
DARD: A Multi-Agent Approach for Task-Oriented Dialog Systems

Aman Gupta*
Carnegie Mellon University
amangupt@cs.cmu.edu

Anirudh Ravichandran
Amazon
aniravic@amazon.com

Ziji Zhang
Amazon
czhangzi@amazon.com

Swair Shah
Amazon
shahswai@amazon.com

Anurag Beniwal
Amazon
beanurag@amazon.com

Narayanan Sadagopan
Amazon
sdgpn@amazon.com

Abstract

Task-oriented dialogue systems are essential for applications ranging from customer service to personal assistants and are widely used across various industries. However, developing effective multi-domain systems remains a significant challenge due to the complexity of handling diverse user intents, entity types, and domain-specific knowledge across several domains. In this work, we propose DARD (Domain Assigned Response Delegation), a multi-agent conversational system capable of successfully handling multi-domain dialogs. DARD leverages domain-specific agents, orchestrated by a central dialog manager agent. Our extensive experiments compare and utilize various agent modeling approaches, combining the strengths of smaller fine-tuned models (Flan-T5-large & Mistral-7B) with their larger counterparts, Large Language Models (LLMs) (Claude Sonnet 3.0). We provide insights into the strengths and limitations of each approach, highlighting the benefits of our multi-agent framework in terms of flexibility and composability. We evaluate DARD using the well-established MultiWOZ benchmark, achieving state-of-the-art performance by improving the dialogue inform rate by 6.6% and the success rate by 4.1% over the best-performing existing approaches. Additionally, we discuss various annotator discrepancies and issues within the MultiWOZ dataset and its evaluation system. The source code can be found at the link²

1 Introduction

In recent research, significant efforts have been made to build systems that involve planning and communication between various specialized agents to perform complex tasks [1, 2, 3]. These agents are, in turn, backed by instruction-tuned open-source LLMs, external APIs, or other simpler tools. Various tasks such as logical reasoning [4, 5], societal simulations [6, 7], software development [8] have seen remarkable improvement in performance using these multi-agent framework methods.

*Work performed during an internship at Amazon

²<https://anonymous.4open.science/r/DARD-A-Multi-Agent-Approach-for-Task-Oriented-Dialog-Systems-8470/>

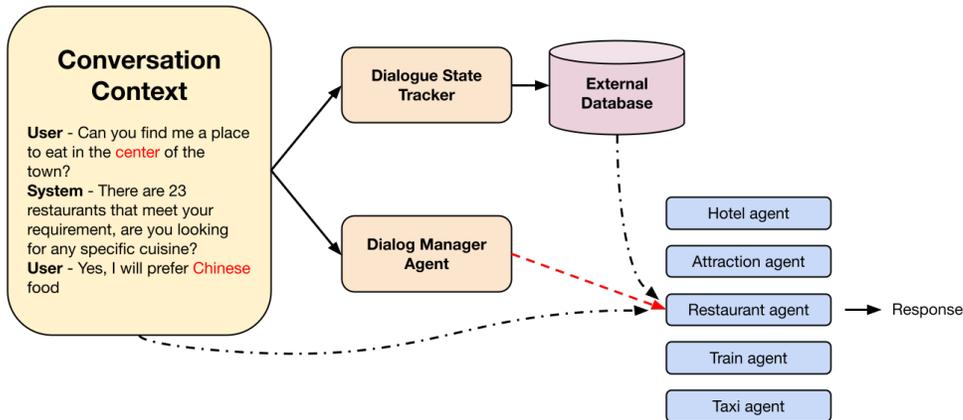


Figure 1: Overview of *DARD* for end-to-end response generation on MultiWOZ. The current diagram shows a conversation assigned to the restaurant agent but in general, it can be assigned to any of the domain agents

In this work, we explore the potential of agentic design in Task-Oriented Dialogue Systems (TODS). TODS are prevalent in real-world applications, such as customer service, e-commerce, and commercial voice assistants like Amazon Alexa and Google Assistant. Multiple research efforts have been made to curate high-quality labeled datasets to aid in developing systems that can handle end-to-end task-oriented dialogs [9, 10, 11]. One of the most well-established and widely used TOD datasets amongst them is the MultiWOZ dataset [9]. The dataset contains 10k+ single and multi-domain conversations spanning 7 domains of attraction, hospital, hotel, restaurant, taxi, train, and police. After the original release, multiple corrected versions (2.1 - [12], 2.2 - [13], 2.3 - [14], 2.4 - [15]) of the MultiWOZ dataset have been published, each addressing specific issues. We primarily experiment with the MultiWOZ 2.2 [13] version, as it had the most established benchmark [16] and is the latest recognized version as per the official repository³. The main tasks performed on the MultiWOZ dataset are Dialogue State Tracking (DST), which involves tracking predefined slots and their values in the context, and Response Generation, which involves predicting the system’s response to the latest user message.

