Dialog Acts for Task-Driven Embodied Agents

Spandana Gella[*], **Aishwarya Padmakumar**[*], **Patrick Lange, Dilek Hakkani-Tur**
Amazon Alexa AI
{sgella,padmakua,patlange,hakkanit}@amazon.com

Abstract

Embodied agents need to be able to interact in natural language – understanding task descriptions and asking appropriate follow up questions to obtain necessary information to be effective at successfully accomplishing tasks for a wide range of users. In this work, we propose a set of dialog acts for modelling such dialogs and annotate the *TEACh* dataset that includes over 3,000 situated, task oriented conversations (consisting of 39.5k utterances in total) with dialog acts. *TEACh-DA* is one of the first large scale dataset of dialog act annotations for embodied task completion. Furthermore, we demonstrate the use of this annotated dataset in training models for tagging the dialog acts of a given utterance, predicting the dialog act of the next response given a dialog history, and use the dialog acts to guide agent's non-dialog behaviour. In particular, our experiments on the *TEACh* Execution from Dialog History task where the model predicts the sequence of low level actions to be executed in the environment for embodied task completion, demonstrate that dialog acts can improve end task success rate by up to 2 points compared to the system without dialog acts.

1 Introduction

Natural language communication has the potential to significantly improve the accessibility of embodied agents. Ideally, a user should be able to converse with an embodied agent as if they were conversing with another person and the agent should be able to understand tasks specified at varying levels of abstraction and request for help as needed, identifying any additional information that needs to be obtained in follow up questions. Human-human dialogs that demonstrate such behavior are critical to the development of effective human-agent communication. Annotation of such dialogs with dialog acts is beneficial to better understand common conversational situations an agent will need to handle (Gervits et al., 2021). Dialog acts can also be used in building task oriented dialog systems to plan how an agent should react to the current situation (Williams et al., 2014).

In this paper, we design a dialog act annotation schema for embodied task completion based on the dialogs of the *TEACh* dialog corpus (Padmakumar et al., 2021). *TEACh* is a dataset of over 3,000 situated text conversations between human annotators role playing a user (*Commander*) and a robot (*Follower*) collaborating to complete household tasks such as making coffee and preparing breakfast in a simulated environment. The tasks are hierarchical, resulting in agents needing to understand task instructions provided at varying levels of abstraction across dialogs. The human annotators had a completely unconstrained chat interface for communication, so the dialogs reflect natural conversational behavior between humans, not moderated by predefined dialog acts or turn taking. Additionally, the *Follower* had to execute actions in the environment that caused physical state changes which were examined to determine whether a task was successfully completed. We believe that these annotations will enable the study of more realistic dialog behaviour in situated environments, unconstrained by turn taking.

Summarizing our contributions:

- We propose a new schema of dialog acts for task-driven embodied agents. This consists of 18 dialog acts capturing the most common communicative functions used in the *TEACh* dataset.
- We annotate the *TEACh* dataset according to the proposed schema to create the *TEACh-DA* dataset.
- We investigate the use of the proposed dialog acts in an extensive suite of tasks related to language understanding and action prediction for task-driven embodied agents.

[*]These two authors contributed equally.

We establish baseline models for classifying the dialog act of a given utterance in our dataset and predicting the next dialog act given an utterance and conversation history. Additionally, we explore whether dialog acts can aid in plan prediction - predicting the sequence of object manipulations the agent needs to make to complete the task, and Execution from Dialog History (EDH) - where the agent predicts low level actions that are executed in the virtual environment and directly evaluated on whether required state changes were achieved.

2 Related Work

Dialog act annotations are common in language-only task-oriented dialog datasets, and are commonly used to plan the next agent action in dialog management or next user action in user simulation (Williams et al., 2014; Budzianowski et al., 2018; Schuster et al., 2019; Hemphill et al., 1990; Feng et al., 2020; Byrne et al., 2019). Many frameworks have been proposed to perform such annotations. Some examples are DAMSL (Dialog Act Markup in Several Layers) and ISO (International Organization for Standardization) standard (Core and Allen, 1997; Young, 2007; Bunt et al., 2009; Mezza et al., 2018). Such standardization of dialog acts across applications has been shown to be beneficial for improving the performance of dialog act prediction models (Mezza et al., 2018; Paul et al., 2019).

Most task-oriented dialog (TOD) applications and dialog act coding standards assume that the tasks to be performed can be fully specified in terms of slots whose values are entities (Young, 2007). However, we find that if we need to adopt a slot-value scheme for multimodal task-oriented dialog datasets such as *TEACh*, much of the information that needs to be conveyed is not purely in the form of entities. For example, If an utterance providing a location of an object: "the cup is in the drawer to the left of the sink" is to be coded at the dialog act level simply as an `INFORM` act, it could for example have a slot value called `OBJECT_LOCATION` but the value of this would need to refer to most of the utterance, i.e. "the drawer to the left of the sink". Hence, we define more fine-grained categories, such as `InfoObjectLocAndOD` (information on object location and other details) in *TEACh-DA*. These categories are designed in a way so that they could be re-purposed into broader dialog act category and intent/slot in the future by merging categories, if needed. As in a TOD, inform would be the DA tag, intent could be `inform_object_location` or `object_location` could be slot category. Thus, we combine the use of many standardized dialog acts such as `Greetings`, `Acknowledge`, `Affirm / Deny` with domain-specific finer grained dialog acts replacing the typical `Inform` and `Request` dialog acts.

Additionally, since the *TEACh* dataset is not constrained by turn taking or a pre-defined dialog flow, sometimes a single utterance may perform multiple communicative functions. To address this, similar to Core and Allen 1997, we allow multiple dialog acts per utterance and require annotators to mark utterance spans corresponding to each dialog act.

