Partially Observable Stochastic Reference Games

Adam Vogel (av@cs.stanford.edu)
Computer Science Department
Stanford University

Dan Jurafsky (jurafsky@stanford.edu)
Linguistics Department
Stanford University

Abstract

We present a model of the production and interpretation of referring expressions as a *partially observable stochastic game* (POSG), a multi-agent probabilistic decision-making model. Agents acting in POSGs maintain a model of the interlocutor's beliefs and intentions, capturing several pragmatic effects on the semantics of referring expressions: audience design, conversational grounding, and underspecification.We present the theory of POSGs and discuss algorithms for tracking the beliefs of interlocutors, decision making, and learning.

Introduction

The production and interpretation of referring expressions (REs) in situated, task-oriented dialog is inherently a collaborative process (Clark & Wilkes-Gibbs, 1986). Furthermore, uncertainty pervades this collaboration: uncertainty as to the beliefs and intentions of interlocutors, the identity and properties of objects, and the content of conversation. This paper models the production and interpretation of referring expressions as a *partially-observable stochastic game* (POSG), a multi-agent probabilistic decision making framework which captures all of these aspects.

Modern single-agent decision-theoretic models of conversation such as the Partially Observable Markov Decision Process (POMDP) (Young et al., 2010) represent the interlocutor as a part of the environment, which restricts their application to literal language interpretation. By explicitly modeling the conversational partner as a rational agent, POSGs afford pragmatic inferences – single-agent models capture *what* was said, but not *why* it was said. Not only what was said but further the alternative REs that an agent did not choose figure into interpretation.

Agents acting in POSGs must maintain not only their own beliefs of the state of the world, but further the beliefs of their conversational partners. The knowledge of the other agent figures deeply into the production of referring expressions in the form of *audience design* (Clark & Murphy, 1982). Agents acting in POSGs maintain an estimate of the *beliefs* of their interlocutor, represented as a probability distribution over a state space, and their *intentions*, modeled as a policy, or mapping from beliefs to actions.

Speakers must balance the brevity of their referring expressions against their vagueness. Real-world REs, when taken out of conversational context, are frequently ambiguous: speakers use task-specific constraints, the conversational history, and conceptual pacts to produce and interpret REs. Despite this nature, previous work frequently considers reference as a one-shot task (Dale & Reiter, 1996), ignoring the richness of clarification and refashioning.

The main drawback of POSGs is the intractability of inference algorithms, which are NEXP-complete (Bernstein, Givan, Immerman, & Zilberstein, 2000). Despite this hardness result, research into approximation algorithms for decision making in POSGs is an active endeavor (Kumar & Zilberstein, 2009). It is still an open question as to which is better: the exact solution to a single-agent model versus an approximate solution to a richer multi-agent model. Natural conversation has several properties that make it computationally simpler than the general worst case problem: participants frequently align their beliefs through grounding and other implicit feedback, restricting the set of beliefs that agents must consider.

In the remainder we present the partially-observable stochastic game formalism, give an example conversation paired with a concrete POSG, and discuss learning and inference in these models.

Illustrative Example

We now consider an example which illustrates several phenomena in referring expressions that our model captures. A chef has just bought a robot for helping with food preparation. The robot is unfamiliar with the kitchen, and must utilize clarification and pragmatic inferences to cope in this new environment. Consider the following dialog sample, where the chef and helper are baking a cake:

1. Chef: Let's mix the cake batter
2. Helper: Sure
3. Chef: Could you get the bowl?
4. Helper: The mixing bowl?
5. Chef: Yeah, the one from the cupboard
6. Helper: [Goes to the cupboard, opens it, finds the bowl and grabs it, comes back in the kitchen]
7. Chef: Put it on the counter here

This simple example demonstrates several properties of referring expression generation and interpretation:

- Grounding to establish common ground: utterance (2) confirms that both participants have the same goal.

- Pragmatic constraints on referring expressions: in utterance (4) the helper thinks that the chef likely wants a mixing bowl but confirms it to be sure.

- Tailoring referring expressions to listener knowledge: in utterance (5) the chef adds the location information of the bowl as she is unsure that the helper knows where it is.

- Implicit confirmation: after utterance (7) the helper is sure it got the right bowl, or else the chef would have added a correction.