We propose a **Domain Assigned Response Delegation** (*DARD*), a framework that involves multiple specialized domain agents invoked by a central dialog manager agent, based on the dialog context. These domain-specific agents use conversational context and relevant entities from an external database to generate a response for the latest user message (see Figure 1). We experiment with fine-tuned Flan-T5-Large [17], Mistral-7B [18], and prompted Claude Sonnet 3.0⁴ models as our domain agents and also a prompted Claude Sonnet 3.0 as a dialog manager agent. We present further details of our experiments in the section 2. We further systematically compare the performance of our approaches with other top-performing approaches on the benchmark in the section 3. To rigorously validate our method and its performance, we conduct comprehensive error analyses and also present the challenges and limitations of the MultiWOZ dataset and its evaluation system in section 4. The following are the key contributions and insights we present through our work:

- We introduce *DARD* (Domain-Assigned Response Delegation), an ensemble of domain-specific agents that improve the state-of-the-art dialog inform rate by 6.6% and success rate by 4.1% on the MultiWOZ benchmark.
- Our study presents a detailed comparison of performance between fine-tuned (Mistral-7B, Flan-T5-Large) vs Prompted (Claude Sonnet 3.0) models in the context of dialog agents and single-agent vs multi-agent approaches for task-oriented dialogs

³<https://github.com/budzianowski/multiwoz>

⁴<https://www.anthropic.com/news/claude-3-family>

- Our analysis reveals multiple annotator discrepancies present in the MultiWOZ dataset, impacting the performance of models for DST and response generation tasks.

2 Methodology

2.1 MultiWOZ 2.2 Dataset

The MultiWOZ 2.2 [13] dataset is an improved version of MultiWOZ 2.1 [12] correcting DST annotations of 17.3% of utterances and ontology issues associated with some of the slot values. The dataset contains 10,437 conversations divided between train (8,437), validation (1,000), and test (1,000) sets. Each conversation contains alternating turns between the user and the system utterances. Although the dataset comprises 7 domains, the domains of hospital and police are only present in the training data. Following the work of [19, 20], we remove the conversation of these domains from the training set. The domains of restaurant, hotel, taxi, and train are bookable as the users can ask the system about booking these for them. The dataset also contains an external database for each domain, which contains a list of entries and their attributes. The system responses are supposed to suggest and offer a reservation for only these entries. Ground truth data comprises detailed annotations of dialog states, system responses, and conversation goals, which are used to compare and evaluate the model-predicted responses.

2.2 Scoring Metrics

We follow the standard and the widely used metrics for the tasks of DST and Response Generation presented in the work⁵ [16].

DST The DST performance is measured by Joint State Accuracy (JSA) between the predicted and the ground-truth states. Each slot consists of a triplet of (domain, slot key, and slot value), and multiple such slots can be present in a given dialog history. Some of these slots like `hotel-stars` are categorical in nature so can take only a fixed set of values while others like `hotel-name` can take any value based on conversation history. The binary JSA values are computed at each user turn of the conversation and involve comparison between all predicted states and the ground truth states. The domain name and slot key must match exactly, but a fuzzy margin is applied when comparing slot values to ignore minor syntactic differences.

End-to-end Response Generation The overall TODS pipeline, including DST and response generation, is evaluated using Inform, Success, and BLEU scores. The **Inform** rate is a conversation-level binary variable that indicates whether the user is presented with the correct venues according to their constraints. For example, for the conversation in Figure 1, if the system provides the user with a restaurant located in the center of town that serves Chinese food, it will be considered a correctly informed conversation. A multi-domain conversation will be considered correctly informed if the right venues are presented in each domain. Furthermore, a conversation has a **Success** rate of one if, firstly, the conversation has an inform rate of one (the user was provided the right venue) and secondly, the user was presented with the right attributes about the venue. For instance, in the above example, if the user was provided with the correct restaurant and later during the conversation, the user asks about the restaurant’s phone number, postcode, etc., the conversation would be considered successful if this information was presented to the user and unsuccessful otherwise. Like the inform rate, success metric is also binary and is computed at a conversation level. To evaluate the quality of the generated text, **BLEU scores** are calculated between delexicalized predicted responses and ground truth system responses. This delexicalization process ensures that model predictions are not unfairly penalized when they provide information about a suitable venue that differs from the specific venue chosen in the ground truth response, as multiple venues may satisfy user constraints. The following definition of the combined score is used to compare approaches in the MultiWOZ benchmark.

$$\text{Combined Score} = \frac{(\text{Inform} + \text{Success})}{2} + \text{BLEU Score}$$

Additionally, some metrics to compare textual richness are also compared.

