Taking Action Towards Graceful Interaction: The Effects of Performing Actions on Modelling Policies for Instruction Clarification Requests

Anonymous ACL submission

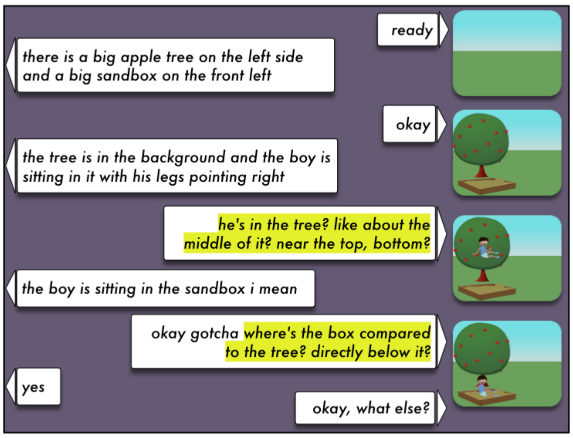
Abstract

Clarification requests are a mechanism to help solve communication problems in instruction-following interactions. Despite their importance, even skilful models struggle with producing or interpreting such repair acts. In this work, we show that even well-motivated, Transformer-based models fail to learn a good policy for *when to ask* Instruction CRs (iCRs), while the task of determining *what to ask about* can be more successfully predicted. We test three hypotheses concerning the effects of action taking as an auxiliary task for iCR policies, concluding that, while its contribution is limited, some information can be extracted from prediction uncertainty. Considering the implications of these findings, we further discuss the shortcomings of the data-driven paradigm for learning meta-communication acts.

1 Introduction

The concept of *graceful interaction* (Hayes and Reddy, 1979, 1983) was proposed as a set of skills that machines should exhibit to properly engage in cooperative dialogue with humans, among which are being able to ask for, understand and offer clarification. More than forty years later, the ineptitude of large language models and voice assistants to handle underspecifications and to properly process or produce clarification requests (CR) is still being documented (Larsson, 2017; Kuhn et al., 2022; Li et al., 2023; Deng et al., 2023). It is also one of the acknowledged limitations of the currently prevailing commercial chat-optimised LLM.[1]

Given that they are modulated for instructions, this seems to be a peculiar fault: CRs are a crucial mechanism used to repair misunderstandings in instruction following interactions (Benotti, 2009), as we see in Figure 1. On second thoughts, it comes as no surprise. Clarification exchanges are meta-communication acts that do not normally appear in non-interactive data (Kuhn et al., 2022) and are also relatively rare in dialogue data, with an empirical frequency of 4% to 11% of turns (Purver et al., 2001; Benotti and Blackburn, 2021; Madureira and Schlangen, 2023b). Therefore, it is still unclear to what extent CR strategies can be learnt with data-driven approaches (Benotti and Blackburn, 2021).



ready

there is a big apple tree on the left side and a big sandbox on the front left

okay

the tree is in the background and the boy is sitting in it with his legs pointing right

he's in the tree? like about the middle of it? near the top, bottom?

the boy is sitting in the sandbox i mean

okay gotcha where's the box compared to the tree? directly below it?

yes

okay, what else?

Figure 1: Clarification requests posed by an instruction follower, demonstrating uncertainty on deciding what actions to take. From: CoDraw dialogue game 8198, CC BY-NC 4.0, scene from Zitnick and Parikh (2013).

As we discuss in §2, many existing CR datasets, despite their utility for applications like conversational search (Keyvan and Huang, 2022; Rahmani et al., 2023), either have not been collected via real interactions or are synthetic, so that learnt CR policies may not correspond to genuine human behaviour. Moreover, current best-performing data-driven models are still not doing very well in deciding when to ask for clarification (see §2).

Recently, Madureira and Schlangen (2023b,a) have argued that CoDraw (Kim et al., 2019) is a rich resource for Instruction CRs (iCR), naturally produced as a by-product of game playing, as in the

[1] In the blogpost releasing chatGPT, the limitations section says: "*Ideally, the model would ask clarifying questions when the user provided an ambiguous query. Instead, our current models usually guess what the user intended.*". Source: https://openai.com/blog/chatgpt.

example in Figure 1. The dataset offers a balance between size (in comparison to well-curated but small corpora) and retaining ecological validity (as opposed to massive datasets collected or crafted artificially). iCRs emerge at Clark's 4th level of communication (Clark, 1996), *i.e.* at the level of uptake (Schlöder and Fernández, 2014). Supposing their underlying strategies can emerge from data, we can reasonably assume that action-taking is a key component in modelling policies for deciding when and what to repair. However, one major drawback in the proposed baseline models is the overhearer paradigm, where models are trained to predict when to ask iCRs in a process detached from the actual actions required by the game.

Contributions Given that background, this work aims to expand the boundaries of the open question of learning meta-communication acts from human data. We do that by (i) proposing a well-motivated model for learning *when to ask* iCRs; (ii) taking another step towards a more realistic CoDraw agent by defining and modelling the task of *what to ask* about; and (iii) testing three hypotheses to study the effect of action-taking in learning iCR policies, verifying whether a measure of certainty can be used to probe for iCR abilities and inform predictions.

2 Related Work

Learning *when to ask* questions The problem of knowing when to ask questions in an interaction appears in various contexts. Relevant work has been done in language-aided visual navigation (Nguyen and Daumé III, 2019; Thomason et al., 2020; Chi et al., 2020; Nguyen et al., 2022), in which the agent must take actions in an environment and decide when to ask for help, where RL is a suitable method. Similar policies are necessary in interactive settings like visual dialogue games that require deciding when to stop asking (Shekhar et al., 2018) or incremental predictions on when to answer a question (Boyd-Graber et al., 2012).

Modelling clarification requests A vast literature exists on describing and modelling clarification strategies (Purver et al., 2003; Gabsdil, 2003; Schlangen, 2004; Rodríguez and Schlangen, 2004; Rieser and Lemon, 2006; Stoyanchev et al., 2013, *inter alia*). In the age of neural network-based NLP, the problem has commonly been broken down into various tasks that are learnt from data: *When to ask* (Narayan-Chen et al., 2019; Aliannejadi et al., 2021; Shi et al., 2022; Kiseleva et al., 2022), *what to ask* about (Braslavski et al., 2017; Aliannejadi et al., 2021; Hu et al., 2020), and how to generate (Kumar and Black, 2020; Gervits et al., 2021; Majumder et al., 2021) or select/rank appropriate CRs (Rao and Daumé III, 2018; Aliannejadi et al., 2019; Mohanty et al., 2023). Ideally, these tasks should be tied into a single agent, but several works are still approaching the problem in a "task-framed" fashion without integration of all capabilities (Schlangen, 2021).

Modelling policies for *when to ask* for clarification in instruction following is far from being a solved problem, with models performing well below the ceiling. The performance in the Minecraft Dialogue dataset is 0.63 accuracy for the CR class (Shi et al., 2022). In the recent IGLU challenge (Kiseleva et al., 2022), the best model in the leaderboard[2] reaches 0.75 weighted average F1 Score. In predicting underspecification for code generation, the highest performance is 0.78 binary F1Score (Li et al., 2023). In Codraw-iCR, the baseline achieves a similarly suboptimal 0.34 average precision (Madureira and Schlangen, 2023b). These policies are failing to fully capture the human behaviour from data. Specifically for CoDraw, many reasons are plausible, which we address in §5.

Another open issue is how to collect high-quality CR data in enough amounts for machine learning purposes. In the annotated Minecraft Dialogue Corpus (Narayan-Chen et al., 2019; Shi et al., 2022), TEACh dataset (Padmakumar et al., 2022; Gella et al., 2022) and CoDraw (Kim et al., 2019; Madureira and Schlangen, 2023b,a), CRs occur by own initiative of the players in real, multi-turn interaction, ranging from hundreds to less than ten thousand identified CR utterances. Still in the same size range, the IGLU dataset (Kiseleva et al., 2022; Mohanty et al., 2022) has been collected in a setting that avoids pairing up players, with a one-shot opportunity to ask for clarification (and without a partner to answer it and allow further actions).

