Combining Cognitive Modeling and Reinforcement Learning for Clarification in Dialogue

Baber Khalid* **Malihe Alikhani**** **Matthew Stone***
Rutgers University* and University of Pittsburgh**
firstname.lastname@rutgers.edu* malihe@pitt.edu**

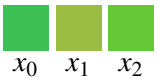
Abstract

In many domains, dialogue systems need to work collaboratively with users to successfully reconstruct the meaning the user had in mind. In this paper, we show how cognitive models of users' communicative strategies can be leveraged in a reinforcement learning approach to dialogue planning to enable interactive systems to give targeted, effective feedback about the system's understanding. We describe a prototype system that collaborates on reference tasks that distinguish arbitrarily varying color patches from similar distractors, and use experiments with crowd workers and analyses of our learned policies to document that our approach leads to context-sensitive clarification strategies that focus on key missing information, elicit correct answers that the system understands, and contribute to increasing dialogue success.

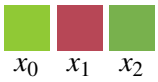
1 Introduction

As dialogue systems move into richer domains, they must increasingly cope with situations where objects don't have generally-known names, where people describe concepts in incompatible ways, and where language use is fundamentally creative and uncertain (Furnas et al., 1987). People succeed in such situations thanks to their collaborative interactions, which let the speaker and the audience share responsibility for reaching a satisfactory mutual understanding (Clark, 1996). This paper contributes to a broader project of building dialogue systems that can do the same.

We focus specifically on the problem of asking targeted, effective clarification questions with creative language. The difficulty of this problem is highlighted by the interactions shown in Figure 1, taken from the human-subjects evaluations reported in Section 6. In these dialogues, our system (playing the role of *matcher* in the referential communication task of Monroe et al. (2017)) uses varying clarification questions to follow up initial color descriptions provided by crowd workers (playing the role of *director*), and is thereby able to successfully distinguish the target color patches from their contextual distractors.



director:	lusher green
matcher:	sea green or bright green?
director:	sea green
matcher:	[Selects]



director:	light green
matcher:	Do you mean lime green?
director:	yes
matcher:	[Selects]

Figure 1: Our system (matcher) interacting with crowd workers (director). System and users are presented with the same color patches in random order; the director must identify x_0 in a text chat conversation. At left, the system's clarification contrasts two alternative referents (*lush*, as it happens, is outside its vocabulary). At right, the system's clarification redescribes the most likely alternative. The choice is based on a learned strategy, described in Section 5, that takes into account both the system's uncertainty in context and likely patterns of user response. Both interactions here are successful.

In formulating such questions, the system must resolve the specific uncertainty it faces in context; there are many options about how to present that uncertainty to users. Meanwhile, depending on the context and the form and content of the clarification question, the director may not understand the question as intended—as both human–human conversations in this domain (Monroe et al., 2017) and evaluation data with our system confirm. Thus, for the system to synthesize a question with a good outcome, it needs accurate models of the interpretations, interactive strategies and responses of human interlocutors, to guide its behavior. We meet this challenge by integrating a reinforcement learning approach to dialogue policy with cognitive models to describe the actions and outcomes available to the system.

In particular, our work relies on a new, probabilistic model of contributions to dialogue, as planned by the user and understood by the system. We construct this model by analyzing human behavior and profiling system performance on a benchmark dataset of human–human conversations, drawing on data collection and modeling work from the recent dialogue literature (Monroe et al., 2017; McMahan and Stone, 2020). We use this model at planning time to roll out simulated interactions that track user deliberation, anticipate user dialogue moves at the level of utterance content, and predict task outcomes. We use deep Q-learning (Mnih et al., 2015) to estimate the effectiveness of different actions as a function of dialogue state. Our approach culminates in a context-sensitive policy that decides whether and how to present clarification options to users, contingent on the ambiguity of user input and the predicted outcomes of different resolution strategies.

Our work is the first to intelligently deploy a range of clarification strategies, selected to reduce system uncertainty in a targeted and effective way, and exploiting the creative use of diverse vocabulary to describe alternatives. We explain the contrast with previous approaches in Section 2. We describe the experimental framework that we build on and extend in Section 3, and describe our reinforcement learning approach in Section 4. We demonstrate the effectiveness of our approach with two kinds of experiments. In Section 5, we analyze reinforcement learning outcomes to document the success of the learning process and characterize the intuitively satisfying policies that result. In Section 6, we describe human-subjects evaluations of these strategies that indicate that adaptive clarification improves the system's success and that reinforcement learning improves user satisfaction. These experiments also enable us to quantify the match between simulation predictions and observed human behavior. Together, these evaluations substantiate the claim that our approach combines effective communication, targeted clarification, and expressive vocabulary.