⁵https://github.com/Tomiinek/MultiWOZ_Evaluation

2.3 Approach

2.3.1 DST Prediction

We start by building a DST pipeline for MultiWOZ, which takes the dialog history as input and outputs the slots present in the context. We run all of our experiments with Flan-T5-large, Mistral-7B, and Claude Sonnet 3.0 as our base models, as they represent a comprehensive range of model sizes, spanning from sub-1B parameters to mid-sized 7B models and larger LLMs, allowing us to assess performance across varying scales.

Single fine-tuned Agent This approach fine-tunes a language model to generate the list of slots given the input. We used the models Mistral-7B and Flan-T5-large for this approach. For model training, we treat each segment of the dialogue history up to each user utterance as a separate example, instead of updating a running dialogue state with each new utterance. This approach offers two key advantages: i) It prevents the accumulation of errors that could occur with subsequent dialogs ii) It better handles scenarios present in MultiWOZ datasets where previous dialogue slots are cleared, a situation where continuously updating dialogue states would fail.

Prompted LLM In this approach, we presented the Claude Sonnet 3.0 model with detailed instructions for the DST task. These instructions include a list of possible slots to track, all possible values categorical slots can take, explicit output format, and some in-context examples from which the models can learn. We present Claude with 50 random in-context examples selected at random from the training set.

Multi-Agent DARD In this approach, we fine-tune domain-specific DST models. We train a separate distinct model for each domain, focusing exclusively on tracking the slots relevant to that domain. To prepare the training data, we segregate the slots by domain; for instance, if a training sample contains slots from both the attraction and train domains, we add that sample to the training data of both domain models, with outputs consisting of slots from the respective domain only. During the testing phase, we first pass the context to a dialog manager agent, which outputs all domains whose slots are present. We then invoke the respective domain models to obtain the final set of slots. We use a prompted Claude 3.0 Sonnet LLM as a dialog manager agent to inform us about the domains present in the dialog context.

2.3.2 Response Generation Pipeline

The response generation pipeline involves predicting the delexicalized system utterance given the conversation context up to the previous user utterance. Similar to the DST pipeline, we choose to experiment with a single-agent approach and a domain-specific multi-agent approach.

Single fine-tuned Agent This approach fine-tunes a single model to handle conversations across all domains, providing it with both the dialog context and details of venues meeting user criteria. Appendix A.1 presents an example of the input provided to the model. These venue details, obtained by querying the database using predictions from the best DST pipeline, include the number of matching venues and specifics of one of them, if any. Adding these venue details allows the model to tailor responses based on the number of available options, whether suggesting a single match, asking for more preferences when multiple venues fit, or informing the user when no matches are found.

Multi-Agent DARD For this approach, we again use models that are specialized to respond to queries from specific domains. We experimented with fine-tuned models (Mistral-7B & Flan-T5-Large) and instructed Claude Sonnet 3.0 as our domain agents. The Claude-based agent is prompted with detailed information about the list of possible delexicalized tags to be used, generic instructions on how to respond, and some in-context examples. Appendix A.5 presents the prompt used for restaurant agents. The generic instructions on how to respond were designed using manual observation of training data from each domain. We use the Sonnet 3.0 dialog manager agent to determine which domain agent will be best equipped to respond based on the conversation context. We then pass the conversation context and venue details to the delegated agent to generate the system response. Since we can choose any type of agent for each domain, we also experiment with and select the best-performing agent from among Claude Sonnet 3.0, Flan-T5-Large, and Mistral-7B agents based on their combined scores on the validation set.