Other procedures have been used to collect CR data in larger amounts. Massive datasets are DialFRED (Gao et al., 2022), created via crowdsourcing with workers who are explicitly asked to generate a question, and answer it, for a situation they are not actually involved with. In neighbour domains like virtual assistance, conversational search and code generation, large-scale datasets containing

[2] Reported in the NeurIPS 2022 IGLU challenge platform.

CRs have been constructed with data augmentation methods (Aliannejadi et al., 2021), user simulation (Kottur et al., 2021), templates (Li et al., 2023) and crawling QA online forums (Rao and Daumé III, 2018; Kumar and Black, 2020). These strategies can reflect CR form and facilitate data collection but abstract away the fundamental triggers of Instruction CRs (joint effort, real-time interaction and action-taking), being arguably not suitable for learning CR policies for instruction following.

Evaluating CR mechanisms in dialogue models Acquiring more natural data alone is no silver bullet; appropriate evaluation methods are also necessary. Weighted-average F1 Score is not the most suitable performance measure for *when to ask* policies because the proportion of CRs is imbalanced, so high scores may rely mostly on the prediction of the negative class. Average precision is more adequate for imbalanced binary classification (Saito and Rehmsmeier, 2015) and bypasses the need for a fixed threshold, but still only captures a limited dimension of the problem.

We need more evaluation campaigns and methods to shed light on what a model has actually learnt with respect to CR strategies. Some initiatives towards more detailed assessment are in progress. Chiyah-Garcia et al. (2023) evaluate the abilities of multimodal models to process CRs in coreference resolution by interpreting the difference in the object-F1 score at turns before and after a CR as the improvement provided by incorporating the clarification; they also analyse results by considering various CR properties. In the realm of LLMs, Kuhn et al. (2022) employ an evaluation technique to prompt LLMs to first detect whether a question is ambiguous and generate a CR before answering it, concluding that the models can detect ambiguity but even so do not generally attempt to repair it. Deng et al. (2023) use "proactive prompting", an intermediate step to first induce the LLM to predict whether the appropriate dialogue act is to ask for clarification or to directly answer a question. The best LLM achieves only 0.28 F1 Score.

3 Definitions

CoDraw (Kim et al., 2019) is a multimodal dialogue game where an instruction follower must use a gallery of 28 (out of 58) cliparts to reconstruct a scene (Zitnick and Parikh, 2013) they cannot see. They exchange text messages in a turn-based fashion with an instruction giver, who sees the original scene but has no access to the state of the reconstructed scene, except for one chance to peek at it during the game. The available actions are adding or deleting, moving, flipping and resizing cliparts in a canvas. Game success is measured by a scene similarity score. The authors collected 9.9k such dialogues, containing around 8k iCRs (11.3% of the game turns), annotated by Madureira and Schlangen (2023b,a) both under CC BY-NC 4.0. We can define types of instruction follower models for this game regarding their CR capabilities:

1. Overhearer: *A model that observes the current game state (dialogue context and scene) to predict when to ask iCRs, without any additional game-play actions or linguistic decisions.*
2. Action-Taker: *A model that plays the game by only taking clipart actions, without iCR decisions.*
3. iCR-Action-Taker: *An Action-Taker with the extra decision of when to ask iCRs.*
4. Full agent: *A model that makes all game-play decisions, including natural language generation.*

The Overhearer is a common paradigm in which models resemble an observer of the actual player, deciding what to do *as if it were in their shoes*. It is a rather rough simplification of a full-fledged agent, for which Action-Takers are an intermediate step examined in this work.

The task of *when to ask* for iCRs has been formalised by Madureira and Schlangen (2023b). In short, given the game state up to the last instruction giver utterance, the instruction follower has to decide whether to ask for clarification. This policy is modelled as a function f_{when} that maps the game state to a probability of asking an iCR at the current turn. Additionally, once the decision to ask for clarification has been made, a player should also know what objects and possible actions or attributes are subject to clarification at that point. We thus define the subsequent task of *what to ask* about: at an iCR turn, a function f_{what} outputs the probability of asking an iCR about each object in the gallery.[3]

4 Hypotheses

In this section, we motivate and state the three hypotheses we test.

Chiyah-Garcia et al. (2023) argue that auxiliary learning objectives of detecting objects' attributes in a scene, as implemented by Lee et al. (2022),

[3] We leave the decision of what *attributes* to mention and which form to realise for ongoing work on iCR generation.

are useful for referential CRs at Clark's 3rd level. Our expectation is that action prediction should be equivalently relevant for 4th level iCRs, which emerge when deciding how to act. More concretely, an iCR-Action-Taker should have a more genuine motivation to decide to ask in comparison to an Overhearer. However, experiments in the Minecraft dataset point to the opposite direction: Generating action sequences slightly harmed the accuracy of the Overhearer on *when to ask* (Shi et al., 2022). To further investigate it, our first hypothesis is:

Hypothesis 1: *An iCR-Action-Taker can learn a more accurate policy for predicting* ***when to ask*** *an iCR than an Overhearer.*

Here, we can also test whether action *detection* has a similar effect, by letting the model learn to detect actions given the scene before and after, as in Rojowiec et al. (2020). It is a framing even more equivalent to Lee et al. (2022), since, in their model, the attributes are already available in the images.

In the next step, we aim to investigate if an Action-Taker trained without any explicit iCR signal still builds representations that encode the need for repair. The study done by Xiao and Wang (2019) on quantifying uncertainty in NLP tasks shows that the examined models output higher data uncertainties for more difficult predictions. Besides, Yao et al. (2019) propose the assumption that if a model is uncertain about a prediction, it is more likely to be an error. They use uncertainty as a score to decide whether the prediction requires user clarification in the context of semantic parsing. Based on that, we conjecture that the need for repair should manifest as less certainty in the Action-Taker's decisions.[4] Thus, we set the linking hypothesis that certainty is expressed in the probability it assigns to taking action, or not, at a given turn. It is a reasonable assumption, because the objective function is expected to push the predictions to be either 0 or 1, so predictions close to 0.5 can be seen as indecisive. Therefore, the second hypothesis we test is:

Hypothesis 2: *At iCR turns, an Action-Taker (trained with no explicit iCR signal) predicts actions with less certainty than at other turns. Similarly, less certainty is expected for actions upon objects that are subject to iCRs than for other objects.*

Finally, iCR policies for *when to ask* should be grounded in a fine-grained representation of what exactly is unclear. Thus our last hypothesis is:

Hypothesis 3: *A pre-trained iCR-Action-Taker can learn a more accurate policy for predicting* ***what to ask about*** *in iCR turns than an Overhearer.*

5 Models

In this section, we present the model architecture we propose for our experiments. It incorporates techniques from top-flight models in recent multimodal dialogue challenges, namely IGLU (Kiseleva et al., 2022) and SIMMC 2.0 (Kottur et al., 2021), to address five of the limitations of the baseline model (iCR-baseline) by Madureira and Schlangen (2023b), some of them already acknowledged by the authors. We also refer to the original CoDraw model (CoDraw-orig) by Kim et al. (2019), which, however, did not include the instruction follower's utterances in the game. The basic architecture of the iCR-Action-Taker is illustrated in Figure 2 (see Appendix for specifications).

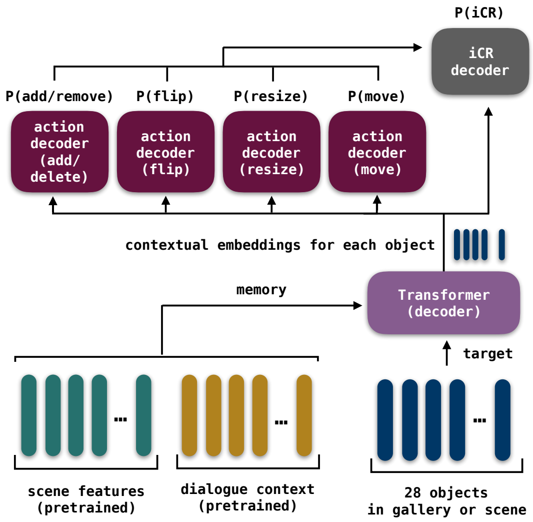

P(iCR)
iCR decoder
P(add/remove)
P(flip)
P(resize)
P(move)
action decoder (add/ delete)
action decoder (flip)
action decoder (resize)
action decoder (move)
contextual embeddings for each object
memory
Transformer (decoder)
target
scene features (pretrained)
dialogue context (pretrained)
28 objects in gallery or scene

Figure 2: The basic structure of our iCR policy models. The Overhearer contains no action decoders, whereas the Action-Taker contains no iCR decoder.

