Exploring the Impact of ChatGPT on Task-Oriented Dialogue Systems: Benefits and Challenges

Anonymous ACL submission

Abstract

Large language models (LLMs) like ChatGPT have demonstrated the ability to perform a variety of natural language processing (NLP) tasks. However, it's unclear whether ChatGPT can serve as a task-oriented dialogue system. In this paper, we evaluate the impact of ChatGPT on task-oriented dialogue (TOD) systems and perform a comprehensive analysis to learn its benefits and challenges. We find that ChatGPT performs well on relatively simple dialogue understanding tasks such as intent detection and slot filling, but fails to understand complex multi-turn conversations and interact with KB in dialogue state tracking and response generation. Future LLM-based TOD work should pay more attention to (1) incorporating domain knowledge (2) understanding complex instructions (3) modeling long-term memory (4) interacting with external knowledge bases. [1]

1 Introduction

Large language models (LLMs) (Brown et al., 2020a; Ouyang et al., 2022; Touvron et al., 2023) have achieved significant performance on various natural language process (NLP) tasks. Their superior zero-shot learning capability enables a new paradigm of NLP research and applications by prompting LLMs without finetuning. Recently, the ChatGPT[2] LLM released by OpenAI has attracted much attention from the research community. Through RLHF training (Ouyang et al., 2022), ChatGPT has impressive capabilities in various aspects, including generating high-quality responses, rejecting unsafe questions, and self-correcting previous errors based on subsequent conversations.

Despite its rapidly increasing worldwide attention, we need to figure out how to evaluate the potential risks behind ChatGPT. Previous efforts have studied various aspects of ChatGPT in law (Choi et al., 2023), ethics (Shen et al., 2023), reasoning (Bang et al., 2023), robustness (Wang et al., 2023a) and arithmetic (Yuan et al., 2023). However, there is a lack of comprehensive research on the impact of ChatGPT on task-oriented dialogue (TOD) systems (Ni et al., 2021). Different from the existing open-domain conversation scenarios of ChatGPT, TOD aims to accomplish a specific task or goal, such as making a reservation or booking a flight by interacting with a knowledge base (KB). It contains semantic understanding, long context modeling, querying the KB and decision-making. Applying ChatGPT to TOD is a nontrivial task that requires both commonsense reasoning and expert knowledge. Therefore, in this paper, we focus on the impact of ChatGPT on task-oriented dialogue systems and perform a comprehensive analysis to learn its benefits and challenges.

Current task-oriented dialogue systems are commonly divided into two categories: pipeline-based and end-to-end. The former build a TOD system by designing multiple functional modules, including Natural Language Understanding (Goo et al., 2018b; He et al., 2020b; Xu et al., 2020; He et al., 2020c), Dialogue State Tracking (Wu et al., 2019; Gao et al., 2019), Policy Learning (Peng et al., 2018; Liu et al., 2021), and Natural Language Generation (Peng et al., 2020). Although these modules can achieve good performance in their respective tasks using the state-of-the-art neural networks, they can't be jointly optimized and make it difficult to transfer modular TOD systems to another domain. The latter (Peng et al., 2021; Su et al., 2021; He et al., 2022a) use only one end-to-end generative model to perform both knowledge base retrieval and response generation in a multi-task paradigm. In this paper, we follow the two standard settings to build LLM-based TOD systems. We hope to provide new insights for the future development of TOD in the era of large language models.

[1] We will open-source our code and all the evaluation results after blind review to facilitate future explorations.