Method	Joint Slot Accuracy
Existing Works - Fine Tuned	
SDP-DST [19]	57.3
TOATOD [21]	63.79
D3ST [24]	57.8
DAIR [25]	59.9
LUAS _R [22]	65.4
LUAS _{R+G} [22]	66.3
<hr/>	
LLaMa-7B [23]	55.37
LDST [23]	60.65
<hr/>	
Fine Tuned Flan-T5-Large	
Single Agent	58.9
Multi-Agent DARD	63.6
<hr/>	
Claude Sonnet 3.0	
Instruction Prompted	45.8
<hr/>	
Fine Tuned Mistral-7B	
Single Agent	66.0
Multi-Agent DARD	63.1

Table 1: In this table, we compare our DST approach with the top-performing existing models on the benchmark. * - Numbers for these models are taken from the work of [23]

3 Results

Table 1 compares the results of our DST experiments with the existing best-performers [21, 22, 23]. We divide the existing work primarily into those that use fine-tuned models and those that use prompting LLM methods. We observe that for the fine-tuned Flan-T5-Large model, the performance is much better with domain-specific agents than with a single agent. However, for a fine-tuned Mistral-7B, the performance is nearly the same with the two approaches, and the single model performs a little better. We notice that Claude’s performance was poorer than that of the fine-tuned models. Our approach of using a single fine-tuned Mistral-7B model performs better than most existing works, second only to the method followed by [22]. [22] fine-tune a LLaMA-2 7B model on the DST dataset, first on the original training data itself (LUAS_R), which leads to similar performance as that of fine-tuned Mistral-7B, and then on both original training data and augmentations (LUAS_{R+G}), producing marginally better performance than a fine-tuned Mistral-7B models. The prompt-based approaches described in [23] performed better than our prompting-based Claude Sonnet method. We believe this stark difference in performance is due to the fact that the work in [23] employs a single-slot return approach, where the model predicts each slot individually for each conversation context, resulting in 30 calls in total for each sample’s complete prediction. This approach simplifies the task for the LLM, as it only needs to check for one slot in the context at a time, unlike our method, which requires consideration of all possible slots. However, the single-query method requires roughly 30 times more LLM calls.

Table 2 shows the evaluation results of the end-to-end response generation pipeline of the methods that we tried and also some of the best-performing models on the benchmark. We present the evaluation metrics of Inform, Success, and BLEU scores, along with the overall comparison of combined scores. As can be seen from the table, the inform and success rates of Claude-based agents are much better than those of existing works and fine-tuned agents; however, the opposite is true for the BLEU scores. Since the multi-agent framework allows the composability of using different agents for different domains, we select the best-performing agents based on domain-wise performance on the validation set. This combination achieves better information, success rate, and combined score than the existing works. In terms of textual richness measured by Conditional Bigram Entropy (CBE), #unique words, and #unique 3 grams, we observe that Claude-based agents are much more lexically diverse than fine-tuned models and other works. We present a detailed discussion of the results in the next section.

Method	BLEU	Inform	Success	Combined Score	CBE	# words	# 3-grams
Existing Works							
GALAXY [26]	19.64	85.4	75.7	100.2	1.75	295	2275
TOATOD [21]	17.04	90.0	79.8	101.9	-	-	-
RewardNet [27]	17.6	87.6	81.5	102.2	1.99	423	3942
Mars [28]	19.9	88.9	78.0	103.4	1.65	288	2264
KRLS [29]	19.0	89.2	80.3	103.8	1.90	494	3884
DiactTOD [30]	17.5	89.5	84.2	104.4	2.00	418	4477
Fine Tuned Flan-T5-Large							
Single Agent	13.0	51.1	44.5	60.8	1.7	383	2658
Multi-Agent DARD	15.6	82.8	70.7	92.3	1.82	354	2871
Claude Sonnet 3.0							
Multi-Agent DARD	9.5	95.6	88.0	101.3	2.37	1197	13742
Fine Tuned Mistral-7B							
Single Agent	15.6	81.0	63.6	87.9	2.75	930	12552
Multi-Agent DARD	15.2	78.8	61.2	85.2	2.79	993	13317
Domain Selective Agents							
Best Domain Agent - DARD	12.1	96.6	88.3	104.6	2.33	1098	10991

Table 2: This table shows the performance comparison between our approaches vs the best-performing models on the End-to-end response generation pipeline. The last three metrics indicate the textual richness. CBE denotes the conditional bigram entropy, # words and # 3-grams are the number of unique words and trigrams, respectively.

4 Discussions

In this section, we compare and present some key insights based on various experiments that we tried.

