Modeling Perspective-Dependent Ambiguity in Collaborative Dialogue

Anonymous ACL submission

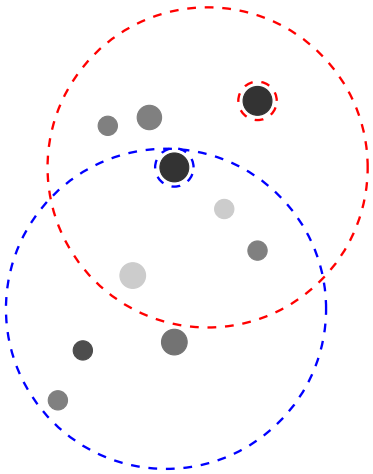
Abstract

Errors in reference resolution can occur if a dialogue agent reasons incorrectly about what their partner can see. We present a dialogue planner that reasons about its partner's perspective in order to produce referring expressions that the partner resolves correctly. The dialogue planner models its partner's perspective as a latent variable, embedded in a partner model that is used for both model-based planning and incorporating evidence from partner responses into the agent's belief over the partner's perspective. We evaluate the dialogue planner on ONECOMMON, a dialogue game where players have large differences in perspectives. In automatic evaluation of plan proposals, reasoning about the partner's perspective as a latent variable results in a 52% reduction in the total number of incorrectly resolved plans compared to a state-of-the-art baseline (Fried et al., 2021).

1 Introduction

Ambiguity is a key impediment to collaborative dialogue. Basic actions, such as reference resolution and generation, are difficult in the presence of ambiguity. One cause of ambiguity is differences in perspectives (Brown-Schmidt and Hanna, 2011). Partners with different perspectives may make errors in reference resolution because they mistakenly assume others' contexts are similar to their own. We refer to ambiguity induced by different perspectives as *perspective-dependent ambiguity*.

We provide an example of perspective-dependent ambiguity in Figure 1. The example is based on ONECOMMON, a collaborative reference game with a high degree of perspective-dependent ambiguity (Udagawa and Aizawa, 2019). In ONECOMMON, two partners are given different but overlapping views of a game board and must jointly identify a shared dot through dialogue. The game terminates after each partner selects a dot.



A: I have one large black dot by itself. Do you have it?
P: Yes, I do have that.
A: Let's pick that one.
P: <select> red
A: <select> blue

Figure 1: An example dialogue from a robot agent and human partner playing ONECOMMON, taken from the evaluation of Fried et al. (2021). The players have different but overlapping views of a shared board, which contains dots of different shapes and sizes. Players must collaborate through dialogue in order to find and select a shared dot. This dialogue ends in failure, as the players did not account for perspective-ambiguity and prematurely selected different dots.

The game is a success if they select the same dot and a failure otherwise. Perspective-ambiguity stems from the different but overlapping views, and the degree of ambiguity depends on the number of overlapping dots. In this example, the agent's description (*one large black dot*) is specific in their view, but actually resolves to a different dot in their partner's view — unknown to both the agent and partner. The incorrect resolution and confirmation has an immediate irrevocable consequence: The two players shortly select different dots, failing the game.

The false positive confirmation in the previous

example is an instance of an error resulting from using an egocentric heuristic to resolve ambiguity (Keysar et al., 2000). Rather than build or reason about common ground, the set of shared information or knowledge (Clark, 1996), dialogue participants utilizing an egocentric heuristic resolve referring expressions based solely on their own perspective, ignoring perspective-dependent ambiguity altogether. The egocentric heuristic fails to account for both the fact that only a subset of the agent's referential domain is shared and that their partner's referential domain has unobserved privileged items, which only the partner can see. While cognitively cheap, the egocentric heuristic can lead to systemic errors. Such errors are more pronounced with larger differences in perspectives, resulting in larger degrees of perspective-dependent ambiguity. Errors of this form can result in irrevocable mistakes that do not get corrected in follow-up interactions (Keysar et al., 2000).

