth (Eqn. 4) of $k = 0$ rather than $k = 1$, which means that PIP$_{\text{-prag}}$ only draws inferences based on the literal semantics of messages. PIP$_{\text{-prag}}$ loses 0.21 in average log-likelihood per action, highlighting the importance of pragmatics in modeling human behavior.

Planning. Let PIP$_{\text{-plan}}$ be PIP, but looking ahead only $f = 1$ step when computing the expected utility (Eqn. 3) rather than $f = 2$. With a shorter future horizon, PIP$_{\text{-plan}}$ tries to give as much information as possible at each turn, whereas human players tend to give information about their state incrementally. PIP$_{\text{-plan}}$ cannot capture this behavior and allocates low probability to these kinds of dialogue. PIP$_{\text{-plan}}$ has an average log-likelihood which is 0.05 lower than that of PIP, highlighting the importance of planning.

Inference. Let PIP$_{\text{-infer}}$ be PIP, but only looking at the last utterance ($b = 1$) rather than the full history ($b = \infty$). The results here are more nuanced. Although PIP$_{\text{-infer}}$ actually performs better than PIP on all games, we find that PIP$_{\text{-infer}}$ is worse than PIP by an average log-likelihood of 0.15 in predicting messages after time step 3, highlighting the importance of inference, but only in long games. It is likely that additional noise involved in the inference process leads to the decreased performance when backward looking inference is not actually needed.

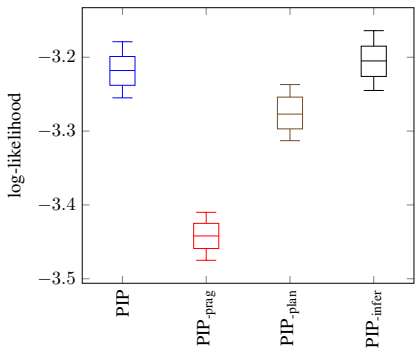
6 Related Work and Discussion

Our work touches on ideas in game theory, pragmatic modeling, dialogue modeling, and learning communicative agents, which we highlight below.

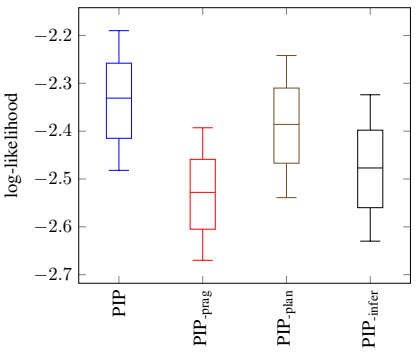
Game theory. According to game theory terminology (Shoham and Leyton-Brown, 2008), InfoJigsaw is a non-cooperative (there is no offline optimization of the player's policy before the game starts), common-payoff (the players have the same utility), incomplete information (the players have private state) game with the sequential actions. One related concept in game theory related to our model is rationalizability (Bernheim, 1984; Pearce, 1984). A strategy is rationalizable if it is justifiable to play against a completely rational player. Another related concept is epistemic games (Dekel and Siniscalchi, 2015; Perea, 2012). Epistemic game theory studies the behavioral implications of rationality and mutual beliefs in games.

It is important to note that we are not interested in notions of global optima or equilibria; rather, we are interested in modeling human behavior. Restricting words to a very restricted natural language has been studied in the context of language games (Wittgenstein, 1953; Lewis, 2008; Nowak et al., 1999; Franke, 2009; Huttegger et al., 2010).

Rational speech acts. The pragmatic component of PIP is based on Rational Speech Act framework (Frank and Goodman, 2012; Golland et al., 2010),



(a) Performance over all games and all rounds.



(b) Performance over messages after round 3.

	PIP	PIP$_{\text{-prag}}$	PIP$_{\text{-plan}}$	PIP$_{\text{-infer}}$
k (pragmatics)	1	0	1	1
f (planning)	2	2	1	2
b (inference)	∞	∞	∞	1
rank all	17.1	19.3	17.2	16.9
rank ≥ 3	10.4	10.8	11.6	13.1

(c) Top: parameter setup. Bottom: expected ranking of human messages according to the different ablations

Figure 12: Performance on ablations of PIP. Average log-likelihood per message, the whiskers show 90% confidence intervals. PIP has better performance of ablation of planning and pragmatics over all rounds. Looking only one step backward has a better performance in the first few rounds but it is worse after round 3.