4.1 Analysis of Claude’s DST performance

We conducted a thorough analysis of errors made by Sonnet 3.0 and discovered that most mistakes stemmed from its tendency to track slots in both user and system utterances, whereas the ground truths typically only include slots from user utterances. Table 3 categorizes these errors. We define over-prediction as cases where groundtruth slots are a subset of predicted slots, under-prediction where groundtruth slots are a superset of predictions, both mismatch when neither is a subset or superset of the other, and value match error when predictions have an incorrect value for any slot. The table reveals that in over half (54%) of cases, predicted responses contain more slots than the groundtruth. Examining about 100 random samples from this set showed that this occurs because Sonnet responses track slots from system utterances as well. An example of this is provided in Appendix A.2. Further investigation uncovered annotation inconsistencies in the MultiWOZ 2.2 dataset itself, with some annotators only tracking slots from user utterances while others included slots from both user and system utterances. This issue is also highlighted in [31], which offers a detailed analysis of the extent of this problem in the MultiWOZ 2.2 dataset. To address this inconsistency, [31] corrected the DST labels in over 70% of the dialogs in the dataset

Error Type	Percentage
Over-prediction	54%
Under-prediction	11%
Both Mismatch	13%
Value Match Error	22%

Table 3: Classification of the errors made by Claude Sonnet 3.0

4.2 Single Agent Vs Multi-Agent Approaches

DST We observe that the JSA of the domain-specific fine-tuned Flan-T5-Large model is 4.6% better than that of a single model. However, for the fine-tuned Mistral-7B, the difference is not as big, and

the single-agent approach performs better. We attribute this observation to the fact that Mistral-7B is a more powerful and larger model ($\sim 7B$ params) compared to Flan-T5-Large ($\sim 780M$ params), hence it is more capable of modeling slots of all domains in a single model. While having domain specialization is useful for the smaller Flan model, it brings no advantages for the bigger Mistral-7B model. A possible drawback of the domain-specific agents is that they are trained on a smaller amount of data than a single agent, and they also do not support cross-domain transfer learning.

Overall TODS Just like the DST pipeline, we get similar observations for the fine-tuning-based approaches, i.e., we get large (31.5%) improvements with domain-specific agents for Flan-T5-Large while Mistral-7B sees a slight decrease (2.7%) in performance. In terms of textual richness, we notice that multi-agent approaches have great lexical diversity. Intuitively, this aligns with our expectations as domain-specific agents are more likely to learn and use domain-specific corpora, leading to higher overall diversity.

Overall, the DARD-based multi-domain approach offers significant flexibility and composability for TODS. This framework serves as a versatile plug-and-play environment for various domain agents, allowing us to select the most effective agents for each domain. For instance, in the case of the MultiWOZ dataset, Claude 3.0 Sonnet worked best for the attraction, hotel, and restaurant domains, while Mistral-7B worked best for the train domain, and Flan-T5-Large worked best for the taxi domain. The multi-agent structure improves interpretability and simplifies the development of improved domain-specific agents. Throughout the development process, we can monitor domain-wise accuracies, focusing on improving underperforming domains without affecting others. Additionally, the domain-expert approach enables runtime performance optimization by employing smaller, faster models for simpler domains while utilizing larger language models for more complex ones. However, these advantages rely on the crucial assumption that the dialog manager agent can accurately assign the appropriate agent to each task. While this assumption held true for the MultiWOZ dataset due to its distinct and exclusive domains, it may not universally apply to TODS with overlapping domains in real-world scenarios.

4.3 Fine-tuned Vs Prompted Models

The BLEU score of prompted Claude is considerably lower compared to that of fine-tuned models and existing works. However, the opposite is true for the inform and success rates. The low BLEU score of Claude’s responses can be explained by the fact that LLM-generated responses follow different speaking styles and vocabulary. Additionally, the model prompt contains only 8-10 (<1%) in-context examples from the training data, while the fine-tuned models are trained on the entire dataset (100%). Hence, the fine-tuned models have a better understanding of responding as an annotator resulting in a higher BLEU score. While the responses from prompted LLMs may have low BLEU scores these responses are preferred by human evaluators [32] [33]. These studies suggest that the low BLEU scores is due to lack of grounding with the dataset. Appendix A.3 presents an example of a conversation using Claude-based agents with a low BLEU score that nonetheless demonstrates reliable inform and success rates.

We also analyzed conversations where fine-tuned agents failed to achieve inform and success rates, but Claude agents succeeded. In 52% of these cases, the fine-tuned model failed to inform because none of its responses contained any suggested venues. We found that this issue arises due to another annotator disagreement in the dataset: while some annotators preemptively suggest venue names in earlier utterances, others ask users about additional preferences first before suggesting them with a particular venue. Figure 2 illustrates this discrepancy. For the same query (museums in the center) with 11 matching venues, annotator-1 immediately recommended a specific venue, while annotator-2 informed the user about the available choices and asked if they had a particular venue in mind. Both methods are valid, but fine-tuned models sometimes adopt the preference-asking way when the ground truth expects an immediate suggestion. As a result, in subsequent interactions, the model assumes a venue name was already mentioned based on the ground truth and doesn’t repeat it. Consequently, no responses contain a venue name, leading the evaluation system to conclude that the agent failed to suggest any venues. Table 4 presents this variation in annotator behavior. For a larger number of venues, most annotators prefer asking for further preferences before making a suggestion, but some directly suggest a name. This often causes the fine-tuned models to miss out on making any suggestions. This issue does not occur with Claude agents, as they are prompted to do both: make a suggestion while also asking for user preferences.