We tackle the problem of perspective-dependent ambiguity by directly reasoning about both common ground that has been established through dialogue as well as unobserved aspects of the partner's perspective. We accomplish this by planning and updating our belief over common ground using a model of the partner that accounts for the partner's perspective. In the context of ONECOMMON, reasoning about common ground means reasoning about the shared dots, a natural object of consideration given the goal of finding a dot in common. We propose a partner model that treats the partner's unobserved perspective as a latent variable. Our approach maintains an explicit distribution over the shared dots, which is used to choose what dots to ask about based on their expected information gain under the partner model.

We demonstrate the ability of our method to prevent perspective-dependent ambiguity on ONECOMMON. We compare models using the *plans* (choices of dots to mention) they produce, evaluating whether plans resolve correctly to suitable referents in the partner context. We find that planning with a perspective-aware partner model makes fewer errors than an egocentric baseline from prior work (Fried et al., 2021). On a static validation corpus consisting of 2,341 turns from 518 human-human dialogues, our proposed model achieves a 47% reduction in the total number of incorrectly resolved plans compared to a state-of-the-art baseline (Fried et al., 2021).

2 Related work

The study of perspective and common ground has its roots in pragmatics and linguistics (Stalnaker, 1970; Clark and Marshall, 1981; Clark et al., 1983). Early behavioural experiments studying the egocentric heuristic found that it can lead to irrevocable errors in human studies (Keysar et al., 2000). More recently, computational work by Liu et al. (2021) and Doğan et al. (2020) utilize perspective to generate and understand referring expressions in a fully shared environment, but from different locations. Perspective also captures more than just spatial properties. Partner capability was considered in the context of reference games (Corona Rodriguez et al., 2019), where partner capabilities such as color-blindness were inferred in a single-turn visual reference game. In the referring expression and partner capability work, the perspectives of others were observed and directly conditioned on. Our work is inspired by these linguistic theories and experiments, and also provides a computational model for dealing with differences in perspectives in the dialogue setting. We demonstrate that mistakes due to perspective-dependent ambiguity in a dialogue setting with unobserved partner perspective can be reduced with an explicit latent-variable model of the partner.

Our work also builds on methods for asking informative questions. Prior work in this area formulates information-seeking conversation as a partially observable Markov decision process, and uses a single-turn partner-model based heuristic to generate questions (Rao and III, 2018; Yu et al., 2019). In these settings, there is one information seeker and one information holder. The information holder has a target referent in mind, which they must convey to the information seeker. The unobserved target referent is the only source of partial observability, meaning that information exchanged is reliable up to errors in language. In contrast, the different perspectives in ONECOMMON and resulting perspective-dependent ambiguity have a large effect on communication, often turning unambiguous utterances in the agent's context into ambiguous ones for their partner.

Aside from prior work that utilized single-turn planning with heuristics, other works apply multi-turn model-based planning to task-oriented dialogue, planning directly in the space of language via rollouts or tree-search (Lewis et al., 2017; Yarats and Lewis, 2017; Shridhar and Hsu, 2018;

Jang et al., 2020). Our work reduces perspective-dependent ambiguity by incorporating a distribution over unobserved context into the partner model. While we focus on improving a single-turn utility heuristic, improvements in partner models are complementary to multi-turn planning and can be combined in future work.

Our method builds on a state-of-the-art approach to ONECOMMON (Fried et al., 2021) in three ways: our model does not assume the partner's perspective is identical to the speaker's, the objective for planning is information gain rather than specificity, and we maintain an explicit belief over the partner's perspective. We also focus exclusively on planning with symbolic representations. By planning in symbolic space, we are able to control for ambiguity from choices in language, allowing us to focus on perspective-dependent ambiguity. We leave language generation to future work.

3 Background

Collaborative reference games pair partners together in order to identify and agree on a shared object through natural language dialogue . We focus on ONECOMMON (Udagawa and Aizawa, 2019), which represents a class of collaborative reference games (He et al., 2017; Haber et al., 2019) where only a subset of each player's referential domain is shared, resulting in perspective-dependent ambiguity.