[2] https://openai.com/blog/ChatGPT

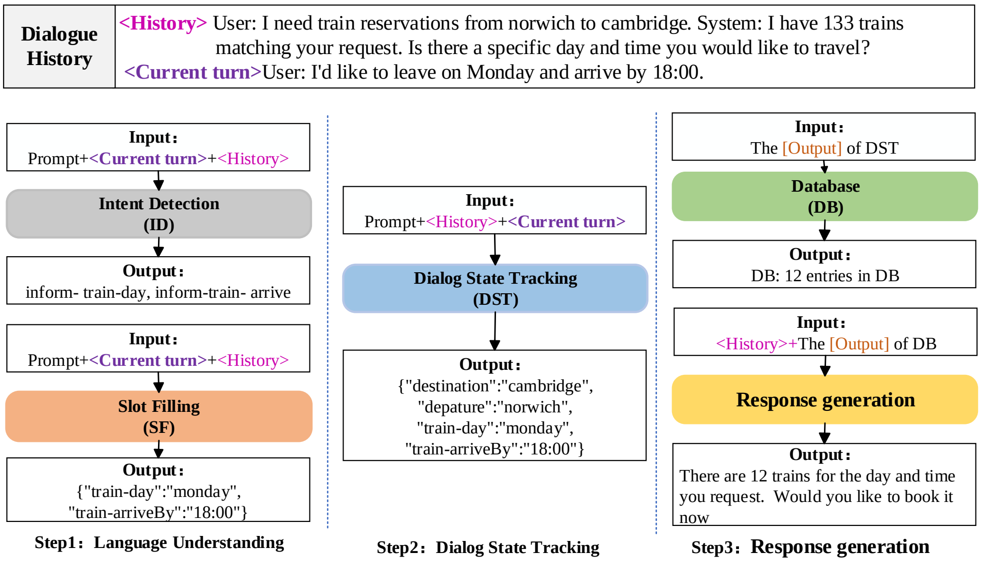

Figure 1: The overall structure of pipeline-based TOD framework.

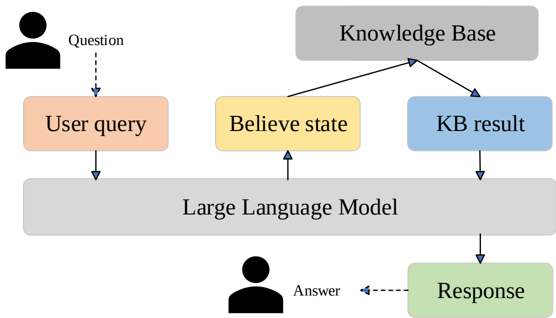

Figure 2: The overall structure of end-to-end TOD framework.

In this work, we introduce an LLM-based TOD framework and evaluate the performance with respect to modular components and end-to-end metrics. Since finetuning these LLMs becomes more expensive and unaffordable, we perform zero-shot evaluation by inferring directly on the test dataset [3]. For pipeline-based modules, we construct each task prompt by combining the task description, the current user query, dialogue history and response format, as shown in Figure 1. Note that we combine dialogue policy learning and natural language generation to a single response generation task similar to He et al. (2022a). For the end-to-end model, we introduce an LLM-based architecture, which first generates a belief state based on the dialogue history, then queries the KB with the generated belief state, and finally generates natural responses. The overall end-to-end architecture is shown in Figure 2. We perform single-domain and multi-domain evaluation using MultiWOZ (Budzianowski et al., 2018).

[3]We also validate the impact of more advanced prompt strategies such as few-shot and CoT (in Appendix E), as well as the bias of different zero-shot templates (in Appendix F).

We mainly compare ChatGPT and text-davince-003 to the existing state-of-the-art finetuning baselines.

Our findings:

- Generally, ChatGPT performs worse than the state-of-the-art models that are fine-tuned on a given TOD task.
- ChatGPT achieves good performance in the single-domain intent detection task but fails to recognize complex multi-domain dialogues.
- For the slot filling task, ChatGPT demonstrates decent performance, and adding few-shot examples can achieve consistent improvements.
- For the dialogue state tracking task, ChatGPT fails to track structured slot-value pairs. We find that ChatGPT can't follow the input instructions and output inappropriate answers.
- ChatGPT does not perform well in generating responses. Although it has strong abilities to understand user goals and generate fluent responses based on existing information, ChatGPT still has weak reasoning abilities and lack long-term memory in multi-turn conversations.
- ChatGPT achieves high fluency scores but lower coherency scores in the end-to-end modeling way. We argue that ChatGPT can not effectively interact with external knowledge bases or learn long dependency.

We believe that future improvements for LLM-based TODs come from the following aspects: (1) Incorporating domain knowledge (2) Understanding complex instructions (3) Modeling long-term memory (4) Interacting with external knowledge bases.