Annotator -1	Annotator -2
<p>USER: Hi, I'm looking for places to go in Cambridge. Preferably a museum in the centre of town.</p> <p>→ 11 attraction venues met the USER's criteria</p> <p>Ground truth Response: you could check out the NAME on ADDRESS.</p> <p>mul1050.json</p>	<p>USER: I also want to go to a museum in the center part of town.</p> <p>→ 11 attraction venues met the USER's criteria</p> <p>Ground truth Response: I have COUNT places. did you have one in specific in mind?</p> <p>pmul1463.json</p>

Figure 2: Two samples from the MultiWOZ 2.2 dataset which show how different annotators respond to the users

Num. Eligible Venues	Venue Suggested
Large (≥ 10)	24.5%
Medium (≥ 5 and < 10)	35.4%
Small (< 5)	55.6%

Table 4: Percentage of times when the annotators immediately recommended user a venue, based on different numbers of venues

5 Conclusion

In this work, we propose a DARD (Domain Assigned Response Delegation), a multi-agent framework to build Task-Oriented Dialogue Systems (TODS). The DARD framework includes a central dialog manager agent that assigns tasks to domain-specific agents, which then handle and solve these tasks. We evaluate this approach using the Dialogue State Tracking (DST) and Response Generation tasks of the MultiWOZ dataset. For our domain agents, we utilize Flan-T5-Large, Mistral-7B, and Claude Sonnet 3.0 models.

Our findings show that the effectiveness of the multi-agent DARD depends on the type of model used. Smaller models, like Flan-T5-Large, experience significant performance improvements with the multi-agent framework, while larger models, such as Mistral-7B, see a slight decrease in performance. Despite this, DARD offers notable benefits in terms of flexibility, composability, and interpretability, making it a valuable framework for developing more efficient TODS. We also observe that while Claude-based agents achieve state-of-the-art inform and success rates, they have lower BLEU scores due to a lack of alignment with the training samples. However, Claude-generated responses are much more lexically diverse than those from other models. Additionally, our work highlights two significant annotator discrepancies in the MultiWOZ 2.2 dataset. The first discrepancy involves the tracking of dialog slots: while some annotators track only the slots from user utterances, others track slots from both user and system utterances. This inconsistency leads to overprediction when using language models like Claude for Dialogue State Tracking (DST). The second discrepancy relates to the conversation policy. Some annotators preemptively suggest the name of a possible venue to the user, while others first ask for user preferences before suggesting a name. This inconsistency causes fine-tuned models to assume that a venue name was already suggested in a previous utterance, leading them not to mention any name in their predicted responses. As a result, these models often fail to achieve high inform and success rates in conversations.

6 Future Work & Limitations

While our work tests DARD on the MultiWOZ benchmark, additional evaluations on more complex task-oriented dialogue (TOD) datasets, such as the Schema Guided Dataset [10] and SMD [34], are needed for more conclusive insights into the advantages of multi-agent frameworks. Moreover, it would be beneficial to evaluate DARD-based TOD systems within an interactive evaluation framework. Currently, model responses are assessed statically based on the existing conversation, which can

be problematic since user messages often correspond to prior ground truth system messages rather than the predicted responses. To enhance DARD's performance, one possible approach is to provide domain agents with selective context instead of the entire conversation. In multi-domain scenarios, such as switching from the attraction domain to the taxi domain, the inclusion of irrelevant prior utterances may hinder the new agent's performance. A dialog manager could mitigate this issue by determining the active domain and selecting only the relevant context.