Incorporating the gallery The gallery is an informative source in CoDraw (*e.g.* if it contains just one of the three tree cliparts, it is less likely that disambiguation is needed). iCR-baseline does not include the available objects as input, whereas CoDraw-orig uses a symbolic representation assuming all 58 objects are available at any time.

[4] A similar investigation of the predictive uncertainty of instruction follower model in the Minecraft data has been done by Naszad et al. (2022) using length-normalized log-likelihood and entropy of generated action *sequences*. Negative results are reported in an unpublished short manuscript concluding that uncertainty is not a good signal for when to ask CRs.

Both approaches do not correspond to reality, as players only see 28 cliparts. We follow a similar symbolic approach to represent the objects' attributes (presence in the scene, orientation, position, size, pose, facial expression), but only for those at play. The cliparts' features and bounding boxes are projected to a higher-dimensional space following Sadler and Schlangen (2023).

Using contextual word embeddings iCR-baseline relies only on two sentence-level embeddings, one to encode the whole dialogue context and one for the last utterance, both not optimised for the game. To allow the policy to access more fine-grained linguistic information, we make all token-level contextual embeddings available to the player, constructed by a pretrained language model.

Enhancing scene representations iCR-baseline uses a pretrained image encoder. It is unlikely that off-the-shelf encoders fit well to clipart scenes without fine-tuning. Here, we follow the approach in DETR (Carion et al., 2020), employing a ResNet (He et al., 2015) backbone with learnable positional encodings to extract scene features, followed by a trainable convolutional layer to reduce the number of channels. The sequence of image features is then used as part of the input.

Transforming The iCR predictions rely only on pretrained embeddings with a feed forward neural network in iCR-baseline, and CoDraw-orig did not employ Transformers (Vaswani et al., 2017) as a trainable component. Given its leading performance in several scenarios, we bring them more explicitly to the scene, in an approach similar to DETR (Carion et al., 2020). We feed the clipart representations to the decoder, to allow self-attention to build up embeddings of the state of the gallery and scene, without positional encoding due to the arbitrary order of the cliparts. Here, we also rely on the findings by Chiyah-Garcia et al. (2023) that encoding relations between objects and their locations is helpful for CRs. Then, it performs cross-attention with the scene and text. We make text and scene available as one sequence like Lee et al. (2022). Since cross-attention between modalities is a cornerstone in current CR models (Shi et al., 2022, 2023), we also run experiments using the encoder to let text and scene attend to each other. We then end up with a multimodal representation of each clipart in the current context, which is then passed to classifier layers for each prediction.

Action-taking via multi-task learning iCR-baseline is an Overhearer, modelling only the policy of *when to ask* iCRs. To test our hypotheses, we implement (iCR-)Action-Takers that predict the game actions (or detect them, if the updated image is used) via multi-task learning. We take inspiration from Shi et al. (2022) and train a joint encoding for multiple classifiers.[5] We let the action logits (or the real actions via teacher forcing) be part of the input to the iCR decoder. To facilitate evaluation, we add an additional meta-action prediction which is 1 whenever *any* action is made to a clipart.

6 Experiments

For our experiments, we implement variations of Overhearers and (iCR-)Action-Takers, all trained on the CoDraw dataset. We compare results by varying the complexity of the input, which can be comprised of the gallery G, the dialogue context D with varying length, the scene before S_b and after S_a the current actions, and the actual actions A or their logits L_A. To test H1, we compare Overhearers with iCR-Action-Takers and iCR-Action-Detecters (which have access to both scenes), all making predictions for *when to ask* iCRs at turn-level. For H2, we examine the predictions of the Action-Taker using the certainty measure we discuss next. Finally, we test H3 by a similar analysis as in H1, but in the task of *what to ask* about, the iCR prediction is done for each clipart and we only use the turns where iCRs actually occurred in the training data (*i.e.*, we assume the decision to ask for iCR has already been taken). Here, we compare an Overhearer with an iCR-Action-Taker/Detecter whose action modules' parameters are initialised with the best Action-Taker/Detecter checkpoint.

	iCRs		**actions**				
	when	what	any	add/del	move	flip	resize
train	11.24	14.32	5.43	3.11	2.13	0.23	0.42
val	11.84	14.43	5.47	3.11	2.17	0.24	0.39
test	11.26	14.69	5.40	3.12	2.11	0.21	0.39

Table 1: % of the positive labels in the dataset.

Table 1 shows the proportion of each type of label in the dataset. Actions at each turn are sparse (mean=1.65, std=1.69) because only a small subset of the full action space is actually performed.

[5] This is not yet a full-fledged Action-Taker. It makes high level binary decisions on which actions are needed (add/delete, move, resize, flip); a full model would also make the subsequent fine-grained decision of exact positions and sizes.

		When to Ask						What to Ask					
		posing iCRs			taking actions			posing iCRs			taking actions		
	inputs	AP	bF1	mF1	AP	bF1	mF1	AP	bF1	mF1	AP	bF1	mF1
Baseline	D, S_a	.347	-	.645	-	-	-	-	-	-	-	-	-
Overhearer	G	.138	.000	.470	-	-	-	.332	.289	.593	-	-	-
	G, D	.384	.349	.642	-	-	-	.697	.665	.801	-	-	-
	G, D, S_b	.372	.267	.604	-	-	-	.697	.666	.799	-	-	-
	G, D, S_b, S_a	.378	.304	.620	-	-	-	.694	.660	.799	-	-	-
	G, D, A	.372	.404	.662	-	-	-	.711	.683	.810	-	-	-
	G, D, S_b, A	.379	.377	.654	-	-	-	.712	.675	.808	-	-	-
	G, D, S_b, S_a, A	.388	.377	.655	-	-	-	.706	.674	.808	-	-	-
Action-Taker	G	-	-	-	.149	.005	.498	-	-	-	-	-	-
	G, D	-	-	-	.769	.710	.853	-	-	-	.571	.550	.770
	G, D, S_b	-	-	-	.762	.708	.851	-	-	-	.547	.530	.761
iCR-Action-Taker	G, D	.378	.393	.658	.755	.702	.848	**.753**	.688	**.815**	.652	.621	.807
	G, D, L_A	.393	.372	.652	.764	.708	.851	.751	.683	.811	.657	.619	.806
	G, D, S_b	.384	.380	.655	.760	.702	.848	.739	.681	.810	.612	.592	.792
	G, D, S_b, L_A	.378	.311	.625	.771	.709	.852	.743	.684	.812	.630	.600	.796
iCR-Action-Detecter	G, D, S_b, S_a	**.416**	**.418**	**.676**	.859	.763	.880	.733	.684	.811	.834	.730	.862
	G, D, S_b, S_a, L_A	.409	.366	.652	.864	.777	.886	.739	**.689**	.813	.838	.738	.867

Table 2: Main results of average precision, binary F1 Score and macro-average F1 Score for all models in the test set. The inputs are G: gallery, D: dialogue, S_b: scene before the actions, S_a: scene after the actions, A: last gold actions, L_A: predicted logits of the actions. Shaded cells means the models were pre-trained on actions.

