Reference-Centric Models for Grounded Collaborative Dialogue

Daniel Fried[†] **Justin T. Chiu**[‡] **Dan Klein**[†]
[†]Computer Science Division, UC Berkeley
[‡]Department of Computer Science, Cornell Tech
{dfried,klein}@cs.berkeley.edu, jtc257@cornell.edu

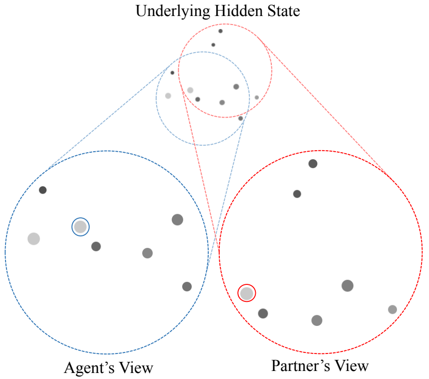
Abstract

We present a grounded neural dialogue model that successfully collaborates with people in a partially-observable reference game. We focus on a setting where two agents each observe an overlapping part of a world context and need to identify and agree on some object they share. Therefore, the agents should pool their information and communicate pragmatically to solve the task. Our dialogue agent accurately grounds referents from the partner's utterances using a structured reference resolver, conditions on these referents using a recurrent memory, and uses a pragmatic generation procedure to ensure the partner can resolve the references the agent produces. We evaluate on the OneCommon spatial grounding dialogue task (Udagawa and Aizawa, 2019), involving a number of dots arranged on a board with continuously varying positions, sizes, and shades. Our agent substantially outperforms the previous state of the art for the task, obtaining a 20% relative improvement in successful task completion in self-play evaluations and a 50% relative improvement in success in human evaluations.

1 Introduction

In grounded dialogue settings involving high degrees of ambiguity, correctly interpreting and informatively generating language can prove challenging. Consider the collaborative dialogue game shown in Figure 1. Each player has a separate, but overlapping, view on an underlying context. They need to communicate to determine and agree on one dot that they share, and both players win if they choose the same dot. To succeed, each participant must—implicitly or explicitly—ground their partner's descriptions to their own context, maintain a history of what's been described and what their partner is likely to have, and informatively convey parts of their own context.

We present a grounded pragmatic dialogue system which collaborates successfully with people on the task above. Figure 1 shows a real example game between our system and a human partner. Our approach is centered around a structured module for perceptually-grounded reference resolution. This reference resolution module plays two roles. First, the module is used to *interpret* the partner's utterances: explicitly predicting which referents (if any) in the agent's context the partner is referring to, for example *a smaller darker grey dot* and *the lightest grey and larger dot*. Second, the reference module is used for *pragmatic generation*: choosing utterances by reasoning about how the partner might interpret them in context. Our pragmatic generation procedure selects referents to describe as well as choosing how to describe them, for example focusing on *the light one* (Figure 1).



A: I have two large light grey dots with a smaller darker dot below and to the right of it
P: I have one large lightest grey dot in my entire view
A: I have two large light grey dots
P: There is a smaller slightly darker grey dot next to the lightest grey and larger dot
A: Yes , let's pick the light one
P: The light grey and large dot?
A: Yes it 's the one . Let's pick that one
P: SELECT red
A: SELECT blue

Figure 1: An example dialogue produced by our system (A) with a human partner (P). The participants have different but overlapping views of a shared board, which contains dots of different shapes and sizes. The partners must collaborate through dialogue in order to find and select a dot that is visible to both.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2130–2147
November 7–11, 2021. ©2021 Association for Computational Linguistics

Much past work that has constructed systems for grounded collaborative dialogue has focused on settings that have asymmetric player roles (Kim et al., 2019; de Vries et al., 2018; Das et al., 2018, 2017), are fully-observable, or are grounded in symbolic attributes (He et al., 2017). In contrast, we focus on the ONECOMMON corpus and task (Udagawa and Aizawa, 2019), which is symmetric, partially-observable, and has relatively complex spatial and perceptual grounding. These traits necessitate complex dialogue strategies such as common grounding, coordination, clarification questions, and nuanced acknowledgment (Udagawa and Aizawa, 2019), leading to the task being challenging even for pairs of human partners.

Past work on ONECOMMON has focused on the subtask of reference resolution (Udagawa and Aizawa, 2020; Udagawa et al., 2020) and only evaluated dialogue systems automatically: using static evaluation on human–human games and self-play evaluations that simulate human partners using another copy of the agent. Our system outperforms this past work on these evaluations. We further confirm these results by performing—for the first time on this task—human evaluations, where we find that our system obtains a 50% relative increase in success rate over a system from past work when paired with human partners. We release code for our system at https://github.com/dpfried/onecommon.

2 Setting

We choose to focus on the ONECOMMON task (Udagawa and Aizawa, 2019) since it is a particularly challenging representative of a class of partially-observable collaborative reference dialogue games (e.g., He et al. 2017; Haber et al. 2019). In this task, two players have different but overlapping views of a game board, which consists of dots of various positions, shades of gray and sizes. The players must coordinate to choose a single dot that both players can see, which is challenging because neither knows which dots the other can see.

Each player's world view, w, consists of a circular view on an underlying board containing between 8 and 10 randomly scattered dots, with continuously varying positions, shades, and sizes (Figure 1). Each player's view contains 7 dots, and the views of the players overlap so that there are between 4 and 6 dots which appear in both views.

We focus on a turn-based version of the dialogue task. In a given turn t, a player may communicate with their partner by either sending an utterance u_t or selecting a dot s. In the event of selection, the partner is notified but cannot see which dot the player has selected. Once a player has selected a dot, they can no longer send messages. The dialogue ends once both players have selected a dot, and is successful if both selected the same one.

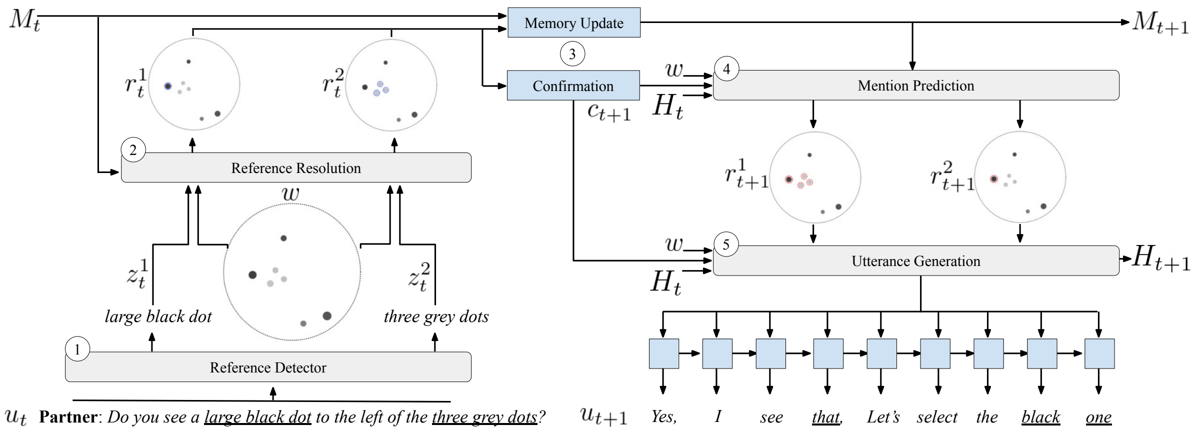
3 Model Structure

Our approach is a modular neural dialogue model which factors the agent's generation process into a series of successive subtasks, all centered on grounding language into referents in the world context. In this section, we describe our model structure, which defines a neural module for each subtask. We then describe our reference-centric pragmatic generation procedure in Section 4.

An overview of the relationship between modules in our model is shown in Figure 2. Each module can condition on neural encodings of the context (the world and past dialogue), as well as the outputs of other modules. We describe our system at a high-level here, then give task-specific implementation details about each component in Section 5.