There exist other multimodal task-oriented dialog datasets that include annotations of dialog acts such as Situated and Interactive Multimodal Conversations (SIMMC 2.0) (Kottur et al., 2021) and Multimodal Dialogues (MMD) (Saha et al., 2018). These are multimodal datasets in the shopping domain that allows users to view products visually, and engage in dialog with an agent where the agent can take actions to refine the products available for the user to view. However, in contrast to the *TEACh* dataset considered in our work, the dialogs are created by first simulating probable dialog flows and then having annotators paraphrase utterances. As such, in these datasets, utterances clearly map to predefined dialog acts and follow patterns expected by the designers. These may not fully cover the range of possible conversational flows that can happen between humans in an unconstrained multimodal context, as can be observed in *TEACh*. The Human Robot Dialogue Learning (HuRDL) corpus includes annotations of human-human multimodal dialogs, with a focus on classifying different types of clarification questions to be used by a dialog agent (Gervits et al., 2021) but it is limited in size - consisting of only 22 dialogs, in contrast to the over 3,000 dialogs in *TEACh*. Another related dataset is MindCraft (Bara et al., 2021) where annotators are periodically asked to answer questions in the middle of the collection of dialog sessions to elicit their belief states. However, belief states do not map directly to utterances and do not directly capture communicative intents, differentiating them from dialog acts.

Prior works propose models for predicting dialog acts given the current utterance and context (Kalch-

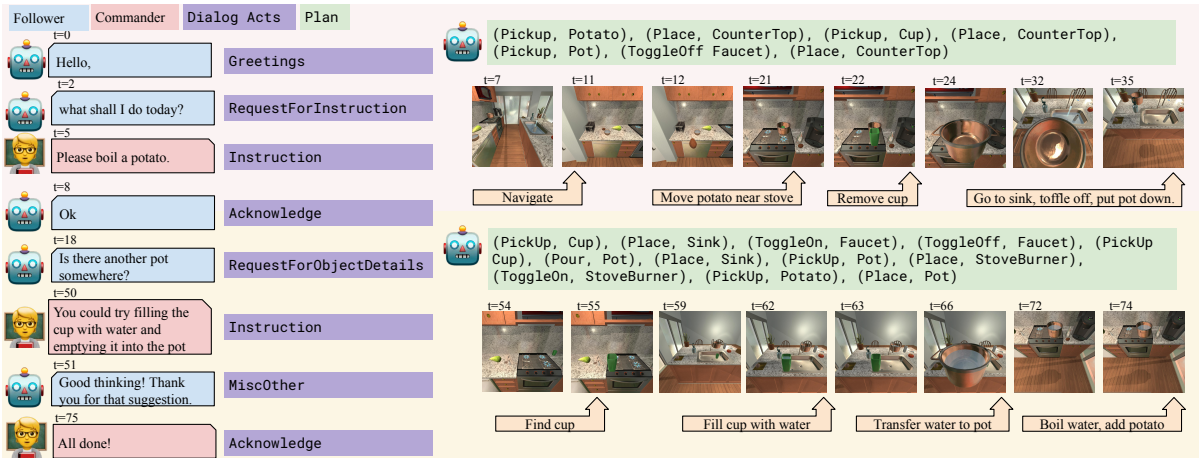

Figure 1: Illustration of example session for the task *Boil Potato* with corresponding dialog acts for each utterance and plans with corresponding actions in the game session.

brenner and Blunsom, 2013; Lee and Dernoncourt, 2016; Ribeiro et al., 2019), dialog acts of previous utterances or both (Paul et al., 2019). We perform similar experiments on our dataset to tag the dialog acts of given utterances and also to predict the dialog acts of future utterances. Due to the limited set of situated dialog datasets annotated with dialog acts, there has been relatively limited work on exploring the benefit of dialog acts on predicting an agent's future behavior in the environment. However, there are works that explore when to engage in a dialog as opposed to acting in the environment (Gervits et al., 2020; Chi et al., 2020; Shrivastava et al., 2021). While we do not directly model this problem, we experiment with the *TEACh* Execution from Dialog History task, where the end of our predicted action sequence would signal the need for another dialog utterance.

3 *TEACh-DA* dataset

The *TEACh* dataset (Padmakumar et al., 2021) consists of situated dialogs between human annotators role playing a user (*Commander*) and robot (*Follower*) collaborating to complete household tasks. In each dialog session, there is a high level task that the *Follower* is expected to accomplish, for example MAKE COFFEE or PREPARE BREAKFAST. Details of the task are known to the *Commander* but not the *Follower*. The *Follower* needs to engage in a dialog with the user to identify the task to be completed, customize the task (for example identify what dishes need to be prepared for breakfast) or obtain additional information such as locations of relevant objects, or more detailed steps needed to accomplish a task, and translate these to actions that can be executed in a simulated environment to complete the task.

In this work, we annotate the *TEACh* dataset with dialog acts (we refer to this new, annotated dataset as *TEACh-DA*) to better understand how language is used in task-oriented situated dialogs. We also explore the usefulness of these dialog acts to develop better agents that can converse in natural language and act in a situated environment for task completion.The *TEACh-DA* dataset consists of 39.5k utterances from 3,000 dialogs, 60% of which are from the *Commander* and the rest from the *Follower*.