Implementation Our implementation uses PyTorch Lightning. We run hyperparameter search and other manual combinations, and then use the configuration that led to the best results in the validation set for the Overhearer-G+D model. The training objective is to minimise a sum of binary cross-entropy losses for each task. Optimisation relies on the Adam algorithm (Kingma and Ba, 2015), with early stopping. Pretrained word embeddings are retrieved from BERT (Devlin et al., 2019).[6]

Evaluation We report test results for the best epoch in the validation set.[7] H1 and H3 are analysed based on the performance on iCR predictions. To facilitate comparison to existing works, we report Average Precision (AP) and binary and macro-average F1-Score (bF1 and mF1) for each model and task (*i.e.* one measure for iCR labels and one for all action labels). To inspect how much information can be extracted from clipart states alone (*e.g.* some cliparts are less often subject to iCRs), we report metrics for a model that only gets the gallery as input. For H2, we need an additional prediction certainty metric. We adapt the classification margin metric used for uncertainty sampling in active learning (Settles, 2012), which is the difference between the probability assigned to the first and the second class, like in Chi et al. (2020). In our binary task, we define it as $|P(iCR) - P(\neg iCR)|$, which is 0 when both are 0.5 (highest uncertainty) and 1 when one or the other is 1 (highest certainty). We analyse whether we can derive a signal for *when to ask* iCRs by finding a decision threshold upon this metric, as in similar works (Yao et al., 2019; Naszad et al., 2022; Khalid and Stone, 2023).

7 Results

Table 2 presents the main results for all experiments. We begin with overall observations, and then walk through the table to analyse the findings for each hypothesis. In the next section, we discuss the implications of these findings.

Firstly, for deciding *when to ask* an iCR, the base Overhearer achieves 0.38 AP and the highest performance comes from the iCR-Action-Detecter with 0.41. This is noticeably higher than the 0.34 Overhearer baseline in Madureira and Schlangen (2023b), but the gain is not as substantial as expected given the improvements in the architecture.[8]

[6] Details of the model, data processing and experiment setup are in the Appendix. Our code is available at `https://anonymous.4open.science/r/codraw-icr-B518/`

[7] The full Transformer encoder-decoder was detrimental in almost all cases, so we report results using only the decoder component. We compared Overhearers using a context from 0 to 5 previous turns. 0 or 1 turns had worse results, but 2 to 5 were almost equivalent, so we report results using 3.

[8] Note that we use the second released version of the annotation, containing a marginally different proportion of iCRs.

When the Overhearer is ablated to have no access to the dialogue, performance drops to close to random, as expected. The addition of scenes before and after the current actions and the inclusion of an explicit signal with the last actions, however, cause only marginal variation and do not really contribute to a better performance. The Action-Taker similarly does not profit from having access to the image. We have no precedent results for the task of *what to ask* about, but even the Overhearer achieves more than .70 AP. Given the imbalance of the labels, we consider it a favourable result, showing this task is easier to model. Introducing iCR decisions does not cause drastic changes to the performance on taking actions for *when to ask*, but fine-tuning on *what to ask* causes a drop, which is probably due to the fine-tuning occurring only on iCR turns. See Appendix for additional analysis.

Hypothesis 1 In H1, we study the effect of action-taking on the decision of *when to ask* iCRs. To analyse it, we compare the results of the Overhearer with the iCR-Action-Taker/-Detecter in the left block of Table 2. Integrating multi-task learning for taking actions is slightly helpful for iCR prediction only if the action decision logits are passed to the iCR classifier. If instead of *predicting* actions we let the model learn the auxiliary task of just *detecting* them from the scenes, the results are better.[9] Interestingly, the magnitude of the positive difference is comparable to the difference (in accuracy) found in the Minecraft dataset (Shi et al., 2022), which was, however, negative. These effects are not large enough to provide us with definite evidence that H1 holds.

Hypothesis 2 For H2, we examine the certainty scores assigned by the Action-Taker to performing *any* action upon each clipart. For the task of *what to ask* about, we compare two distributions: Scores of cliparts subject to iCRs *versus* scores of cliparts not subject to iCRs. For *when to ask* iCRs, we inspect the distributions of the lowest score at turns where iCRs occur *versus* turns where no iCR is made. Using the two-sample Kolmogorov-Smirnov test (Hodges Jr, 1958), we compare the underlying empirical cumulative distributions of the two samples, shown in Figure 1, under the null hypothesis that they are equal, and a two-sided alternative.

Table 3 shows the statistically significant test results. It means that, on the whole, Action-Takers behave differently regarding action certainty for cliparts or turns with iCRs. In Figure 3, we can see that the certainty for non-iCR cliparts is more concentrated around 1 than for cliparts subject to iCRs. Similarly, the distribution of the minimum certainty score at iCR turns is more concentrated at lower values. In that sense, we find support for H2. Still, using these scores directly as a signal for iCR prediction does not result in high AP, in line with the findings by Naszad et al. (2022). This seems to occur because, although the distributions are different, both samples have values in the whole range, with overlap in their standard deviation.

	clipart (what to ask)		**turn (when to ask)**	
	iCR	**non-iCR**	**iCR**	**non-iCR**
mean (std)	.838 (.251)	.952 (.147)	.363 (.283)	.525 (.328)
KS test	.524*		.219*	
AP	.009		.080	

Table 3: Mean (std) of certainty scores for each sample, results of the two-sided Kolmogorov-Smirnov test and average precision. * means p-value < 0.001.

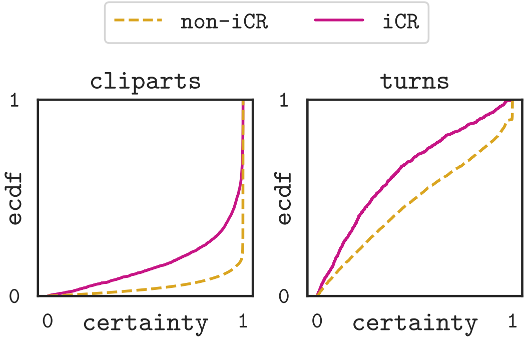


Figure 3: Empirical cumulative distribution function of the certainty of taking actions for each clipart (left) and the minimum by turn (right).

Hypothesis 3 For the last hypothesis, we assess the effect of taking actions in deciding *what to ask* about. Here, we focus on the right columns of Table 2, again comparing the Overhearer with the pretrained iCR-Action-Takers/Detecters. We observe a positive effect of learning to take actions on the iCR policy, with AP increasing from .69 to .75. Differently from the task of *when to ask*, here *predicting* actions leads to better results than merely *detecting* them. The difference is not negligible, which is stronger support in favour of H3 in this context.

[9] This is still plausible, because we can assume that the actual player has taken actions before generating the iCR, as discussed by Madureira and Schlangen (2023b).

8 Discussion

Our multi-task learning setting allowed us to differentiate between understandability and iCR policy. The first refers to learning a mapping from linguistic input to actions, the latter is an additional decision on top of action-taking that regards knowing when the information available to the agent at a given moment is not enough for the current purposes of wanting to commit to an outcome. Learning to take actions does not seem to be a signal informative enough for deciding *when to ask* for iCRs, although it has a more prominent effect on deciding *what to ask* about in iCR turns. Besides, we investigated whether there is a signal in the pure understanding models that predicts what to clarify. Indeed, a model trained without any explicit iCR signal makes predictions whose certainty distribution differ at iCR turns and cliparts. Even though the score cannot be directly used as a predictor of human iCR behaviour, further investigation can be done on extracting an agent's implicit iCR policies, *e.g.* with probing or attribution methods and in-depth analysis of the model's internal states.

The integration of the gallery, the token-level representations, learnable scene features and attention mechanism to construct contextual embeddings of the game state form together a conceptually superior design for our model in relation to the baseline. We expected this more sophisticated architecture, aligned with the latest literature, to lead up to a clear-cut improvement in the task of when to ask iCRs. The fact that the gain is not more than 10% in our main metric compel us to join the ranks of works that question whether the current NLP paradigm (employing imitation learning or behavioural cloning to learn with supervision from limited human data) is the right way to go when it comes to meta-discursive acts (Hayes, 1980; Nguyen et al., 2022; Min et al., 2022; Naszad et al., 2022; Bohg et al., 2023, *inter alia*).