In ONECOMMON, players take turn speaking and listening. At a given turn, an agent sees a referential domain $\mathcal{D}$ consisting of 7 dots. Each dot has a set of features: size, color, and position in the 2D plane. All of the features are continuous. We denote the partner's referential domain, which also consists of 7 dots, as z, and model it as a latent variable. Between 4-6 of these dots are shared by both players, but neither player knows which dots these are. We denote the *shared* intersection $s = \mathcal{D} \cap z$. We denote the partner's *unshared* dots, which are only visible to the partner, as $u = z \setminus s$. The size of s and u is the number of dots contained in each of them: $|s| + |u| = |z| = |\mathcal{D}| = 7$. The goal of the dialogue is to find and select a single dot in common, any element from s. Once an agent selects a dot, the partner does not observe the selected dot, and must select a dot of their own. If both players select the same dot they win, otherwise both players fail.

4 Planning with a partner model

Previous computational approaches to ONECOMMON do not take into account perspective-dependent ambiguity. We attribute a large percentage of game failures in a state-of-the-art approach (Fried et al., 2021) to ONECOMMON to perspective-dependent ambiguity, based on an analysis we perform of a sample of their system outputs. In the most challenging setting, 25% of games where humans were partnered with the previous model fail due to perspective-dependent ambiguity (out of a sample of 20 games taken from the 200 human-robot evaluation games in previous work). This is a direct result of the previous approach's decision to utilize the egocentric heuristic.[1]

Reasoning about the partner's perspective formally as a latent variable encourages the agent to avoid miscommunication by simulating how the partner would respond given different possible perspectives. This contrasts with a typical egocentric heuristic, which assumes the partner's perspective is identical to the agent's.

Central to our method is a partner model that, given the agent plan x and partner perspective z, simulates a possible partner response y. The model has the joint distribution

$$P(y, z \mid x) = P(y \mid z, x)P(z).$$

We define the partner perspective z and partner response y in a task specific manner, detailed in Section 5.

The primary use of this model is planning. At each turn, planning finds the plan x that maximizes the expected information gain (Quinlan, 1986), defined as

$$\operatorname*{argmax}_{x} \mathbb{E}_{y|x} \left[H[z] - H[z \mid y, x] \right],$$

where $H[z]$ is the entropy of the prior over partner perspective and $H[z \mid y, x]$ the posterior. The definition of information gain stems from the interpretation of entropy as uncertainty: the agent wants a plan the reduces their uncertainty about the partner's perspective. Note that $H[z]$ in the definition of information gain is constant with respect to the plan x, meaning it can be dropped from the objective. We provide more specific details in Section 6.1.

[1] Perspective-ambiguity does not account for all failures. Other causes for failure include a lack of attention from the human partner or misinterpretation of language.

A similar computation is used to update the agent's belief over the partner's perspective,

$$P(z \mid x, y) = P(y \mid z, x)P(z) / \sum_z P(y, z \mid x).$$

This is used to recursively update the belief $P(z)$ after observing the partner's response y. In contrast to an egocentric heuristic, where the agent would assume the partner's perspective is identical to their own, explicitly maintaining a belief over the partner's perspective allows the agent to account for the partner's perspective. The belief update requires marginalizing over the partner's perspective z; we provide the details of the belief update in Section 6.2.

5 Representation

We model the effect of perspective z on the partner response y to the agent plan x in ONECOMMON. The agent plan $x \subseteq \mathcal{D}$ is a subset of their referential domain that they ask the partner about. We decompose the partner's perspective and the partner's response, illustrated in Figure 2.