2 Related Work

Our work contributes to the general project of grounding in interactive systems—making sure that system and user have a shared understanding of content in conversation; see Clark and Schaefer (1989) or Traum (1994). We follow the broadly decision-theoretic approach initiated in the early 2000s by Paek and Horvitz (2000), Walker (2000) and others.

Grounding has been a major focus of research in spoken dialogue systems because of the uncertainties associated with speech recognition results, making confirmation strategies particularly important. Optimal decisions have long been guided by user simulations that capture the realistic dynamics of slot filling dialogues; see Young et al. (2013) for review. However, because we work with written rather than spoken language, our concern is uncertainty at deeper levels of interpretation, including such new dimensions of user simulation as the flexible vocabulary and syntax that users adopt, their strategic variability in producing and interpreting utterances, and their deliberation in coordinating with their interlocutors. While a range of work has attempted to model and learn about such referential and speech act ambiguity from dialogue data, including McRoy and Hirst (1995) and DeVault and Stone (2009), these models have not been used to drive decision-theoretic approaches to clarification.

Reinforcement learning in open-domain referential communication has instead focused on simple decisions, such as whether to wait or move forward in incremental dialogue (Manuvinakurike et al., 2017) or asking yes/no questions to resolve ambiguity in visual dialogue (Niu et al., 2019; Shekhar et al., 2019; Zhang et al., 2018). Corpus-based modeling work has shown, however, that natural strategies for collaborative reference are typically substantially more flexible (Clark and Wilkes-Gibbs, 1986; Ginzburg

and Cooper, 2004; Heeman and Hirst, 1995; Schlangen, 2004). Our work aims to learn correspondingly more flexible policies. We argue elsewhere (Khalid et al., 2020) that formal models of discourse coherence give important resources to do this. Although we build on that work here, our prior work has not addressed the problem of learning context-sensitive clarification strategies. Appelgren and Lascarides (2020) also consider semantic grounding of color terms and exploit formal models of discourse coherence to do so; however, their work so far assumes dialogue policies (and primitive motor skills) to be fixed and known in advance.

In rolling out dialogue transcripts using detailed models of interaction, our work may recall utterance-level models of strategic negotiation, such as Jang et al. (2020). An important contrast is that these models learn by self-play: two learning agents interact with and adapt to one another. This is not a good fit for human–agent collaboration since agents can easily learn policies by self-play that work well for other agents but cannot be interpreted by human users. This explains the unique role of cognitive modeling in our approach.

3 Background: Task, Models and Architecture

We work in a constrained referential communication domain, inspired by the human subjects experiments of Clark and colleagues (Clark, 1996). Two participants communicate via text chat. In each round, participants both see a set of three color patches, arrayed in possibly different orders. One participant, the director, gets an indication of a target patch to identify. The other participant, the matcher, can click on the patches. The director and matcher can freely exchange text messages to coordinate on the target object. The round is successful if the matcher selects the indicated target. Monroe et al. (2017) crowdsource a large dataset of human–human conversations in this setup; we call this the Colors-in-Context (CIC) dataset. Figure 2 shows a sample interaction from CIC. Note the matcher's use of a two-alternative comparative clarification question to pin down a more precise understanding of the director.

3.1 Human Strategies

The complexity of the task depends on the visual difficulty of distinguishing the target color from its alternatives. Monroe et al. (2017) set up three conditions, including FAR conditions where all the patches are visually distinctive, CLOSE conditions where all the patches are visually similar, and SPLIT conditions where the target has a single distractor that's visually similar. As Figure 2 shows, more challenging contexts elicit a wide range of utterances, including creative, detailed but nevertheless vague expressions.

