Modeling Perspective-Dependent Ambiguity in Grounded Collaborative Dialogue

Justin T. Chiu
Cornell Tech
jtc257@cornell.edu

Wenting Zhao
Cornell University
wzhao@cornell.edu

Alexander M. Rush
Cornell Tech
arush@cornell.edu

Daniel Fried
Carnegie Mellon University
dfried@cs.cmu.edu

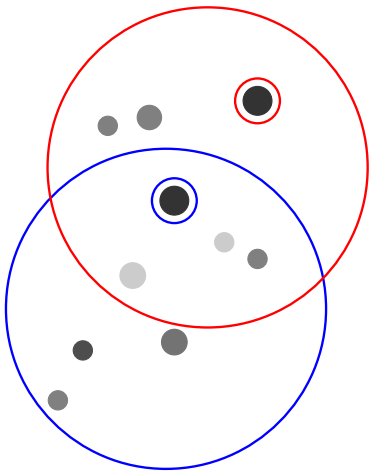
Abstract

Errors in reference generation and resolution can occur if a dialogue agent reasons incorrectly about their partner's perspective. We present, in a collaborative and visually grounded setting, a dialogue planner that infers its partner's perspective to produce referring expressions that the partner resolves correctly. The dialogue planner models the partner perspective as a latent variable, embedded in a partner model that is used for both model-based planning and incorporating evidence from partner responses. We validate our approach on ONECOMMON, a challenging dialogue game where players have large differences in their perspectives. In symbolic self-play, where agents partner with a copy of themselves using symbolic communication, the dialogue planner improves over a planning baseline that does not reason about the partner's full perspective.

1 Introduction

Ambiguity is a key impediment to collaborative dialogue that makes linguistics actions, such as reference resolution and generation, difficult to execute. One cause of ambiguity in a grounded setting is differences in perspectives (Brown-Schmidt and Hanna, 2011). Agents that incorrectly assume their partner shares their perspective may make errors in reference generation and resolution. We refer to ambiguity induced by different perspectives as *perspective-dependent ambiguity*.

Figure 1 provides an example from ONECOMMON, a collaborative reference game with a high degree of perspective-dependent ambiguity (Udagawa and Aizawa, 2019). An agent and partner are given different but overlapping views of a game board and must jointly identify a shared dot through dialogue. 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 have an immediate irrevocable consequence: The agent and partner select different dots, failing the game.



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

This false positive confirmation results from an *egocentric* heuristic (Keysar et al., 2000). Agents utilizing an egocentric heuristic resolve referring expressions based mainly on their own perspective, failing to account for the fact that the perspectives have partial overlap and there are unshared items

which cannot be observed by the agent. 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).

In this work, we incorporate the partner's perspective into a collaborative dialogue system. Our approach centers on a partner model that predicts the partner's response given a question from the agent and the partner's unobserved perspective, modeled as a latent variable. The partner model reasons explicitly over shared and unshared aspects of the partner perspective, marginalizing over the unshared aspects. The agent uses the partner model to plan symbolically, asking maximally informative questions by maximizing the expected information gain with respect to the latent partner perspective. After observing the partner's response to the utterance, the agent uses the partner model to conservatively update its belief over shared perspective.

Incorporating a planning model has a side effect of leading to richer plans than those performed by human players. As such, this approach makes it difficult to use supervised generation models trained on typically egocentric human players. Preliminary results on human evaluation show that a templated generation system is difficult for human partners to understand.

Experiments show that planning through the partner model outperforms a state-of-the-art fully supervised dialogue agent (Fried et al., 2021). The supervised approach chooses plans based on an egocentric heuristic, and is limited by the strategies demonstrated in its training data. When applied to ONECOMMON, our approach utilizes richer plans than those demonstrated by humans and achieves a symbolic[1] selfplay success rate of 85.2%, a large improvement over the prior state-of-the-art language selfplay success rate of 62.4% (Fried et al., 2021).

2 Related work

Perspective in common ground The study of perspective and its influence on *common ground*, the shared knowledge between dialogue agents, has its roots in pragmatics and linguistics (Stalnaker, 1970; Clark and Marshall, 1981; Clark et al., 1983). Behavioural experiments studying the egocentric heuristic found that it can lead to irrevocable errors in human studies (Keysar et al., 2000). 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. In contrast to these works, where the perspective of the partner is known and conditioned on, in our setting the partner perspective is inferred and the environment is partially shared between partners.