In a static dataset of human play, the underlying CR policies of each player may differ by nature and also in visibility in the data. We cannot know with certainty if other humans would have behaved differently at each point than what is realised in the data; consequently, it is hardly possible to set a standard against which to judge the trained model's policy. We are, after all, trying to learn a "customary" policy from what is actually a mixture of policies with observations sampled from various players. It may be the case that we have reached the limits of the generalisable policies we can capture from this data, even though the actual metrics are not close to the ceiling. As Hayes (1980) discussed, graceful interaction requires developers to aim for non-literal aspects of communication that are effective for the human-agent interaction, instead of trying to imitate human patterns exactly. This connects to the over confidence problem in LLMs: In some situations, they should produce an *I don't know* or a CR, but their limited abilities in meta-semantic communication often cause failures.

CRs are not a problem: They are a solution emerging from joint effort. If many bits of information are to be conveyed, the instruction giver may produce minimally sufficient messages and leave it to the addressee to identify gaps. The instruction follower may also take actions that are only approximately good, since mistakes can normally be fixed later. Moreover, crowdworkers seem to lack incentive to try to build perfect reconstructions (see Appendix). Therefore, the iCR signal may not be "out there" in the data, but live in the internal state of the agents. Treating the task as *iid* predictions under supervised learning is also not ideal because actions and iCR decisions are actually made sequentially. Like some works on learning when to ask questions, modelling iCR policies may call for reinforcement learning (see *e.g.* Khalid et al. (2020)), with evaluation methods that capture the effectiveness of the agent's policy for the game, beyond comparison with human behaviour.

9 Conclusion

We have examined the effects of performing actions on learning iCR policies in the CoDraw game. The assumption that learning to take actions would make the underlying *when to ask* policy emerge does not fully hold. Still, we find that prediction certainty of actions differs at iCR turns. Then, if we assume that a given policy has informed us on *when* iCRs have to be made, we show that it is possible to predict *what to ask* about more successfully, with action-taking having a stronger positive effect. Exploring larger datasets with CRs produced as a by-product of action-taking is desired. Still, the suboptimal performance of various SOTA models in deciding *when to ask* for clarification speaks against approaches that seek to imitate human behaviour. We recommend more investigation with RL and evaluation methods that capture the effectiveness of iCR policies in dynamic contexts.

10 Limitations

In this work, out models do not predict all fine-grained game actions. In preliminary experiments, we first attempted to model an agent that predicts all features of each clipart at each turn. However, since most of the 28 available cliparts remain unchanged from one turn to the other, the model could simply learn to output a copy of the current state. We thus opted to turn all tasks into binary predictions for our analysis, as we observed results that are good enough for our purposes, given the imbalanced nature of the actions in the data. A full agent should include the subsequent tasks of deciding where to place cliparts and what exact (discrete) size to set (presence and orientation can be deduced in post-processing with the current version).

Further investigation can be done to improve the performance of the Action-Takers. Since the actions are very sparse, it may be the case that models just learn to detect mentioned cliparts in the utterances. A detailed error analysis should look closer at the predictions and also examine how good the scene similarity scores of the reconstructions are. Instead of predicting probabilities, the model could also output parameters of a distribution from which the actions would be sampled; we do not investigate that option here. Besides, we use a supervised learning approach that treats turns as *iid*. In reality, what the player does in one turn influences its next moves, so other methods like RL could be more appropriate, as we discussed.

Although our models take several epochs to overfit the training data, performance in the validation set saturates very early. The techniques we tried (for instance, dropout, variations of the architecture and filtering the training data) did not lead to better results. We performed a limited hyperparameter search that could be done more extensively in the future.

For the task of *what to ask* about, we did not include the utterances for which the annotation does not provide the reference cliparts due to ambiguity. Still, that happens for very few cases and should not have a considerable impact on the results.

To conclude, we do not have human performance to compare our results to. It would be interesting to collect human data by letting humans decide *when to ask* for clarification and *what to ask* about, so that we can better understand to what extent the task itself is possible for humans acting as overhearers.

References

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.

Luciana Benotti. 2009. Clarification potential of instructions. In *Proceedings of the SIGDIAL 2009 Conference*, pages 196–205, London, UK. Association for Computational Linguistics.

Luciana Benotti and Patrick Blackburn. 2021. A recipe for annotating grounded clarifications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4065–4077, Online. Association for Computational Linguistics.

Jeannette Bohg, Marco Pavone, and Dorsa Sadigh. 2023. Principles of robot autonomy. http://web.stanford.edu/class/cs237b/. [Stanford lecture notes available online; session 12.].

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. 2012. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1290–1301, Jeju Island, Korea. Association for Computational Linguistics.

Pavel Braslavski, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka. 2017. What do you mean exactly? analyzing clarification questions in cqa. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, page 345–348, New York, NY, USA. Association for Computing Machinery.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-tur. 2020. Just ask: An interactive learning framework for vision and language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2459–2466.

Javier Chiyah-Garcia, Alessandro Suglia, Arash Eshghi, and Helen Hastie. 2023. 'what are you referring to?'evaluating the ability of multi-modal dialogue models to process clarificational exchanges. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Prague, Czech Republic. Association for Computational Linguistics.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Yang Deng, Wenqiang Lei, Lizi Liao, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Malte Gabsdil. 2003. Clarification in spoken dialogue systems. In *Proceedings of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*, pages 28–35.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. 2022. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056.

Spandana Gella, Aishwarya Padmakumar, Patrick Lange, and Dilek Hakkani-Tur. 2022. Dialog acts for task driven embodied agents. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 111–123, Edinburgh, UK. Association for Computational Linguistics.

Felix Gervits, Antonio Roque, Gordon Briggs, Matthias Scheutz, and Matthew Marge. 2021. How should agents ask questions for situated learning? an annotated dialogue corpus. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 353–359, Singapore and Online. Association for Computational Linguistics.

Phil Hayes. 1980. Expanding the horizons of natural language interfaces. In *18th Annual Meeting of the Association for Computational Linguistics*, pages 71–74, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Phil Hayes and Raj Reddy. 1979. Graceful interaction in man-machine communication. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 1*, pages 372–374.

Philip J Hayes and D Raj Reddy. 1983. Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies*, 19(3):231–284.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

JL Hodges Jr. 1958. The significance probability of the smirnov two-sample test. *Arkiv för matematik*, 3(5):469–486.

Xiang Hu, Zujie Wen, Yafang Wang, Xiaolong Li, and Gerard de Melo. 2020. Interactive question clarification in dialogue via reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 78–89, Online. International Committee on Computational Linguistics.

Kimiya Keyvan and Jimmy Xiangji Huang. 2022. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Computing Surveys*, 55(6):1–40.

Baber Khalid, Malihe Alikhani, and Matthew Stone. 2020. Combining cognitive modeling and reinforcement learning for clarification in dialogue. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4417–4428, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Baber Khalid and Matthew Stone. 2023. Investigating reinforcement learning for communication strategies in a task-initiative setting. *arXiv preprint arXiv:2308.01479*.

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*, pages 6495–6513, Florence, Italy. 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*.

Julia Kiseleva, Alexey Skrynnik, Artem Zholus, Shrestha Mohanty, Negar Arabzadeh, Marc-Alexandre Côté, Mohammad Aliannejadi, Milagro Teruel, Ziming Li, Mikhail Burtsev, et al. 2022. Iglu 2022: Interactive grounded language understanding in a collaborative environment at neurips 2022. *arXiv preprint arXiv:2205.13771*.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Clam: Selective clarification for ambiguous questions with large language models. *arXiv preprint arXiv:2212.07769*.

Vaibhav Kumar and Alan W Black. 2020. ClarQ: A large-scale and diverse dataset for clarification question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7296–7301, Online. Association for Computational Linguistics.

Staffan Larsson. 2017. User-initiated sub-dialogues in state-of-the-art dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 17–22, Saarbrücken, Germany. Association for Computational Linguistics.

Haeju Lee, Oh Joon Kwon, Yunseon Choi, Minho Park, Ran Han, Yoonhyung Kim, Jinhyeon Kim, Youngjune Lee, Haebin Shin, Kangwook Lee, and Kee-Eung Kim. 2022. Learning to embed multi-modal contexts for situated conversational agents. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 813–830, Seattle, United States. Association for Computational Linguistics.