We follow McMahan and Stone (2020) in modeling such strategies through cognitive models of choice under uncertainty. McMahan and Stone (2020) first train a large-scale semantic model of color descriptions using Randall Monroe's crowdsourced collection of free text descriptions of color patches, as curated by McMahan and Stone (2015). Then, building on prior work in computational pragmatics, they describe a number of psychologically-plausible inference methods for predicting the effectiveness of semantic content in identifying a target referent. For example, speakers might try to anticipate and shape the listener's interpretive reasoning while drawing on shared and familiar meanings (Monroe and Potts, 2015), speakers might try to signal innovative meanings that would name the target unambiguous (Meo et al., 2014), or speakers might rely exclusively on established and unambiguous meanings, as in classic

director:	the very dull red
matcher:	lighter or darker
director:	lighter

Figure 2: An example from the Colors in Context (CIC) dataset (Monroe et al., 2017) of the director and matcher coordinating so that the matcher can click on the correct color patch (x_0).

approaches to generating referring expressions (Dale and Reiter, 1995). Each of these reasoning methods is implemented as a model that produces a probability distribution over a diverse (but finite) set of descriptive expressions. McMahan and Stone (2020) fit a latent variable model that best explains the dataset of Monroe et al. (2017) with a mixture of such strategies. The result is a probability distribution

$$P(w_k|x_i,C)$$

describing the likelihood that a human speaker will use the term w_k to identify a target x_i in the context C of a specific set of three patches to be distinguished. McMahan and Stone (2020) have made their code and models available at `https://go.rutgers.edu/ugycm1b0` to enable follow-up work such as ours. Our system uses their models in three ways: it uses the model to sample likely user utterances in new dialogue contexts; it uses the model to estimate the likely target objects associated with user utterances; and it uses the model to plan natural system utterances that would be likely understood.

3.2 System Architecture

Our system tracks the organization and flow of dialogue using a novel formal model of the collaborative reference task. The principles behind this model are described in full in Khalid et al. (2020). We have released our implementation, including a pre-trained RL model, at `https://go.rutgers.edu/tc7k14b`, to enable replicability of our results. We highlight key aspects of the implementation here.

In brief, unlike approaches based on an information-state update approach (Larsson and Traum, 2000), collaborative discourse theory (DeVault et al., 2005), or a flat state-space (Heeman, 2007), we model utterances as making abstract moves that attach into an evolving discourse structure, as in formal approaches such as SDRT (Lascarides and Asher, 2009). One advantage of this approach is that it enables the system to represent a hierarchical dialogue structure, which is important for maintaining context through clarification episodes, without cumbersome logic for plan recognition or stack manipulation.

3.2.1 Understanding Utterances

Our understanding module is based on a probabilistic context free grammar (PCFG): the rules are handcrafted based on CIC development data; rule weights are automatically estimated based on the likelihood that estimated parses identify the actual target in CIC training data. At run time, we use an A* chart parser to find the most likely parse of an utterance, while drawing on heuristics for partial parsing to handle utterances with unseen vocabulary and syntax. Parses are associated with contributions to logical form that introduce discourse referents, characterize them in terms of the basic color predications of McMahan and Stone (2015), and use logical operations and coherence relations to link these predications into a knowledge graph representing the content of the discourse.

3.2.2 Modeling Conversational State

Utterance understanding feeds into a symbolic representation of discourse content. To track progress on referential communication, we define a probabilistic semantic evaluation operation to link that discourse content to the visual context. The input is the structure of speaker commitments. The output is a posterior distribution over candidate color patches for the target. Following Khalid et al. (2020), this distribution takes the form of a probability distribution:

$$P(x_i|w^0,\ldots,w^{t-1},w^t,C)$$

where w^t represents the descriptive constraint on the target associated with the tth speaker commitment, x_i ranges over the candidate color patches in the scene, and C indicates the dependency of the calculation on a specific visual context, in this case given by three color patches. (Note that this posterior is calculated based purely on the contributions from the director; since the matcher does not know the target, their contributions typically don't constrain the referent but instead, as with clarification questions, shape the course of the conversation.)

3.2.3 Modeling Color Descriptions

Color descriptions are interpreted using the EM model of McMahan and Stone (2020). In generating color descriptions of a target in context, we also sample from the distribution generated by the EM model (which, although sometimes ambiguous, seems to give the most human-like results).

4 Modeling Dialogue Decisions

Like many reinforcement learning frameworks, our approach starts from the models of dialogue state and action described in Section 4.1 and induces a policy that chooses an action based on dialogue state. Choice is based on outcome; we use the user simulation described in Section 4.2 to predict the outcomes of actions, and measure the success of the dialogue based on the reward function described in Section 4.3. We contrast two configurations: a basic one invites the system to choose whether to ask a confirmatory clarification question or to proceed with its best interpretation; the more substantive one invites the system to choose one of three forms of clarification question: a confirmatory clarification question *CQ1* involving one description *A?* (e.g., *do you mean the mauve?*), a disjunctive clarification question *CQ2* involving two alternatives *A or B?* (e.g., *bright purple or pink?*), and an exhaustive clarification question *CQ3* describing all the objects in the context *A or B or C?* (e.g., *lime green or grey green or green?*).