We find that other dialog act frameworks for multimodal datasets (Gervits et al., 2021; Kottur et al., 2021; Saha et al., 2018) tend to be domain specific and do not cover all utterance types that would be beneficial for embodied task completion. Hence, we propose a new set of dialog acts for embodied task completion based on the communicative functions we observe in the *TEACh* dataset. Whenever possible, for utterances that are not very specific to the *TEACh* task, we have borrowed dialog acts from prior work. These include dialog acts related to generic chit chat such as `Greetings`, `Affirm`, `Deny` and `Acknowledge` (Paul et al., 2019).

In total, we defined 18 dialog acts that covered all utterances in *TEACh*. Our careful analysis of utterances in *TEACh* data lead to 5 broader categories of dialog acts as shown in Table 1.

- Generic: Acts that fall under conventional dialog such as opening and closing of dialog,
- Instruction Related: Which represent the utterances related to actions that should be per-

Dialog Act	Category	Example	Count	*Commander*(%)	*Follower*(%)
`Instruction`	Instruction	fill the mug with coffee	11019	99.4	0.6
`ReqForInstruction`	Instruction	what should I do today?	4043	0.7	99.3
`RequestOtherInfo`	Instruction	How many slices of tomato?	675	0.75	99.25
`RequestMore`	Instruction	Is there anything else to do	503	0.2	99.80
`InfoObjectLocAndOD`	Object/Location	knife is behind the sink	6946	99.4	0.6
`ReqForObjLocAndOD`	Object/Location	where is the mug?	2010	0.3	99.70
`InformationOther`	Object/Location	Mug is already clean	1148	88.76	11.24
`AlternateQuestions`	Object/Location	yellow or blue mug?	123	27.65	72.35
`Acknowledge`	Generic	perfect	7421	21.38	78.62
`Greetings`	Generic	hello	2565	44.01	55.9
`Confirm`	Generic	Should I clean the cup?	726	25.75	74.25
`MiscOther`	Generic	ta-da	607	52.22	47.78
`Affirm`	Generic	Yes	460	78.26	21.74
`Deny`	Generic	No	161	72.92	26.08
`FeedbackPositive`	Feedback	great job	2745	97.12	2.88
`FeedbackNegative`	Feedback	that is not correct	46	95.65	4.35
`OtherInterfaceComment`	Interface	Which button opens drawer	486	60.09	39.91
`NotifyFailure`	Interface	not able to do it	408	3.68	96.32

Table 1: Dialog act labels, total number of utterances and frequencies per speaker type in overall corpus.

formed in the environment to accomplish the household task.

- Object/Location related: Represents requests and information seeking utterances related to objects that need to be handled or manipulated for the specific *TEACh* task. Many of these are on the specifics of object location (where to find it, where to place it) and queries on disambiguation related to objects or their locations.
- Interface Related: Utterances related to *TEACh* data annotation itself (`NotifyFailure` and `OtherInterfaceComment`)
- Feedback related: Utterances used to provide feedback (both positive and negative) on navigation, object manipulation and in general task execution.

We hired expert annotators who are fluent in English to annotate utterances from the *TEACh* dataset with our dialog acts. Annotators were shown the complete dialog and asked to annotate each utterance with the most appropriate dialog act. When an utterance had multiple dialog acts applicable, annotators were asked to divide the utterance into spans and annotate each span with a single dialog act label. We observed that 7% of the utterances were segmented to have multiple dialog acts. To measure the quality of the annotations, on a small subset of 235 utterances (17 dialogs), we collected annotations from two annotators. On this subset, we observed a Cohen's kappa score of 0.87. We include an example *TEACh* session in Figure 1 for the task *Boil Potato* containing dialog act acctions for each utterance.

Similar to many task-oriented dialogs, we observe a strong correlation between the speaker role (*Commander* or *Follower*) and the dialog act of an utterance. For example, the majority of the inform utterances are from *Commander* i.e., where *Commander* gives instructions or informs object locations or other details on the task, whereas majority of the request utterances (instructions, object locations etc.) are from *Follower*. In Table 1, we present the set of dialog acts, definitions and their frequency distributed across *Commander* and *Follower* utterances. We observe that some communicative functions such as clarification of ambiguity are relatively infrequent in this dataset. We group together such rare functions into a single dialog act `MiscOther`.

4 Experiments

In this section, we explore how dialog acts can be used for various modeling tasks including predicting the agent's future behavior in the environment. We explore the following tasks (i) dialog act classification: predicting the dialog act of an utterance; (ii) future turn dialog act prediction given dialog history; (iii) given *TEACh* dialog history, predicting a plan for the task and (iv) given dialog history and the past actions in environment, predicting the entire sequence of low-level actions to be executed in the *TEACh* environment to complete the task (Execution from Dialog History (EDH) benchmark from Padmakumar et al. 2021). Note that *TEACh*

Utterance (Utt)	the bowl is in the microwave
Utt + ST	<<Commander>> the bowl is in the microwave
Utt + DH	how can i help <<TURN>> please serve 1 slice of tomato in a bowl <<TURN>> where can i find a bowl <<TURN>> the bowl is in the microwave
Utt + DH + DA-E	how can i help <<ReqForInstruction>> <<TURN>> please serve 1 slice of tomato in a bowl <<Instruction>> where can i find a bowl <<ReqForObjLocAndOD>> <<TURN>> the bowl is in the microwave <<InfoObjectLocAndOD>>
Utt + ST + DH + DA-E	<<Follower>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>> <<TURN>> <<Follower>> where can i find a bowl <<ReqForObjLocAndOD>> <<TURN>> <<Commander>> the bowl is in the microwave <<InfoObjectLocAndOD>>

Figure 2: Sample input to dialog act prediction or next turn dialog act prediction models showing incorporation of speaker and dialog history