References

- [1] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.
- [2] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents, 2023.
- [3] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi-agent debate, 2024.
- [4] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.
- [5] Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning, 2024.
- [6] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. Sotopia: Interactive evaluation for social intelligence in language agents, 2024.
- [7] Zhao Kaiya, Michelangelo Naim, Jovana Kondic, Manuel Cortes, Jiaxin Ge, Shuying Luo, Guangyu Robert Yang, and Andrew Ahn. Lyfe agents: Generative agents for low-cost real-time social interactions, 2023.
- [8] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework, 2023.
- [9] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [10] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset, 2020.
- [11] Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In Kallirroi Georgila, Matthew Stone, Helen Hastie, and Ani Nenkova, editors, *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A., June 2014. Association for Computational Linguistics.
- [12] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, H el ene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France, May 2020. European Language Resources Association.
- [13] Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In Tsung-Hsien Wen, Asli Celikyilmaz, Zhou Yu, Alexandros Papangelis, Mihail Eric, Anuj Kumar, I nigo Casanueva, and Rushin Shah, editors, *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online, July 2020. Association for Computational Linguistics.

- [14] Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation, 2021.
- [15] Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In Oliver Lemon, Dilek Hakkani-Tur, Junyi Jessie Li, Arash Ashrafzadeh, Daniel Hernández Garcia, Malihe Alikhani, David Vandyke, and Ondřej Dušek, editors, *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360, Edinburgh, UK, September 2022. Association for Computational Linguistics.
- [16] Tomáš Nekvinda and Ondřej Dušek. Shades of BLEU, flavours of success: The case of MultiWOZ. In Antoine Bosselut, Esin Durmus, Varun Prashant Gangal, Sebastian Gehrmann, Yacine Jernite, Laura Perez-Beltrachini, Samira Shaikh, and Wei Xu, editors, *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online, August 2021. Association for Computational Linguistics.
- [17] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.
- [18] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.
- [19] Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, 2021.
- [20] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy, July 2019. Association for Computational Linguistics.
- [21] Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. Multi-task pre-training for plug-and-play task-oriented dialogue system. 2022.
- [22] Cheng Niu, Xingguang Wang, Xuxin Cheng, Juntong Song, and Tong Zhang. Enhancing dialogue state tracking models through llm-backed user-agents simulation, 2024.
- [23] Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. Towards llm-driven dialogue state tracking, 2023.
- [24] Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. Description-driven task-oriented dialog modeling, 2022.
- [25] Tianjian Huang, Shaunak Halbe, Chinnadhurai Sankar, Pooyan Amini, Satwik Kottur, Alborz Geramifard, Meisam Razaviyayn, and Ahmad Beirami. Robustness through data augmentation loss consistency, 2023.
- [26] Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

- [27] Yihao Feng, Shentao Yang, Shujian Zhang, Jianguo Zhang, Caiming Xiong, Mingyuan Zhou, and Huan Wang. Fantastic rewards and how to tame them: A case study on reward learning for task-oriented dialogue systems, 2023.
- [28] Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. Mars: Modeling context state representations with contrastive learning for end-to-end task-oriented dialog, 2023.
- [29] Xiao Yu, Qingyang Wu, Kun Qian, and Zhou Yu. Krls: Improving end-to-end response generation in task oriented dialog with reinforced keywords learning, 2023.
- [30] Qingyang Wu, James Gung, Raphael Shu, and Yi Zhang. Diacttod: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems, 2023.
- [31] Kun Qian, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. Annotation inconsistency and entity bias in multiwoz, 2022.
- [32] Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. Guiding large language models via directional stimulus prompting, 2023.
- [33] Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [34] Mihail Eric and Christopher D. Manning. Key-value retrieval networks for task-oriented dialogue, 2017.

A Appendix / supplemental material

A.1 Example Input provided for Response Generation

USER: I need a place to dine in the center thats expensive
SYSTEM: I have several options for you; do you prefer African, Asian, or British food?
USER: Any sort of food would be fine, as long as it is a bit expensive. Could I get the phone number for your recommendation?

Number of restaurants that meet the user’s criteria: 33

One of them is the following:

```
<restaurant>
address - 106 Regent Street City Centre
area - centre
food - indian
id - 19214
name - curry garden
phone - 01223302330
postcode - cb21dp
pricerange - expensive
type - restaurant
</restaurant>
```

A.2 Claude DST Response Error

Conversation Context:

USER: My friends said that I must visit Corpus Christi, can you please give me some more information about it?

SYSTEM: I certainly can. Corpus Christi is a college located in the **center** of town, on King's Parade. There's a two pound entrance fee. Do you need any additional information?

USER: Yes. What is the postcode?

SYSTEM: The postcode is cb21rh. Do you need anything else?

USER: Are there any African restaurants in the centre?

SYSTEM: I have one listing for bedouin would that work for you?

USER: Is this listing in the expensive price range?

Ground-Truth Slots:

restaurant: area: centre, food: african, name: bedouin, pricerange: expensive

attraction: name: corpus christi, type: college

Claude Predicted Slots:

restaurant: area: centre, food: african, name: bedouin, pricerange: expensive

attraction: **area: centre**, name: corpus christi, type: college

A.3 Claude Response Generation Example Conversation

USER: I would like to try saigon city.