3.1 Context Encodings

Our modules can condition on encodings of (i) the past utterances $u_{1:t}$ in the dialogue, represented as a memory vector H_t produced by a word-level recurrent encoder and (ii) the continuous dots in the world context w, produced by the entity encoding network of Udagawa and Aizawa (2020), which produces a vector $w(d)$ for each dot d encoding the dot's continuous attributes as well as its pairwise attribute relationships to all other dots in the context (Santoro et al., 2017). (i) and (ii) both follow Udagawa and Aizawa (2020). To explicitly encourage the model to retain and use information about the history of referents mentioned by both players, which affects the choice of future referents as well as the selection of dot at the end of the game, we also use (iii) a structured recurrent **referent memory** grounded in the context. This memory, inspired by He et al. (2017), has one representation for each dot d in the agent's view, $M_t(d)$, which is updated based on the referents predicted in turn t. See Section 5.4 for details.



Partner: *Do you see a large black dot to the left of the three grey dots?*

Yes, I see that, Let's select the black one

Figure 2: In a given turn, an agent first identifies referring expressions in their partner's utterance u_t using the reference detector (1). Each reference is then resolved with the reference resolution module (2), which uses encoded representations $z^{1:K_t}$ of the reference segments and the world context w. The referents are then used to update the referent memory M_t, and cross-referenced against the agent's own dots to confirm whether or not the agent can also see them (3). Given the referent memory M_t and confirmation variable c_{t+1}, the mention prediction module (4) produces a sequence of dot configurations $z_{t+1}^{1:K_{t+1}}$ to mention. Finally, the utterance generation module (5) uses the dialog history H_t, confirmation variable, and attended representations of the selected mentions and world context to generate a response u_{t+1}.

3.2 Decomposing Turns into Subtasks

We assume turn $t+1$ in the dialogue has the following generative process (numbers correspond to Figure 2). Steps (1) and (2) identify and resolve referring expressions in the partner's utterance u_t; step (3) updates the memory and determines whether the model can confirm any referents from the partner's utterance; steps (4) and (5) produce the agent's next utterance u_{t+1}.

(1) First, a sequence of K_t (with $K_t \geq 0$) referring expressions are identified in u_t using the **reference detector** tagging model of Udagawa and Aizawa (2020)[1], and encodings $\mathbf{z}_t = z_t^{1:K_t}$ are obtained for them by pooling features from a recurrent utterance encoder.

(2) Then, the referring expressions are grounded. From each referring expression's features z^k, we predict a referent r^k, which is the set of zero or more dots in the agent's own view which are described by the referring expression. For example, the referring expression *three gray dots* corresponds to a single referent containing three dots. A **reference resolution** module $P_R(\mathbf{r}_t \mid \mathbf{z}_t, w, M)$, where $\mathbf{r}_t = r_t^{1:K_t}$, predicts a sequence of referents, one for each referring expression.

(3) Given these referents, the agent updates the referent memory M_t using the predicted referents and constructs a discrete **confirmation variable** c_{t+1}, which indicates whether the agent can confirm in its next utterance that it has all the referents the partner is describing (e.g., *Yes, I see that*). c_{t+1} takes on one of three values: NA if no referring expressions were in the partner's utterance, YES if all of the partner's referring expressions have referents that are at least partially visible in the agent's view, and NO otherwise.

(4) The agent chooses a sequence of referents to mention next using a **mention prediction** module $P_M(\mathbf{r}_{t+1} \mid c_{t+1}, M_{t+1}, H_t, w)$.

(5) Finally, the next utterance u_{t+1} is produced using an **utterance generation** module $P_U(u_{t+1} \mid \mathbf{r}_{t+1}, c_{t+1}, H_t, w)$, also updating the word-level recurrent memory H_{t+1}.

At the end of the dialogue (turn T), the agent selects a dot s using a **choice selection** module $P_S(s \mid H_T, M_T, w)$ (not shown in Figure 2).[2]

Modules that predict referents (reference resolution, mention selection, and choice selection) are all implemented using a structured conditional random field (CRF) architecture (Section 5.2), with independent parameterizations for each module.

Our model bears some similarities to Udagawa and Aizawa (2020)'s neural dialogue model for this task: both models use a reference resolution module[3] and both models attend to similar encodings

[1] Udagawa and Aizawa refer to this as a *markable detector* given their work's focus on referent annotation.

[2] The choice selection module is invoked when the utterance generation model predicts a special <SELECT> token, following Udagawa and Aizawa (2020).

[3] Our model, however, uses a structured CRF while Udagawa and Aizawa's model does not use structured output modeling.

of the dots in the agent's world view ($w(d)$) when generating language. Crucially, however, our decomposition of generation into subtasks results in a factored, hierarchical generation procedure: our model identifies and then conditions on previously-mentioned referents from the partner's utterances,[4] maintains a structured referent memory updated at each utterance, and explicitly predicts which referents to mention in each of the agent's own utterances. In Section 4, we will see how factoring the generation procedure in this way allows us to use a pragmatic generation procedure, and in Section 6 we find that each of these components improves performance.

4 Pragmatic Generation

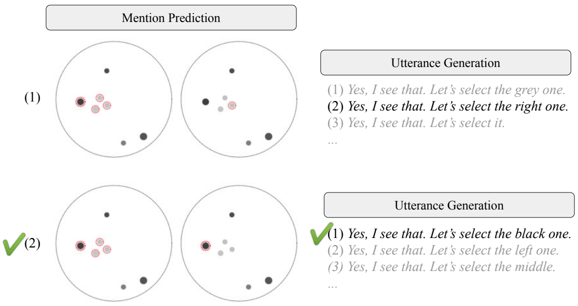
The modules as described above can be used to generate the next utterance u_{t+1} using the predictions of $P_M(\mathbf{r}_{t+1})$ and $P_U(u_{t+1}|\mathbf{r}_{t+1})$ (omitting other conditioning variables from the notation for brevity; see Section 3 for the full conditioning contexts). This section describes an improvement, *pragmatic generation*, to this process. Referents and their expressions should be relevant in the dialogue and world context, but they should also be discriminative (Dale, 1989): allowing the listener to easily understand which referents the speaker is intending to describe. Our pragmatic generation approach, based on the Rational Speech Acts (RSA) framework (Frank and Goodman, 2012; Goodman and Frank, 2016), uses the reference resolution module, $P_R(\mathbf{r}_{t+1}|u_{t+1})$, to predict whether the partner can identify the intended referents. This encourages selecting referents that are easy for the partner to identify and describing them informatively in context.[5]

We use the following objective over referents $\mathbf{r}$ and utterances u for a given turn:

$$\begin{aligned} &\arg\max_{\mathbf{r},u} \; L(\mathbf{r}, u) \\ &\quad L(\mathbf{r}, u) = P_M(\mathbf{r})^{w_M} \cdot P_U(u|\mathbf{r})^{w_S} \cdot P_R(\mathbf{r}|u)^{w_L} \end{aligned} \tag{1}$$

where w_M, w_S, and w_L are hyperparameters.

Partner: *Do you see a large black dot to the left of the three grey dots?*



(1) *Yes, I see that. Let's select the grey one.*
(2) *Yes, I see that. Let's select the right one.*
(3) *Yes, I see that. Let's select it.*
...

✓(1) *Yes, I see that. Let's select the black one.*
(2) *Yes, I see that. Let's select the left one.*
(3) *Yes, I see that. Let's select the middle.*
...

Figure 3: Agents optimize for a combination of fluency and informativity during pragmatic utterance generation (Section 4 and Algorithm 1). A set of paired candidate referents (from the mention prediction module) and utterances (from the utterance generation module) is rescored using $L(\mathbf{r}, u)$ (Equation 1), a weighted geometric mean of scores from the mention prediction, utterance, and reference resolution modules. The pair of referent and utterance that maximizes this score is chosen as a response.

This objective generalizes the typical RSA setup (as implemented by the weighted pragmatic inference objective of e.g., Andreas and Klein 2016 and Monroe et al. 2017), which chooses *how* to describe a given context (i.e., choosing an utterance u), to also choose *what* context to describe (i.e., choosing the referents $\mathbf{r}$). Our objective also models the tradeoff, explored in past work on referring expression generation (Dale, 1989; Jordan and Walker, 2005; Viethen et al., 2011), between producing utterances relevant in the discourse and world context and producing utterances that are discriminative. We use P_U and P_M to model discourse and world relevance, P_S to model discriminability, and the weights w to empirically model the tradeoff between them.