The full partner model which we use for planning in ONECOMMON is

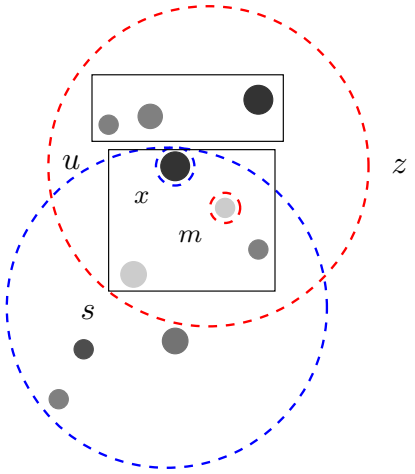
$$\begin{aligned} &P(y \mid x, z)P(z) \\ &= P(c, m, v \mid x, s, u)P(s, u) \\ &= P(v \mid c, m)P(c \mid x, s, u)P(m \mid s, u) \\ &\quad \cdot P(u \mid s)P(s), \end{aligned}$$

where the partner perspective z is decomposed into shared and unshared dots s and u, and the partner response y is decomposed into confirmation c, mentions (a set of reference dots) m, and utterance text v. We detail each of these decompositions in this section.

5.1 Partner perspective

For perspective prior $P(z) = P(s)P(u \mid s)$, we decompose the unobserved partner perspective z into the shared dots s and unshared dots u. The decomposition of perspective into shared and unshared dots allows the agent to reason separately about shared dots, which the agent can actually see, and unshared dots, which the agent will never see and can only hear about from the partner.

Formally, shared dots are a subset $s \subseteq \mathcal{D}$ of the agent's own referential domain. The model



v = Yes, is there a small grey one as well?

c = Yes $\quad m =$

Figure 2: The partner model for ONECOMMON. The agent's referential domain $\mathcal{D}$ is represented by the blue circle, and the partner's unobserved domain $z = (s, u)$ by the red. The shared dots s are in the the overlap of the two domains, while the unobserved dots u are only in the red circle. The partner response $y = (c, m, v)$ has confirmation c, set of mentioned dots m (circled in red), and v given above.

observes all dot features for shared dots in s: size, color, and spatial position. We make a simplifying assumption that sizes and colors are bucketed into $B = 3$ buckets, and the relative spatial position of two dots is one of four buckets: above-left, above-right, below-left, or below-right. The prior $P(s)$ is uniform over all subsets of $\mathcal{D}$, and has no learned parameters. We update the prior after observing a partner response y, which we detail in Section 6.2 on inference.

The unshared dots u are represented as a set of dots of size $|\mathcal{D}| - |s|$ dots, with a limited set of features: only size and color, but not spatial position. We do not model the relative spatial features of u to limit the computational complexity of inference. Given the dependence on the size of s, $P(u \mid s)$ is uniform over dot features for $|u| = |\mathcal{D}| - |s|$ dots and has no learned parameters.

5.2 Partner response

For the partner response model,

$$\begin{aligned} P(y \mid x, z) &= P(c, m, v \mid x, z) \\ &= P(c \mid x, z)P(m \mid z)P(v \mid c, m), \end{aligned}$$

we decompose the response y into a boolean confirmation c, a set of mentioned dots m, and utterance

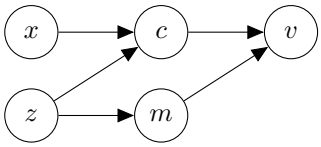


Figure 3: The graphical model for the decomposed partner response model. Given an agent plan x and partner perspective z, the model generates the partner confirmation c, mentioned dots m, and utterance text v.

text v describing the confirmation and mentions. Given the agent's plan x, the partner confirms x if x can be resolved under their perspective z, given by the confirmation model $P(c \mid x, z)$. Additionally, the partner can give unsolicited information about dots, unrelated to the agent's plan x, as mentions m from the mention model $P(m \mid z)$. The words in utterance v are then generated independently from one another given either c or m from the utterance model $P(v \mid c, m)$. We illustrate the graphical model in Figure 3.

In the partner response model, we assume the utterance v does contain any information that is not represented in c or m. While c and m contain all necessary information, only the utterance v is observed. However, the confirmation and mentions are easily predicted from v with an approximate posterior.