2 LLM for Pipeline-based TOD

2.1 Intent Detection

2.1.1 Task Description

The intent detection task plays a critical role in natural language understanding and constitutes a vital technology for the development of TOD systems (Young et al., 2013). Its objective is to facilitate accurate comprehension of user intents within the dialog system. It can be further classified into single-intent detection and multi-intent detection. Multi-intent detection pertains to scenarios where a user query may encompass more then one intent (Kim et al., 2017; Gangadharaiah and Narayanaswamy, 2019; Qin et al., 2020). In this paper, we evaluate the multi-intent detection capability of ChatGPT.

2.1.2 Related Work

The state-of-the-art intent detection methods use pre-trained models (Devlin et al., 2018; Cer et al., 2018; Jiang et al., 2020). In addition, researchers have explored techniques such as semi-supervised pre-training, response selection tasks, and sentence similarity matching to improve the performance of intent detection (Wu et al., 2020; He et al., 2022b). Zeng et al. (2022a) introduce Semi-Supervised Knowledge-Grounded Pre-training. They use Roberta as the backbone and utilize the dialog history as input. The hidden state of the [CLS] token is used to predict the results, with the learning objective being binary cross entropy. We use it as our finetuning baseline in this paper.

2.1.3 Experiment Setup

We utilize MultiWOZ2.1 for the evaluation [4]. We extract the user intent for each utterance from the dialog_act in the log of user turns. The intent consists of three components: "Action," "Domain," and "Entity," in the format of 'Action-Domain-Entity.' In total, we have 64 intents, and we present the detailed statistics in Appendix Table 8. We employ three commonly used metrics in multi-label classification tasks: Precision, Recall, and F1 score.

2.1.4 Prompt Engineering

We design the prompt to guide ChatGPT in identifying user intents. We provide an instruction that includes a task description and the supported intent labels. ChatGPT is provided with the instruction, the user's current utterance, and the conversation history. Our prompt takes the following format <Task description><Utterance for text><Dialog history><Response Format>. The complete prompt is presented in Appendix Figure 3.

[4]Due to the cost of ChatGPT API calls, we limit the number of test samples for each task to around 100, consistent with previous works such as Bang et al. (2023).

Domain	Model	Precision	Recall	F1
Attraction	baseline	91.49	93.48	89.58
	text-davinci-003	20.86	27.18	23.6
	ChatGPT	69.57	66.67	68.08
Hotel	baseline	78.02	79.78	76.34
	text-davinci-003	12.27	21.51	15.62
	ChatGPT	63.11	69.89	66.37
Restaurant	baseline	96.64	94.74	98.63
	text-davinci-003	28.26	35.62	31.51
	ChatGPT	73.68	76.71	75.17
Taxi	baseline	95.74	97.83	93.75
	text-davinci-003	32.81	43.75	37.5
	ChatGPT	63.33	79.17	70.37
Train	baseline	90.16	90.16	90.16
	text-davinci-003	35.23	50.82	41.61
	ChatGPT	59.09	63.93	61.42
Multi	baseline	79.90	81.26	78.58
	text-davinci-003	20.86	27.18	23.6
	ChatGPT	32.1	38.45	35.46

Table 1: Comparison of intent detection performance between ChatGPT and baseline

2.1.5 Results

Table 1 represents the comparison of LLMs (ChatGPT and text-davinci-003) with the finetuning baseline in three metrics. The results indicate a significant gap between current LLMs and the baseline. This can be attributed to the conflict between the general knowledge of LLM and the domain-specific knowledge required for intent detection. ChatGPT outperforms text-davinci-003 due to its superior dialogue understanding capability.

We identify five types of errors made by ChatGPT, as shown in the Table 2. The most common error is returning intent from the dialogue history. We suspect that this may be due to ChatGPT's difficulty in understanding longer instructions or mistaking the user's intent from the history as the potential intent for the current turn. ChatGPT also tends to anticipate the user's needs, which can be attributed to the deviation from human understanding of instructions. It struggles with understanding labels and identifying real-world entities, which can be attributed to its lack of specific domain knowledge. Additionally, ChatGPT tends to miss key information when the input is too long.