Given the combinatorially-large spaces of possible $\mathbf{r}$ and u, we rely on an early-stopping approximate search, which to our knowledge is novel for RSA. The search (illustrated in Figure 3) iterates through the highest probability structured referent sequences $\mathbf{r}$ under the mention prediction module P_M (Figure 3 shows the top two) and for each $\mathbf{r}$ sampling N_u utterances u from the utterance generation module (Figure 3 shows three u per $\mathbf{r}$). If the maximum of these $(\mathbf{r}, u)$ pairs under L is better than an early-stopping threshold value τ, we return the pair. Otherwise, we continue on to the next $\mathbf{r}$. If more than N_r referent sequences have been evaluated, we return the best $(\mathbf{r}, u)$ pair found so far. See Appendix C for pseudocode and a discussion of robustness to the threshold τ.

[4] Udagawa and Aizawa used the reference resolution module only to define an auxiliary loss at training time.

[5] Note that the reference resolution model, which has access to the agent's own view and not the partner's, can only approximate whether the referents are identifiable by the partner; nevertheless we find that it is beneficial for pragmatic generation. Future work might explore also inferring and using the partner's view.

5 Module Implementations

As described so far, our system is applicable to a range of partially-observable grounded collaborative referring expression dialogue tasks (e.g., He et al. 2017; Haber et al. 2019). In this section, we describe implementations of our systems' modules, some of which are tailored to OneCommon.

5.1 Reference Detection

We identify a sequence of referring expressions in the utterance using the **reference detector** of Udagawa and Aizawa (2020), a BiLSTM-CRF tagger (Huang et al., 2015). Then, following Udagawa and Aizawa (2020), we obtain features z^k for each of the K referring expressions in the utterance (for use in the reference resolution model) with a bidirectional recurrent encoder, using learned weights to pool the encodings at the referring expression's boundaries as well as the end of the utterance.[6]

5.2 Structured Reference Resolution

We use a structured reference resolution module to ground the referring expressions identified above: identifying dots in the agent's own view described by each expression. Grounding referents in this domain involves reasoning not only about attributes of individual dots but also spatially and comparatively within a single referring expression (e.g., *a line of three dots*) or across referring expressions (e.g., *a large grey dot left of a smaller dot*).

To predict a sequence of referents $\mathbf{r} = r^{1:K}$ from the K referring expression representations $z^{1:K}$ extracted above, we use a linear-chain CRF (Lafferty et al., 2001) with neural potentials to parameterize $P_R(r^{1:K}|z^{1:K}, w, M)$. This architecture generalizes the reference resolution and choice selection models of Udagawa and Aizawa (2020) and Udagawa et al. (2020) to model, in the output structure, relationships between dots, both inside and across referring expressions.

There are three different types of potentials, designed to model language-conditioned features of individual dots d in a referent r, ϕ; relationships within a referent, ψ, and transitions between successive referents, ω. Given these potentials, the distribution is parameterized as

$$P(r^{1:K}|z^{1:K}) \propto$$
$$\exp\left(\sum_k f(r^k, z^k) + \psi(r^k, z^k) + \omega(r^{k:k+1}, z^{k:k+1})\right),$$

where $f(r, z) = \sum_{d \in r} \phi(d, z)$, and we've omitted the dependence of all terms on M and w for brevity. We share all module parameters across the two subtasks of resolving referents for the agent and for the partner.[7]

Individual Dots. Dot potentials ϕ model the correspondence between language features z^k and individual dots represented by encodings $w(d)$, as well as discourse salience using the dot-level memory $M(d)$ that tracks when the dot d has been mentioned:[8]

$$\phi(d, z^k) = \text{MLP}_\phi([M(d), z^k, w(d)])$$

Dot Configurations. Configuration potentials $\psi(r^k, z^k)$ model the correspondence between language features and the set of all active dots in the agent's view for a referent r^k. These potentials further decompose into (1) pairwise potentials between active dots in the configuration, which relate the language embedding z^k to attribute differences between dots in the pair (including as relative position, size, and shade) and (2) a potential on the entire configuration, which relates the language embedding to an embedding for the count of active dots in the configuration. See Appendix A.1 for more detail.

Configuration Transitions. Transition potentials $\omega(r^k, r^{k+1}, z^k, z^{k+1})$ model the correspondence between language features and relationships between referring expressions, e.g., *to the left of* in *the black dot to the left of the triangle of gray dots*. See Appendix A.1 for details.

5.3 Confirmations

When applied to the partner's utterances, the reference resolution module gives a distribution over which referents the *partner* is likely to be referring to in the *agent's* own context. If the agent can identify the referents its partner is describing, it should be able to confirm them, both in the dots it

[6] The bidirectional encoder only has access to the utterances that have been produced so far, i.e., $u_{1:t}$ when the agent is generating utterance u_{t+1}.

[7] Parameters are shared for efficiency; sharing had little effect on performance in preliminary experiments.

[8] For the subtasks of reference resolution and choice selection, dot potentials are the same as the *attention module* used by Udagawa and Aizawa (2020), with the addition of the memory M.

talks about next (e.g., choosing to refer to one of the same dots the partner identified) and in the text of the utterances (e.g., *yes, I see it*). The discrete-valued confirmation variable (defined in Section 3) models this, taking the value NA if no referring expressions were identified in the partner's utterance, YES if all of the $K > 0$ referring expressions have a non-empty referent (at least one dot predicted in the agent's context) and NO otherwise.

5.4 Referent Memory

The memory state is composed of one state vector $M_t(d)$ for each dot in the agent's own context. These dot states are updated using the referents identified in each utterance. This update is parameterized using a decoder cell, which is applied separately to each dot state:

$$M_{t+1}(d) = \text{RNN}_C(M_t(d), \iota(d, \mathbf{r}_t))$$

where ι is a function that extracts features from the predictive distribution over referents from the previous utterance, representing mentions of dot d in the referring expressions. We implement the cell using a GRU (Cho et al., 2014). See Appendix A.2 for more details.

5.5 Mention Selection

The mention selection subtask requires predicting a sequence of referents to mention in the agent's next utterance, $P_M(\mathbf{r}_{t+1} \mid u_{1:t}, M_{t+1}, c_{t+1}, w)$. To produce these referents, we use the same structured CRF architecture as the reference resolution module P_R. However, we use separate parameters from that module, and instead of the referring-expression inputs $\mathbf{z}$ use a sequence of vectors $x^{1:K_{t+1}}$ produced by a decoder cell RNN_M, implemented using a GRU (Cho et al., 2014). The decoder conditions on the dialogue context representation H_t from the end of the last utterance, a learned vector embedding for the confirmation variable c_{t+1}, and a mean-pooled representation of the memory $m = \frac{1}{|d|}\sum_d M(d)$:

$$x^k = \text{RNN}_M(x^{k-1}, [H_t, c_{t+1}, m])$$

We obtain the number of referents K_{t+1} by predicting at each step k whether to halt from each x^k using a linear layer followed by a logistic function.

5.6 Choice Selection

To parameterize the choice selection module $P_S(s \mid u_{1:T}, M_T, w)$, we again reuse the CRF architecture, with independent parameters from reference resolution and mention selection modules, replacing reference resolution's inputs $z^{1:K}$ with the dialogue context representation H_T from the end of the final utterance in the dialogue. Since only a single dot needs to be identified, we use only the CRF's individual dot potentials ϕ, removing ψ and ω. This is equivalent to the choice selection model (TSEL) used by Udagawa and Aizawa (2020) if the recurrent memory M_T is removed.

5.7 Utterance Generation

The utterance generation module $P_U(u_{t+1}|\mathbf{r}_{t+1}, c_{t+1}, H_t, w)$ is a sequence-to-sequence model. The module first encodes the sequence of dot encodings $w(d)$ for dots in the referents $z_{t+1}^{1:K_{t+1}}$ (predicted by the mention selection module) to produce encodings $y_t^{1:K}$. Words in the utterance are then produced one at a time using a recurrent decoder that has a hidden state initialized with a function that combines $y_t^{1:K}$, the dialog context H_t, and a learned embedding for the discrete confirmation variable c_{t+1}. The decoder has two attention mechanisms over: (i) dot encodings $w(d)$, following Udagawa and Aizawa (2020), and (ii) the sequence of encoded referents $y_t^{1:K_{t+1}}$. See Appendix A.3 for details.

6 Experiments

We compare our approach to past systems for the ONECOMMON dataset. While our primary evaluation is to evaluate systems on their success rate on the full dialogue game when paired with human partners (Section 6.4), we also compare our system to past work, and ablated versions of our full system, using the automatic evaluations of past work.

6.1 Models

We compare our full system (FULL) to ablated versions of it that successively remove: (i) the referent memory, ablating explicit tracking of referents mentioned (F–MEM) and (ii) the structured potentials ψ, γ in the reference resolution and mention selection modules (F–MEM–STRUC), removing explicit modeling of relationships within and across referents. We also compare to a reimplementation of the system of Udagawa and Aizawa (2020), which we found obtained better performance than their reported results in all evaluation conditions due to implementation improvements. See Appendix A.5.

Model	Choice Acc.	Ref. Resolution Acc.	 Ex.
U&A (2020)	69.3±2.0	86.4±0.4	35.0±2.0
U+ (2020)	–	86.0±0.3	54.9±0.8
F–MEM–STRUC	71.6±0.9	87.7±0.2	44.3±0.5
F–MEM	70.9±1.2	92.6±0.2	76.2±0.5
FULL	83.3±1.2	93.3±0.2	78.2±0.5
Human	90.8	96.3	86.9

Table 1: Accuracies for predicting the dot selected at the end of the game (Choice Acc.) and resolving referents from utterances produced in the agent's own perspective (dot-level accuracy Acc. and exact match Ex.) in 10-fold cross-validation on the corpus. Our FULL model outperforms all past work on the dataset: U&A (2020) is our reimplementation of Udagawa and Aizawa (2020), and U+ (2020) are taken from Udagawa et al. (2020). Human scores are annotator agreements (Udagawa and Aizawa, 2019).

We obtain supervision for all components of the systems by training on the referent-annotated corpus of 5,191 successful human–human dialogues collected by Udagawa and Aizawa (2019; 2020). See Appendix A.6 for training details. We train one copy of each model on each of the corpus's 10 cross-validation splits. We report means and standard deviations across the splits' models, except in human evaluations where we use a single model.

6.2 Corpus Evaluation

Following Udagawa and Aizawa (2020), we evaluate models' accuracy at (1) predicting the dot chosen at the end of the game (Choice Acc.) using P_S and (2) resolving the referents in utterances from the human partner in the dialogue who had the agent's view (Ref Resolution dot-level accuracy Acc. and exact match accuracy Ex.) using P_R.

We see in Table 1 that our FULL model improves substantially on past work, including the work of Udagawa et al. (2020), who augment their referent resolution model with numeric features. Our structured reference resolver is able to learn these features in its potentials ψ (in addition to other structured relationships), and improves exact match from 44% to 76% compared to the ablated version of our system. Our recurrent memory helps in particular for the choice selection task, improving from 71% to 83% accuracy.

We also compare the performance of our full and ablated systems on the tasks of resolving the partner's referring expressions and mention prediction, with results given in Appendix B.

Model	#Shared=4	#Shared=5	#Shared=6
U&A (2020)	50.7±2.0	66.0±1.9	83.5±1.5
F–MEM–STRUC	42.3±2.1	57.0±2.1	75.4±1.1
F–MEM	52.6±1.5	67.1±1.9	84.1±1.6
FULL	58.5±2.7	71.6±2.9	86.8±1.8
FULL+PRAG	62.4±2.2	74.7±2.7	90.9±1.4
Human	65.8	77.0	87.0

Table 2: Task success rates in automatic self-play evaluations, by difficulty of context (the number of items shared in the players' views). Our FULL model outperforms past work: U&A (2020) is our tuned reimplementation of Udagawa and Aizawa (2020). Human shows success rates of trained human annotators in collecting the dataset (Udagawa and Aizawa, 2019).

6.3 Evaluation in Self-Play

To evaluate systems on the full dialogue task, we first use self-play, where a system is partnered with a copy of itself, following Udagawa and Aizawa (2020). We evaluate systems on 3,000 world contexts, stratified into contexts with 4, 5, and 6 dots overlapping between the two agents' views, with 1,000 contexts in each stratification.

Table 2 reports average task success (the fraction of times both agents chose the same dot at the end of the dialogue) averaged across the 10 copies of each model trained on the cross-validation splits. As in the corpus evaluation, we see substantial improvements to our system from the structured referent prediction and the recurrent reference memory. Our Full system, without pragmatic generation, improves over the system of Udagawa and Aizawa (2020) from 51% to 58% in the hardest setting, with a further improvement to 62% when adding our pragmatic generation procedure.

6.4 Human Evaluation

Finally, we perform human evaluation by comparing system performance when playing with workers from Amazon's Mechanical Turk (MTurk). To conduct evaluation, we used 100 world states from the #Shared=4 partition, and collected 718 complete dialogues by randomly pairing worker with one of the following three: our best-performing model in self-play (FULL+PRAG), the model from Udagawa and Aizawa (2020), or another worker.

In order to ensure higher quality dialogues, and following Udagawa and Aizawa (2019), we filtered workers by qualifications, showed workers a game tutorial before playing, and prevented dots from being selected within the first minute of the game. We

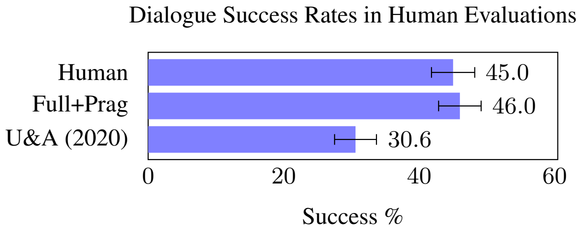


Figure 4: Success rates of systems on the full dialogue game task when paired with human partners. Error bars show standard errors. Our FULL+PRAG system achieves a 50% relative performance improvement over past work (U&A 2020: Udagawa and Aizawa 2020).

paid workers \$0.30 per game, with a bonus of \$0.15 if the dialogue was successful. See Appendix E for sample dialogues.

We compare systems based on the percentage of successful dialogues. The results, in Figure 4, corroborate the trends observed in self-play. Both the models of U&A (2020) and our FULL+PRAG perform worse against humans than against agent partners in the automatic self-play evaluation, illustrating the importance of performing human evaluations. However, the trend is preserved, and we see that the FULL+PRAG system substantially outperforms the U&A (2020) model, resulting in a 50% relative improvement in task success rate. This difference is statistically significant at the $p \leq 0.05$ level using a one-tailed t-test.

6.5 Success by Human Skill Level

In Section 6.4, we compared our systems to a human population of MTurk workers. However, human populations themselves vary greatly based on many factors, including the day and time workers are recruited, training and feedback given to workers, and worker retention. One difference between our worker population and the population that produced the dataset is training. When collecting the dataset, Udagawa and Aizawa (2019) performed manual and individualized coaching of their MTurk workers which made them more effective at the game: giving players personalized feedback on how to improve their game strategies, e.g., “please ask more clarification questions.”[9] Manual coaching produced a high-quality corpus by increasing players’ skill and obtained a success rate of 66%; however coaching would make human evaluations difficult to replicate across works due to the labor, cost, and variability that coaching involves.