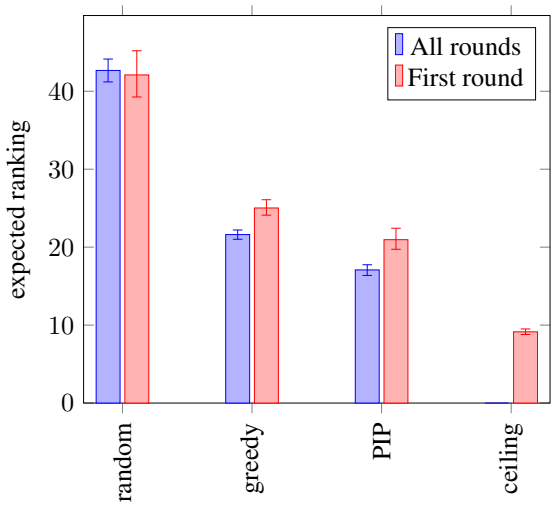


Figure 13: Expected ranking of the human messages according to different policies. Error bars show 90% confidence intervals.

which defines recurrences capturing how one agent reasons about another. Similar ideas were explored in the precursor work of Golland et al. (2010), and much work has ensued (Smith et al., 2013; Qing and Franke, 2014; Monroe and Potts, 2015; Ullman et al., 2016; Andreas and Klein, 2016).

Most of this work is restricted to production and comprehension of a single utterance. Hawkins et al. (2015) extend these ideas to two utterances (a question and an answer). Vogel et al. (2013b) integrates planning with pragmatics using decentralized partially observable Markov processes (DEC-POMDPs). In their task, two bots should find and co-locate with a specific card. In contrast to InfoJigsaw, their task can be completed without communication; their agents only communicate once sharing the card location. They also only study artificial agents playing together and were not concerned about modeling human behavior.

Learning to communicate. There is a rich literature on multi-agent reinforcement learning (Busoniu et al., 2008). Some works assume full visibility and cooperate without communication, assuming the world is completely visible to all agents (Lauer and Riedmiller, 2000; Littman, 2001); others assume a predefined convention for communication (Zhang and Lesser, 2013; Tan, 1993). There is also some work that learns the convention itself (Foerster et al., 2016; Sukhbaatar et al., 2016; Lazaridou et al., 2017; Mordatch and Abbeel, 2018). Lazaridou et al. (2017) puts humans in the loop to make the communication more human-interpretable. In comparison to these works, we seek to predict human behavior instead of modeling artificial agents that communicate with each other.

Dialogue. There is also a lot of work in computational linguistics and NLP on modeling dialogue. Allen and Perrault (1980) provides a model that in-

fers the intention/plan of the other agent and uses this plan to generate a response. Clark and Brennan (1991) explains how two players update their common ground (mutual knowledge, mutual beliefs, and mutual assumptions) in order to coordinate. Recent work in task-oriented dialogue uses POMDPs and end-to-end neural networks (Young, 2000; Young et al., 2013; Wen et al., 2017; He et al., 2017). In this work, instead of learning from a large corpus, we predict human behavior without learning, albeit in a much more strategic, stylized setting (two words per utterance).

7 Conclusion

In this paper, we started with the observation that humans use language in a very contextual way driven by their goals. We identified three salient aspects—planning, inference, pragmatics—and proposed a unified model, PIP, that captures all three aspects simultaneously. Our main result is that a very simple, context-independent literal semantics can give rise via the recurrences to rich phenomena. We study these phenomena in a new game, InfoJigsaw, and show that PIP is able to capture human behavior.

Reproducibility

All code, data, and experiments for this paper are available on the CodaLab platform at `https://worksheets.codalab.org/worksheets/0x052129c7afa9498481185b553d23f0f9/`.

Acknowledgments

We would like to thank the anonymous reviewers and the action editor for their helpful comments. We also thank Will Monroe for providing valuable feedback on early drafts.

References

James F. Allen and C. Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143–178.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1182.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *Association for Computational Linguistics (ACL)*, pages 1545–1554.

B. Douglas Bernheim. 1984. Rationalizable strategic behavior. *Econometrica: Journal of the Econometric Society*, pages 1007–1028.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 38(2):156–172.