Confirmation The confirmation $c \in \{\text{no}, \text{yes}\}$ indicates whether the partner confirms that agent's plan x is resolvable to either s or u. We make a simplifying assumption that x cannot be split between both s and u: either x resolves to s or u based on comparing the features of x to subsets of s or u. This yields the disconfirmation probability

$$\begin{aligned} &P(c = \text{no} \mid x, s, u) \\ &= P(c = \text{no} \mid x, s) P(c = \text{no} \mid x, u) \\ &= \theta \mathbf{1}(x \not\subseteq s) \cdot \theta \mathbf{1}(x \not\subseteq u), \end{aligned}$$

where θ is the probability of the partner confirming given successful resolution of x. See Appendix A for the details of feature-based resolution.

Mentions The partner mentions $m \subseteq \mathcal{D}$ are any dots from the agent's domain that are mentioned in the partner's response y. We assume the mentions are independent from the confirmation c and agent plan x, so the mentions chosen from $P(m \mid s, u)$ condition only on the partner's perspective. Similar to the confirmation model, we assume partner mentions are drawn from either partition s or u uniformly, and a configuration from the chosen the respective partition uniformly. As shown in the graphical model in Figure 3, m is independent from x given c.

Utterance The utterance v is generated given the confirmation and mentions, $P(v \mid c, m)$. The utterance model is not used during planning. Instead, we directly use a supervised approximate posterior $q(c, m \mid v)$ to predict confirmations and mentions.

For confirmation prediction from the partner utterance $q(c \mid v)$, we perform a second stage of fine-tuning of a RoBERTa model (Liu et al., 2019), originally fine-tuned on sentiment (Heitmann et al., 2020), using 147 annotated dialogue turns. We annotate each turn as a confirmation, disconfirmation, or no response.

For mention resolution from the partner utterance $q(m \mid v)$, we use the argmax prediction from the reference resolution model from prior work (Udagawa and Aizawa, 2019; Fried et al., 2021), then remove the dots mentioned in the agent plan x from the previous turn.

6 Inference

In this section, we describe how the partner model is used to plan the agent's generations and update beliefs about the partner's perspective.

6.1 Planning

At a given turn, the agent chooses a plan using expected information gain, a heuristic also used in prior work (Yu et al., 2019). For example, asking about a triangle and getting a confirmation would result in a large amount of information gain, as the partner must have all three dots. However, the probability that the patner has all three dots may be small. A disconfirmation would be less informative, as only one of three dots must be missing in order for the plan to be unresolvable. In this setting, it may be better to ask about a pair of dots, which is more likely to be resolved and also results in higher information gain if disconfirmed, as the agent knows that at least one of only two dots is missing.

Rather than optimize the information gain with respect to the full partner perspective z, we instead optimize information gain with respect to just the shared dots s:

$$\operatorname*{argmin}_{x} \mathbb{E}_{c|x} \left[H(s \mid x, c) \right].$$

As the mentions m and utterance v are not in the Markov blanket of s, the information gain from observing them is not affected by the agent's choice of x. While this objective does not optimize for the information gain with respect to the unshared dots, it still considers the full partner perspective by marginalizing over the unshared dots u. Accounting for the unshared dots in the partner perspective in this manner encourages the agent to choose plans that are less likely to suffer from perspective-dependent ambiguity, by avoiding dots which are more likely to have counterparts which only the partner can see. We leave gaining information about unshared dots to future work.

Our information gain objective requires the following quantities: the marginal confirmation distribution $P(c \mid x)$, as well as the posterior intersection distribution

$$P(s \mid x, c) = P(c \mid x, s)P(s)/P(c \mid x).$$

The marginal confirmation distribution is the key term in both, and is obtained by marginalizing over both s and u in the partner model. Given the parameterization of the partner confirmation model from Section 5, we can compute the marginal disconfirmation probability

$$P(c = \text{no} \mid x) = \sum_s P(c = \text{no}, s \mid x) \sum_u P(c = \text{no}, u \mid x, s).$$

Since c is binary-valued, this also gives us $P(c = \text{yes} \mid x)$. For each plan x, whether the plan resolves to s can be pre-computed, yielding a resolution function. This must be computed once per context $\mathcal{D}$, and can be computed ahead-of time before a dialogue takes place. For details on this computation and its time complexity, see Appendix A. Given the precomputated resolution function, the marginal disconfirmation probability can be computed in $O(|2^{\mathcal{D}}|)$ computation, which is the number of values s. There are $|2^{\mathcal{D}}|$ possible plans x, resulting in $O(|2^{\mathcal{D}}|^2)$ computation. In ONECOMMON, there are 7 dots so $|2^{\mathcal{D}}| = 128$. The marginal probability of resolving x to unshared dots can be precomputed for all x, s pairs with the same amount of computation, as shown in Appendix B.