	Valid seen	Valid unseen	Test seen	Test unseen
Utterance	85.59	83.74	85.88	83.59
+Speaker Tags (ST)	87.98	85.91	87.55	85.73
+ Dialog History (DH)	86.7	84.66	86.48	84.25
+ DH + DA-E	**88.6**	**86.32**	88.35	**86.09**
+ DH + ST + DA-E	88.35	86.15	**88.54**	85.89
Follower utterances only				
Utterance	83.12	79.58	84.86	83.85
+Speaker Tags (ST)	86.84	82.26	88.33	**87.71**
+Dialog History (DH)	86.52	84.13	86.67	84.53
+ DH +DA-E	**88.62**	**85.87**	88.82	86.56
+ DH + ST + DA-E	88.32	85.79	**89.22**	86.3
Commander utterances only				
Utterance	87.16	86.71	86.5	83.42
+ Speaker Tags (ST)	**88.70**	**88.52**	**87.08**	84.42
+ Dialog History (DH)	87.11	81.03	85.79	83.49
+ DH + DA-E	88.55	87.90	86.69	**84.84**
+ DH + ST + DA-E	88.42	87.4	86.15	84.79

Table 2: Dialog Act prediction accuracy scores for whole *TEACh-DA* dataset. We also report accuracy scores for *Follower* and *Commander* utterances separately.

has two validation and two test splits each - seen and unseen. These refer to visual differences between the environments in which gameplay sessions occurred. With the exception of the EDH experiment, since we only focus on language, we do not expect significant differences between the seen and unseen splits.

4.1 Dialog Act Classification

Dialog Act classification is the task of identifying the general intent of the user utterance in a dialog. While dialog act classification has been well explored in both task-oriented dialogs and open-domain dialogs, it is still an under explored problem in human-robot dialogs (Gervits et al., 2020). We study the *TEACh* dataset to predict the dialog act for a given utterance. We experimented with fine-tuning a large pre-trained language model *RoBERTa-base* for the classification of dialog acts[1]. We expect the speaker role (*Follower* or *Commander*) and the dialog context to be important for predicting the intent of an utterance. To test this, we predict dialog acts with different input formats (shown in Figure 2) ablating the value of speaker and context information (DH: all the previous utterances in the dialog, ST: speaker tags, DA-E: ground-truth dialog act tags of all the previous utterances in the dialog). We present our results in Table 2. Similar to prior studies on dialog act classification for task-oriented dialogs, we observe that both the speaker tags and dialog history help in predicting the correct dialog act for a given utterance, and the best performance is observed when both of them are used.

In *TEACh*, the distribution of dialog acts varies with the speaker role (*Commander* vs. *Follower*) as shown in Table 1. To understand the accuracy of the models on utterances of each speaker role, we also present results separated by speaker role in Table 2. We observed that both speaker tags and dialog history with previous turn dialog acts helped identifying dialog acts for *Follower* utterances. For *Commander* utterances both speaker tags and dialog history gave marginal improvements.

[1]We also experimented with *BERT-base* and *TOD-BERT* but observed *RoBERTa-base* performed consistently better

	Valid seen	Valid unseen	Test seen	Test unseen
DH	42.62	42.44	43.55	41.07
DH + ST	56.23	54.68	54.69	53.27
DH + DA-E	56.05	55.58	56.49	53.45
DH + ST + DA-E	**56.72**	**56.14**	**56.28**	**54.99**
Follower utterances only				
DH	30.73	28.64	31.41	29.06
DH +ST	51.67	49.3	54.11	52.34
DH + DA-E	50.19	50.28	54.72	52.24
DH + ST + DA-E	**52.17**	**50.35**	**54.72**	**53.44**
Commander utterances only				
DH	49.27	51.08	50.07	48.08
DH + ST	58.78	58.05	55.01	53.82
DH + DA-E	59.33	58.9	57.4	54.16
DH + ST + DA-E	**59.26**	**59.77**	**57.11**	**55.89**

Table 3: Predict next utterance Dialog Act given dialog history. We also report results when next utterance is *Commander* and *Follower* separately. Speaker Tags: Additional to current utterance speaker tag we also provide next utterance speaker information.

4.2 Next Dialog Act Prediction

In end-to-end dialog models, predicting the desired dialog act for the next turn is useful for response generation (Tanaka et al., 2019). Predicting the dialog act of the next response in *TEACh* will provide insights into a model's ability to provide appropriate dialog responses. This is particularly useful for *Follower* utterances to enable the agent to identify when to ask for more instructions or additional information to accomplish a sub-task. We modeled this as a classification task where we provide dialog history until a particular turn as input and predict the dialog act of the next turn. In addition to providing dialog history, we also tested this to see if providing next turn speaker information will improve the performance of the model. Similar to our dialog act classification model in Section 4.1 we fine-tuned a *RoBERTa-base* model for predicting the dialog act of the next utterance. In Table 3, we present results for next dialog act prediction. We observe a significant improvement in the performance for next dialog act prediction when the next utterance is from the *Follower* and the speaker information or previous utterances dialog act is added to the input. We hypothesize that the accuracy in this task is low compared to similar tasks in other task-oriented dialog datasets because this dataset does not enforce turn taking. The *Commander* or *Follower* may break up a single intent into multiple utterances and one may anticipate the next response from the other before it is asked. For example, if the *Commander* has asked the *Follower* to slice a tomato, the *Commander* may expect that the *Follower* is likely to then ask for the locations of the tomato or the knife and may start providing this information before the *Follower* has asked for it. Further, the *Commander* or *Follower* may have responded directly to visual cues or actions taken by the other in the environment. Hence, visual or environment information is likely also important for predicting future dialog acts.