Haau-Sing (Xiaocheng) Li, Mohsen Mesgar, André Martins, and Iryna Gurevych. 2023. Python code generation by asking clarification questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14287–14306, Toronto, Canada. Association for Computational Linguistics.

Brielen Madureira and David Schlangen. 2023a. “Are you telling me to put glasses on the dog?” Content-grounded annotation of instruction clarification requests in the CoDraw dataset. *arXiv preprint arXiv:2306.02377*.

Brielen Madureira and David Schlangen. 2023b. Instruction clarification requests in multimodal collaborative dialogue games: Tasks, and an analysis of the CoDraw dataset. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2303–2319, Dubrovnik, Croatia. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what’s missing and what’s useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312, Online. Association for Computational Linguistics.

So Yeon Min, Hao Zhu, Ruslan Salakhutdinov, and Yonatan Bisk. 2022. Don’t copy the teacher: Data and model challenges in embodied dialogue. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9361–9368, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shrestha Mohanty, Negar Arabzadeh, Julia Kiseleva, Artem Zholus, Milagro Teruel, Ahmed Awadallah, Yuxuan Sun, Kavya Srinet, and Arthur Szlam. 2023. Transforming human-centered ai collaboration: Redefining embodied agents capabilities through interactive grounded language instructions.

Shrestha Mohanty, Negar Arabzadeh, Milagro Teruel, Yuxuan Sun, Artem Zholus, Alexey Skrynnik, Mikhail Burtsev, Kavya Srinet, Aleksandr Panov, Arthur Szlam, Marc-Alexandre Côté, and Julia Kiseleva. 2022. Collecting interactive multi-modal datasets for grounded language understanding.

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*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Kata Naszad, Michiel Van Der Meer, Kim Taewoon, and Putra Manggala. 2022. Learning to ask timely questions in a collaborative grounded language understanding task. unpublished two-page abstract.

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)*, pages 684–695, Hong Kong, China. Association for Computational Linguistics.

Khanh X Nguyen, Yonatan Bisk, and Hal Daumé Iii. 2022. A framework for learning to request rich and contextually useful information from humans. In *International Conference on Machine Learning*, pages 16553–16568. PMLR.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025.

Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2001. On the means for clarification in dialogue. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2003. On the means for clarification in dialogue. In *Current and new directions in discourse and dialogue*, pages 235–255. Springer.

Hossein A. Rahmani, Xi Wang, Yue Feng, Qiang Zhang, Emine Yilmaz, and Aldo Lipani. 2023. A survey on asking clarification questions datasets in conversational systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2698–2716, Toronto, Canada. Association for Computational Linguistics.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Verena Rieser and Oliver Lemon. 2006. Using machine learning to explore human multimodal clarification strategies. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 659–666, Sydney, Australia. Association for Computational Linguistics.

Kepa Joseba Rodríguez and David Schlangen. 2004. Form, intonation and function of clarification requests in german task-oriented spoken dialogues. In *Proceedings of Catalog (the 8th workshop on the semantics and pragmatics of dialogue; SemDial04)*.

Robin Rojowiec, Jana Götze, Philipp Sadler, Henrik Voigt, Sina Zarrieß, and David Schlangen. 2020. From “before” to “after”: Generating natural language instructions from image pairs in a simple visual domain. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 316–326, Dublin, Ireland. Association for Computational Linguistics.

Philipp Sadler and David Schlangen. 2023. Pento-DIARef: A diagnostic dataset for learning the incremental algorithm for referring expression generation from examples. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2106–2122, Dubrovnik, Croatia. Association for Computational Linguistics.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

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

David Schlangen. 2021. Targeting the benchmark: On methodology in current natural language processing research. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 670–674, Online. Association for Computational Linguistics.

Julian Schlöder and Raquel Fernández. 2014. Clarification requests at the level of uptake. In *Proceedings of the 18th Workshop on the Semantics and Pragmatics of Dialogue - Poster Abstracts*, Edinburgh, United Kingdom. SEMDIAL.

Burr Settles. 2012. *Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning*. Springer Cham.

Ravi Shekhar, Tim Baumgärtner, Aashish Venkatesh, Elia Bruni, Raffaella Bernardi, and Raquel Fernandez. 2018. Ask no more: Deciding when to guess in referential visual dialogue. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1218–1233, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to execute actions or ask clarification questions. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2060–2070, Seattle, United States. Association for Computational Linguistics.

Zhengxiang Shi, Jerome Ramos, To Eun Kim, Xi Wang, Hossein A Rahmani, and Aldo Lipani. 2023. When and what to ask through world states and text instructions: Iglu nlp challenge solution. *arXiv preprint arXiv:2305.05754*.

Svetlana Stoyanchev, Alex Liu, and Julia Hirschberg. 2013. Modelling human clarification strategies. In *Proceedings of the SIGDIAL 2013 Conference*, pages 137–141, Metz, France. Association for Computational Linguistics.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7322–7329.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. 2019. Model-based interactive semantic parsing: A unified framework and a text-to-SQL case study. 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)*, pages 5447–5458, Hong Kong, China. Association for Computational Linguistics.

C Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016.

A Additional Analysis

Here we present additional analysis. Figure 4 illustrates the distribution of the number of actions per turn. Table 4 presents the average precision for each type of action, which are aggregated in Table 2. Figure 5 show the boxplots for the distribution of certainty scores, to aid visualising that they have different shapes in each sample.

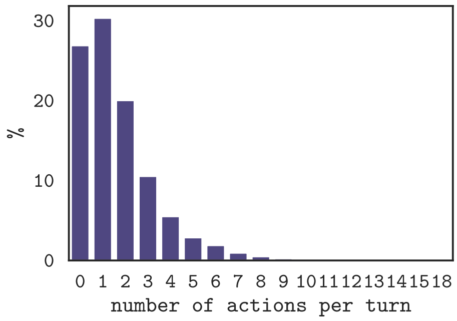


Figure 4: Empirical distribution of the number of actions per turn in the CoDraw dataset.

	add/del	move	flip	resize
Action-Taker G, D	.875	.617	.367	.531
iCR-Action-Taker G, D	.865	.600	.398	.539
Action-Detecter G, D, $S_{a,b}$	.976	.644	.414	.636
iCR-Action-Detecter G, D, $S_{a,b}$	.974	.642	.423	.626

Table 4: Detailed performance of the Action-Takers and Action-Detecters for *when to ask*. Values are the average precision for each type of action in the test set.

B Reproducibility

In this section, we provide details of our data preprocessing and implementation. For precise details, please check the available code. Here, we provide a brief overview of each component and the justification of some decisions.

B.1 Data

We used the annotation released in the file `codraw-icr-v2.tsv`[10] to identify iCRs and mentioned cliparts. We followed the train-val-test splits as in the original CoDraw data. The *ambiguity classes* introduced by the authors were treated as follows: If an iCR was about an ambiguous but concrete class, we assigned the positive iCR label to all objects in the gallery that belong to that class. For instance, for `hat_group`, all hats in the gallery were treated as positive cases. The general ambiguity class, used for unclear cases, was ignored in our labelling. This occured in 318 iCRS. The whole dataset was used in all experiments, except for the tasks of *what to ask* about, for which only the turns containing iCRs were included for all splits.

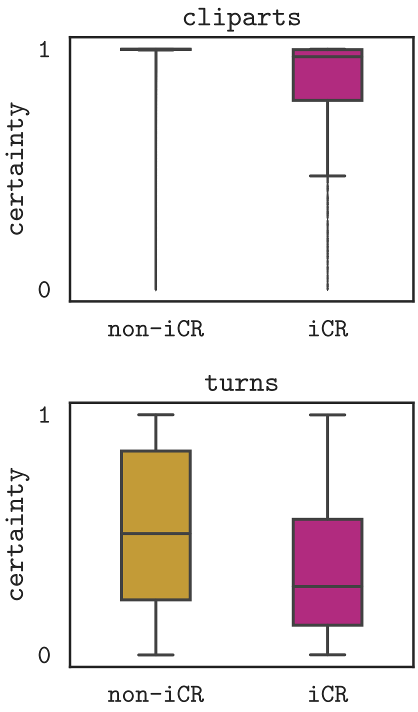


Figure 5: Empirical distribution of the certainty of taking actions for each clipart (top) and the minimum by turn (bottom).