6.2 Belief update

Planning requires a prior over the partner's perspective, $P(z) = P(s)P(u \mid s)$. This prior must be updated at every turn, given new evidence from the partner's response y to the agent's plan x. We only update the distribution over the shared dots $P(s)$, and leave the distribution over unshared dots $P(u \mid s)$ uniformly distributed over size and color features. This is a conservative assumption which forces the model to avoid plans that could be incorrectly resolved to any unshared dot configuration. The belief update over s is given by

$$P(s \mid x, c, m) = \frac{\sum_u P(c, u \mid x, s)P(m \mid s, u)P(s)}{\sum_{s,u} P(c, u \mid x, s)P(m \mid s, u)P(s)}.$$

We use the argmax predictions of the supervised approximate posterior $c, m = \text{argmax}_{c',m'}\, q(c', m' \mid v)$ for the belief update.

7 Experimental setup

We evaluate our method on the ONECOMMON dataset (Udagawa and Aizawa, 2019). We perform automatic evaluation of plans by evaluating whether the agent plan is incorrectly resolved by the partner model. A plan is incorrectly resolved if the partner does not resolve the plan to any of the agent's intended referents. We evaluate this without language by directly feeding the feature representation of plans from the agent to the partner. Formally, given a mention plan x, the partner-resolved referents are $l(x) \subseteq z$, and the agent's intended referents $s(x) \subseteq \mathcal{D}$. A successful plan is either unresolvable, $l(x) = \emptyset$, or one of the intended referents matches the resolved referents, $l(x) \cap s(x) \neq \emptyset$. An incorrectly resolved plan does get resolved to some referents by the partner, $l(x) \neq \emptyset$, but not to any of the intended referents: $l(x) \cap s(x) = \emptyset$.

We generate plans given dialogue history from a validation split from prior work (Fried et al., 2021).[2] For each turn in the human-generated dialogue, we generate a plan from our model and label that plan as either a success or failure using the procedure above. We evaluate on 518 validation games.

We refer to our proposed method as Planner, which maintains an explicit belief state over shared dots, performs information gain based planning,

[2] Prior work used 10-fold cross-validation. We use models from prior work trained on one fold, and evaluate our approach on the validation set for that fold. In particular, we use fold 1 from prior work.

Model	Num dots shared			Total
	4	5	6	
Planner	14	29	13	56
Egocentric planner	176	138	64	378
Fried et al. (2021)	54	47	17	118

Table 1: The number of incorrectly resolved plans in automatic feature-based evaluation on 519 static validation dialogues. A lower number of incorrect plans is better.

Model	I	C	U	S
Planner	56	926	1359	0
Egocentric planner	378	1262	701	0
Fried et al. (2021)	118	1125	565	533

Table 2: Finer-grained automatic evaluation of plans on all static validation dialogues. For each turn of a dialogue, the 1-best plan was labelled as incorrect, coarse, unresolvable, or specific (I, C, U, S).

and is aware of perspective-dependent ambiguity. We refer to an ablated version of our approach as Egocentric Planner, which does not reason about perspective-dependent ambiguity but does maintain an explicit belief over shared dots and uses information gain to plan. As a baseline, we compare to prior work's plan proposals, which do not account for perspective-dependent ambiguity (Fried et al., 2021). The baseline model chooses plans based on the round-trip probability from an utterance back to the plan. The baseline agent chooses 8 plans and 64 utterances for each plan, then chooses the plan and utterance pair with the highest probability of recovering the plan given the utterance using a reference resolution model applied only to the agent's own context.