In terms of *action*, ChatGPT occasionally confuses *Inform* and *Request*. As for *domain*, ChatGPT achieves a relatively high recall rate, but errors can still occur. For example, if a user informs the des-

Error Type	Ratio
Return intents from the historical dialog.	27.9%
Make anticipatory judgments about the user's intent.	19.7%
Inability to recognize the Name or Type in the user's requests.	16.4%
Miss the information in the utterance.	16.4%
Ambiguous label semantics	19.7%

Table 2: Error type and relevant ratio of intent detection from ChatGPT.

Error Type	Ratio
Boundary Error	26.2%
Misclassification Error	9.2%
Overprediction	50.8%
Underprediction	13.8%

Table 3: Error types of slot filling for ChatGPT.

tination of a taxi is a restaurant, ChatGPT may recognize it as *Inform-Restaurant-Name*. However, the *entity* is where more errors occur, such as missing information provided by the user, recognizing information from the dialogue history, and anticipating additional information that the user may need. Based on this, we speculate that ChatGPT's performance would be good for coarse-grained intent detection, but for the fine-grained labels we set, its performance is limited by the **lack of domain knowledge**, **overuse of general knowledge**, and **the impact of input length on the results**. These three points are the directions for optimizing ChatGPT's performance in the TOD multi-intent detection.

2.2 Slot Filling

2.2.1 Task Description

The slot filling (SF) task is a critical component in the task-oriented dialog system which aims to identify task-related slot types in certain domains (Fujii et al., 1998). Given an input utterance $X = \{x_1, x_2, ..., x_N\}$, where N represents the length of X, we adopt a triple $y_i = \{l, r, t\} \in Y$ to represent the $i-th$ entity that appears in X, where Y represents all the entity triplets in X , and l, r denote the entity boundaries, while t denotes the entity type.

2.2.2 Related Work

The slot filling model has undergone several stages of improvement throughout its development, gradually evolving from initial sequence labeling-based methods to generation-based approaches.(Yao et al., 2014; Liu and Lane, 2016; Goo et al., 2018a; He et al., 2020a; Wang et al., 2021) Large-scale language models (LLMs) (Brown et al., 2020a; Chowdhery et al., 2022) have demonstrated impressive in-context learning capabilities and have achieved promising results across various NLP tasks. Similarly, LLMs have been proven to be effective in the slot filling task (Xu et al., 2022; Wang et al., 2023b).

2.2.3 Experiment Setup

In this experiment, we evaluate the performance of Large Language Models (LLMs) on the slot filling task using the MultiWOZ 2.1 dataset. The specific distribution of slots and labels in each domain in the experiment is presented in Appendix Table 8. We compare ChatGPT against the following models for slot filling: Text-davinci-003(Brown et al., 2020b) the latest model in the Davinci series with 175B parameters and PSSAT (Dong et al., 2022) a strong fine-tuned baseline using bert. To measure the performance of the model, we use precise, recall and F1 score as our automatic evaluation metric.

2.2.4 Prompt Engineering

We designed a prompt for LLM to guide ChatGPT to identify the slots. We provide a task description, predefined slot categories, examples and dialogue history for ChatGPT as input and ChatGPT outputs slot : category pairs. The complete prompt is presented in Appendix Figure 4.

2.2.5 Results

LLMs (Text-davinci-003, ChatGPT) exhibit relatively poorer performance in the slot filling task compared to the baseline model PSSAT. We attribute this to the fact that LLMs are typically pre-trained on large-scale, general-domain corpora, which makes it challenging to perform well on specific domain data in zero-shot scenarios. Specifically, ChatGPT demonstrates significant differences in accuracy compared to the baseline model PSSAT, indicating that it still faces challenges in accurately identifying slots. In terms of bad cases, this is manifested by ChatGPT tending to over-predict slots (51.9%), labeling some non-slot words as slots. Additionally, ChatGPT frequently mispredicts slot boundaries (26.0%), resulting in lower accuracy. We believe both issues arise due to knowledge confusion caused by the mismatch between the knowledge acquired through pre-training LLMs and the specific problems being addressed.