Perspective 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 colorblindness were inferred in a single-turn visual reference game. While their approach did not model partner perspectives, we propose to directly model the partner perspectives as a latent variable.

Informative question generation Work in question generation formulates information-seeking conversation as a partially observable Markov decision process. There is one information seeker and one information holder. Methods use a single-turn partner-model heuristic to generate questions (Rao and III, 2018; Yu et al., 2019). In this setting, 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 perspective into ambiguous ones for the partner.

Task-oriented dialogue Task-oriented dialogue uses multi-turn model-based planning. Here planning is performed 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). While we use a single-turn planning heuristic, improvements in partner models are complementary to multi-turn planning and can be combined in future work.

Reference generation There is a long line of work on referring expression generation (Heeman, 1991; Dale and Reiter, 1995; Jordan and Walker, 2005; Krahmer and van Deemter, 2012; Takmaz et al., 2020, *inter alia*), particularly in grounded set-

[1] Symbolic selfplay for ONECOMMON allows agents to directly communicate the symbolic representation of plans instead of describing them in natural language.

tings (Dale and Viethen, 2009; Mao et al., 2016; Yu et al., 2017; Takmaz et al., 2020, *inter alia*). The OneCommon (Udagawa and Aizawa, 2019) dialogue setting poses challenges for generation due to its partial observability and the non-identical views of its participants. Our method is most closely related to the state-of-the-art approach to ONECOMMON (Fried et al., 2021) which forms the backbone of our method. We focus on relaxing the egocentric assumptions of this method by incorporating symbolic planning and generation.

3 Partner Modeling in Reference Games

Collaborative reference games pair an agent and a partner in order to agree on a shared object through natural language dialogue. At each turn, the agent or partner may decide to terminate the game and make a selection. Once either the agent or their partner terminates, the other player must also act (without observing the other's choice). If both agent and partner agree, both win; otherwise, both fail.

Our approach to reference games separates planning of utterances (choosing what to talk about) from surface realization (choosing how to say it). At each turn, our agent produces an utterance plan x by using a *partner model*, which simulates the partner's possible responses, y, given their hidden perspective, z. The agent uses the partner model to infer the partner's perspective and predict the partner's responses to the agent's plans. We first give an overview of the partner model and planning procedure in this section.

Partner model We model the partner's perspective as a latent variable, infer the value for this variable over the course of a game, and use it to plan generation. This contrasts with a typical egocentric heuristic, as used in Fried et al. (2021), which assumes the partner's perspective is identical to the agent's.

The partner model predicts a distribution over the partner response y given the agent plan x under the latent common ground z , and decomposes as:

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

Planning The agent uses the partner model to plan what to say next, by choosing the plan x that maximizes the expected information gain (Lindley, 1956) about the common ground z, defined as

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

where $H[z]$ is the entropy of the prior[2] and $H[z \mid x, y]$ the posterior, which requires marginalizing over z.

Belief update After observing the partner response y, the agent updates its belief $p(z)$ over the common ground with Bayes' rule:

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

This is performed iteratively after each turn, and requires marginalizing over possible shared perspectives.

Selection After gathering information through planning and incorporating information through belief updates, the agent must decide when it has built enough common ground, collaboratively identifying a shared dot with its partner. We set a threshold on the belief entropy, $H[z]$, which determines when the agent should transition from information gathering to ending the game.

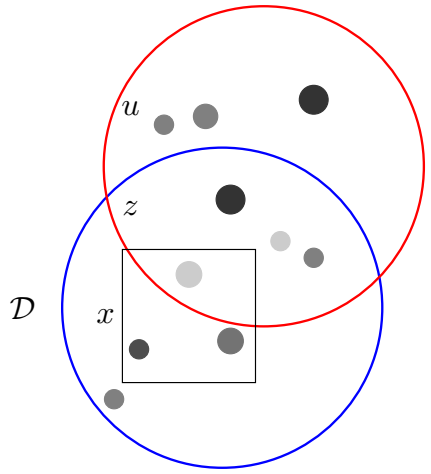
4 Planning in ONECOMMON

We focus on applying partner modeling to 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 perspective is shared, resulting in perspective-dependent ambiguity.