We also perform a finer-grained evaluation of plans from the Planner, Egocentric Planner, and baseline from Fried et al. (2021). Rather than dividing plans into just incorrectly resolved or successful, we further divide successful plans into coarse, unresolvable, or specific. Coarse plans are ones that resolve to multiple speaker configurations, e.g. $|s(m)| > 1$, but whose overlap with the listener resolved configurations is not empty $s(m) \cap l(m) \neq \emptyset$. Unresolvable plans are those that cannot be resolved to any configuration in the listener context, with $l(m) = \emptyset$. Specific plans resolve to the same single configuration in both the speaker and listener domains $s(m) = l(m), |s(m)| = 1$.

8 Results

RQ1: Does accounting for partner perspective result in fewer incorrectly resolved plans? Directly modeling and marginalizing over unobserved partner perspective results in fewer errors than an egocentric heuristic, as shown in Table 1. The number of incorrectly resolved plans from Planner is much lower than the Egocentric Planner, an 85% reduction. In automatic evaluation of plan proposals, reasoning about the partner's perspective as a latent variable results in a 47% reduction in the total number of incorrectly resolved plans compared to a state-of-the-art baseline The Planner model also results in fewer incorrectly resolved plans compared to prior work, a 52% reduction, demonstrating the benefit of accounting for partner perspective. This trend holds up across various levels of perspective-dependent ambiguity, as determined by the number of shared dots.

We note that the absolute number of resolution errors is small relative to the total number of turns. This is expected, as the egocentric baseline is allowed to repeat any dots referenced by their partner. Additionally, outside of static evaluation, a single incorrect plan may result in immediate failure regardless successful plans in the preceding turns.

Interestingly, the baseline model of Fried et al. (2021), which uses an egocentric heuristic with a specificity objective in planning, outperforms the Egocentric Planner, which uses an information gain objective in planning. We hypothesize that this is due to the baseline model's tendency to repeat plans mentioned or confirmed by the partner. Less exploration in the baseline model of Fried et al. (2021) means fewer risks, which could result in less success overall when agents play the full game.

RQ2: How do the plans from Planner differ from human plans and Fried et al. (2021)? We provide a more detailed breakdown of plans in Table 2. Plans from the proposed model are largely unresolvable, compared to the specific plans of the egocentric model. This is a result of the planning objective, rather than perspective modeling. The proposed model chooses plans via information gain, which results in many unresolvable plans due to exploration. The egocentric model is trained via supervised learning on human-human dialogues,

implying that its training data does not have a similar bias towards exploration.

We hypothesize that information gathering will be helpful for overall game-playing, as it will prevent premature selection. However, the predominance of unresolvable plans implies that an agent must also determine when it has gathered enough information and switch behaviours to selecting a specific dot.

9 Future Work

For future work, the current model will be extended with template-based text generation. Preliminary experiments using utterance generation methods from Fried et al. (2021) found that there were issues with both fidelity and distribution shift. For fidelity, the utterance generation model from prior work mixed up relative positioning such as left or right. Additionally, the distribution of mention plans generated by Planner is different from the training data, with a preference for multi-dot compound references. We will therefore use template generation methods to ensure fidelity and also avoid domain shift.

In addition to text generation, the current approach needs to be able to determine when it has gathered enough information and select a dot to terminate the game. We will switch away from information gathering once the entropy of the belief distribution over shared dots is low enough. Once the model is relatively certain about which dots are shared, selection can be reduced to picking a shared triangle, or other specific structure that is unlikely to be incorrectly resolved, and referring to a particular dot contained in that triangle.

Incorporating text generation and entropy-based selection will enable us to evaluate the Planner approach on selfplay and human evaluation, allowing us to answer whether directly modeling perspective-dependent ambiguity improves gameplay on ONECOMMON.

References

Sarah Brown-Schmidt and Joy E. Hanna. 2011. Talking in another person's shoes: Incremental perspective-taking in language processing. *Dialogue Discourse*, 2:11–33.