4.1 State Representation and Action Sets

Our action set includes four meaningful actions for the system: *CQ1*, *CQ2*, *CQ3* and *S* (for selection). Parameters for these actions are instantiated by rules based on the learned models of McMahan and Stone (2020). In particular, *S* requires a target object x_i. The system chooses x_i to maximize the posterior probability $P(x_i|w^0,\ldots,w^{t-1},w^t,C)$ given the dialogue content so far. Meanwhile, the *CQ* actions require descriptions w_n that identify an appropriate target x_i. To compute these descriptions, we use the mixture model from McMahan and Stone (2020) to compute a probability $P(w_n|x_i,C)$, exclude terms w_k that already occur in the dialogue, renormalize, and sample from the resulting distribution. Our action set also includes user actions: *I* for identifying a target using a description w_n; a yes answer *Y* and a no answer *N* for clarification questions, a repeat action *R* using a description from a previous *CQ* and an unrecognized case *U* where not even a partial interpretation is available.

The state representation used by the RL model for training purposes is a concatenation of posterior distribution $P(x_i|w^0,\ldots,w^{t-1},w^t,C)$, one-hot encoding representation of actions which have been taken so far, a turn counter, and a Boolean feature representing if the parser was unable to completely parse the whole sentence. It is given as follows:

$$s_t = \{\forall x_i P(x_i|w^0,\ldots,w^{t-1},w^t,C) : [a_1,a_2,...,a_t] : TurnCounter : IncompleteParse\}$$

We train the model to choose between clarification requests and select actions. We also include an invalid state reached after dialogue exceeds a maximum dialogue length of 15 turns.

4.2 User Simulation

Algorithm 1 shows the logic that we use to simulate the strategy of the user director interacting with the system. In all cases, the basic flow of the director simulation is the same: the director has to contribute an appropriate description to try to characterize the target t in the context C corresponding to a sample interaction d from the CIC dataset. In many cases, this means sampling an appropriate new description W. The most general way we use is to sample from the EM mixture probability distribution $P(W|t,C)$ as modeled by McMahan and Stone (2020).

Meanwhile, if the director is responding to a clarification question, the director also has the option of confirming with the repetition of an element A from the matcher's proposed descriptions. The candidate descriptions for this purpose are drawn in an input list L storing the descriptions used in the previous iteration. The response A is appropriate if the true target t is assigned the highest probability among the candidates under the EM mixture posterior distribution $P(x|A,C)$. In this case we say t is the *referent* of A in C. In the case of perfect communication, the director would simply describe the target using the first appropriate description from the list $L+W$. This description is the initial assignment to *Content* in Algorithm 1.

Algorithm 1 Director Simulation. The director's job is to come up with an appropriate characterization *Content* of the target to continue the interaction; *Content* defaults to a new description W. Complexity comes from allowing for director errors (via a *mistake*) and from anticipating system failures (via parse errors and out-of-vocabulary *OOV* items).

Require: CIC data item d with context C; original initial director utterance U; target patch t
Require: Description W of target t in context C (generated from EM model or sampled as U)
Require: Move M and description list L of options from prior utterance (both possibly empty)

After First director turn opportunity:
  if U does not have a complete parse **then**
    return $Move(U)$
  else
    return $Identify(W)$
$MistakeRates = \{CQ1 : 0.35, CQ2 : 0.20, CQ3 : 0.05\}$
$OOVRate = 0.1$
$Content \leftarrow$ First $\{A$ in $L : referent(A, C) = t\} + W$
$mistake \leftarrow random() < MistakeRates[M]$
if *mistake* **then**
  $Content \leftarrow$ *random choice* $(L + W - Content)$
if $Content = W$ **then**
  if $random() < OOVRate$ **then**
    return $No, Identify(OOV)$
  else
    return $No, Identify(W)$
else
  return $R(Content)$

Accurately anticipating user behavior, however, requires modeling two crucial possibilities: the system may not understand the user, and conversely the user may not understand the system. To handle the possibility where the system does not understand the user, we distinguish the initial contribution and subsequent ones. For initial contributions, we choose to always retain problematic utterances from the original CIC data. For subsequent contributions, we have a 10% chance that new descriptions from the user are an out-of-vocabulary item that leads to a uniform posterior over the targets.