Partially Observable Stochastic Games

We now proceed to introduce our model and illustrate how it captures these inferences. At a high level, POSGs are similar to other modern decision making frameworks: they compose into a state space, which represents the physical state of the world and the intentions of agents acting in it, beliefs of agents, which are probability distributions over this state space, a set of actions which the agents can execute to change the world, a transition model which describes how these actions change the state, and a reward function which models the goals of the task at hand.

Definition

A *partially observable stochastic game* (POSG) is a tuple $(I, S, b_0, A, O, T, \Omega, R)$ where

- I is a finite set of n agents. In this paper we restrict attention to two agents.

- S is a set of states.

- $b_0 \in \Delta(S)$ is the initial state distribution.

- A is a set of actions.

- O is a set of observations, which provide evidence of the state of the world

- $T(s'|, s, a)$ is a state transition distribution, which represents the dynamics of the world.

- $\Omega(o|s', a)$ is the observation distribution, which gives the probability of a given observation after executing action a and transitioning to state s'.

- $R : S \times \vec{A} \to \mathscr{R}$ is the reward function, which dictates the goals of the agents.

We now discuss each of these constituent parts and give concrete examples.

State S

The state of POSGs for this reference game are formed of three constituent parts:

1. The physical state of the world: objects, their properties, and relations between them

2. The conversational history, in the form of a sequence of dialog acts and the identity of referents

3. The task stack of each agent, a representation of their intentions (Grosz & Sidner, 1986)

The task stack of each agent is always hidden from the other; it can only be inferred indirectly. In simple reference games the physical state of the world is joint knowledge and fully observable, but in actual applied settings with noisy sensors and partial information it can be hidden. Similarly, the conversational history is uncertain to each agent: although an agent always knows what it meant by a statement, the mapping from surface forms to dialog acts is uncertain.

The task stack is a representation of the goals of an agent. In simple reference games the goal of one agent is to get another to identify a given object, who aims to identify the correct object. We write states as tuples (world, conversational history, task stack), for example

$$\begin{aligned} s = (&\text{mixing\_bowl}(x_1) \wedge \text{location}(x_1, x_2) \wedge \text{cupboard}(x_2), \\ &(\text{push}(\text{mix}(t_1) \wedge \text{cake\_batter}(t_1)), \text{“Let's mix the cake batter”})_C, \\ &[\text{mix}(t_1) \wedge \text{cake\_batter}(t_1)]_C, []_H) \end{aligned}$$

Here the chef has stated her intention to mix the cake batter, and adds this goal to her task stack with the push action. The physical state of the world is simplified here to just the mixing bowl. The helper currently has an empty task stack, but will soon add the goal of getting the mixing bowl. Note that the logical variables in a state, such as x_1, are not a shared representation, only used internally by each agent. Thus they cannot communicate using x_1 directly, a departure from (Golland, Liang, & Klein, 2010).

Action A

Speech actions are composed of a dialog act paired with a surface natural language expression. We consider the following dialog acts which are adapted from (Thomason, Stone, & DeVault, 2006):

- ACK: acknowledge a previous conversational act

- ynq(p, x): ask a yes or no question if object x has property p, where x can be an arbitrary referring expression

- assert(p, x): state that object x has property p

- push(g): add a goal to the task stack, where g is a conjunctive logical formula over the state variables

- pop(g): remove a goal from the task stack
- LISTEN: the agent remains silent

An example speech action is (ynq(mixing_bowl(t_1)), "The mixing bowl?"). Agents may add or remove goals from their task stack, using push(x) and pop(x), which may be paired with an empty surface form.

Lastly, agents also have physical actions they can execute:

- goto(x): Move to a location
- open(x): Open a container
- pickup(x): Pickup an object
- putdown(x, l): Place an object in a location

Observation $\Omega(o|s,a)$

When an agent executes a speech action, the interlocutor observes a noisy estimate of the surface form. This is where noise from speech recognition can be modeled, similar to (Young et al., 2010). From $a =$ (ynq(mixing_bowl(t_1)), "The mixing bowl?"), the helper observes a distribution over surface forms and their corresponding dialog acts. The linguistic content can be represented by an N-best list from a speech recognizer, and the mapping from language to a dialog act is a semantic parser. The observation model also yields information about the physical world. From the example above, when the helper opens the cabinet it might observe the presence of the mixing bowl, as well as the presence of other utensils, but again in an uncertain manner.