The gallery and scene representation was constructed using features in a similar fashion as the original paper. Each clipart was assigned integers for its identifier, size (three categories), orientation (two categories), presence in the current scene (a binary feature), pose (seven categories) and facial expression (5 categories), as well as five features for its position (x and y coordinates of its centre, width, height and area in the canvas). We set features (except pose and facial expression) to a special category 0 for objects that are not in the scene. All cliparts for the boy and the girl were collapsed into one class for each, and their facial expressions and poses were turned into features in the symbolic representation, as in original paper. Other cliparts were assigned a "not-applicable" class for these two

[10] https://osf.io/gcjhz/files/osfstorage

features. To define the bounding boxes, rescaling the sizes was done according to the AbstractScenes documentation.

Actions were defined as either addition/deletion or edits. Edits meant flip, resize and move. If a clipart was added or deleted, we did not consider changes to its orientation, position and size with respect to the gallery (in order to avoid that the model only learnt the edits that occur due to an addition or deletion). Actions were defined by comparing the state of the gallery in a turn in relation to its state in the previous turn. For initial turns and some cases where the scene string was not available in the dataset, we set the scene to empty and use the gallery in adjacent turns (since the gallery should remain the same across the game). We also introduced an "acted upon" action that is positive whenever any type of action occurs upon a clipart.

Text embeddings were retrieved from `bert-base-uncased`, licensed under Apache 2.0. Following Shi et al. (2022), we concatenate the instruction giver and follower utterances using special tokens before each speaker. Special tokens `<TELLER>` and `<DRAWER>` were appended before the instruction giver and follower, respectively. The last utterance from the instruction follower was appended to the beginning of the utterance of the instruction giver, so that potential previous iCRs are encoded with their responses, if given immediately. Embedding sequences were padded with zeros to the right to an empirical length of 80 tokens. When context is used, the previous turns are appended to the left of the last instruction and, if necessary, padded with zeros to the left, so that the most recent turn is always at the same position in the input.

B.2 Implementation

The models were implemented with Python (v3.10.12), PyTorch[11] (v1.13.1) and Pytorch Lightning[12] (v2.0.8), in Linux 5.4.0-99-generic with processor x86_64 on an NVIDIA GeForce GTX 1080 Ti GPU with CUDA (v11.6). The pre-trained ResNet model was retrieved from torchvision[13] (v0.14.1) and the pre-trained BERT model was loaded from HuggingFace transformers[14] (v4.29.2).

Optimisation was done with the Adam algorithm (Kingma and Ba, 2015), using `BCEWithLogitsLoss` with `reduction` set to sum and the argument `pos_weight` to 2 for each task. The total loss used for backpropagation was a sum of all task losses.

Early stopping was implemented using a patience of 8 epochs and the minimum delta of 0.001 for maximisation of a monitored metric. Metrics were computed using torchmetrics[15] (v0.11.4). The monitored metric varied according to the task: If iCRs were predicted, we tracked the binary average precision of iCR labels; otherwise, we tracked the binary average precision of the meta-action class. The maximum number of epochs was set to 30. The checkpoint that lead to best performance in the validation set was saved and loaded to run the tests. Comet[16] was used to manage experiments and to perform hyperparameter search.

Hyperparameter search was performed with the base model (*i.e.* an Overhearer that gets only the dialogue and the gallery representation as input and predicts only *when to ask* iCRs). We used comet's Bayes algorithm as well as a few manual selections of hyperparameters, and opted for the model with highest iCR binary average precision in the validation set. Table 5 shows the final hyperparameter configuration used in all experiments.

We did not keep records of all experiments during development. For the final run, we run 43 experiments during tuning and 102 for the analysis. The duration varied from 5 minutes (the random baseline) to 06h16m (the iCR-Action-Detecter using the full Transformer), without including the time for data preparation. The number of parameters varied according to the model. The turn-level Overhearer without scenes had 5,008,923 and with both scenes 29,054,299 (5,546,267 learnable). The turn-level iCR-Action-Taker without scenes had 5,339,168, and the iCR-Action-Detecter had 29,384,544 (5,876,512 learnable).

To enable reproducibility, we set the use of use deterministic algorithms to `True` in PyTorch and used Lightning's `seed_everything` method with a fixed random seed. Despite this, according to the documentation, some methods cannot be forced to be deterministic in PyTorch when using CUDA.[17]

[11] https://pytorch.org/

[12] https://lightning.ai/pytorch-lightning

[13] https://pytorch.org/vision/stable/models.html

[14] https://huggingface.co/bert-base-uncased

[15] https://torchmetrics.readthedocs.io/en/latest/

[16] https://www.comet.com

[17] https://pytorch.org/docs/1.13/generated/torch.use_deterministic_algorithms.html#torch.use_deterministic_algorithms

hyperparameter	type	options	selected
accumulate gradient	discrete	1, 2, 5, 10, 25	1
batch size	discrete	16, 32, 64, 128, 256	32
clipping	discrete	0, 0.25, 0.5, 1, 2.5, 5	1
context length	integer	min=1, max=5	3
dropout	discrete	0.1, 0.2, 0.3	0.1
d_model	discrete	128, 256, 512	256
hidden_dim	discrete	32, 64, 128, 256, 512, 1024	256
hidden_dim_trf	discrete	256, 512, 1024	2048
learning rate	discrete	0.1, 0.01, 0.001, 0.0001, 0.003, 0.0003, 0.00001, 0.0005	0.0001
lr scheduler	bool	True, False	False
lr step	integer	min=1, max=10	-
n heads	discrete	1, 2, 4, 8, 32	16
n layers	float	min=1, max=6	3
n reload datasets	float	min=1, max=10	1
pos weight	float	min=0.8, max=3	2
pre-trained text embeddings	categorical	bert-base-uncased, roberta-base, distilbert-base-uncased	bert-base-uncased
random seed	integer	min=1, max=54321	12345
weight decay	discrete	1, 0.1, 0.01, 0.001, 0.0001	0.
weighted loss	bool	True, False	False

Table 5: Hyperparameters: Investigated options and selected values. Note that the search did not extensively cover all possibilities for each hyperparameter.

Details of the models' components Let d_model be the dimension used for the Transformer. First of all, an embedding of the gallery and scene state is constructed. Embedding layers are used for a clipart's identifier, orientation, presence, size, face and pose states with dimensions d_model-100, 10, 10, 10, 20 and 20, respectively. The position is embedded with a linear layer that maps its centre coordinates, area, width and height to 30 dimensions. All embedded features are concatenated so as to create a representation with dimensions 28 (number of cliparts) by d_model. We used only the decoder of the Transformer, which gets the gallery representation as "target" and the instruction tokens (whose dimensions were reduced with a linear layer and, if applicable, the sequence was concatenated to the scene features) summed to positional encodings as "memory". The decoder performs self-attention in the gallery and then cross-attention with the memory. Scenes are encoded following Carion et al. (2020)'s implementation, but we first preprocess the scene according to the pretrained model's documentation. The scene is then fed into a pre-trained ResNet50 followed by a trainable convolutional layer that reduces the number of channels to the same dimension used for the Transformer. Then, the height and width dimensions are flattened and the result is added to learnable position embeddings, with a dropout layer. The probabilities (for iCRs or actions) are predicted by taking each output of the Transformer (*i.e.* one representation for each clipart in the gallery) and passing it through a feed-forward network with the following sequential layers: leaky ReLU, dropout, linear, leaky ReLU and linear. For predicting turn-level iCRs, the representations of all cliparts are averaged. If the action-taking logits or teacher forcing is used, they are appended to the input. The outputs are logits that are converted to probabilities using the sigmoid function.