To handle the possibility that the user does not understand the system, we associate an error probability with each of the clarification questions. We set these probabilities to 0.35, 0.20 and 0.05 in response to *A?*, *A or B?*, and *A or B or C?*, based on approximate rates in the CIC development data. (In light of the evaluation results presented in Section 6, it would be interesting to refine this model based on observed rates when people interact with the system.) In case of error, the director randomly uses one of the alternative descriptive contents maintained as a candidate by the algorithm.

4.3 Reward Function

Our initial reward function is designed as a proof of concept for our approach, to embody the commonly observed tradeoff between dialogue length and task success (Walker et al., 1997). Ultimately, we would like to optimize user satisfaction. User satisfaction is not observable, but it can be approximated, following Walker et al. (1997), by correlating user's reported preferences with other features of the dialogue such as length and task success, based on empirical observations. However, this requires the kind of evaluation results we present in Section 6, which were of course not available during our initial experiments. Instead, we crafted an artificial reward function with typical characteristics, penalizing both dialogue length and task failure. In particular, we operationalize dialogue length in terms of the number of distinct types of referring expressions that are used in the dialogue. Clarification questions may have one, two or three referring expressions and are penalized accordingly. User answers are penalized only if they

introduce new referring expressions into the dialogue. Thus, the cost associated with each action and other rewards and penalties in our reward function is given below:

$$R = \begin{cases} 0.95 & \text{Select and Success} \\ -1 & \text{Select and Failure} \\ -0.05 & \text{Clarification: } A? \\ -0.1 & \text{Clarification: } A \text{ or } B? \\ -0.15 & \text{Clarification: } A \text{ or } B \text{ or } C? \\ -0.05 & \text{New Color Description} \\ -1 & \text{Dialogue exceeds maximum length} \end{cases}$$

These individual actions rewards are accumulated as the dialogue proceeds and the model is optimized to maximize the average cumulative reward.

5 RL Experiment

To optimize the system's policy, we use the DQN algorithm (Mnih et al., 2015). In DQN, we have two neural nets: a policy network, θ_p, and a target network, θ_t. Both networks estimate the expected long-term reward of taking action a in state s, but the policy network is adjusted dynamically via back-propagation during training while the target network updates more slowly.

We use experience replay memory. We repeatedly sample actions as preferred by the policy network under an ε-greedy policy with exponential decay. This gives a transition from state s to s' and an immediate reward $r(s,a,s')$. This transition is inserted into a experience replay buffer so that a mini-batch of transitions can be sampled from it for training. We make sure each sampled mini-batch always includes the most recent transition (Zhang and Sutton, 2017).

In training, the target network is used to get an estimate over the future utility of the next state s', which is used to minimize the temporal difference error δ of the policy network:

$$\delta = Q^{\theta_p}(s,a) - (r(a) + \gamma * max_{a' \in A} Q^{\theta_t}(s',a')),$$

where A is the action set and γ is the discount factor; δ drives an update to θ_p by back-propagation using the Huber loss. We update the target network after each iteration using a smoothing parameter τ (Fujimoto et al., 2018):

$$\theta_t^{i+1} = \theta_p^i * \tau + \theta_t^i * (1 - \tau)$$

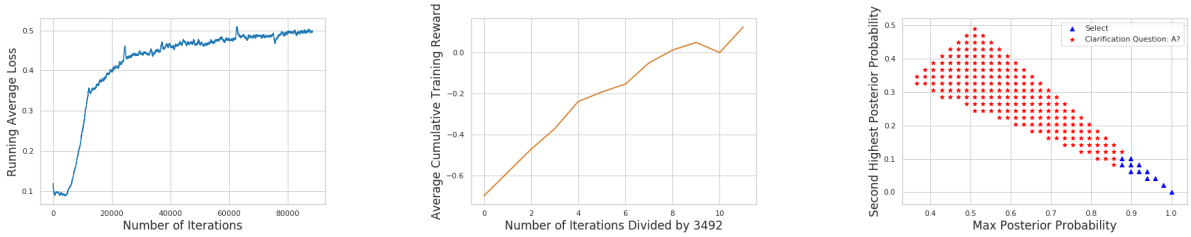
5.1 Training details

We trained the model on 5000 conversations from the CIC dataset and used 500 unseen CIC game samples from the test to evaluate the model performance. Our experience memory is a circular buffer of size $15,000$. We set $\gamma = 0.99$, $\alpha = 0.000075$, $\tau = 0.1$, and use *Adam* for parameter adjustment with a mini-batch of size 64. After each pass over the training set, the performance of model over unseen samples was evaluated and training was stopped when model had not improved for the past 5 epochs and difference of average loss over unseen samples between consecutive epoch was less than 10^{-2}. For consecutive iterations when absolute difference of loss over unseen samples was greater than 10^{-2}, learning rate was decreased by a factor of 0.1.