In ONECOMMON, the agent's known perspective $\mathcal{D}$ consists of 7 dots in its view. Each dot has a set of features: size, color, and position in the 2D plane. All features are continuous. The main challenge of the game is that the partner perspective z is also a view of 7 dots. Between 4–6 of those dots are shared with the agent perspective which we denote as the common ground z. Additionally there are a set of unshared dots u the fill out the partner perspective. An example is given in Figure 2.

The agent communicates with the partner by producing an utterance plan, x, which it then describes in natural language. This plan is a subset of the dots in the agent view, $x \subseteq \mathcal{D}$, that the agent will ask the partner about. The partner gives a response y to the plan x, given their perspective z. In ONECOMMON, the partner responds in natural language; however, the partner model only models the partner response as a confirmation $y \in \{\text{YES}, \text{NO}\}$, obtained by classifying natural language responses.

[2]The prior entropy $H[z]$ in the definition of information gain is constant with respect to the plan x, and can be dropped from the objective.



Do you have a triangle of one gray dot ...

$y = \text{NO}$

Figure 2: In ONECOMMON, the agent's perspective $\mathcal{D}$ is represented by the large blue circle, and the partner's unobserved perspective by the red. The shared dots z are in both perspectives, while the unshared dots u are only in the red circle. The agent plan x is given by the dots in the box, and also described in language. The partner response y is a binary confirmation.

Exact planning is intractable because the objects in the partner perspective have continuous-valued features. In this section, we describe simplifying assumptions for the partner model and inference procedure that make planning tractable.

4.1 Partner model

We build a partner model by factor the common ground z and partner response y as illustrated in Figure 2. Formally,

$$\begin{aligned} p(y \mid x) &= \sum_{z} p(y \mid x, z) p(z) \\ &= \sum_{z,u} p(y \mid x, z, u) p(z) p(u), \end{aligned}$$

where we introduce the latent variable u representing the unshared dots in the partner perspective.

The shared dot prior, $p(z)$, is a distribution over subsets of $\mathcal{D}$, indicating which dots in the agent perspective $\mathcal{D}$ are shared with the partner. The model $p(z)$ is initially uniform over dot subsets at the start of a game, but is updated given evidence from the partner response y at the end of each turn, $p(z \mid x, y)$. For notational simplicity we focus on the first turn.

The unshared dot prior, $p(u)$, is a distribution over the remaining partner dots. Since the dots in u are unobserved by the agent, we parameterize $p(u \mid s)$ using a uniform distribution over discretized features for each dot. We ignore spatial features for dots in u and discretize the other originally continuous features: size and color. [3]

The confirmation model, $p(y \mid x, z)$, checks whether a partner will confirm or deny the agent plan. Given fully realized z and u confirmation of a plan x is performed by matching the features of x to z and u. See Appendix A for the details of feature-based resolution. In order to avoid fully enumerating both z and u jointly, the model reasons separately about z and u by making the simplifying assumption that plans are fully in z or u. This means that the model will deny if part of x is in z, while the remainder is in u (and x is not fully contained in either z or u):

$$\begin{aligned} p(y = \text{NO} \mid x) &= \sum_{z,u} p(y = \text{NO} \mid x, z, u) p(z, u) \\ &= \sum_{z} p(y = \text{NO} \mid x, z) p(z) \\ &\quad \cdot \sum_{u} p(y = \text{NO} \mid x, u) p(u). \end{aligned}$$

The partner confirms or denies accurately with probability θ, a hyperparameter.

4.2 Inference

During inference, we need to compute $p(y \mid x)$ for all plans x, which can be done in two steps: First, we marginalize over the unshared dots u. This marginalization can be expressed in closed form. For the details, see Appendix B. Second, we marginalize over the possible set of shared dots z. The computational cost of marginalization is the size of the power set of $\mathcal{D}$, $O(2^{|\mathcal{D}|})$.

We utilize this distribution to compute the posterior on the common ground z, after observing a partner response to a plan,

$$p(z \mid x, y) = \frac{p(z, y \mid x)}{p(y \mid x)}.$$

This posterior then allows us to perform optimization over plans with respect to the expected information gain, as well as update our beliefs given the partner response.