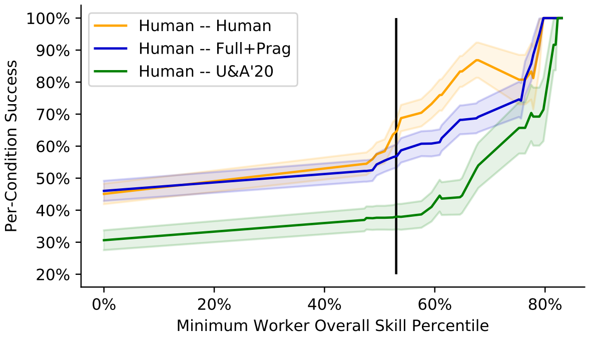


Figure 5: Success rates of human players against each system type, and other humans, with progressive filtering of humans by their overall success rate (across all conditions) along the x-axis. Shaded regions give standard errors. Our FULL+PRAG system outperforms past work (U&A 2020) at all levels.[10]

In this section, we run a sweep of system comparisons of the form of Section 6.4, but on increasingly select sub-populations of MTurk workers. Results are shown in Fig. 5. The x-axis gives the minimum skill percentile for a worker’s games to be retained (with a worker’s skill defined to be their average success across all games; see Appendix D for an alternative), so that the far left of the graph shows all workers (corresponding to the numbers in Fig. 4), the far right shows only those workers who won all of their games, and the black vertical line marks the player filtering needed to obtain a human-human success rate comparable to Udagawa and Aizawa (2019). Our FULL+PRAG system outperforms the model of Udagawa and Aizawa (2020) at all player skill levels.[10] This result shows that, while more accomplished workers’ overall success rates can be much higher than the success rate of our general worker population, in all cases the ordering between the two systems remained the same.

7 Related Work

Goal-oriented dialog. The modular approach that we use reflects the pipelined approach often used in goal-oriented dialogue systems (Young et al., 2013). Recent work on neural systems has also used structured and memory-based approaches (Bordes et al., 2017; He et al., 2018) including

[9] Udagawa and Aizawa also manually removed around 1% of dialogues where workers did not follow instructions. While we do not perform post-hoc manual filtering of the dialogues, in order to avoid introducing systematic bias that would favor or disfavor one of the systems we compare, an inspection of a subset of our collected dialogues indicates a similarly high fraction of our workers were making a good effort at the task.

[10] Until, by necessity, the point where filtering removes all workers who lost a game against any system. Differences between U&A’20 and FULL+PRAG are significant at the $p \leq 0.05$ level by a one-tailed t-test for minimum worker overall skills up to the 68th percentile.

tracking entities identified in text (Williams et al., 2017; He et al., 2017). We also find improvements from an entity-centric approach with a structured memory, although our domain involves more challenging entity resolution and generation due to the spatial grounding.

Referring expressions. A long line of past work on referring expression grounding has tackled generation (Dale, 1989; Dale and Reiter, 1995; Viethen et al., 2011; Krahmer and van Deemter, 2012), interpretation (Schlangen et al., 2009; Liu et al., 2013; Kennington and Schlangen, 2015) or both (Heeman, 1991; Mao et al., 2016; Yu et al., 2017). Closest to ours is the work of Takmaz et al. (2020), which builds models for reference interpretation and generation in the rich PhotoBook corpus (Haber et al., 2019), focusing on a non-interactive setting with static evaluation on reference chains extracted from human-human dialogues.

Collaborative games. The closest work on dialogue systems for collaborative grounded tasks has focused on tasks with different properties from ours, as discussed in Section 1. A closely related task to the shared visual reference game we pursue here is the PhotoBook task (Haber et al., 2019), although a dialogue system has not been constructed for it. Other work on grounded collaborative language games includes collection games (Potts, 2012; Suhr et al., 2019), navigation and interactive question games (Thomason et al., 2019; Nguyen and Daumé III, 2019; Ilinykh et al., 2019), and construction tasks (Wang et al., 2017; Kim et al., 2019; Narayan-Chen et al., 2019).

Pragmatics. Our approach to pragmatics (Grice, 1975) builds on a large body of work in the RSA framework (Frank and Goodman, 2012; Goodman and Frank, 2016), which models how speakers and listeners reason about each other to communicate successfully. The most similar applications to ours in past work on computational pragmatics have been to single-turn grounded reference tasks (rather than dialogue), with much smaller and unstructured spaces of referents than ours,[11] such as discriminative image captioning (Vedantam et al., 2017; Andreas and Klein, 2016; Cohn-Gordon et al., 2018) and referent identification (Monroe et al., 2017; McDowell and Goodman, 2019; White et al., 2020). Explicit speaker–listener models of pragmatics have also been used for dialogue, and while these approaches plan or infer across multiple turns (which our work does not do explicitly), they have either involved ungrounded settings (Kim et al., 2020) or constrained language (Vogel et al., 2013; Khani et al., 2018).

8 Conclusion

We presented a modular, reference-centric approach to a challenging partially-observable grounded collaborative dialogue task. Our approach is centered around a structured referent grounding module, which we use (1) to interpret a partner's utterances and (2) to enable a pragmatic generation procedure that encourages the agent's utterances to be able to be understood in context. We perform, for the first time, human evaluations on the full dialogue task, finding that our system cooperates with people substantially more successfully than a system from past work and—in aggregate—achieves a success rate comparable to pairs of human partners.

While our results are encouraging, there is still much room for improving all systems in their interactions with people on this challenging task. As the examples in Appendix E illustrate, people use sophisticated conversational strategies to build common ground (Clark and Wilkes-Gibbs, 1986; Traum, 1994; Clark, 1996) when they interact with each other, producing utterances that play multiple conversational roles and performing complex reasoning. To better plan utterances (Cohen and Perrault, 1979) and more accurately infer the partner's state (Allen and Perrault, 1980), we suspect it will be helpful to extend the single-step pragmatic utterance planning and implicit inference procedures that we use here: planning over longer time horizons, performing more explicit reasoning under uncertainty, and learning richer models of the full range of speech acts that people use. Future work might continue to explore these directions on this task and other similarly challenging tests of collaborative grounding.

Acknowledgments

We are grateful to Takuma Udagawa for sharing information about the corpus collection, and to Yoav Artzi, Chris Potts, Aida Nematzadeh, the Berkeley NLP group, and the anonymous reviewers for helpful suggestions and feedback. This work was supported by DARPA through the XAI program and by a Google PhD Fellowship to DF.

[11] Our setting has 2^7 possible referents for each referring expression in the dialogue.

References

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

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, Texas. Association for Computational Linguistics.

John Langshaw Austin. 1962. *How to Do Things With Words*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Herbert H. Clark. 1996. *Using Language*.

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

Philip R. Cohen and C. Raymond Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3).

Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2018. Pragmatically informative image captioning with character-level inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Robert Dale. 1989. Cooking up referring expressions. In *27th Annual Meeting of the Association for Computational Linguistics*.

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

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating New York City through grounded dialogue. *ArXiv preprint*, abs/1807.03367.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*.

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

H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The PhotoBook dataset: Building common ground through visually-grounded dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Peter A. Heeman. 1991. Collaborating on referring expressions. In *29th Annual Meeting of the Association for Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *ArXiv preprint*, abs/1508.01991.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Meet up! a corpus of joint activity dialogues in a visual environment. In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*.

Pamela W. Jordan and Marilyn A. Walker. 2005. Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24(1).

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference* *on Natural Language Processing (Volume 1: Long Papers)*.

Fereshte Khani, Noah D. Goodman, and Percy Liang. 2018. Planning, inference and pragmatics in sequential language games. *Transactions of the Association for Computational Linguistics*, 6.

Hyunwoo Kim, Byeongchang Kim, and Gunhee Kim. 2020. Will I sound like me? improving persona consistency in dialogues through pragmatic self-consciousness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jin-Hwa Kim, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh. 2019. CoDraw: Collaborative drawing as a testbed for grounded goal-driven communication. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*.

Changsong Liu, Rui Fang, Lanbo She, and Joyce Chai. 2013. Modeling collaborative referring for situated referential grounding. In *Proceedings of the SIGDIAL 2013 Conference*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*.

Bill McDowell and Noah Goodman. 2019. Learning from omission. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Will Monroe, Robert X.D. 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.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

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

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Tim Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a Bayesian filtering model that is sensitive to disfluencies. In *Proceedings of the SIGDIAL 2009 Conference*.