To train the model for the case where system can ask only one type of clarification question, we represent both the target and policy networks as linear models. For the multiple clarification strategy experiment, we represent the networks as two layered networks with ReLU activation applied on the output of hidden layer.

5.2 Learning Results

We train two types of RL models one which can choose between a *Confirmation question* and a *Select* action, and another which can choose between *Three Clarification* strategies and a *Select* action.



(a) Average Train Loss (b) Average Training Reward (c) Learned Policy

Figure 3: Learning to choose between Confirmation Question and Selection: (a) running average of loss during training; (b) running average of reward during training; and (c) a map of action decisions by the matcher in response to the director's initial utterance, as a function of the interpretive ambiguity across the three candidate referents.

	Learned Policy	Always Select	Always Clarify
Success Rate	0.92	0.83	0.95
Number of Turns	2.83	2.00	4.44
Average Reward	0.66	0.55	0.56
Clarification Round 1	0.30	0.00	1.00

Table 1: Characteristics and performance for the learned policy in simulation

Choosing Between Confirmation and Selection: As a first case study, we run an RL training session given the choice of actions *CQ1* and *S*. Figure 3 shows the training loss, average reward progression during training, and a policy visualization for this experiment.

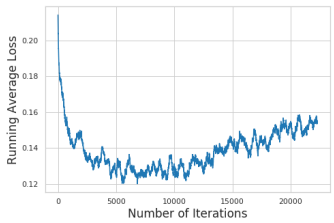
The policy visualization is particularly indicative of the learned behavior in the experiment and deserves close attention. The director's initial utterance is typically an identification action offering a description of the target square. All such cases lead to comparable state representations at the next system move, differing only in the probabilities they assign to the alternative target patches. Since there are three candidates, the space of probabilities is a 2-simplex, which we visualize here in order of weight creating an arrowhead shape: the greatest probability p (visualized on the x axis) can range from one third to 1, the second-largest probability q (visualized on the y axis) must be at least $(1-p)/2$ but cannot exceed $1-p$, and the third probability is $1-p-q$. At each point, we can query the policy with corresponding state representations to get a predicted action. As Figure 3(c) shows, the policy prefers selection at the tip of the arrow, where the choice is clear, and clarification elsewhere.

Table 1 draws the comparison of performance parameters of the learned policy against two baseline policies: **1)** Always Select at first turn and **2)** Always clarify at first turn. The profile of the learned policy shows that the RL framework learns an appropriate balance between dialogue length and task success to maximize average reward. This experiment shows that our approach can replicate established methods for using RL to optimize dialogue thresholds.

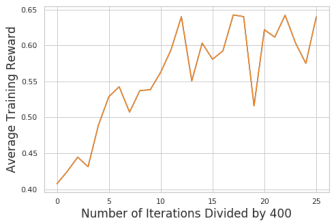
Choosing Between Multiple Clarification Strategies: Now we run an RL training session to decide among all the available actions: *CQ1*, *CQ2*, *CQ3* and *S*. Again, we visualize the average train loss, reward and the learned policy visualization: see Figure 4.

Here too, the policy visualization is particularly informative. As before, the tip of the arrowhead shows selection. Now there is a peripheral area at the lower right where there is strong evidence for one dominant interpretation, but still substantial ambiguity: here the system asks the lightweight confirmation question. Meanwhile, at the top of the wedge, we have cases where the top two referents dominate the options and the third can be effectively discounted. Here the learned policy asks the two-alternative clarification question. The remainder of the policy space has sufficient evidence for all the possibilities that the best option is to use an exhaustive clarification question.

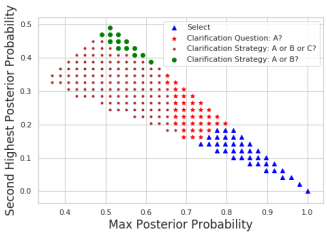
As the visualization shows, the decision thresholds for the different choices combine information



(a) Average Train Loss



(b) Average Training Reward



(c) Learned Policy

Figure 4: Plots showing the performance of multi-clarification learned policy: (a) running average of loss during training; (b) running average of reward during training; and (c) a map of action decisions by the matcher in response to the director's initial utterance, as a function of the interpretive ambiguity across the three candidate referents.

director:	Purple
Matcher:	Light purple, bright purple or blue?
Director:	bright purple
Matcher:	[SELECT]

Figure 5: An example conversation of the user with our system that demonstrates the effectiveness of the use of clarification questions for the successful completion of the task.

from all the candidate referents. As an indication of the responsiveness of these patterns to dialogue outcomes, Table 2 contrasts the performance of the learned policy in simulation to a baseline strategy that considers only the top candidate (but which we handcrafted to ensure that it beat uniform baselines): Select when $P_{max} >= 0.95$, clarify using strategy *A?* when $P_{max} >= 0.80$, clarify using strategy *A or B?* when $P_{max} >= 0.60$ and clarify using strategy *A or B or C?*

	Learned Policy	Baseline: Mixed Policy
Success Rate	0.90	0.85
Average Length	2.63	3.70
Average Rewards	0.73	0.58
Clarification Rate	0.24	0.49

Table 2: Baseline multi-clarification strategy versus multi-clarification policy learned through RL.

6 Human Evaluation

We ran human evaluation experiments to assess how effective learned policies were in human interactions. We are interested not only in task success, but in the kinds of questions the system asks, the kinds of answers it gets, and users' impressions of the system.

Protocol. This study was conducted with the approval of our human subjects review committee. We recruited 60 subjects through Amazon Mechanical Turk. Participants were all US citizens, gave written consent, and were compensated at an estimated rate of USD 15 an hour. Each subject played 4 games with the system out of which 2 were selected from the *close* condition, 1 from the *far* condition and 1 from the *split* condition. After each trial, subjects were asked to rate the performance of the system on a scale of 0 to 5 and leave us feedback. We conducted experiments with a baseline system, which clarifies whenever the probability of the most likely referent is less than 0.95, as well as with the two RL systems reported in Section 5.

Results. Figure 5 shows an example conversation of the user with our system that shows how the use of clarification leads to a natural and effective conversation in this task. The rates of successful trials are reported in Table 3. We can see from the results of the human evaluation that the single-clarification system is comparable with the rule-based baseline. Although we see an improvement in the results of the multi-clarification model, the results of the t-tests show that the differences are not statistically significant ($p > 0.05, t > 0.942$). The more capable RL models received increasingly high overall ratings. The average overall ratings of the baseline model, the RL-SignalClarification model, and the RL-MultiClarification model are respectively 3.70, 4.33, and 4.66. The differences between the ratings are statistically significant according to the results of the t-tests ($p < 0.01, t > 11.172$).

		condition			average
		far	close	split	
human evaluation	Baseline	0.947	0.685	0.941	0.816
	RL-SingleClarification	0.952	0.750	0.809	0.827
	RL-MultiClarification	1.0	0.772	1.0	0.865

Table 3: Success rates in human evaluation trials and the expected success rates.

The distribution of different types of clarification questions in human evaluation trials shows that our analysis of system policy carries over to live trials. Out of total dialogue moves conducted by the model, clarification questions were asked 46% of the time. Out of these, the first type of clarification question "A?" was asked 54.5% of the time and the second and third type appeared 18.2% and 27.3% respectively. Further, we study whether clarification questions elicit correct answers. In human evaluation trials, the accuracy of the "A?" clarification question is 57% and the accuracy of the "A or B?" and the "A or B or C" questions are respectively 71% and 76%. These differences underscore the importance of modeling human errors at the level of responses to individual dialogue moves.

In comparing the simulation model to actual performance, we observed that further breakdowns are necessary to capture the inherent task difficulty. The RL model expects relatively flat performance across the different conditions: 0.884 for close, 0.817 for far and 0.827 for split. In fact the differences are much greater—a natural direction for improvement as we iterate this methodology.

7 Discussion and Conclusion

In this paper, we have shown how a model of user utterances, capturing word choice and anticipating both user errors and system difficulties, can be used to optimize the fine-grained interactive strategy to resolve interpretive ambiguities in referential communication.