Planning Planning optimizes the expected information gain with respect to the common ground z:

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

[3] We discretize size and color uniformly into 3 buckets based on their absolute range across ONECOMMON.

Computing $p(y \mid x)$ has cost $O(2^{|\mathcal{D}|})$, while there are also $O(2^{|\mathcal{D}|})$ plans.[4] As a result, optimizing this objective takes $O(2^{2|\mathcal{D}|})$ computation, and is performed in less than second on CPU in real-time.

Selection To determine when to select a dot, the agent uses a threshold on the entropy $H[z]$, given by the hyperparameter τ. The agent them communicates which dot to select by describing the configuration of four dots with the highest marginal probability of being shared, as well as the dot within that configuration that is most likely to be shared. The agent then selects the described dot.

5 Experimental setup

We evaluate our method, the Partner Planner, on the ONECOMMON dataset (Udagawa and Aizawa, 2019). We perform two evaluations: First, we evaluate the number of incorrectly resolved plans generated by the Partner Planner given a static, natural language dataset. Second, we evaluate the Partner Planner in dynamic, symbolic selfplay.

Static plan evaluation In order to show that resolving perspective-dependent ambiguity reduces errors, 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 plan is not empty and 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, who is a Partner Planner.

We generate plans given natural language dialogue history from a validation split of ONECOMMON following prior work (Fried et al., 2021).[5] 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, which have different numbers of shared dots: either 4, 5, or 6. Fewer dots result in perspective-dependent ambiguity: there are more unshared dots, which the partner may incorrectly resolve plans to.

In order to process the natural language dialogue, we use a classifier to extract binary partner confirmations, as well as any dots they mention, from natural language. For confirmation prediction, 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 partner mention prediction, we use the mention prediction network from Fried et al. (2021). The Partner Planner incorporates partner mentions into its belief by treating them as a confirmed plan.

Symbolic selfplay We also evaluate the Partner Planner on symbolic selfplay, where it plays the full ONECOMMON game with a copy of itself using symbolic communication. We evaluate only on the setting with the most perspective-dependent ambiguity, 4 shared dots. In symbolic selfplay, agents must perform planning, belief updates, and selection. The Partner Planner is able to exactly communicate confirmations and dot features (bucketed size, color, and relative positions). We compare the game success rate of the Partner Planner to success rate of the baseline by (Fried et al., 2021) and human performance on language selfplay. Language inherently has more noise than symbolic representations, meaning symbolic selfplay is an upper bound on performance with language.

Systems We focus evaluation on the Partner Planner, which reasons about common ground and unshared dots. We consider an ablated version, which does not model unshared dots.

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, which is an egocentric heuristic and also does not account for uncertainty over shared dots. 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 give the utterance using a reference resolution model applied only to the agent's own context.

Hyperparameters For the Partner Planner, we set the response faithfulness probability $\theta = 0.95$. We determine the selection entropy threshold by running grid search over $\tau \in \{1, 1.5, 2\}$ on symbolic selfplay, and pick the value with the highest success rate.

[4] Plans x are subsets of $\mathcal{D}$ that the agent would like to ask the partner about.

[5] 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.

Agent	# shared 4	5	6	Total
Partner Planner	26	12	5	43 (12)
-unshared	70	61	22	153 (22)
Fried et al. (2021)	28	17	4	49 (17)

Table 1: The number of incorrectly resolved plans in automatic feature-based evaluation on 519 static validation dialogues (2,341 turns). The total number of errors made in the first turn of a dialogue is shown in parentheses for each agent. A lower number of incorrect plans is better.

Agent	Success	Avg # turns
Partner Planner	85.2%	10.10
-unshared	77.0%	11.69
Fried et al. (2021)	62.4%	-
Human	65.8%	4.97

Table 2: The success rate of different agents in selfplay on the hardest setting of ONECOMMON, with 4 shared dots. The Partner Planner and ablated version communicate symbolically, while the Fried et al. (2021) baseline and human performance use language. A higher success rate is better. The human performance is from the ONECOMMON dataset (Udagawa and Aizawa, 2019).

6 Analysis

Static plan evaluation Directly modeling and marginalizing over unobserved partner perspective results in fewer errors than the egocentric heuristic from Fried et al. (2021), as shown in Table 1. The number of incorrectly resolved plans from Planner is much lower than an ablated planner that does not model unshared dots, an 72% reduction. The Planner model also results in fewer incorrectly resolved plans compared the baseline from Fried et al. (2021), a smaller 12% reduction, demonstrating the benefit of accounting for partner perspective.

We hypothesize that the Fried et al. (2021) baseline makes fewer errors because it is able to repeat plans mentioned in the static dialogue. The Partner Planner does not often repeat plans, as there is little information gained. When restricted to the plan proposals from the first turn of a static human-demonstrated dialogue, where no plans can be copied, the Partner Planner outperforms the baseline by a larger margin, as shown in Table 1. 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. We hypothesize that this is because of the nature of static evaluation. Static evaluation considers next step plan proposals given human dialogue, preventing agents from fully steering dialogue themselves. Our next experiments demonstrate that the Partner Planner has strong performance in the symbolic dialogue setting.

Symbolic selfplay The Partner Planner achieves strong performance in symbolic selfplay, as shown in Table 2. The ablated version, which does not model unshared dots, also performs well, but worse than the full Partner Planner. This demonstrates the utility of modeling unshared perspective.

Both the full and ablated Partner Planner outperform the baseline of Fried et al. (2021) and coached human performance from the training data of Udagawa and Aizawa (2019), demonstrating the utility of partner modeling. Much of this success comes from the ability to control the belief entropy selection heuristic, as shown by the performance of the ablated Partner Planner over human performance. The selection heuristic encourages the Partner Planner to be more patient and gather more information before selecting than most human participants, reflected in the average number of turns per game.

7 Future Work

For future work, the current model will be extended with text generation in order to play ONECOMMON with humans. Preliminary experiments using utterance generation methods from Fried et al. (2021) found that the plans proposed by the Partner Planner were out-of-distribution for supervised methods, resulting in many errors in generation. Further preliminary experiments with a templated generation system found that the resulting templates were too difficult to understand for human partners. Future work will experiment with generation systems that account for utterance processing cost, balancing information cost and fluency.

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 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.

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

Robert Dale and Jette Viethen. 2009. Referring expression generation through attribute-based heuristics. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 58–65, Athens, Greece. Association for Computational Linguistics.

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.

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

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.

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).

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.

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

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.

D. V. Lindley. 1956. On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27(4):986 – 1005.

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.

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*.

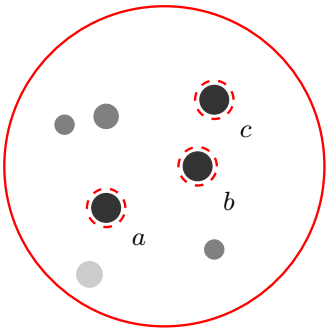
Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *CoRR*, abs/1805.04655.

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.

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)*.

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.



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

Figure 3: An example of feature-based resolution. The above feature representation for a pair of dots resolves to dot configurations $\{(a, b), (a, c), (b, c)\}$.

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

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*.

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 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 3.

Given a plan x, feature-based 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 $O(|\mathcal{D}|^{|x|})$. This can be precomputed at the start of a dialogue.

When resolving x to unshared dots, we ignore spatial features.

B Resolution to unshared dots

The partner model, with the assumption that x cannot be split between z and u, is given by

$$
\begin{aligned}
&p(y = \textsc{no} \mid x) \\
&= \sum_{z,u} p(y = \textsc{no} \mid x, z, u) p(z) p(u) \\
&\approx \sum_{z,u} p(y = \textsc{no} \mid x, z) p(z) p(y = \textsc{no} \mid x, u) p(u) \\
&= \sum_{z} p(y = \textsc{no} \mid x, z) p(z) \sum_{u} p(y = \textsc{no} \mid x, u) p(u) \\
&= \sum_{z} p(y = \textsc{no} \mid x, z) p(z) \sum_{u} p(y = \textsc{no}, u \mid x).
\end{aligned}
$$

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

$$
\sum_{u} p(y = \textsc{no}, u \mid x) = 1 - \theta \binom{|u|}{|x|} \frac{B^{2 \cdot (|u| - |x|)}}{B^{2 \cdot |x|}},
$$

where B is the feature bucket size, given $|u| \geq |x|$. This relies on the assumption that spatial features are ignored when resolving to unshared dots.