Agents can also observe properties of the physical world. Vision provides a noisy estimate of the location, color, and other properties of objects. Range finders and odometers provide estimates of the agent's location with respect to obstacles and a world map. Partially observable statistical decision making models were first introduced to deal with sensor uncertainty in robotics (Thrun, Burgard, & Fox, 2005).

Transition $T(s'|a,s)$

The transition distribution $T(s'|a,s)$ models the dynamics of the world. This represents the ramifications of both physical and dialog acts. Speech actions change the conversational history in a deterministic way: the most recent speech act is appended to the conversational history. Physical actions, such as locomotion or grasping objects, succeed with some probability and can change the physical state of the world. Lastly, intentional actions push or pop items from the task stack. These transitions are never perfectly observed and instead must be inferred from observation.

Reward $R(s,a)$

Agents are rewarded for forming and completing task goals in a cooperative fashion. That is, both the helper and the chef gain utility by making the cake, and also by fulfilling intermediate goals they form along that path. The goal of the task at hand is represented by a reward function $R(s,a)$ which maps from states and actions to a real number. The goal of the agents is to maximize the reward accrued over a given trial.

The simplest reward function was simply task completion: the agents receive a large positive reward when they complete the cake and a small negative reward otherwise to encourage brevity. Although in theory this is enough to give optimal behavior, it has several drawbacks.

Firstly, learning in decision-making models with a long time between action and reward makes the credit-attribution problem much harder. It is difficult to discern which actions taken throughout a trial were truly good. Given enough experience acting in the domain agents can eventually learn this, but the actual time taken may be impractical. To ameliorate this problem, we can introduce intermediate rewards which capture useful intermediate behaviors in the task at hand, an approach called *reward shaping* (Ng, Harada, & Russell, 1999). For example, we can allocate a reward to executing actions which result in the completion of a goal on the task stack.

Secondly, the actual behavior humans exhibit in dialog can deviate from that prescribed by models such as ours. Another approach to designing reward functions is to estimate the reward functions that humans implicitly follow. Examples include PARADISE (Walker, Litman, Kamm, & Abella, 1997), which learns an evaluation function which correlates well with human satisfaction judgments. In reinforcement learning more generally, the task of inferring a reward function from demonstrated behavior is called *inverse reinforcement learning*(Ng & Russell, 2000). Applying current inverse reinforcement learning algorithms to dialog data is not straightforward without annotating the dialog acts that humans use.

Belief Tracking

The computational demands of acting in POSGs breaks into two components: belief tracking and decision making. Each agent maintains its own beliefs and an estimate of their interlocutor's $(b,b') \in \Delta(S \times S)$. Starting from the initial belief distribution b_0, the agent uses its sequence of actions and observations to continually update this belief distribution.

Let b_C represent the chef's belief and b_H represent the helper's. Each agent maintains an estimate of (b_C, b_H). When the chef executes a speech action a and observes o, she must update both her own beliefs and her estimate of the helper's.

$$b_C^{t+1}(s') = \frac{\sum_{s \in S} \Omega(o|a,s') T(s'|a,s) b_C^t(s)}{\sum_{s'' \in S} \sum_{s \in s} \Omega(o|a,s'') T(s''|a,s) b_C^t(s)} \quad (1)$$

The observation function Ω has a simple structure in this case: it is 1 when o matches the surface form of a and 0 otherwise. Similarly, $T(s'|a,s)$ is 1 for the state formed by adding a to the conversational history in s, and 0 otherwise. Lastly, since o is independent of s, $\Pr(o|b,a) = 1$. Additionally, the chef must update her model of the helper's beliefs, which proceeds in a similar manner.

In the second case, the helper must update his beliefs after hearing a surface form o. To do so he must form an estimate of the dialog act a that the chef executed.

$$\Pr(a|o,s) = \frac{\sum_{s' \in S} \Omega(o|a,s') T(s'|a,s) \pi(a|s)}{\sum_{a' \in A} \sum_{s' \in S} \Omega(o|a',s') T(s'|a',s) \pi(a'|s)} \quad (2)$$

Here $\pi(a|s)$ is the policy of the other agent, which describes how to act in a given state. We discuss how the policy is computed in the next section. Given this estimate of the underlying dialog act, the helper updates its own beliefs as

$$b_H^{t+1}(s') = \frac{\sum_{s \in S} \sum_{a \in A} \Omega(o|a,s') T(s'|a,s) \Pr(a|o,s) b_H^t(s)}{\Pr(o|b_H^t, a)} \quad (3)$$

In practice there are too many possible actions and states to efficiently compute these quantities. Given an observation o, we can consider only actions which have a non-zero observation likelihood given our speech recognition and semantic parsing. We can use a similar approach to the sum over states by only considering states for which we have a non-negligible belief. Other approaches to simplifying belief tracking include pruning states with low belief and collapsing similar states into a single belief state.