4.3 Plan Prediction

In robotics, task planning is the process of generating a sequence of symbolic actions to guide high-level behavior of a robot to complete a task (Ghallab et al., 2016). In this experiment, we consider a simple plan representation where a task plan consists of a sequence of object manipulations that need to be completed in order for the task to be successful. An example is included in Figure 3 When executing such a plan, the robot will need to navigate to required objects and additional steps may be required based on the state of the environment (for example if the microwave is too full, the robot may need to partially clear it first).

However, it should be possible to generate the plan for a task based on the dialog alone. We explore two settings for this

- `Game-to-Plan`: Given the entire dialog from a gameplay session, predict the plan - that is, all object interaction actions taken during that gameplay session.
- `Dialog-History-to-Plan`: Given a portion of dialog history from a gameplay session, predict the object interaction actions that need to occur until the next dialog utterance.

The `Game-to-Plan` setting is more likely to be useful for post-hoc analysis of such situated interactions after they have occurred, whereas the `Dialog-History-to-Plan` setting can be used to build an embodied agent that engages in dialog with a user and executes actions in a virtual environment based on information obtained in the dialog. At any point in time, such an agent would predict the next few object interactions to be accomplished given the dialog history so far, complete

Language Input:	
DH	how can i help <<TURN>> please serve 1 slice of tomato in a bowl <<TURN>> where can i find a bowl <<TURN>> the bowl is in the microwave
DH + DA	<<Follower>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>> <<TURN>> <<Follower>> where can i find a bowl <<TURN>> <<Commander>> the bowl is in the microwave
DH + DA + Filter	<<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>>
Language Output:	
Pickup Tomato -- Place CounterTop -- Pickup ButterKnife -- Slice Tomato -- Place CounterTop – Pickup TomatoSliced -- ToggleOff Microwave -- Open Microwave -- Place Bowl -- Pickup Bowl	

Figure 3: Sample input and output for plan prediction showing incorporation of speaker and dialog act information.

	Game-to-Plan											
	Percentage of valid plans				Plan tuple precision				Plan tuple recall			
	Valid seen	Valid unseen	Test seen	Test unseen	Valid seen	Valid unseen	Test seen	Test unseen	Valid seen	Valid unseen	Test seen	Test unseen
DH	24.31	**30.39**	**28.18**	28.69	72.67	**73.93**	73.48	**78.53**	37.06	**34.35**	37.46	**36.00**
+ DA	25.97	23.86	19.89	26.83	**75.29**	73.0	**74.81**	77.52	**38.18**	33.7	**39.28**	35.31
+ Filter	**37.57**	29.41	27.62	**32.94**	71.29	70.94	69.80	75.45	34.33	31.61	35.45	33.42
	Dialog-History-to-Plan											
DH	**23.76**	23.69	25.41	24.45	72.97	73.47	**75.65**	**78.64**	36.38	34.06	**39.11**	**36.53**
+ DA	24.31	**30.39**	**28.18**	**28.69**	72.67	**73.93**	73.48	78.53	**37.06**	**34.35**	37.46	36.0
+ Filter	26.52	23.69	25.41	28.01	**73.66**	69.88	71.67	74.33	36.08	31.29	35.83	33.12

Table 4: Plan prediction results. Using dialog act information helps increase the fraction of valid generated plans but not as much with plan precision or recall.

them and then use another module that makes use of subsequent dialog act prediction (section 4.2) to engage in further dialog with the user.

We model plan prediction as a sequence to sequence task where the input consists of the dialog / dialog history, and the output as a sequence of alternating object interaction actions (eg: `Pickup`, `Place`, `ToggleOn`) and object types (eg: `Mug`, `Sink`). We experiment with augmenting the dialog history with dialog act information (+ DA information) and filtering the input dialog to only contain utterance segments annotated as being of type `Instruction` (+ filter) We fine-tune a `BART-base` model for this task and evaluate different experimental conditions on the following metrics:

- Fraction of valid plans: Fraction of generated output sequences that consist of alternating valid actions and object types. (For example `(Pickup, Mug), (Place, Sink) (ToggleOn, Faucet)` is a valid sequence while `(Pickup, Mug) (Sink) (ToggleOn, Faucet)` and `(Pickup, Mug) (Place) (ToggleOn, Faucet))` are not due to the missing action for `Sink` and the missing object for `Place` respectively.
- Precision of (action, object) tuples: We identify a valid object type followed by a valid action as an (action, object) tuple and precision is the fraction of such tuples in the generated output present in the ground truth plan.
- Recall of (action, object) tuples: Recall is the fraction of (action, object) tuples in the ground truth plan present in the generated output.

The results are included in Table 4. We notice that addition of dialog act information and filtering to relevant dialog acts improves performance in some splits but not others. More improvements are seen in the `Dialog-History-to-Plan`

DH	how can i help <<TURN>> please serve 1 slice of tomato in a bowl
DH + ST	<<Follower>> how can i help <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl
DH + ST + DA-E	<<Follower>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>>
DH + DA-E	how can i help <<ReqForInstruction>> <<TURN>> please serve 1 slice of tomato in a bowl <<Instruction>>
DH + ST + DA-SE	<<Follower>> <<ReqForInstruction>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> <<Instruction>> please serve 1 slice of tomato in a bowl <<Instruction>>

Figure 4: Language Input Variants for EDH.

	EDH Validation				**EDH Test**			
	Seen		*Unseen*		*Seen*		*Unseen*	
Language Input	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]
DH	7.9 [1.0]	7.1 [3.3]	6.7 [0.4]	3.9 [1.5]	10.5 [0.5]	7.9 [3.2]	7.5 [0.7]	5.6 [1.9]
+ ST	6.7 [0.5]	7.4 [2.8]	6.7 [0.8]	4.0 [1.5]	9.8 [0.9]	8.3 [2.9]	7.1 [0.8]	6.6 [1.7]
+ DA-E	8.5 [0.6]	**8.2 [3.3]**	6.7 [0.5]	**5.0 [1.9]**	**12.2 [1.2]**	8.6 [3.7]	7.4 [0.8]	6.1 [2.3]
+ DA-SE	7.8 [1.8]	6.4 [4.0]	7.2 [0.6]	4.6 [1.6]	11.0 [0.7]	**10.1 [4.3]**	**7.7 [0.8]**	6.2 [1.8]
+ ST + DA-SE	**8.7 [1.0]**	7.3 [2.6]	**7.5 [0.8]**	4.4 [1.8]	9.9 [0.7]	8.0 [2.9]	7.0 [0.7]	**7.2 [2.2]**

Table 5: We experiment whether addition of speaker or dialog act information improves performance of the Episodic Transformer (E.T.) model on the Execution from Dialog History (EDH) task. In most cases, speaker information is not found to be beneficial but adding dialog acts at the end or start and end of an utterance is seen to provide small improvements in performance.

setting compared to the `Game-to-Plan` setting. We hypothesize that this is because the model is able to automatically identify the dialog act from the utterance text and hence does not need it to be explicitly specified.

4.4 Execution from Dialog History

The Execution from Dialog History (EDH) task defined in the Padmakumar et al. 2021 is an extension of the above task. Instead of simply predicting important object interactions, given dialog history and past actions in the environment, a model is expected to predict a full sequence of low level actions to accomplish the task described in the dialog. Action sequences predicted by the model are executed in the virtual environment and models are evaluated based on how many required object state changes are accomplished. The metrics used for this task include the fraction of successful state changes (goal condition success rate or GC), the fraction of sessions for which all state changes were accomplished (success rate or SR) and Trajectory Length Weighted versions of these metrics that multiply the metrics with the ratio of the ground truth path length to the predicted path length - where a lower value of the trajectory weighted metric suggests that the model used longer sequences of actions to accomplish the same state changes.

We borrow the Episodic Transformer (E.T.) model proposed in Padmakumar et al. 2021 and vary the language input (with a baseline of just the dialog history (DH)) by adding speaker tags (+ST) and ground-truth dialog act tags at the start (+DA-S), end (+DA-E) or both (+DA-SE). We present the results for selected set of experiments in Table 5. We observe small performance improvements on success rate of up to 2 points when the language input is marked up with dialog acts, either at the end or start and end of an utterance, but less benefit is observed from speaker information. We believe that stronger improvements will likely be observed when using a more modular approach (eg: (Min et al., 2021)) where it is easier to decouple the effects of errors arising from language understanding from those arising from navigation which is the most difficult component when predicting such

low-level actions (Blukis et al., 2022; Jia et al., 2022; Min et al., 2021).

5 Conclusion

We propose a new dialog act annotation framework for embodied task completion dialogs and use this to annotate the *TEACh* dataset - a dataset of over 3,000 unconstrained, situated human-human dialogs. We evaluate baseline models for predicting dialog acts of utterances, demonstrate that predicting future dialog acts from past ones is much more difficult in dialog datasets that are not constrained by turn taking. Towards guiding agent actions in the environment beyond dialog, we show explore the benefit of dialog acts in the generation of plans, and improve end-to-end performance in the *TEACh* Execution from Dialog History task.

6 Future Work

Unlike the majority of dialog datasets, situated or otherwise, utterances in the *TEACh* dataset are not constrained by a pre-designed dialog act schema or by turn taking. We observe that this makes it much more difficult than expected to predict subsequent dialog acts given past ones - the predictability of which has been typically used to design dialog simulators (Schatzmann and Young, 2009; Keizer et al., 2010). We believe that annotation of this large and more natural dataset will aid in the development of more realistic dialog simulators, which can in turn result in the development of more natural dialog agents. Further, in *TEACh*, visual cues or actions taken by the agent in the environment might play an important role for predicting future dialog acts. This would be an interesting direction to explore for future. Finally, we hypothesize that there is considerable scope in using such annotated dialog acts to develop modular models for embodied task completion that involve better language understanding, and to generate realistic situated dialogs for data augmentation.

References

Cristian-Paul Bara, CH-Wang Sky, and Joyce Chai. 2021. Mindcraft: Theory of mind modeling for situated dialogue in collaborative tasks. In *Proceedings of EMNLP 2021*, pages 1112–1125.

Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. 2022. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.

Harry Bunt, Dirk K. J. Heylen, Catherine Pelachaud, Roberta Catizone, and David R. Traum. 2009. The dit++ taxanomy for functional dialogue markup. In *EDAML@AAMAS, Workshop Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 13–24.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. 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 4516–4525.

Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-Tür. 2020. Just ask: An interactive learning framework for vision and language navigation. In *AAAI 2020*, pages 2459–2466. AAAI Press.

Mark G. Core and James F. Allen. 1997. Coding dialogs with the damsl annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128.

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 SIGDIAL 2021*, pages 353–359.

Felix Gervits, Ravenna Thielstrom, Antonio Roque, and Matthias Scheutz. 2020. It's about time: Turn-entry timing for situated human-robot dialogue. In *Proceedings of the SIGDIAL 2020*, pages 86–96.