Ground truth: great choice. when can i make the reservation for?

Claude Response: NAME is an FOOD restaurant located at ADDRESS in the AREA area. their phone number is PHONE. would you like me to book a table for you?

BLEU: 1.76

USER: 5 people on a Wednesday at 19:45.

Ground truth: you got it. here is your reference number: REFERENCE.

Claude Response: booking confirmed for NAME on DAY at TIME for COUNT people. your reference number is REFERENCE.

BLEU: 10.93

USER: Thank you for help. That's everything I needed.

Ground truth: Have a great day!

Claude Response: you're welcome. thank you for contacting us!

BLEU: 4.76

Avg. Conversation BLEU: **5.81**

A.4 Fine-Tuned Models didn't offer any suggestion

USER: I am coming to visit and will be in the centre area. I would like an attraction to visit.

SYSTEM: sure, how about NAME?

Predicted Response: there are COUNT attractions in the AREA. what type of attraction are you interested in?

USER: Sounds interesting! What is the address?

SYSTEM: Located at ADDRESS .

Predicted Response: the address is ADDRESS

None of the predicted response present any museum

A.5 Claude Based Restaurant Agent Prompt

<instructions>

You are supposed to act a system that assists users with their queries about finding and booking restaurants.

You will be provided with the following information:

- Conversation History between the USER and the SYSTEM
- Number of restaurants that meet the user's requirements. You will also be provided with details of one of the restaurant that meet the user's requirements, if any

You have to generate the following:

- Delexicalized response to the last USER message
- Values of the delexicalized tokens you used in your response

</instructions>

<Delexicalization>

You are allowed to use the following delexicalized tokens:

1. [restaurant_name] - for the name of the restaurant
2. [restaurant_food] - for the food cuisine of the restaurant
3. [restaurant_pricerange] - for the price range of the restaurant
4. [restaurant_area] - for the area of the restaurant
5. [restaurant_address] - for the address of the restaurant
6. [restaurant_phone] - for the phone number of the restaurant
7. [restaurant_postcode] - for the postcode of the restaurant
8. [restaurant_choice] - used at instances where you provide user with a choice of multiple restaurants
9. [restaurant_ref] - used to provide the booking reference to the user
10. [restaurant_booktime] - used to provide the booking time to the user
11. [restaurant_bookday] - used to provide the booking day to the user
12. [restaurant_bookpeople] - used to provide the number of people you have booked the restaurant for

</Delexicalization>

<how to respond>

These are the specific guidelines to follow while responding to the user:

1. Do not write very long responses for the user it should be at max 1 to 2 lines
2. Always inform the user about the things they asked for in their last utterance, it can be things like phone number, address, postcode, etc
3. Whenever you book a restaurant you should always provide the booking reference to the user, also inform that their table will be held for 15 minutes
4. In the instances where multiple restaurants meet user's requirements, you should inform the user about the count using [restaurant_choice] token and also try to suggest user one of the restaurant using [restaurant_name] token
5. Before you book the restaurant be sure to ensure that the user has provided the information about bookpeople, bookday and booktime. If they haven't first ask them to provide these information
6. If you notice that none of the restaurant meet user's criteria, just politely inform them that no restaurant is found that satisfies the user's query
7. If the user's done with their query, just say thanks for contacting us and end the conversation.
8. You may encounter cases where the restaurant that has been suggested to the user in the conversation has not been provided to you as example this will be because you are only presented with one of the examples. In these cases you have to respond normally to the user as if you know information about the suggested restaurant

</how to respond>

<output format>

You have to strictly follow the following format while generating the response:

Response: The delixicalized response to the last USER utterance

Token_values: [delixicalized tokens] - [value],

Reasoning: The reasoning behind the response

</output format>

<examples>

In this section you will be presented with some of the examples you can learn from:

<examples_about_suggesting_restaurants>

This subsection will provide you examples of how to suggest restaurants to the user.

.....

</examples_about_suggesting_restaurants>

<examples_no_restaurant_found>

This subsection contains examples where no restaurant was found meeting user's criteria

...

</examples_no_restaurant_found>

<examples_about_booking>

This subsection will contain examples related to user queries about booking restaurants

...

</examples_about_booking>

<examples_ending_conversation>

This subsection will contain examples related to ending the conversation

...

</examples_ending_conversation>

</examples>

Now its your turn to answer, generate the response to the following conversation history: