Emergence of Gricean Maxims from Multi-Agent Decision Theory

Adam Vogel, Max Bodoia, Christopher Potts, and **Dan Jurafsky**
Stanford University
Stanford, CA, USA
{acvogel,mbodoia,cgpotts,jurafsky}@stanford.edu

Abstract

Grice characterized communication in terms of the *cooperative principle*, which enjoins speakers to make only contributions that serve the evolving conversational goals. We show that the cooperative principle and the associated maxims of relevance, quality, and quantity emerge from multi-agent decision theory. We utilize the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) model of multi-agent decision making which relies only on basic definitions of rationality and the ability of agents to reason about each other's beliefs in maximizing joint utility. Our model uses cognitively-inspired heuristics to simplify the otherwise intractable task of reasoning jointly about actions, the environment, and the nested beliefs of other actors. Our experiments on a cooperative language task show that reasoning about others' belief states, and the resulting emergent Gricean communicative behavior, leads to significantly improved task performance.

1 Introduction

Grice (1975) famously characterized communication among rational agents in terms of an overarching *cooperative principle* and a set of more specific maxims, which enjoin speakers to make contributions that are truthful, informative, relevant, clear, and concise. Since then, there have been many attempts to derive the maxims (or perhaps just their effects) from more basic cognitive principles concerning how people make decisions, formulate plans, and collaborate to achieve goals. This research traces to early work by Lewis (1969) on signaling systems. It has recently been the subject of extensive theoretical discussion (Clark, 1996; Merin, 1997; Blutner, 1998; Parikh, 2001; Beaver, 2002; van Rooy, 2003; Benz et al., 2005; Franke, 2009) and has been tested experimentally using one-step games in which the speaker produces a message and the hearer ventures a guess as to its intended referent (Rosenberg and Cohen, 1964; Dale and Reiter, 1995; Golland et al., 2010; Stiller et al., 2011; Frank and Goodman, 2012; Krahmer and van Deemter, 2012; Degen and Franke, 2012; Rohde et al., 2012).

To date, however, these theoretical models and experiments have not been extended to multi-step interactions extending over time and involving both language and action together, which leaves this work relatively disconnected from research on planning and goal-orientation in artificial agents (Perrault and Allen, 1980; Allen, 1991; Grosz and Sidner, 1986; Bratman, 1987; Hobbs et al., 1993; Allen et al., 2007; DeVault et al., 2005; Stone et al., 2007; DeVault, 2008). We attribute this in large part to the complexity of Gricean reasoning itself, which requires agents to model each other's belief states. Tracking these as they evolve over time in response to experiences is extremely demanding. Our approach complements slot-filling dialog systems, where the focus is on managing speech recognition uncertainty (Young et al., 2010; Thomson and Young, 2010).

However, recent years have seen significant advances in multi-agent decision-theoretic models and their efficient implementation. With the current paper, we seek to show that the Decentralized Par-

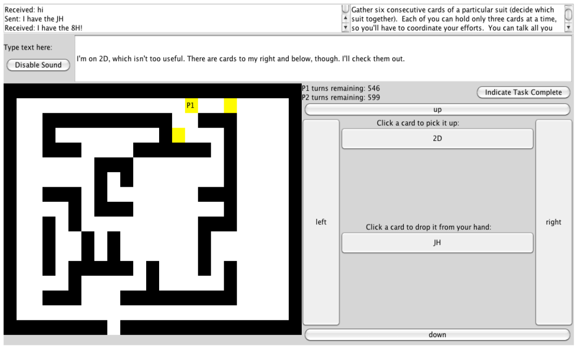
Proceedings of NAACL-HLT 2013, pages 1072–1081,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

tially Observable Markov Decision Process (Dec-POMDP) provides a robust, flexible foundation for implementing agents that communicate in a Gricean manner. Dec-POMDPs are multi-agent, partially-observable models in which agents maintain belief distributions over the underlying, hidden world state, including the beliefs of the other players, and speech actions change those beliefs. In this setting, informative, relevant communication emerges as the best way to maximize joint utility.

The complexity of pragmatic reasoning is still forbidding, though. Correspondingly, optimal decision making in Dec-POMDPs is NEXP complete (Bernstein et al., 2002). To manage this issue, we introduce several cognitively-plausible approximations which allow us to simplify the Dec-POMDP to a single-agent POMDP, for which relatively efficient solvers exist (Spaan and Vlassis, 2005). We demonstrate our algorithms on a variation of the Cards task, a partially-observable collaborative search problem (Potts, 2012). Spatial language comprises the bulk of communication in the Cards task, and we discuss a model of spatial semantics in Section 3. Using this task and a model of the meaning of spatial language, we next discuss two agents that play the game: *ListenerBot* (Section 4) makes decisions using a single-agent POMDP that does not take into account the beliefs or actions of its partner, whereas *DialogBot* (Section 5) maintains a model of its partner's beliefs. As a result of the cooperative structure of the underlying model and the effects of communication within it, DialogBot's contributions are relevant, truthful, and informative, which leads to significantly improved task performance.

2 The Cards Task and Corpus

The Cards corpus consists of 1,266 transcripts[1] from an online, two-person collaborative game in which two players explore a maze-like environment, communicating with each other via a text chat window (Figure 1). A deck of playing cards has been distributed randomly around the environment, and the players' task is to find six consecutive cards of the same suit. Our implemented agents solve a simplified version of this task in which the two agents must both end up co-located with a single card, the Ace of Spades (AS). This is much simpler than the six-card version from the human–human corpus, but it involves the same kind of collaborative goal and forces our agents to deal with the same kind of partial knowledge about the world as the humans did. Each agent knows its own location, but not his partner's, and a player can see the AS only when co-located with it. The agents use (simplified) English to communicate with each other.



Received: hi
Sent: I have the JH
Received: I have the 8H!

Gather six consecutive cards of a particular suit (decide which suit together). Each of you can hold only three cards at a time, so you'll have to coordinate your efforts. You can talk all you

I'm on 2D, which isn't too useful. There are cards to my right and below, though. I'll check them out.

P1 turns remaining: 546
P2 turns remaining: 599

Figure 1: The Cards corpus gameboard. Player 1's location is marked "P1". The nearby yellow boxes mark card locations. The dialogue history and chat window are at the top. This board, the one we use throughout, consists of 231 open grid squares.

3 Spatial Semantics

Much of the communication in the Cards task involves referring to spatial locations on the board. Accordingly, we focus on spatial language for our artificial agents. In this section, we present a model of spatial semantics, which we create by leveraging the human–human Cards transcripts. We discuss the spatial semantic representation, how we classify the semantics of new locative expressions, and our use of spatial semantics to form a high-level state space for decision making.

3.1 Semantic Representation

Potts (2012) released annotations, derived from the Cards corpus, which reduce 599 of the players' statements about their locations to formulae of the form $\delta(\varphi_1 \wedge \cdots \wedge \varphi_k)$, where δ is a *domain* and $\varphi_1, \ldots, \varphi_k$ are semantic literals. For example, the utterance "(I'm) at the top right of the board" is annotated as BOARD(top $\wedge$ right), and "(I'm) in bottom

[1] Released by Potts (2012) at http://cardscorpus.christopherpotts.net

of the C room" is annotated as C_room(bottom). Table 1 lists the full set of semantic primitives that appear as domain expressions and literals.

Because the Cards transcripts are so highly structured, we can interpret these expressions in terms of the Cards world itself. For a given formula $\sigma = \delta(\varphi_1 \wedge \cdots \wedge \varphi_k)$, we compute the number of times that a player identified its location with (an utterance translated as) σ while standing on grid square (x,y). These counts are smoothed using a simple 2D-smoothing scheme, detailed in (Potts, 2012), and normalized in the usual manner to form a distribution over board squares $\Pr((x,y)|\sigma)$. These grounded interpretations are the basis for communication between the artificial agents we define in Section 4.

BOARD, SQUARE, right, middle, top, left, bottom, corner, approx, precise, entrance, C_room, hall, room, sideways_C, loop, reverse_C, U_room, T_room, deadend, wall, sideways_F

Table 1: The spatial semantic primitives.

3.2 Semantics Classifier

Using the corpus examples of utterances paired with their spatial semantic representations, we learn a set of classifiers to predict a spatial utterance's semantic representation. We train a binary classifier for each semantic primitive φ_i using a log-linear model with simple bag of words features. The words are not normalized or stemmed and we use whitespace tokenization. We additionally train a multi-class classifier for all possible domains δ. At test time, we use the domain classifier and each primitive binary classifier to produce a semantic representation.

3.3 Semantic State Space

The decision making algorithms that we discuss in Section 4 are highly sensitive to the size of the state space. The full representation of the game board consists of 231 squares. Representing the location of both players and the location of the card requires $323^3 = 12,326,391$ states, well beyond the capabilities of current decision-making algorithms.

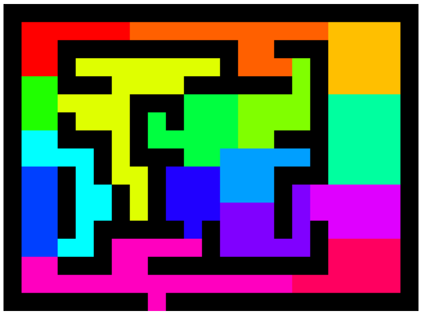


Figure 2: Semantic state space clusters with $k = 16$.

To ameliorate this difficulty, we cluster squares together using the spatial referring expression corpus. This approach follows from research that shows that humans' mental spatial representations are influenced by their language (Hayward and Tarr, 1995). Our intuition is that human players do not consider all possible locations of the card and players, but instead lump them into semantically coherent states, such as "the card is in the top right corner." Following this intuition, we cluster states together which have similar referring expressions, allowing our agents to use language as a *cognitive technology* and not just a tool for communication.

For each board square (x,y) we form a vector $\phi(x,y)$ with $\phi_i(x,y) = \Pr((x,y)|\sigma_i)$, where σ_i is the i^{th} distinct semantic representation in the corpus. This forms a 136-dimensional vector for each board square. We then use k-means clustering with a Euclidean distance metric in this semantic space to cluster states which are referred to similarly.

Figure 2 shows a clustering for $k = 16$ which we utilize for the remainder of the paper. Denoting the board regions by $\{1,\ldots,N_{\text{regions}}\}$, we compute the probability of an expression σ referring to a region r by averaging over the squares in the region:

$$\Pr(r|\sigma_i) \propto \sum_{(x,y)\in \text{ region } r} \frac{\Pr((x,y)|\sigma_i)}{|\{(x,y)|(x,y) \in \text{ region } r\}|}$$

4 ListenerBot

We first introduce ListenerBot, an agent that does not take into account the actions or beliefs of its partner. ListenerBot decides what actions to take using a *Partially Observable Markov Decision Process* (POMDP). This allows ListenerBot to track its beliefs about the location of the card and to incorporate linguistic advice. However, ListenerBot does not produce utterances.

A POMDP is defined by a tuple $(S, A, T, O, \Omega, R, b_0, \gamma)$. We explicate each component with examples from our task. Figure 3(a) provides the POMDP influence diagram.

States S is the finite state space of the world. The state space S of ListenerBot consists of the location of the player p and the location of the card c. As discussed above in Section 3.3, we cluster squares of the board into N_{regions} semantically coherent regions, denoted by $\{1, \ldots, N_{\text{regions}}\}$. The state space over these regions is defined as

$$S := \{(p,c) | p,c \in \{1, \ldots, N_{\text{regions}}\}\}$$

Two regions r_1 and r_2 are called *adjacent*, written $\text{adj}(r_1, r_2)$, if any of their constituent squares touch.

Actions A is the set of actions available to the agent. ListenerBot can only take physical actions and has no communicative ability. Physical actions in our region-based state space are composed of two types: traveling to a region and searching a region.

- $\text{travel}(r)$: travel to region r
- search: player exhaustively searches the current region

Transition Distributions The transition distribution $T(s'|a,s)$ models the dynamics of the world. This represents the ramifications of physical actions such as moving around the map. For a state $s = (p,c)$ and action $a = \text{travel}(r)$, the player moves to region r if it is adjacent to p, and otherwise stays in the same place:

$$T((p',c')|\text{travel}(r),(p,c)) = \begin{cases} 1 & \text{adj}(r,p) \wedge p' = r \\ & \wedge c = c' \\ 1 & \neg\text{adj}(r,p) \wedge p = p' \\ & \wedge c = c' \\ 0 & \text{otherwise} \end{cases}$$

Search actions are only concerned with observations and do not change the state of the world:[2]

$$T((p',c')|\text{search},(p,c)) = \mathbb{1}\left[p' = p \wedge c' = c\right]$$

The travel and search high-level actions are translated into low-level (up, down, left, right) actions using a simple A* path planner.

Observations Agents receive observations from a set O according to an observation distribution $\Omega(o|s',a)$. Observations include properties of the physical world, such as the location of the card, and also natural language utterances, which serve to indirectly change agents' beliefs about the world and the beliefs of their interlocutors.

Search actions generate two possible observations: o_{here} and $o_{\neg\text{here}}$, which denote the presence or absence of the card from the current region.

$$\Omega(o_{\text{here}}|(p',c'),\text{search}) = \mathbb{1}\left[p' = c'\right]$$
$$\Omega(o_{\neg\text{here}}|(p',c'),\text{search}) = \mathbb{1}\left[p' \neq c'\right]$$

Travel actions do not generate meaningful observations:

$$\Omega(o_{\neg\text{here}}|(p',c'),\text{travel}) = 1$$

Linguistic Advice We model linguistic advice as another form of observation. Agents receive messages from a finite set Σ, and each message $\sigma \in \Sigma$ has a *semantics*, or distribution over the state space $\Pr(s|\sigma)$. In the Cards task, we use the semantic distributions defined in Section 3. To combine the semantics of language with the standard POMDP observation model, we apply Bayes' rule:

$$\Pr(\sigma|s) = \frac{\Pr(s|\sigma)\Pr(\sigma)}{\sum_{\sigma'} \Pr(s|\sigma')\Pr(\sigma')} \tag{1}$$

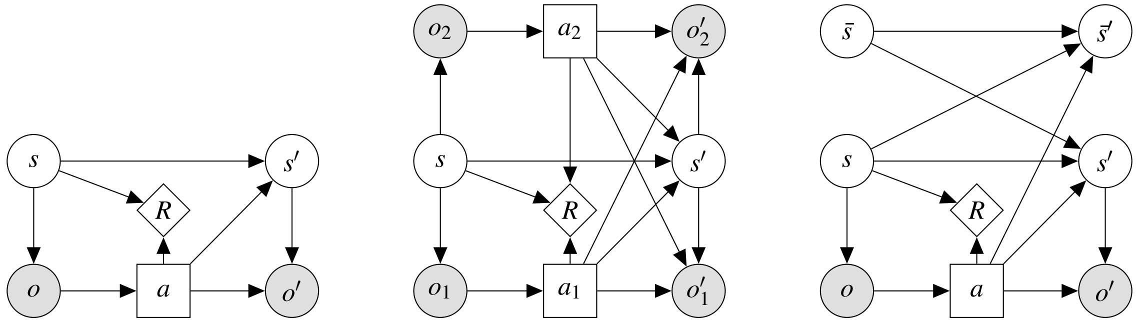
The prior, $\Pr(\sigma)$, can be derived from corpus data. By treating language as just another form of observation, we are able to leverage existing POMDP solution algorithms. This approach contrasts with previous work on communication in Dec-POMDPs, where agents directly share their perceptual observations (Pynadath and Tambe, 2002; Spaan et al., 2008), an assumption which does not fit natural language.

Reward The reward function $R(s,a) : S \to \mathbb{R}$ represents the goals of the agent, who chooses actions to maximize reward. The goal of the Cards task is for both players to be on top of the card, so any action that leads to this state receives a high reward R^+. All other actions receive a small negative reward R^-, which gives agents an incentive to finish the task as quickly as possible.

$$R((p,c),a) = \begin{cases} R^+ & p = c \\ R^- & p \neq c \end{cases}$$

Lastly, $\gamma \in [0,1)$ is the discount factor, specifying the trade-off between immediate and future rewards.

[2] $\mathbb{1}[Q]$ is the indicator function, which is 1 if proposition Q is true and 0 otherwise.



(a) ListenerBot POMDP

(b) Full Dec-POMDP

(c) DialogBot POMDP

Figure 3: The decision diagram for the ListenerBot POMDP, the full Dec-POMDP, and the DialogBot approximation POMDP. The ListenerBot (a) only considers his own location p and the card location c. In the full Dec-POMDP (b), both agents receive individual observations and choose actions independently. Optimal decision making requires tracking all possible histories of beliefs of the other agent. In diagram (c), DialogBot approximates the full Dec-POMDP as single-agent POMDP. At each time step, DialogBot marginalizes out the possible observations $\bar{o}$ that ListenerBot received, yielding an *expected belief state* $\bar{b}$.

Initial Belief State The initial belief state, $b_0 \in \Delta(S)$, is a distribution over the state space S. ListenerBot begins each game with a known initial location p_0 but a uniform distribution over the location of the card c:

$$b_0(p,c) \propto \begin{cases} \frac{1}{N_{\text{regions}}} & p = p_0 \\ 0 & \text{otherwise} \end{cases}$$

Belief Update and Decision Making The key decision making problem in POMDPs is the construction of a policy $\pi : \Delta(S) \to A$, a function from beliefs to actions which dictates how the agent acts. Decision making in POMDPs proceeds as follows. The world starts in a hidden state $s_0 \sim b_0$. The agent executes action $a_0 = \pi(b_0)$. The underlying hidden world state transitions to $s_1 \sim T(s'|a_0, s_0)$, the world generates observation $o_0 \sim \Omega(o|s_1, a_0)$, and the agent receives reward $R(s_0, a_0)$. Using the observation o_0, the agent constructs a new belief $b_1 \in \Delta(S)$ using Bayes' rule:

$$\begin{aligned} b_{t+1}^{a_t,o_t}(s') &= \Pr(s'|a_t, o_t, b_t) \\ &= \frac{\Pr(o_t|a_t, s', b_t)\Pr(s'|a_t, b_t)}{\Pr(o_t|b_t, a_t)} \\ &= \frac{\Omega(o_t|s', a_t)\sum_{s\in S} T(s'|a_t, s) b_t(s)}{\sum_{s''}\Omega(o_t|s'', a_t)\sum_{s\in S} T(s''|a_t, s) b_t(s)} \end{aligned}$$

This process is referred to as *belief update* and is analogous to the forward algorithm in HMMs. To incorporate communication into the standard POMDP model, we consider observations $(o, \sigma) \in O \times \Sigma$ which are a combination of a perceptual observation o and a received message σ. The semantics of the message σ is included in the belief update equation using $\Pr(s|\sigma)$, derived in Equation 1:

$$b_{t+1}^{a_t,o_t,\sigma_t}(s') = \frac{\Omega(o|s',a)\frac{\Pr(s'|\sigma)\Pr(\sigma)}{\sum_{\sigma'\in\Sigma}\Pr(s'|\sigma')\Pr(\sigma')}\sum_{s\in S} T(s'|a,s) b_t(s)}{\sum_{s''\in S}\Omega(o|s'',a)\frac{\Pr(s''|\sigma)\Pr(\sigma)}{\sum_{\sigma'\in\Sigma}\Pr(s''|\sigma')\Pr(\sigma')}\sum_{s\in S} T(s''|a,s) b_t(s)}$$

Using this new belief state b_1, the agent selects an action $a_1 = \pi(b_1)$, and the process continues.

An initial belief state b_0 and a policy π together define a Markov chain over pairs of states and actions. For a given policy π, we define a *value function* $V^\pi : \Delta(S) \to \mathbb{R}$ which represents the expected discounted reward with respect to that Markov chain:

$$V^\pi(b_0) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}[R(b_t, a_t)|b_0, \pi]$$

The goal of the agent is find a policy π^* which maximizes the value of the initial belief state:

$$\pi^* = \arg\max_\pi V^\pi(b_0)$$

Exact computation of π^* is PSPACE-complete (Papadimitriou and Tsitsiklis, 1987), making approximation algorithms necessary for all but the simplest problems. We use Perseus (Spaan and Vlassis, 2005), an anytime approximate point-based value iteration algorithm.

5 DialogBot

We now introduce DialogBot, a Cards agent which is capable of producing linguistic advice. To decide when and how to speak, DialogBot maintains a distribution over its partner's beliefs and reasons about the effects his utterances will have on those beliefs. To handle these complexities, DialogBot models the world as a *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) (Bernstein et al., 2002). See Figure 3(b) for the influence diagram. The definition of Dec-POMDPs mirrors that of the POMDP, with the following changes.

There is a finite set I of agents, which we restrict to two. Each agent takes an action a_i at each time step, forming a joint action $\vec{a} = (a_1, a_2)$. Each agent receives its own observation o_i according to $\Omega(o_1, o_2|a_1, a_2, s')$. The transition distributions $T(s'|a_1, a_2, s)$ and the reward $R(s, a_1, a_2)$ both depend on both agents' actions.

Optimal decision making in Dec-POMDPs requires maintaining a probability distribution over all possible sequences of actions and observations $(\bar{a}_1, \bar{o}_1, \ldots, \bar{a}_t, \bar{o}_t)$ that the other player might have received. As t increases, we have an exponential increase in the belief states an agent must consider. Confirming this informal intuition, decision making in Dec-POMDPs is NEXP-complete, a complexity class above P-SPACE (Bernstein et al., 2002). This computational complexity limits the application of Dec-POMDPs to very small problems. To address this difficulty we make several simplifying assumptions, allowing us to construct a single-agent POMDP which approximates the full Dec-POMDP.

Firstly, we assume that other agents do not take into account our own beliefs, i.e., the other agent acts like a ListenerBot. This bypasses the infinitely nested belief problem by assuming that other agents track one less level of nested beliefs, a common approach (Goodman and Stuhlmüller, 2012; Gmytrasiewicz and Doshi, 2005).

Secondly, instead of tracking the full tree of possible observation histories, we maintain a point estimate $\bar{b}$ of the other agent's beliefs, which we term the *expected belief state*. Rather than tracking each possible observation/action history of the other agent, at each time step we marginalize out the observations they could have received. Figure 4 compares this approach with exact belief update.

Thirdly, we assume that the other agent acts according to a variant of the Q_{MDP} approximation (Littman et al., 1995). Under this approximation, the other agent solves a fully-observable MDP version of the ListenerBot POMDP, yielding an MDP policy $\bar{\pi} : S \rightarrow A$. This critically allows us to approximate the other agent's belief update using a specially formed POMDP, which we detail next.

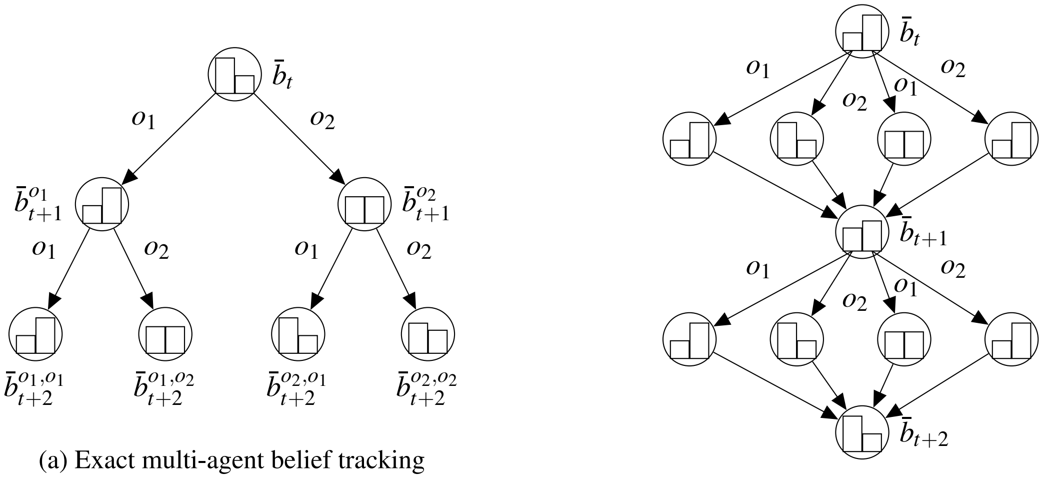
State Space To construct the approximate single-agent POMDP from the full Dec-POMDP problem, we formulate the state space as $S \times S$. (See Figure 3(c) for the influence diagram.) We write a state $(s, \bar{s}) \in S \times S$, where s is DialogBot's beliefs about the true state of the world, and $\bar{s}$ is DialogBot's estimate of the other agent's beliefs.

Transition Distribution The main difficulty in constructing the approximate single-agent POMDP is specifying the transition distribution $T((s', \bar{s}')|a, (s, \bar{s}))$. To address this, we break this distribution into two components: $T((s', \bar{s}')|a, (s, \bar{s})) = \bar{T}(\bar{s}'|s', a, (s, \bar{s}))T(s'|a, s, \bar{s})$. The first term dictates how DialogBot updates its beliefs about the other agent's beliefs:

$$\begin{aligned}\bar{T}(\bar{s}'|s',a,(s,\bar{s})) &= \Pr(\bar{s}'|s',a,(s,\bar{s}))\\ &= \sum_{\bar{o}\in O} \Pr(\bar{s}'|a,\bar{o},\bar{s},s)\Pr(\bar{o}|s',a,\bar{\pi}(\bar{s}))\\ &= \sum_{\bar{o}\in O} \left(\frac{\Omega(\bar{o}|\bar{s}',a,\bar{\pi}(\bar{s}))T(\bar{s}'|a,\bar{\pi}(\bar{s}),\bar{s})}{\sum_{\bar{s}''}\Omega(\bar{o}|\bar{s}'',a,\bar{\pi}(\bar{s}))T(\bar{s}''|a,\bar{\pi}(\bar{s}),\bar{s})} \right.\\ &\qquad \left. \times\, \Omega(\bar{o}|s',a,\bar{\pi}(\bar{s})) \vphantom{\frac{a}{b}}\right)\end{aligned}$$

We sum over all observations $\bar{o}$ the other agent could have received, updating our probability of $\bar{s}'$ as ListenerBot would have, multiplied by the probability that ListenerBot would have received that observation, $\Omega(\bar{o}|s', \bar{\pi}(\bar{s}))$. The Q_{MDP} approximation allows us to simulate ListenerBot's belief update in $\bar{T}(\bar{s}'|s', a, (s, \bar{s}))$. Exact belief update would require access to $\bar{b}$: by using $\bar{\pi}(\bar{s})$ we can estimate the action that ListenerBot would have taken.

In cases where $\bar{s}$ contradicts s such that for all $\bar{o}$ either $\Omega(\bar{o}|s', \bar{\pi}(\bar{s})) = 0$ or $\Omega(\bar{o}|\bar{s}', \bar{\pi}(\bar{s})) = 0$, we redistribute the belief mass uniformly: $\bar{T}(\bar{s}'|s', a, (s, \bar{s})) \propto$


$\bar{b}_t$
o_1
o_2
$\bar{b}^{o_1}_{t+1}$
$\bar{b}^{o_2}_{t+1}$
$\bar{b}^{o_1,o_1}_{t+2}$
$\bar{b}^{o_1,o_2}_{t+2}$
$\bar{b}^{o_2,o_1}_{t+2}$
$\bar{b}^{o_2,o_2}_{t+2}$
$\bar{b}_{t+1}$
$\bar{b}_{t+2}$

(a) Exact multi-agent belief tracking

(b) Approximate multi-agent belief tracking

Figure 4: Exact multi-agent belief tracking compared with our approximate approach. Each node represents a belief state. In exact tracking (a), the agent tracks every possible history of observations that its partner could have received, which grows exponentially in time. In approximate update (b), the agent considers each possible observation and then averages the resulting belief states, weighted by the probability the other agent received that observation, resulting in a single summary belief state $\bar{b}_{t+1}$. Under the Q$_{\text{MDP}}$ approximation, the agent considers what action the other agent would have taken if it completely believed the world was in a certain state. Thus, there are four belief states resulting from $\bar{b}_t$, as opposed to two in the exact case.

$1\ \forall \bar{s'} \neq \bar{s}$. This approach to managing contradiction is analogous to logical belief revision (Alchourronón et al., 1985; Gärdenfors, 1988; Fermé and Hansson, 2011).

Speech Actions Speech actions are modeled by how they change the beliefs of the other agent. The effects of a speech actions are modeled in $\bar{T}(\bar{s'}|s', a, (s, \bar{s}))$, our model of how ListenerBot's beliefs change. For a speech action $a = \text{say}(\sigma)$ with $\sigma \in \Sigma$,

$$\bar{T}(\bar{s'}|s', a, (s, \bar{s})) = \sum_{\bar{o} \in O} \Bigg(\frac{\Omega(\bar{o}|\bar{s'}, a, \bar{\pi}(\bar{s}))\,\Pr(\sigma|\bar{s'})\,T(\bar{s'}|a, \bar{\pi}(\bar{s}), \bar{s})}{\sum_{\bar{s''}} \Omega(\bar{o}|\bar{s''}, a, \bar{\pi}(\bar{s}))\,\Pr(\sigma|\bar{s''})\,T(\bar{s''}|a, \bar{\pi}(\bar{s}), \bar{s})} \times \Omega(\bar{o}|s', a, \bar{\pi}(\bar{s})) \Bigg)$$

DialogBot is equipped with the five most frequent speech actions: BOARD(middle), BOARD(top), BOARD(bottom), BOARD(left), and BOARD(right). It produces concrete utterances by selecting a sentence from the training corpus with the desired semantics.

Reward DialogBot receives a large reward when both it and its partner are located on the card, and a negative cost when moving or speaking:

$$R((p, c, \bar{p}, \bar{c}), a) = \begin{cases} R^+ & p = c \wedge \bar{p} = c \\ R^- & p \neq c \vee \bar{p} \neq c \end{cases}$$

DialogBot's reward is not dependent on the beliefs of the other player, only the true underlying state of the world.

6 Experimental Results

We now experimentally evaluate our semantic classifiers and the agents' task performance.

6.1 Spatial Semantics Classifiers

We report the performance of our spatial semantics classifiers, although their accuracy is not the focus of this paper. We use 10-fold cross validation on a corpus of 577 annotated utterances. We used simple bag-of-words features, so overfitting the data with cross validation is not a pressing concern. Of the 577 utterances, our classifiers perfectly labeled 325 (56.3% accuracy). The classifiers correctly predicted the domain δ of 515 (89.3%) utterances. The precision of our binary semantic primitive classifiers was $\frac{969}{1126} = .861$ and recall $\frac{969}{1242} = .780$, yielding F_1 measure .818.

6.2 Cards Task Evaluation

We evaluated our ListenerBot and DialogBot agents in the Cards task. Using 500 randomly generated initial player and card locations, we tested each combination of ListenerBot and DialogBot partners. Agents succeeded at a given initial position if they both reached the card within 50 moves. Table 2 shows how many trials each dyad won and how many high-level actions they took to do so.

Agents	% Success	Moves
LB & LB	84.4%	19.8
LB & DB	87.2%	17.5
DB & DB	90.6%	16.6

Table 2: The evaluation for each combination of agents. LB = ListenerBot; DB = DialogBot.

Collaborating DialogBots performed the best, completing more trials and using fewer moves than the ListenerBots. The DialogBots initially explore the space in a similar manner to the ListenerBots, but then share card location information. This leads to shorter interactions, as once the DialogBot finds the card, the other player can find it more quickly. In the combination of ListenerBot and DialogBot, we see about half of the improvement over two ListenerBots. Roughly 50% of the time, the ListenerBot finds the card first, which doesn't help the DialogBot find the card any faster.

7 Emergent Pragmatics

Grice's original model of pragmatics (Grice, 1975) involves the cooperative principle and four maxims: quality ("say only what you know to be true"), relation ("be relevant"), quantity ("be as informative as is required; do not say more than is required"), and manner (roughly, be clear and concise).

In most interactions, DialogBot searches for the card and then reports its location to the other agent. These reports obey quality in that they are made only when based on actual observations. The behavior is not hard-coded, but rather emerges, because only accurate information serves the agents' goals. In contrast, sub-optimal policies generated early in the POMDP solving process sometimes lie about card locations. Since this behavior confuses the other agent and thus has a lower utility, it gets replaced by truthful communication as the policies improve.

We also capture the effects of relation and the first clause of quantity, because the nature of the reward function and the nested belief structures ensure that DialogBot offers only relevant, informative information. For instance, when DialogBot finds the card in the lower left corner, it alternates saying "left" and "bottom", effectively overcoming its limited generation capabilities. Again, early sub-optimal policies sometimes do not report the location of the card at all, thereby failing to fulfill these maxims.

We expect these models to produce behavior consistent with manner and the second clause of quantity, but evaluating this claim will require a richer experimental paradigm. For example, if DialogBot had a larger and more structured vocabulary, it would have to choose between levels of specificity as well as more or less economical forms.

8 Conclusion

We have shown that cooperative pragmatic behavior can arise from multi-agent decision-theoretic models in which the agents share a joint utility function and reason about each other's belief states. Decision-making in these models is intractable, which has been a major obstacle to achieving experimental results in this area. We introduced a series of approximations to manage this intractability: (i) combining low-level states into semantically coherent high-level ones; (ii) tracking only an averaged summary of the other agent's potential beliefs; (iii) limiting belief state nesting to one level, and (iv) simplifying each agent's model of the other's beliefs so as to reduce uncertainty. These approximations bring the problems under sufficient control that they can be solved with current POMDP approximation algorithms. Our experimental results highlight the rich pragmatic behavior this gives rise to and quantify the communicative value of such behavior. While there remain insights from earlier theoretical proposals and logic-based methods that we have not fully captured, our current results support the notion that probabilistic decision-making methods can yield robust, widely applicable models that address the real-world difficulties of partial observability and uncertainty.

Acknowledgments

This research was supported in part by ONR grants N00014-10-1-0109 and N00014-13-1-0287 and ARO grant W911NF-07-1-0216.

References

Carlos E. Alchourronón, Peter Gärdenfors, and David Makinson. 1985. On the logic of theory change: Partial meets contradiction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530.