Malik Ghallab, Dana Nau, and Paolo Traverso. 2016. *Automated planning and acting*. Cambridge University Press.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Zhiwei Jia, Kaixiang Lin, Yizhou Zhao, Qiaozi Gao, Govind Thattai, and Gaurav Sukhatme. 2022. Learning to act with affordance-aware multimodal neural slam. *arXiv preprint arXiv:2201.09862*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126.

Simon Keizer, Milica Gasic, Filip Jurcicek, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Parameter estimation for agenda-based user simulation. In *Proceedings of the SIGDIAL 2010 Conference*, pages 116–123.

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 EMNLP*, pages 4903–4912.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *NAACL HLT*, pages 515–520.

Stefano Mezza, Alessandra Cervone, Evgeny A. Stepanov, Giuliano Tortoreto, and Giuseppe Riccardi. 2018. Iso-standard domain-independent dialogue act tagging for conversational agents. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3539–3551.

So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. 2021. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2021. Teach: Task-driven embodied agents that chat. *arXiv preprint arXiv:2110.00534*.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952.

Shachi Paul, Rahul Goel, and Dilek Hakkani-Tür. 2019. Towards universal dialogue act tagging for task-oriented dialogues. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, pages 1453–1457. ISCA.

Eugénio Ribeiro, Ricardo Ribeiro, and David Martins de Matos. 2019. Deep dialog act recognition using multiple token, segment, and context information representations. *J. Artif. Intell. Res.*, 66:861–899.

Amrita Saha, Mitesh Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jost Schatzmann and Steve Young. 2009. The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of NAACL-HLT*, pages 3795–3805.

Ayush Shrivastava, Karthik Gopalakrishnan, Yang Liu, Robinson Piramuthu, Gökhan Tür, Devi Parikh, and Dilek Hakkani-Tür. 2021. VISITRON: visual semantics-aligned interactively trained object-navigator. *CoRR*, abs/2105.11589.

Koji Tanaka, Junya Takayama, and Yuki Arase. 2019. Dialogue-act prediction of future responses based on conversation history. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 197–202.

Jason D Williams, Matthew Henderson, Antoine Raux, Blaise Thomson, Alan Black, and Deepak Ramachandran. 2014. The dialog state tracking challenge series. *AI Magazine*, 35(4):121–124.

Steve Young. 2007. Cued standard dialogue acts. *Report, Cambridge University Engineering Department*, 2007.

A Further Experiment Details

A.1 Dialog Act Classification and Next Turn Dialog Act Prediction

Both for dialog act classification and next turn dialog act prediction models, we finetune a `RoBERTa-base` model for multiclass classification with 18 classes (our target number of dialog acts). For all the experiments were run using Huggingface library and the publicly available pre-trained models. Additional to the utterance we provide dialog-context and speaker information (mentioned as dialog history (DH) and Speaker Info (SI)) and train the classifiers for a maximum sequence length of 512 tokens. When the input exceeds 512 tokens we truncate from left i.e., we keep the most recent context. We use a batch size of 16 per GPU and accumulate gradients across 4 GPU instances. We use a learning rate of $2e-05$ and train for 5 epochs.

A.2 Plan Prediction

For the plan prediction task, we finetune a `bart-base` model, treating the problem as sequence to sequence prediction. A sample input and output from the `Game-to-Plan` version of the task are included below:

Sample Input:

```
what do I do? <<TURN>> making
coffee <<TURN>> grab a mug
<<TURN>> where is tyhe mug?
<<TURN>> on the counter next to
you <<TURN>> empty, and wash
<<TURN>> should I wash the mug
<<TURN>> place in coffee maker
after cleaning <<TURN>> yes
<<TURN>> okay <<TURN>> turn on
water <<TURN>> turn off <<TURN>>
place in coffee maker next to
sink <<TURN>> empty first
<<TURN>> turn on <<TURN>> great
job....we're done... <<TURN>>
```

Sample Output:

```
Pickup Mug Pour SinkBasin Place
SinkBasin ToggleOn Faucet
ToggleOff Faucet Pickup Mug Pour
SinkBasin Place CoffeeMachine
ToggleOn CoffeeMachine
```

Note that we do not include any punctuation in the output sequence to demarcate (action, object) tuples and instead post process the generated sequence deleting any action not followed by an object or object not preceded by an action for evaluation. Also, while we use $\langle\langle$`TURN`$\rangle\rangle$ in the above example to demarcate turns, in actual implementation, the default BART separator token is used.

All experiments are run using the HuggingFace library and pretrained models [2]. We use a batch size of 2 per GPU accumulating gradients from batches on 4 GPUs of an AWS 'p3.8xlarge' instance leading to an effective batch size of 8. Training was done for 20 epochs. We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 1e-08$ and weight decay of 0.01. We use a learning rate of $5e-05$ with a linear warmup over 500 steps. Where necessary, we right-truncate the input to the model's limit of 1024 tokens as we believe that when an incomplete conversation must be used, the model may be able to infer most of the necessary steps from the task information which is likely to be indicated by the first few utterances of the conversation.

The primary hyperparameter tuning we experimented with involved the position at which the dialog act was inserted relative to the utterance, which was one of

- START_OF_SEGMENT - Start of the utterance segment
- END_OF_SEGMENT - End of the utterance segment
- START_END_SEGMENT - Start and end of the utterance segment

and the format used to insert dialog act information, which was one of

- NO_CHANGE_TEXT - The name of the dialog act is inserted in Camel case as a part of the input text to the model.
- FILTER - Retain only utterances marked with the dialog act INSTRUCTION. Additionally, the name of the dialog act is inserted in Camel case as a part of the input text to the model.
- TAGS_IN_TEXT - The name of the dialog act in Camel case is surrounded by $\langle\langle\rangle\rangle$.
- TAGS_SPL_TOKENS - The name of the dialog act in Camel case is surrounded by $\langle\langle\rangle\rangle$ and this is specified as being a special token so that it does not get split by the tokenizer.
- SPLIT_WORDS_TEXT - The name of the dialog act is split into individual words (for example, REQUESTFORINSTRUCTION becomes "request for instruction") and these are inserted into the text.