Decision Making

Given the formal definitions of a Partially Observable Stochastic Game given above, an agent must decide how to act in a given situation. More precisely, agents compute a *policy*, or distribution over actions given beliefs, which describes how they will act. The goal of the agents is to choose a policy which maximizes the expected reward accumulated over a trial. These decisions must typically balance the immediate reward gained from taking an action versus the expected future rewards.

In fully observable decision processes, we define the *value* of a state s under a policy π as the expected reward of starting in state s and behaving as π:

$$V^\pi(s) = \mathop{\mathbb{E}}_{\pi}\{R(s,\pi(s))|s\}$$

Using Bellman's equation, we can represent the value function as the well-known recursion

$$V^\pi(s) = R(s,\pi(s)) + \sum_{s' \in S} P(s'|s,\pi(s)) V^\pi(s')$$

Generalizing this to the partially observable case, we define the value of a belief state as

$$\begin{aligned} V^\pi(b) &= \sum_{s \in S} b(s) V^\pi(s) \\ &= \sum_{s \in S} b(s) \left(R(s,\pi(b)) + \sum_{s' \in S, o \in O} P(s',o|s,\pi(b)) V^\pi(b^{a,o}) \right) \end{aligned}$$

Given the policies (π_1, π_2) of two agents and a joint belief state (b_1, b_2), let $a_i = \pi(b_i)$ be the actions each agent chooses. Then we define the joint value function of their policies and beliefs as

$$\begin{aligned} V^{(\pi_1,\pi_2)}(b_1,b_2) = \sum_{s \in S} b_1(s) b_2(s) (R(s,a_1,a_2) \\ + \sum_{s',o_1,o_2} P(s',o_1,o_2|s,a_1,a_2) V^{(\pi_1,\pi_2)}(b_1^{(a_1,o_1)}, b_2^{(a_2,o_2)})) \end{aligned}$$

The state space space we have considered here is extremely large: it contains the set of all possible conversations. Since the same dialog rarely happens twice, computing and storing a value function for the whole state space is infeasible. Instead, we represent the value of taking an action a in state s as

$$\begin{aligned} Q(s,a) &= R(s,a) + \sum_{s'} T(s'|a,s) V(s') \\ &\approx \theta^{\mathrm{T}} \phi(s,a) \end{aligned}$$

where $\phi(s,a)$ is a vector of features which captures salient aspects of s and a. Moving to the partially observable case,

$$Q(b,a) = \sum_{s \in S} b(s) \theta^{\mathrm{T}} \phi(s,a)$$

We can learn the weights θ using a partially-observable variant of SARSA. First we initialize θ to arbitrary small values. While acting with respect to our current estimate of $Q(s,a)$, we update θ for each state transition (b,a,o,b',a') as follows:

$$\begin{aligned} \theta = \theta + \sum_s b(s) \phi(s,a) \sum_{s'} b'(s') (R(s,a) + \theta^{\mathrm{T}} \phi(s',a') \\ - \theta^{\mathrm{T}} \phi(s,a)) \end{aligned}$$

This update still requires that we sum over the whole state space. As a further approximation we can only consider states for which $b(s)$ is non-negligible.

Decision making in general multi-agent games is difficult given the dependence on the other agent's policy. However, the models we are considering have a cooperative structure where agents share a common reward function. This introduces the notion of a *best response*, the set of best policies an agent can choose given knowledge of the other's policy:

$$\begin{aligned} B(\pi_2,b_1,b_2) = \{\pi_1 : \Delta(S) \to A | \\ \forall \pi' . V^{(\pi_1,\pi_2)}(b_1,b_2) \geq V^{(\pi',\pi_2)}(b_1,b_2)\} \end{aligned}$$

Given a value function V agents can prune dominated strategies using linear programming(Hansen, Bernstein, & Zilberstein, 2004). A policy π_j can be pruned if there exists a probability distribution Pr over policies for which

$$\forall s \in S . \sum_{k \neq j} \Pr(\pi_k) V^{\pi_k}(s) \geq V^{\pi_j}(s)$$

Agents prune policies using *iterative elimination of dominated policies*, where they iteratively prune policies until

arriving at an optimum. In common-payoff games there is guaranteed to be a Pareto-optimal Nash equilibrium which is a set of best response policies that maximize the payoffs to all agents (Emery-Montemerlo, Gordon, Schneider, & Thrun, 2004).

Learning

The previous exposition assumed that the model was known to each agent. However, the question remains how to learn the parameters for the transition distribution $T(s'|a,s)$ and observation distribution $\Omega(o|s',a)$. Choosing the states S, actions A, and observations O is a design decision. The main components which remain to be learned are $T(s'|a,s)$, the state transition distribution, and $\Omega(o|a,s')$, the observation distribution. In the reference games we consider here T is deterministic: speech acts are added to the conversational history, intentional actions modify the goal stack, and physical actions always succeed.

Estimating $\Omega(o|a,s')$ is more difficult. Determining the surface form of a given sensor input o is trivial in the case of text input, and is the well-studied speech recognition problem in the case of audio input. However, the speech acts underlying the surface linguistic forms are never observed. Explicit annotation of dialog acts is a difficult and time-consuming endeavor. Recent advances in semantic parsing suggest that it is possible to learn semantic representations using only task completion as a learning signal (Liang, Jordan, & Klein, 2009). In short, rather than being concerned with learning a semantic representation which matches expert annotation, the agent learns a semantic parser which yields good performance on the end task. DeVault and Stone (2007) present a method for learning to disambiguate conversational history by using unannotated dialog data. They cleverly utilize clarifications as implicit training data: when their system has a single interpretation of the dialog history, they treat previous ambiguous states that led to the unambiguous state as valid inferences.

A Return to the Kitchen

We now return to the example dialog from the beginning of the paper to show how the components of the model interact.

1. Chef: Let's mix the cake batter
2. Helper: Sure
3. Chef: Could you get the bowl?
4. Helper: The mixing bowl?
5. Chef: Yeah, the one from the cupboard
6. Helper: [Goes to the cupboard, opens it, finds the bowl and grabs it, comes back in the kitchen]
7. Chef: Put it on the counter here

For simplicity we assume perfect speech recognition. The physical state of the world is known to each participant at the start of the trial, but the chef is not sure of the helper's beliefs.

$$\text{mixing\_bowl}(x_1) \wedge \text{location}(x_1,x_2) \wedge \text{cupboard}(x_2) \wedge \text{serving\_bowl}(x_3) \wedge \text{location}(x_3,x_2)$$

The chef starts the trial off with the goal of making the cake batter and informs the helper of this in the first utterance. The helper observes this utterance and uses its semantic parser to estimate the underlying dialog act

$$a = (\text{push}(\text{mix}(t_1) \wedge \text{cake\_batter}(t_1)), \text{“Let's mix the cake batter”})$$

using Equation 2. Under the helper's model of the chef's policy, it further infers that the chef would only say this if she had the goal of mixing the cake batter, yielding the task stack

$$[\text{mix}(t_1) \wedge \text{cake\_batter}(t_1)]_C, []_H\}.$$

At this point in the dialog the chef is unsure that the helper heard and understood her request and therefore her beliefs of the helper's beliefs contains uncertainty. When the helper utters “Sure” in the next turn, it serves to make the chef believe that the helper understood correctly, establishing common ground.

With the stage set, the chef proceeds in establishing mutual goals by asking the helper to get the bowl [1]. The observation of the helper yields two possible referents of “the bowl”, the mixing bowl and the serving bowl.

$$a_1 = (\text{pickup}(x_1), \text{“Could you get the bowl?”})$$
$$a_2 = (\text{pickup}(x_2), \text{“Could you get the bowl?”})$$

Using the $\Pr(a|o,s)$ term in Equation 2, the helper infers that the chef was likely referring to the mixing bowl (i.e. $\Pr(a_1|\text{“Could you get the bowl?”},s) > \Pr(a_2|\text{“Could you get the bowl?”},s)$), but is not sure. She weighs the cost of proceeding anyway against the cost of asking a clarifying question and opts for the latter, asking “The mixing bowl?”. The chef recognizes this as a yes/no question about the identity of t_1 in the conversational history, and confirms this with “yeah”. Since she is further unsure of the helper's beliefs about the location of the mixing bowl, she adds “the one from the cupboard.”

[1] Note that the chef uses the indirect form of the request, asking a “could” question. In a richer model involving explicit representation of the possibility of actions, the helper could infer that the chef would only ask a “could” question if she wanted the helper to perform the action.

At this point the belief of the helper is

$$s = \Big(\text{mixing\_bowl}(x_1) \wedge \text{location}(x_1, x_2) \wedge \text{cupboard}(x_2) \wedge \text{serving\_bowl}(x_2) \wedge \text{cupboard}(x_2),$$
$$(\text{push}(\text{mix}(t_1) \wedge \text{cake\_batter}(t_1)), \text{“Let's mix the cake batter”})_C$$
$$(\text{ACK}, \text{“Sure”})_H, (\text{pickup}(x_1), \text{“Could you get the bowl?”})_C,$$
$$(\text{ynq}(\text{mixing\_bowl}(x_1)), \text{“The mixing bowl?”})_H,$$
$$(\text{assert}(\text{mixing\_bowl}(x_1)), \text{“Yeah”})_C$$
$$(\text{assert}(\text{location}(x_1, \text{cupboard})), \text{“the one from the cupboard”})_C,$$
$$[\text{mix}(t_1) \wedge \text{cake\_batter}(t_1)]_C, [\text{pickup}(x_1)]_H\Big)$$

Given the helper's task stack of getting the mixing bowl, which is now of known location and identity, the helper executes a series of physical actions to get the mixing bowl. Upon returning the bowl to the chef, who sees the mixing bowl, asks the helper to put it on the counter. Again using Equation 2, the helper deduces that if it got the wrong bowl that the chef would have added a correction, and is now confident that it is correct.

Conclusion

In this paper we demonstrated how referring expression generation and interpretation can be cast as decision making and belief tracking in a partially observable stochastic game. In comparison to single-agent models, POSGs enable the modeling of the more subtle aspects of dialog.

Although currently theoretical, our future work is to implement this theory in a robot. Physical agents must cope with not only linguistic uncertainty but also uncertainty as to the state of the world. Although the POSG formalism is rather complicated and has high computational cost, we believe it has a unique descriptive power for the reasoning agents use to converse with others about their shared world.

Acknowledgments

This research was partially supported by the National Science Foundation via a Graduate Research Fellowship.

References

Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2000). The complexity of decentralized control of markov decision processes. In *Mathematics of operations research* (p. 2002).

Clark, H. H., & Murphy, G. L. (1982). Audience Design in Meaning and Reference. In *Language and comprehension* (Vol. 9, pp. 287–299). Elsevier.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, *22*(1), 1–39.

Dale, R., & Reiter, E. (1996). The role of the gricean maxims in the generation of referring expressions. *CoRR*, *cmp-lg/9604006*.

DeVault, D., & Stone, M. (2007). Learning to interpret utterances using dialogue history. In *Proceedings of DECALOG: The 2007 workshop on the semantics and pragmatics of dialogue.*

Emery-Montemerlo, R., Gordon, G., Schneider, J., & Thrun, S. (2004). Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of int. joint conference on autonomous agents and multi agent systems* (pp. 136–143).

Golland, D., Liang, P., & Klein, D. (2010, October). A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 410–419). Cambridge, MA: Association for Computational Linguistics.

Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, *12*(3), 175-204.

Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *Proceedings of the nineteenth national conference on artificial intelligence* (p. 709-715). San Jose, California.

Kumar, A., & Zilberstein, S. (2009). Dynamic programming approximations for partially observable stochastic games. In *Proceedings of the twenty-second international FLAIRS conference* (p. 547-552). Sanibel Island, Florida.

Liang, P., Jordan, M. I., & Klein, D. (2009). Learning semantic correspondences with less supervision. In *Association for computational linguistics and international joint conference on natural language processing (acl-ijcnlp).*

Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *In proceedings of the sixteenth international conference on machine learning* (pp. 278–287). Morgan Kaufmann.

Ng, A. Y., & Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In P. Langley (Ed.), *Icml* (p. 663-670). Morgan Kaufmann.

Thomason, R. H., Stone, M., & DeVault, D. (2006). Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In *Ohio state pragmatics initiative.*

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press. Hardcover.

Walker, M. A., Litman, D. J., Kamm, C. A., & Abella, A. (1997). Paradise: A framework for evaluating spoken dialogue agents. In *Acl'97* (p. 271-280).

Young, S., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., et al. (2010). The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, *24*(2), 150-174.