Evaluation The threshold for the F1-Scores was set to 0.5. We did not include the meta-action label in the main results for taking actions to avoid inflating the performance; it was only used for the analysis for H2, done on the Action-Taker+G, D. Metrics for the evaluation were computed with sklearn[18] (v1.0.2) and the plots were generated with seaborn (v0.12.2) and matplotlib[19] (v3.7.1). The hypothesis test was performed with SciPy[20] (v1.11.1) `stats.ks_2samp` method with a two-sided alternative.

C CoDraw Examples

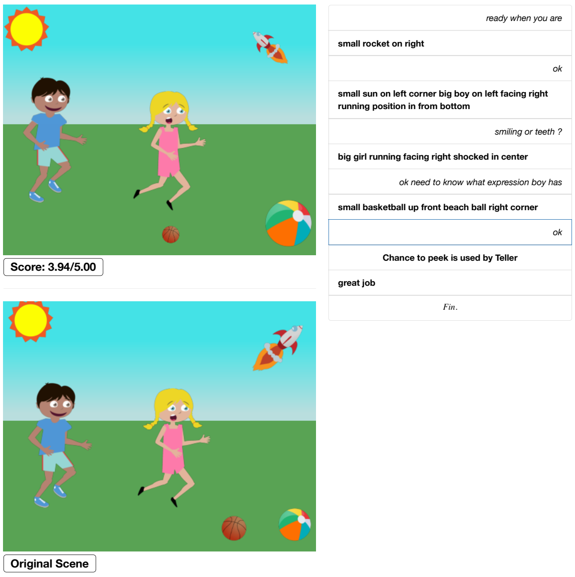
Figures 6-9 exemplify strategies of crowdworkers, showing various levels of commitment to playing the game well.[21] Scenes at the top are the state of the reconstructions at the highlighted turns.

[18] https://scikit-learn.org/stable/index.html

[19] https://matplotlib.org/

[20] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html

[21] The images are generated with the CoDraw Dataset Visualizer, developed by @jnhwkim at https://github.com/facebookresearch/CoDraw.



ready when you are

small rocket on right

ok

small sun on left corner big boy on left facing right running position in from bottom

smiling or teeth ?

big girl running facing right shocked in center

ok need to know what expression boy has

small basketball up front beach ball right corner

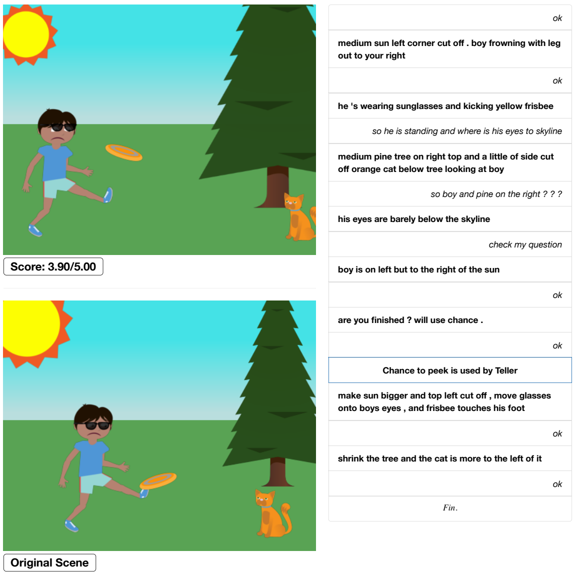
ok

Chance to peek is used by Teller

great job

Fin.

Figure 6: Even peeking, the instruction giver does not inform the instruction follower that the reconstruction is not totally correct: The orientation of the rocket is wrong, as well as the position of the basketball and the size of the two balls. From: CoDraw dialogue game 488, CC BY-NC 4.0, scene from Zitnick and Parikh (2013).



ok

medium sun left corner cut off . boy frowning with leg out to your right

ok

he 's wearing sunglasses and kicking yellow frisbee

so he is standing and where is his eyes to skyline

medium pine tree on right top and a little of side cut off orange cat below tree looking at boy

so boy and pine on the right ? ? ?

his eyes are barely below the skyline

check my question

boy is on left but to the right of the sun

ok

are you finished ? will use chance .

ok

Chance to peek is used by Teller

make sun bigger and top left cut off , move glasses onto boys eyes , and frisbee touches his foot

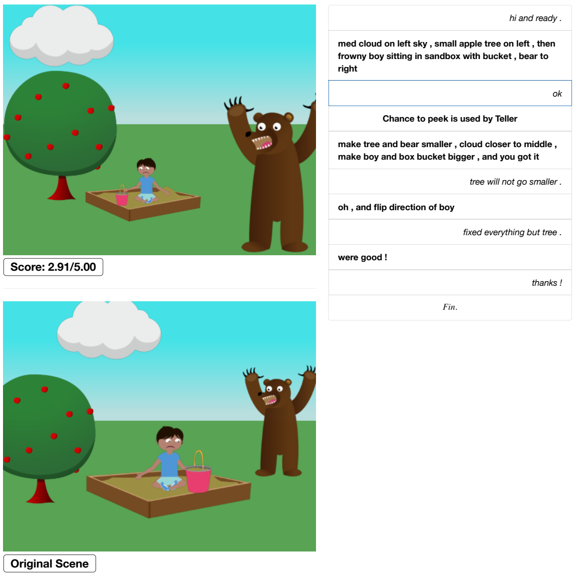
ok

shrink the tree and the cat is more to the left of it

ok

Fin.

Figure 7: A more careful instruction giver uses two turns to try to repair even minor details after the peek, like the slightly wrong position of the sunglasses. From: CoDraw dialogue game 198, CC BY-NC 4.0, scene from Zitnick and Parikh (2013).



hi and ready .

med cloud on left sky , small apple tree on left , then frowny boy sitting in sandbox with bucket , bear to right

ok

Chance to peek is used by Teller

make tree and bear smaller , cloud closer to middle , make boy and box bucket bigger , and you got it

tree will not go smaller .

oh , and flip direction of boy

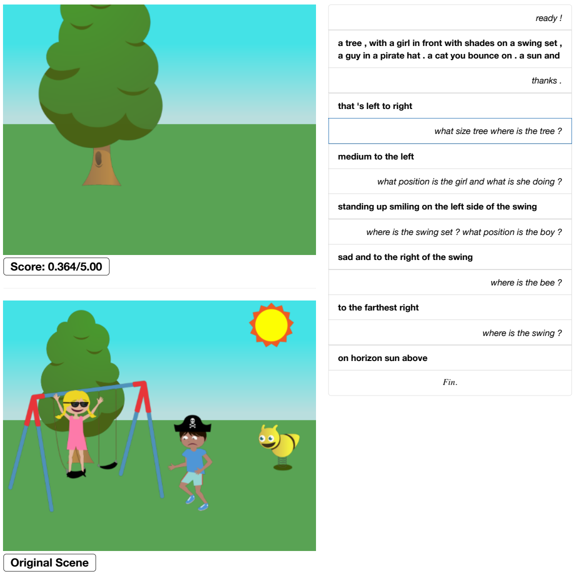
fixed everything but tree .

were good !

thanks !

Fin.

Figure 8: The instruction follower gets underspecified instructions at the first turn (for instance, nothing is said about the orientation of the boy and his position with respect to the bucket), but acts even so without asking for clarification. From: CoDraw dialogue game 3835, CC BY-NC 4.0, scene from Zitnick and Parikh (2013).



ready !

a tree , with a girl in front with shades on a swing set , a guy in a pirate hat . a cat you bounce on . a sun and

thanks .

that 's left to right

what size tree where is the tree ?

medium to the left

what position is the girl and what is she doing ?

standing up smiling on the left side of the swing

where is the swing set ? what position is the boy ?

sad and to the right of the swing

where is the bee ?

to the farthest right

where is the swing ?

on horizon sun above

Fin.

Figure 9: The instruction giver provides underspeficied instructions at the first turn. Instead of taking all actions immediately, the instruction follower does many rounds of clarification. From: CoDraw dialogue game 4286, CC BY-NC 4.0, scene from Zitnick and Parikh (2013).