We also tuned whether speaker information was passed to the model. None of the format, position or speaker tag choices were found to consistently outperform the other.

For the DH rows in table 4, neither the position, nor the format of dialog acts is relevant as no dialog act information is used. We also do not filter utterances. The best +DA row in the `Game-to-Plan` setting used dialog acts in format SPLIT_WORDS_TEXT in position END_OF_SEGMENT with speaker tags. The best +Filter row in the `Game-to-Plan` setting used dialog acts in format START_END_SEGMENT

[2] https://huggingface.co/

without speaker tags. The best +DA row in the `Dialog-History-to-Plan` setting used dialog acts in format SPLIT_WORDS_TEXT in position START_OF_SEGMENT without speaker tags. The best +Filter row in the `Dialog-History-to-Plan` setting used dialog acts in format END_OF_SEGMENT without speaker tags.

A.3 Execution from Dialog History

We adapt the Episodic Transformer (E.T.) model first introduced in (Pashevich et al., 2021) and used for baseline experiments in (Padmakumar et al., 2021) on the TEACh dataset. We keep all training parameters constant from (Padmakumar et al., 2021) and primarily experiment with the input format as described in the main paper. Unlike our previous experiments, since the language encoder of the E.T. model is trained from scratch using only the vocabulary present in the training data, we insert dialog acts and speaker indicators as individual tokens in the input that will be treated identically to other text tokens.

B Dialog Acts

In Table 6 we add further examples for each dialog act (for both *Follower*and *Commander*) from different *TEACh* tasks to demonstrate the difference in type of utterances we observe in the dataset.

Dialog Act	Task	Agent: Example
`Instruction`	Water Plant	*Commander*: The plant by the sink needs to be watered
	Plate Of Toast	*Commander*: please slice bread and toast 1 slice
	Plate Of Toast	*Commander*: lets make a slice of toast
`InfoObjectLocAndOD`	Plate Of Toast	*Commander*: knife is in the fridge
	Plate Of Toast	*Commander*: the clean plate is on the white table
	Clean All X	*Commander*: right cabinet under the sink
`Acknowledge`	Make Coffee	*Commander*: we are done!
	Clean All X	*Follower*: Plate is clean
	N Slices Of X In Y	*Follower*: found it
`ReqForInstruction`	Put All X On Y	*Follower*: how can I help
	Put All X On Y	*Follower*: what are my directions
	Plate Of Toast	*Follower*: what is my task today
`FeedbackPositive`	Plate Of Toast	*Commander*: good job
	Put All X In One Y	*Commander*: that's it good job
	Water Plant	*Commander*: thank you its seems to be done
`Greetings`	Make Coffee	*Commander*: Hi how are you today?
	Water Plant	*Follower*: Good day
	Boil X	*Commander*: Good morning
`ReqForObjLocAndOD`	Clean All X	*Follower*: where is the dirty cookware?
	Plate Of Toast	*Follower*: Can you help me find knife?
	Put All X In One Y	*Follower*: where is the third one?
`InformationOther`	Make Coffee	*Commander*: Don't take martini glass
	Boil X	*Commander*: You keep walking past them
	Boil X	*Commander*: That looks cooked already
`Confirm`	Put All X In One Y	*Follower*: was that everything
	Salad	*Commander*: you can see the toaster right?
	N Slices of X in Y	*Follower*: Shall I turn off the water?
`RequestOtherInfo`	Breakfast	*Follower*: how many slices of each?
	Clean All X	*Follower*: what pieces?
	Plate Of Toast	*Follower*: shall i take it to the toaster now
`MiscOther`	Sandwich	*Commander*: One sec
	Salad	*Commander*: Common!!
	Breakfast	*Commander*: Thant's my bad...Sorry
`RequestMore`	N Cooked Slices Of X In Y	*Follower*: Is there anything more I can help with?
	Salad	*Follower*: what else would you like me to do
	Clean All X	*Follower*: Any more tasks?
`OtherInterfaceComment`	Plate of Toast	*Follower*: Finish and report a bug?
	Clean All X	*Follower*: refresh the page
	Put All X On Y	*Follower*: connection is slow
`Affirm`	Water Plant	*Commander*: yes, you can use the green cup
	Breakfast	*Commander*: yes, toast the bread
	Put All X On Y	*Commander*: yes please
`NotifyFailure`	Make Coffee	*Follower*: It's not turning on the coffee.
	N Slices Of X In Y	*Follower*: tomato won't fit in those
	Sandwich	*Follower*: can't seem to grab the knife in cabinet
`Deny`	Make Breakfast	*Commander*: No don't toast the bread
	Salad	*Commander*: don't
	Plate of Toast	*Commander*: don't think so
`AlternateQuestions`	N Cooked Slices Of X In Y	*Follower*: Do I boil it or slice it?
	Clean All X	*Follower*: To the left or right of the stove?
	Make Coffee	*Follower*: This mug or the other one?
`FeedbackNegative`	Make Coffee	*Commander*: you don't have the correct mug
	N cooked Slices of X in Y	*Commander*: task not complete
	Plate of Toast	*Commander*: wrong plate

Table 6: Example utterances for Dialog act labels that could be observed in different *TEACh* tasks from *Commander*and *Follower*.