Herbert H. Clark and Susan E. Brennan. 1991. *Grounding in Communication*. Perspectives on Socially Shared Cognition.

Eddie Dekel and Marciano Siniscalchi. 2015. *Epistemic game theory*, volume 4. Handbook of Game Theory with Economic Applications.

Jakob Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2137–2145.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336:998–998.

Michael Franke. 2009. *Signal to act: Game theory in pragmatics*. Institute for Logic, Language and Computation.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 410–419.

Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818–829.

Herbert P. Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3:41–58.

Robert X. D. Hawkins, Andreas Stuhlmüller, Judith Degen, and Noah D. Goodman. 2015. Why do you ask? Good questions provoke informative answers. In *Proceedings of the Thirty-Seventh Annual Conference of the Cognitive Science Society*.

Robert X. D. Hawkins. 2015. Conducting real-time multiplayer experiments on the web. *Behavior Research Methods*, 47(4):966–976.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Association for Computational Linguistics (ACL)*, pages 1766–1776.

Simon M. Huttegger, Brian Skyrms, Rory Smead, and Kevin J.S. Zollman. 2010. Evolutionary dynamics of Lewis signaling games: Signaling systems vs. partial pooling. *Synthese*, 172(1):177–191.

Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics (TACL)*, 1:193–206.

Martin Lauer and Martin Riedmiller. 2000. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *International Conference on Machine Learning (ICML)*, pages 535–542.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations (ICLR)*.

David Lewis. 2008. *Convention: A philosophical study*. John Wiley & Sons.

Michael L. Littman. 2001. Value-function reinforcement learning in Markov games. *Cognitive Systems Research*, 2(1):55–66.

Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *International Conference on Machine Learning (ICML)*, pages 1671–1678.

Will Monroe and Christopher Potts. 2015. Learning in the Rational Speech Acts model. In *Proceedings of 20th Amsterdam Colloquium*.

Richard Montague. 1973. The proper treatment of quantification in ordinary English. In *Approaches to Natural Language*, pages 221–242.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Martin A. Nowak, Joshua B. Plotkin, and David C. Krakauer. 1999. The evolutionary language game. *Journal of Theoretical Biology*, 200(2):147–162.

David G. Pearce. 1984. Rationalizable strategic behavior and the problem of perfection. *Econometrica: Journal of the Econometric Society*, pages 1029–1050.

Andr'es Perea. 2012. *Epistemic game theory: reasoning and choice*. Cambridge University Press.

Christopher Potts. 2012. Goal-driven answers in the Cards dialogue corpus. In *Proceedings of the 30th West Coast Conference on Formal Linguistics*, pages 1–20.

Ciyang Qing and Michael Franke. 2014. Gradable adjectives, vagueness, and optimal language use: A speaker-oriented model. In *Semantics and Linguistic Theory*, volume 24, pages 23–41.

Yoav Shoham and Kevin Leyton-Brown. 2008. *Multi-agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Nathaniel J. Smith, Noah D. Goodman, and Michael C. Frank. 2013. Learning and using language via recursive pragmatic reasoning about other agents. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3039–3047.

Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2244–2252.

Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *International Conference on Machine Learning (ICML)*, pages 330–337.

Tomer D. Ullman, Yang Xu, and Noah D. Goodman. 2016. The pragmatics of spatial language. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*.

Adam Vogel, Max Bodoia, Christopher Potts, and Daniel Jurafsky. 2013a. Emergence of Gricean maxims from multi-agent decision theory. In *North American Association for Computational Linguistics (NAACL)*, pages 1072–1081.

Adam Vogel, Christopher Potts, and Dan Jurafsky. 2013b. Implicatures and nested beliefs in approximate decentralized-POMDPs. In *Association for Computational Linguistics (ACL)*, pages 74–80.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *European Association for Computational Linguistics (EACL)*, pages 438–449.

Ludwig Wittgenstein. 1953. *Philosophical Investigations*. Blackwell, Oxford.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. In *Proceedings of the IEEE*, number 5, pages 1160–1179.

Steve J. Young. 2000. Probabilistic methods in spoken-dialogue systems. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 358(1769):1389–1402.

Chongjie Zhang and Victor Lesser. 2013. Coordinating multi-agent reinforcement learning with limited communication. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 1101–1108.