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

Herbert H. Clark and Catherine R. Marshall. 1981. Definite knowledge and mutual knowledge. In Aravind K. Joshi, Bonnie L. Webber, and Ivan A. Sag, editors, *Elements of Discourse Understanding*, pages 10–63. Cambridge, UK: Cambridge University Press.

Herbert H. Clark, Robert Schreuder, and Samuel Buttrick. 1983. Common ground at the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior*, 22(2):245–258.

Rodolfo Corona Rodriguez, Stephan Alaniz, and Zeynep Akata. 2019. Modeling conceptual understanding in image reference games. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fethiye Irmak Doğan, Sarah Gillet, Elizabeth J. Carter, and Iolanda Leite. 2020. The impact of adding perspective-taking to spatial referencing during human–robot interaction. *Robotics and Autonomous Systems*, 134:103654.

Daniel Fried, Justin T. Chiu, and Dan Klein. 2021. Reference-centric models for grounded collaborative dialogue. In *Proceedings of EMNLP*.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The photobook dataset: Building common ground through visually-grounded dialogue. *CoRR*, abs/1906.01530.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. *CoRR*, abs/1704.07130.

Mark Heitmann, Christian Siebert, Jochen Hartmann, and Christina Schamp. 2020. More than a feeling: Benchmarks for sentiment analysis accuracy. *Available at SSRN 3489963*.

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.

Boaz Keysar, Dale J. Barr, Jennifer A. Balin, and Jason S. Brauner. 2000. Taking perspective in conversation: The role of mutual knowledge in comprehension. *Psychological Science*, 11(1):32–38. PMID: 11228840.

Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *CoRR*, abs/1706.05125.

Mingjiang Liu, Chengli Xiao, and Chunlin Chen. 2021. Perspective-corrected spatial referring expression generation for human-robot interaction. *CoRR*, abs/2104.01558.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1:81–106.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *CoRR*, abs/1805.04655.

Mohit Shridhar and David Hsu. 2018. Interactive visual grounding of referring expressions for human-robot interaction. *CoRR*, abs/1806.03831.

Robert C. Stalnaker. 1970. Pragmatics. *Synthese*, 22(1-2):272–289.

Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7120–7127.

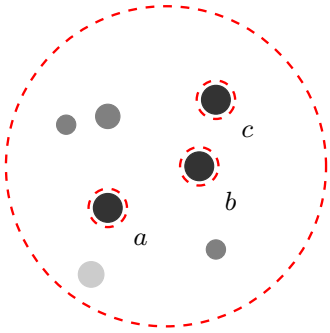
Denis Yarats and Mike Lewis. 2017. Hierarchical text generation and planning for strategic dialogue. *CoRR*, abs/1712.05846.

Lili Yu, Howard Chen, Sida I. Wang, Yoav Artzi, and Tao Lei. 2019. Interactive classification by asking informative questions. *CoRR*, abs/1911.03598.

A Feature-based resolution

Feature-based resolution featurizes a plan x then compares the features to all subsets of dots in the partner domain $\mathcal{D}$. The set of features used for each plan x is given by the shape and size each dot in the plan, bucketed into $B = 3$ bins based on the range of each feature. The pairwise positions, limited to above-left, above-right, below-left, and below-right, are also contained in the feature set. We provide an example of feature-based resolution in Figure 4.

Given a plan x, feature-based resolution resolution must compare all the features of the plan, of which there are $O(|x|^2)$, to all partial permutations of subsets size $|x|$ taken from $\mathcal{D}$, of which there are $O(|\mathcal{D}|^{|x|})$.



Feature representation
Dot 1: Large, dark
Dot 2: Large, dark, below-left Dot 1

Figure 4: An example of feature-based resolution. The above feature representation for a pair of dots resolves to dot configurations $\{(a, b), (a, c), (b, c)\}$.

B Resolution to unshared dots

The probability a plan x resolves to the unshared dots u is

$$\sum_u P(c = \text{yes}, u \mid x, s) = \theta \binom{|u|}{|x|} \frac{B^{2\cdot(|u|-|x|)}}{B^{2\cdot|x|}},$$

where B is the feature bucket size, given $|u| \geq |x|$.