James F. Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom. 2007. PLOW: A collaborative task learning agent. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1514–1519. AAAI Press, Vancouver, British Columbia, Canada.

James F. Allen. 1991. *Reasoning About Plans*. Morgan Kaufmann, San Francisco.

David Beaver. 2002. Pragmatics, and that's an order. In David Barker-Plummer, David Beaver, Johan van Benthem, and Patrick Scotto di Luzio, editors, *Logic, Language, and Visual Information*, pages 192–215. CSLI, Stanford, CA.

Anton Benz, Gerhard Jäger, and Robert van Rooij, editors. 2005. *Game Theory and Pragmatics*. Palgrave McMillan, Basingstoke, Hampshire.

Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840.

Reinhard Blutner. 1998. Lexical pragmatics. *Journal of Semantics*, 15(2):115–162.

Michael Bratman. 1987. *Intentions, Plans, and Practical Reason*. Harvard University Press.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Judith Degen and Michael Franke. 2012. Optimal reasoning about referential expressions. In *Proceedings of SemDIAL 2012*, Paris, September.

David DeVault, Natalia Kariaeva, Anubha Kothari, Iris Oved, and Matthew Stone. 2005. An information-state approach to collaborative reference. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 1–4, Ann Arbor, MI, June. Association for Computational Linguistics.

David DeVault. 2008. *Contribution Tracking: Participating in Task-Oriented Dialogue under Uncertainty*. Ph.D. thesis, Rutgers University, New Brunswick, NJ.

Eduardo Fermé and Sven Ove Hansson. 2011. AGM 25 years: Twenty-five years of research in belief change. *Journal of Philosophical Logic*, 40(2):295–331.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998.

Michael Franke. 2009. *Signal to Act: Game Theory in Pragmatics*. ILLC Dissertation Series. Institute for Logic, Language and Computation, University of Amsterdam.

Peter Gärdenfors. 1988. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press.

Piotr J. Gmytrasiewicz and Prashant Doshi. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:24–49.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 410–419, Cambridge, MA, October. ACL.

Noah D. Goodman and Andreas Stuhlmüller. 2012. Knowledge and implicature: Modeling language understanding as social cognition. In *Proceedings of the Thirty-Fourth Annual Conference of the Cognitive Science Society*.

H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry Morgan, editors, *Syntax and Semantics*, volume 3: Speech Acts, pages 43–58. Academic Press, New York.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Comput. Linguist.*, 12(3):175–204, July.

William G. Hayward and Michael J. Tarr. 1995. Spatial language and spatial representation. *Cognition*, 55:39–84.

Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

David Lewis. 1969. *Convention*. Harvard University Press, Cambridge, MA. Reprinted 2002 by Blackwell.

Michael L. Littman, Anthony R. Cassandra, and Leslie Pack Kaelbling. 1995. Learning policies for partially observable environments: Scaling up. In Armand Prieditis and Stuart J. Russell, editors, *ICML*, pages 362–370. Morgan Kaufmann.

Arthur Merin. 1997. If all our arguments had to be conclusive, there would be few of them. Arbeitspapiere SFB 340 101, University of Stuttgart, Stuttgart.

Christos Papadimitriou and John N. Tsitsiklis. 1987. The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450, August.

Prashant Parikh. 2001. *The Use of Language*. CSLI, Stanford, CA.

C. Raymond Perrault and James F. Allen. 1980. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3–4):167–182.

Christopher Potts. 2012. Goal-driven answers in the Cards dialogue corpus. In Nathan Arnett and Ryan Bennett, editors, *Proceedings of the 30th West Coast Conference on Formal Linguistics*, Somerville, MA. Cascadilla Press.

David V. Pynadath and Milind Tambe. 2002. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:2002.

Hannah Rohde, Scott Seyfarth, Brady Clark, Gerhard Jäger, and Stefan Kaufmann. 2012. Communicating with cost-based implicature: A game-theoretic approach to ambiguity. In *The 16th Workshop on the Semantics and Pragmatics of Dialogue*, Paris, September.

Robert van Rooy. 2003. Questioning to resolve decision problems. *Linguistics and Philosophy*, 26(6):727–763.

Seymour Rosenberg and Bertram D. Cohen. 1964. Speakers' and listeners' processes in a word communication task. *Science*, 145:1201–1203.

Matthijs T. J. Spaan and Nikos Vlassis. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24(1):195–220, August.

Matthijs T. J. Spaan, Frans A. Oliehoek, and Nikos Vlassis. 2008. Multiagent planning under uncertainty with stochastic communication delays. In *In Proc. of the 18th Int. Conf. on Automated Planning and Scheduling*, pages 338–345.

Alex Stiller, Noah D. Goodman, and Michael C. Frank. 2011. Ad-hoc scalar implicature in adults and children. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, Boston, July.

Matthew Stone, Richmond Thomason, and David DeVault. 2007. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. To appear in Donna K. Byron; Craige Roberts; and Scott Schwenter, *Presupposition Accommodation*.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Comput. Speech Lang.*, 24(4):562–588, October.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Comput. Speech Lang.*, 24(2):150–174, April.