The work has a number of limitations. It captures only some of the clarification strategies people expect in this domain, and doubtless more sophisticated domains involve more complex referential strategies and problem solving. Even for color swatches, an important missing case concerns utterances that compare referents rather than characterizing them (*brighter* rather than *bright*). The user simulation is only roughly data driven and captures only some of the factors that influence speaker choice; the reward function likewise represents a coarse approximation to user satisfaction. Ultimately, we would like the system to derive its user models from its own interactions and learn the features that predict user satisfaction and user choice.

More generally, this work is only an initial step towards more flexible learned models of situated, collaborative interaction, and applying the technique to a broader range of problems, including explanation, diagnosis and negotiation, is crucial for the next generation of dialogue applications. We are hopeful about exploring such directions in future work.

Acknowledgement

We thank Brian McMahan for technical assistance with his data and models. This work was supported by NSF IIS-1526723 and CCF-19349243 and has benefited from discussions with Abdeslam Boularias, Alex Lascarides, Amelie Marian, David Pennock, Yongfeng Zhang and anonymous reviewers.

References

Mattias Appelgren and Alex Lascarides. 2020. Interactive task learning via embodied corrective feedback. *Autonomous Agents and Multi-Agent Systems*, 34(2):1–45.

Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13(2):259–294.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

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

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 18:233–263.

David DeVault and Matthew Stone. 2009. Learning to interpret utterances using dialogue history. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 184–192, Athens, Greece, March. Association for Computational Linguistics.

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, Michigan, June. Association for Computational Linguistics.

Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. *ArXiv*, abs/1802.09477.

George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Jonathan Ginzburg and Robin Cooper. 2004. Clarification, ellipsis, and the nature of contextual updates in dialogue. *Linguistics and philosophy*, 27(3):297–365.

Peter A. Heeman and Graeme Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*, 21(3):351–382.

Peter Heeman. 2007. Combining reinformation learning with information-state update rules. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 268–275, Rochester, New York, April. Association for Computational Linguistics.

Youngsoo Jang, Jongmin Lee, and Kee-Eung Kim. 2020. Bayes-adaptive monte-carlo planning and learning for goal-oriented dialogues. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7994–8001. AAAI Press.

Baber Khalid, Malihe Alikhani, Michael Fellner, Brian McMahan, and Matthew Stone. 2020. Discourse coherence, reference grounding and goal oriented dialogue. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Virtually at Brandeis, Waltham, New Jersey, July. SEMDIAL.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340.

Alex Lascarides and Nicholas Asher. 2009. Agreement, disputes and commitment in dialogue. *Journal of Semantics*, 26(2):109–158.

Ramesh Manuvinakurike, David DeVault, and Kallirroi Georgila. 2017. Using reinforcement learning to model incrementality in a fast-paced dialogue game. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 331–341, Saarbrücken, Germany, August. Association for Computational Linguistics.

Brian McMahan and Matthew Stone. 2015. A Bayesian model of grounded color semantics. *Transactions of the Association for Computational Linguistics*, 3:103–115.

Brian McMahan and Matthew Stone. 2020. Analyzing speaker strategy in referential communication. In *The 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2020)*.

Susan W McRoy and Graeme Hirst. 1995. The repair of speech act misunderstandings by abductive inference. *Computational linguistics*, 21(4):435–478.

Timothy Meo, Brian McMahan, and Matthew Stone. 2014. Generating and resolving vague color references. In *SEMDIAL 2014: The 18th Workshop on the Semantics and Pragmatics of Dialogue*, pages 107–115.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

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

Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338.

Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. 2019. Recursive visual attention in visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6679–6688.

Tim Paek and Eric Horvitz. 2000. Conversation as action under uncertainty. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 455–464.

David Schlangen. 2004. Causes and strategies for requesting clarification in dialogue. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 136–143.

Ravi Shekhar, Alberto Testoni, Raquel Fernández, and Raffaella Bernardi. 2019. Jointly learning to see, ask, decide when to stop, and then guesswhat. In *CLiC-it*.

David R Traum. 1994. *A Computational Theory of Grounding in Natural Language Conversation.* Ph.D. thesis, University of Rochester.

Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. *arXiv preprint cmp-lg/9704004*.

Marilyn A Walker. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.

S. Young, M. Gašić, B. Thomson, and J. D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Shangtong Zhang and Richard S. Sutton. 2017. A deeper look at experience replay. *ArXiv*, abs/1712.01275.

Jiaping Zhang, Tiancheng Zhao, and Zhou Yu. 2018. Multimodal hierarchical reinforcement learning policy for task-oriented visual dialog. *arXiv preprint arXiv:1805.03257*.