John R Searle. 1976. A classification of illocutionary acts. *Language in Society*, 5(1).

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Ece Takmaz, Mario Giulianelli, Sandro Pezzelle, Arabella Sinclair, and Raquel Fernández. 2020. Refer, Reuse, Reduce: Generating Subsequent References in Visual and Conversational Contexts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Thora Tenbrink and Reinhard Moratz. 2003. Group-based spatial reference in linguistic human-robot interaction. *Spatial Cognition and Computation*, 6.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. In *Conference on Robot Learning (CoRL)*.

David R. Traum. 1994. A computational theory of grounding in natural language conversation. Technical report, Rochester University Department of Computer Science.

Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Takuma Udagawa and Akiko Aizawa. 2020. An annotated corpus of reference resolution for interpreting common grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Takuma Udagawa, Takato Yamazaki, and Akiko Aizawa. 2020. A linguistic analysis of visually grounded dialogues based on spatial expressions. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Jette Viethen, Robert Dale, and Markus Guhe. 2011. Generating subsequent reference in shared visual scenes: Computation vs re-use. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Adam Vogel, Max Bodoia, Christopher Potts, and Daniel Jurafsky. 2013. Emergence of Gricean maxims from multi-agent decision theory. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Sida I. Wang, Samuel Ginn, Percy Liang, and Christopher D. Manning. 2017. Naturalizing a programming language via interactive learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Julia White, Jesse Mu, and Noah D Goodman. 2020. Learning to refer informatively by amortizing pragmatic reasoning. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*.

Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5).

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L. Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

A Model Details

A.1 Structured CRF

Dot Configurations. Dot configuration potentials $\psi(r, z)$ are composed of two terms: $R(r, z)$ which decomposes into functions of pairwise relationships between the dots (whether active or not) in the context w and the text features z, and $A(r, z)$ which is a function of all active dots in the referent:

$$\psi(r, z) = R(r, z) + A(r, z)$$

The pairwise relationships are

$$R(r, z) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \alpha(r, z, i, j)$$

where N is the number of dots in view (7) and α is a scalar-valued neural function of the text features and whether the dots indexed by i and j are active in the referent r:

$$\alpha(r, z, i, j) = \begin{cases} p(z, i, j)_0, & r(i) \wedge r(j) \\ p(z, i, j)_1, & \neg r(i) \wedge \neg r(j) \\ p(z, i, j)_2, & \text{otherwise} \end{cases}$$

p is a 3-dimensional vector produced by an MLP:

$$p(z, i, j) = \text{MLP}_\psi([w(i) - w(j), z])$$

The active dot potential A is designed to model group properties such as cardinality and common attributes, which other work has found useful on this and similar tasks (Tenbrink and Moratz, 2003; Udagawa et al., 2020). We define the potential as

$$A(r, z) = \text{MLP}_A([w(r), e(r)])$$

where $w(r)$ is the mean of the feature values for the active dots in r, $\frac{1}{|r_{active}|} \sum_{d \in r_{active}} w(d)$ and $e(r)$ is a learned 40-dimensional embedding for the discrete count of active dots in r.

Configuration Transitions. The configuration transition potential $\omega(r^{k:k+1}, z^{k:k+1})$ is similar to the dot configuration potential above but bridges the dots in referents k and $k+1$. It is the sum of two terms: $\omega(r^{k:k+1}, z^{k:k+1}) = S(r^{k:k+1}, z^{k:k+1}) + B(r^{k:k+1}, z^{k:k+1})$. First is S, which decomposes into pairwise relationships between dots across referents r^k and r^{k+1}:

$$S(r^{k:k+1}, z^{k:k+1}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \beta(r^{k:k+1}, z^{k:k+1}, i, j)$$

$$\beta(r^{k:k+1}, z^{k:k+1}, i, j) = \begin{cases} q_0, & r^k(i) \wedge r^{k+1}(j) \\ q_1, & \neg r^k(i) \wedge \neg r^{k+1}(j) \\ q_2, & \text{otherwise} \end{cases}$$

q (short for $q(z^{k:k+1}, i, j)$) is, like p in Dot Configurations, a 3-dimensional vector produced by an MLP:

$$q = \text{MLP}_\omega([w(i) - w(j), z^k - z^{k+1}])$$

Next is B, which is a function of the feature centroids of the active dots in referents k and $k+1$. B is defined analogously to A in Dot Configurations:

$$\begin{aligned} &B(r^{k:k+1}, z^{k:k+1}) \\ &\quad = \text{MLP}_B([w(r^k) - w(r^{k+1}), z^k - z^{k+1}]) \end{aligned}$$

with $w(r)$ again giving the mean of the feature values for the active dots in r. We fix $B(r^{k:k+1}, z^{k:k+1}) = 0$ if $|r^k_{active}| > 3$ or $|r^{k+1}_{active}| > 3$, which had little effect on model accuracy but improves memory efficiency as it substantially reduces the number of group-pairwise relationships that need to be computed.

Inference. We compute the normalizing constant for the CRF distribution by enumerating the possible 2^7 assignments to each r^k to compute the ϕ, ψ, and ω potential terms, which can be performed efficiently on a GPU. To compute the normalizing constant, which sums over all combinations of assignments to these r^k, we use the standard linear-chain dynamic program. In training, we backpropagate through the enumeration and dynamic program steps to pass gradients to the parameters of the potential functions.

A.2 Referent Memory

The function ι collapses predicted values for the dot d over K referents into a single representation for the dot, which we do in two ways: by max- and average- pooling predicted values for d across the K referents. We also obtain the prediction values in two ways: by taking the argmax structured prediction from P_R, and by taking the argmax predictions from each dot's marginal distribution. We found that using these "hard" argmax predicted values gave slightly better results in early experiments than using the "soft" probabilities from P_R. In combination, these give four feature values as the output of $\iota(d, \mathbf{r}_t)$.

A.3 Utterance Generation Module

We first use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to encode the sequence of K referents-to-mention $\mathbf{r}_{t+1} = r^{1:K}_{t+1}$, using the inputs at each position $k \in [1, K]$ a mean-pooled representation of the world context embeddings for the active dots in the referent: $\frac{1}{|r^k_{t+1}|} \sum_{d \in r^k_{t+1}} w(d)$, to produce a sequence of encoded vectors $y^{1:K}_t$. We make gated updates to the decoder's initial state, updating it with (i) a linear projection of the forward and backward vectors for y^1_t and y^K_t, representing the referent context and (ii) an embedding for the discrete confirmation variable c_{t+1}.

A.4 Implementation Choices

For our reimplementation of the system of Udagawa and Aizawa (2020) in a shared codebase with our system, we replace their $tanh$ non-linearities with `ReLU`s and use PyTorch's default initializations for all parameters. These improve performance across all evaluation conditions in comparison to the reported results.

For our system, we use separate word-level recurrent models, a Reader and a Writer, to summarize the dialogue history. The Reader is bidirectional over each utterance, and is used in the reference resolution and choice selection modules. The Writer is unidirectional, and is used in the mention selection and utterance generation modules.

A.5 Hyperparameters

Recurrent Unit Hyperparameters	
Reader GRU size	512
Writer GRU size	512
Mention decoder RNN$_M$ size	512
Referent memory RNN$_C$ size	64
Confirmation embedding c size	512
CRF Hyperparameters	
MLP$_\phi$ hidden layers	2
MLP$_\phi$ hidden size	256
MLP$_\phi$ dropout	0.5
MLP$_\psi$ and MLP$_\omega$ hidden size	64
MLP$_\psi$ and MLP$_\omega$ dropout	0.2
MLP$_\psi$ and MLP$_\omega$ hidden layers	1
MLP$_A$ and MLP$_B$ hidden size	64
MLP$_A$ and MLP$_B$ dropout	0.2
MLP$_A$ and MLP$_B$ hidden layers	1
Generation Hyperparameters	
Sampling temperature in P_U	0.25
# Utterance candidates, N_u	100
# Referent candidates, N_r	20
Mention weight, w_M	0
Speaker weight, w_S	1×10^{-3}
Listener weight, w_L	$1 - w_S$
Early-stopping threshold, τ	0.8

	Partner Refs.		Next Refs
Model	Acc.	Ex.	Ex.
F–MEM–STRUC	87.3±0.3	41.8±0.9	4.8±1.0
F–MEM	90.6±0.3	65.2±1.0	23.5±2.0
FULL	91.2±0.4	67.0±1.0	31.1±1.0

Table 3: Accuracies for resolving referents in the *partner*'s view (dot-level accuracy Acc. and exact match Ex.) and predicting the next referents to mention in the dialogue (Next Refs Ex.) in 10-fold cross-validation on the corpus of human–human dialogues. Our FULL benefits from its recurrent referent memory (outperforming F–MEM) and structured referent prediction module (outperforming F–MEM–STRUC).

A.6 Training Details

For our full system and ablations, we train on each cross-validation fold for 12 epochs using the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 1×10^{-3} and early stopping on the fold's validation set. Our loss function is a weighted combination of losses for the subtask objectives:

$$\mathcal{L} = w_S \log P_S(s|...) + \frac{1}{T}\sum_{t=1}^{T}(\log P_R(r_t|...) + \log P_M(r_t|...) + \log P_U(u_t|...)),$$

where w_S is a hyperparameter which we set to $\frac{1}{32}$ following Udagawa and Aizawa (2020) and we have omitted conditioning contexts from the probability distributions for brevity; see Section 3.2 for the full contexts. We decay the learning rate when the loss plateaus on validation data.

We train models on a Quadro RTX 6000 GPU. Training takes around 1 day for models that use the structured CRF, and several hours without the structured CRF. Self-play evaluation takes around 1 hour.

B Evaluation for Other Subtasks

Table 3 gives performance accuracies for resolving referents in the *partner*'s view (dot-level accuracy Acc. and exact match Ex.) and predicting the next referents to mention in the dialogue (Next Refs Ex.) in 10-fold cross-validation on the corpus of human–human dialogues. We observe improvements from both the recurrent memory (comparing F-Mem to Full) and the structured referent prediction module (comparing F-Mem-Struc to Full) on both tasks.

C Pragmatic Generation

We give pseudocode for the pragmatic generation procedure (Section 4) in Algorithm 1. Figure 3 shows an example, showing 2 referents $\mathbf{r}$ (inputs to the `realize`) function on the left, and 3 utterances u sampled for each referent on the right. Fewer than N_r referent candidates may be evaluated (as in Figure 3) if one $(\mathbf{r}, u)$ pair is found with $L(\mathbf{r}, u) \geq \tau$.

```
hyperparameters: $N_r, N_u, \tau$
function generate($M, u_{1:t-1}, c_t, w$):
    $(\hat{\mathbf{r}}, \hat{u}, \hat{s}) \leftarrow$ (None, None, $-\infty$)
    for $\mathbf{r} \in \text{topk}_{N_r}\ P_M(\mathbf{r}|u_{1:t-1}, M, c, w)$ :
        $u, s \leftarrow$realize($\mathbf{r}$)
        if $s > \hat{s}$ :
            $(\hat{\mathbf{r}}, \hat{u}, \hat{s}) \leftarrow (\mathbf{r}, u, s)$
            if $s \geq \tau$ :
                break
    return $\hat{\mathbf{r}}, \hat{u}$
function realize($\mathbf{r}$):
    for $k \in 1 \ldots N_u$ :
        $u^{(k)} \sim P_U(\cdot \mid \mathbf{r})$
    $\hat{u} \leftarrow \arg\max_{u^{(k)}} L(\mathbf{r}, u^{(k)})$ (Eqn. 1 in Sec. 4)
    $\hat{s} \leftarrow \max_{u^{(k)}} L(\mathbf{r}, u^{(k)})$
    return $(\hat{u}, \hat{s})$
```

Algorithm 1: Our pragmatic generation procedure chooses a sequence of referents $\mathbf{r}$ to describe, and an utterance u to describe them, to optimize the objective $L(\mathbf{r}, u)$ (Equation 1 in Section 4) using candidates from the models P_M and P_U and an early stopping search with threshold τ.

We used self-play evaluation on one of the cross-validation splits to tune the early-stopping threshold τ, selecting from among the values {0.0, 0.6, 0.7, 0.8, 0.9}. The optimal value was $\tau = 0.8$, but the success rate in self-play was fairly robust to the value chosen (including $\tau = 0.0$, which results in performing pragmatic search only over those utterances for the single highest-scoring referent sequence under P_M), with a range of about 2%. We did not evaluate without early-stopping (searching over all candidate reference sequences and utterances) as this would have made generation too computationally expensive to be feasible in both self-play and human evaluations.

D Alternative Skill Analysis

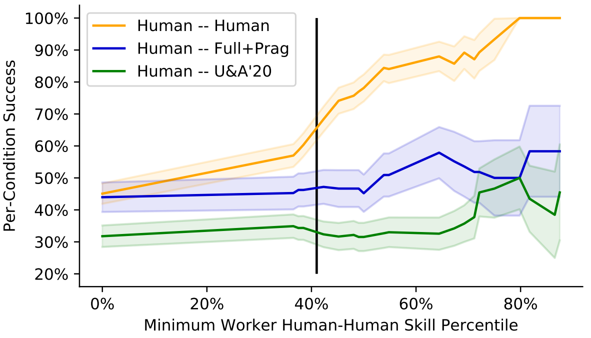


Figure 6: Success rates of human players against each system type, and other humans, with progressive filtering of humans by their overall success rate (when partnered with other humans) along the x-axis. Shaded regions give standard errors. Our FULL+PRAG system outperforms the model from Udagawa and Aizawa (2020) at all levels.[10]

In Section 6.5, we compared systems on increasingly select sub-populations of MTurk workers, selected by their average success across all conditions (whether playing with other humans or one of the two system types). In this section, we run a similar analysis but select workers by their average success when paired with *human partners only*. Results are shown in Figure 6. The x-axis gives the minimum skill percentile for a worker's games to be retained, with skill defined by a worker's average success when paired *with other human workers*. The far left of the graph shows all workers,[12] the far right shows only those workers who won all of their games when paired with other workers, and the black vertical line marks the player filtering needed to obtain a human-human success rate comparable to Udagawa and Aizawa (2019). As we saw in Section 6.5, our FULL+PRAG system outperforms the model of Udagawa and Aizawa (2020) at all worker skill levels. However, focusing on the sub-population of workers who are successful when paired with other humans (the right side of Figure 6) reveals a gap between humans and our system: humans who are successful when partnering with other humans are substantially less successful when partnering with our FULL+PRAG system (and even less successful when partnering with the model of U&A'20). This indicates room for improvement on the task, as we want to build a system that can collaborate as well as humans with any population of human partners.

E Dialogue Examples

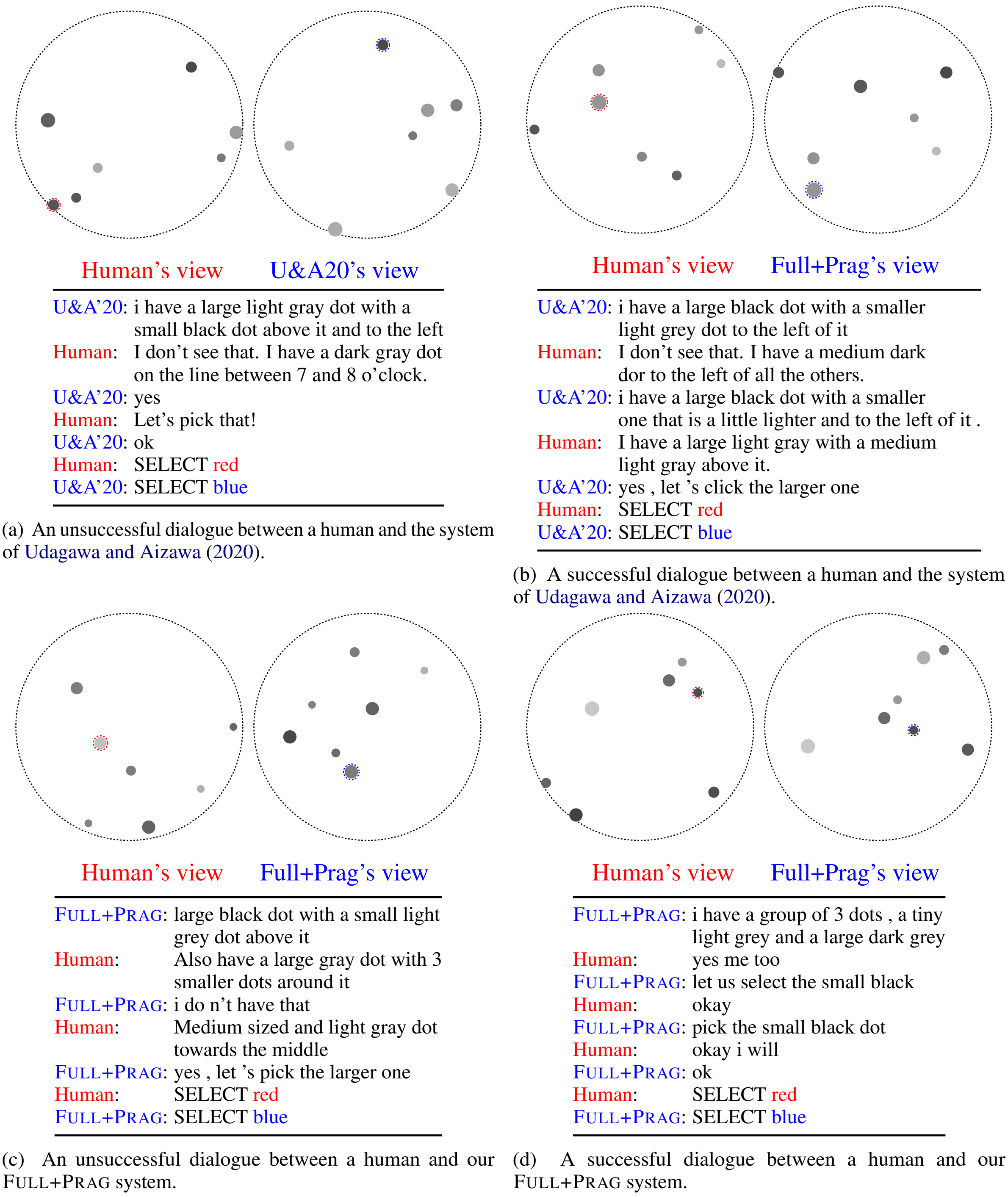
We show one successful and one failed dialogue from our human evaluations (Section 6.4) for each system (Figure 7) and from human–human pairs

[12] After filtering to remove any workers who did not play at least one game with another human worker.

(Figure 8).

As seen in these examples, descriptions from the baseline system (Figures 7a and 7b) typically have a consistent syntactic structure (e.g., “i have a <size> <color> dot with a <size> <color> dot <spatial relation>”) but often do not correspond to the visual context. We suspect that it is difficult for this end-to-end generation model to simultaneously learn which dots to talk about (content selection) and how to describe them (surface realization) with the amount of training data available. Our FULL+PRAG system (Figures 7c and 7d) produces broader and generally more accurate utterances, which we attribute to our factored and pragmatic generation procedure.

Our system’s utterances still have substantial qualitative differences from those in human-human dialogues (Figure 8), which—due to the richness of the task (Udagawa and Aizawa, 2019)—often use more complex strategies. Human strategies can unfold across multiple turns, e.g., introducing information in installments or referring to the same dot in multiple turns without being repetitive, as A does when providing more information about the “light grey dot“ in Figure 8a. Sophisticated strategies are also used even in single turns, e.g., in Figure 8b, B’s utterance “is one on top of the other? if so pick the top one” combines multiple types of speech act (Austin, 1962; Searle, 1976): implicitly acknowledging A’s utterance, asserting new information about the dots in view, and issuing a command.



Human's view U&A20's view

U&A'20: i have a large light gray dot with a small black dot above it and to the left
Human: I don't see that. I have a dark gray dot on the line between 7 and 8 o'clock.
U&A'20: yes
Human: Let's pick that!
U&A'20: ok
Human: SELECT red
U&A'20: SELECT blue

(a) An unsuccessful dialogue between a human and the system of Udagawa and Aizawa (2020).

Human's view Full+Prag's view

U&A'20: i have a large black dot with a smaller light grey dot to the left of it
Human: I don't see that. I have a medium dark dor to the left of all the others.
U&A'20: i have a large black dot with a smaller one that is a little lighter and to the left of it .
Human: I have a large light gray with a medium light gray above it.
U&A'20: yes , let 's click the larger one
Human: SELECT red
U&A'20: SELECT blue

(b) A successful dialogue between a human and the system of Udagawa and Aizawa (2020).

Human's view Full+Prag's view

FULL+PRAG: large black dot with a small light grey dot above it
Human: Also have a large gray dot with 3 smaller dots around it
FULL+PRAG: i do n't have that
Human: Medium sized and light gray dot towards the middle
FULL+PRAG: yes , let 's pick the larger one
Human: SELECT red
FULL+PRAG: SELECT blue

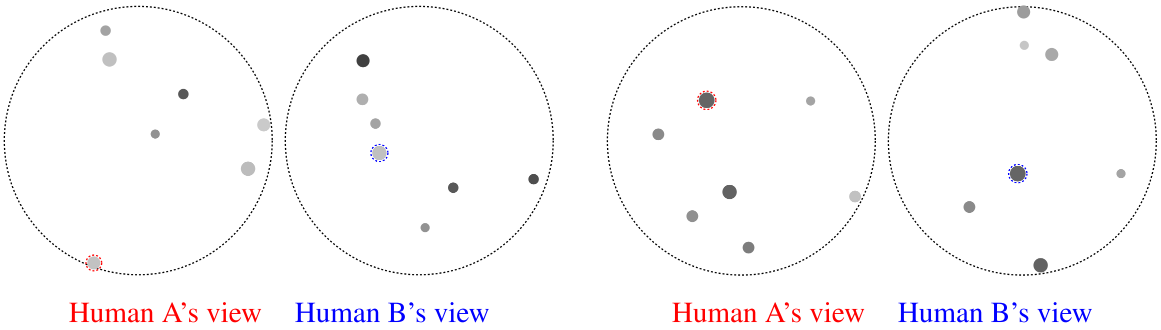
(c) An unsuccessful dialogue between a human and our FULL+PRAG system.

Human's view Full+Prag's view

FULL+PRAG: i have a group of 3 dots , a tiny light grey and a large dark grey
Human: yes me too
FULL+PRAG: let us select the small black
Human: okay
FULL+PRAG: pick the small black dot
Human: okay i will
FULL+PRAG: ok
Human: SELECT red
FULL+PRAG: SELECT blue

(d) A successful dialogue between a human and our FULL+PRAG system.

Figure 7: Example dialogues collected during our human evaluation (Section 6.4) of the dialogue systems. We show one unsuccessful (left) and one successful (right) example for each system. The top row is our reimplementation of Udagawa and Aizawa (2020) and the bottom row is FULL+PRAG, our full system with pragmatic inference.



Human B: three light grey dots in a diagonal line
Human A: i dont have that but i have a black dot neer the top to the right, the only black dot in the circle
Human B: i have two black dots. find something else
Human A: ok i have a light grey dot by itself at the bottom to the left. right on the line
Human B: how big is it
Human A: its one of the bigger ones
Human B: okay just pick it then
Human A: ok
Human B: SELECT blue
Human A: SELECT red

Human B: the smallest lightest grey dot. it's near a larger grey dot
Human A: did you see 3 dark dots in same line
Human B: no
Human A: did you see two larger black dots
Human B: is one on top of the other? if so, pick the top one
Human A: ok
Human B: SELECT blue
Human A: SELECT red

Figure 8: Examples of unsuccessful (left) and successful (right) dialogues collected between pairs of people during our human evaluation (Section 6.4).