To further enhance the contextual learning capabilities of LLM, we incorporate five examples from

Model	Domain	precise	recall	F1
PSSAT	Train	91.67	95.65	93.62
text-davinci-003		44.83	56.52	50.00
ChatGPT		67.86	82.61	74.51
ChatGPT +5example		71.43	86.96	78.43
PSSAT	Taxi	94.74	94.74	94.74
text-davinci-003		44.00	57.89	50.00
ChatGPT		66.67	84.21	74.42
ChatGPT +5example		68.00	89.47	77.27
PSSAT	Restaurant	94.44	100	94.14
text-davinci-003		47.62	58.82	52.63
ChatGPT		70.00	82.35	75.67
ChatGPT +5example		75.00	88.21	81.07
PSSAT	Hotel	96.00	97.96	96.97
text-davinci-003		47.37	55.10	50.94
ChatGPT		67.74	85.71	75.67
ChatGPT +5example		69.35	87.76	77.48
PSSAT	Attraction	94.12	96.97	95.52
text-davinci-003		50.00	60.61	54.80
ChatGPT		69.05	87.88	77.34
ChatGPT +5example		71.43	90.91	80.00
PSSAT	Multi	95.14	97.16	96.14
text-davinci-003		39.08	48.23	43.18
ChatGPT		62.50	85.11	72.07
ChatGPT +5example		65.24	86.52	74.39

Table 4: Slot filling results on MultiWOZ.

the current task domain into the input of ChatGPT. We observe a certain improvement in performance compared to the zero-shot scenario, indicating that LLM can leverage a few domain-specific examples for learning and achieve enhanced effectiveness through contextual learning.

In conclusion, ChatGPT can enhance its performance by learning domain-specific knowledge through the incorporation of domain examples in the input. We believe that further improvements for ChatGPT can be achieved by providing more domain-specific knowledge or conducting domain fine-tuning, which would facilitate better slot recognition and matching between slots and labels.

2.3 Dialog State Tracking

2.3.1 Task Description

Dialogue State Tracking (DST) serves as a crucial component within Task-Oriented Dialogue Systems. Its primary objective is to recognize user intent and the corresponding dialogue attributes, including slots and their respective values (Williams et al., 2016; Eric et al., 2019). During each turn, these attributes are identified, and their accumulation constructs the dialogue state, which directs the system's response. Moreover, the dialogue state plays a pivotal role in retrieving vital information from external databases. This process is essential for constructing efficient TOD Systems.

model	JOINT ACC						
	ave	multi	hotel	rst	taxi	train	attraction
GALAXY	53.97	47.70	62.90	53.33	71.05	71.11	68.57
text-davinci-003	18.31	11.22	11.29	40.00	36.84	55.56	14.29
ChatGPT	23.33	14.03	17.74	53.33	52.63	53.33	28.57

Table 5: Dialog State Tracking results on MWOZ.

2.3.2 Related Work

DST models have progressed through various stages, transitioning from classification-based (Ye et al., 2021; Chen et al., 2020) to generation-based approaches (Heck et al., 2020). Furthermore, researchers have aimed to construct complete end-to-end TOD systems that perform well in DST tasks. For example, SimpleTOD (Hosseini-Asl et al., 2020) cascades sub-tasks for dialogue generation based on pre-trained models and generates belief states through a generation approach. With the arrival of large-scale pre-trained neural language models, generation-based DST models have achieved excellent results without any dependence on domain-specific modules.

2.3.3 Experiment Setup

We sampled 100 dialogues from various domains in MultiWOZ 2.1 to evaluate the models' performance on DST task. Table 8 shows the number of slots involved in each domain, the average belief state length of each dialogue and other information. Additionally, multi-domain dialogues generally involve more slots, longer dialogue length, and a longer belief state that needs to be maintained. This challenge the models' ability to reason in multi-turn dialogues and maintain long-term memory. We used "Joint ACC" (Joint Accuracy) to assess the ability on the DST task. Specifically, for each turn and each slot, the system's predicted result needs to match the true value exactly. Only when all slot predictions match the true values entirely, it is considered correct.

2.3.4 Prompt Engineering

We constructed a prompt for the ChatGPT to complete the DST task. We take instructions, dialogue history, and belief state templates as inputs, and ChatGPT outputs the current turn's belief state. For multi-round conversations, they will be divided into rounds, and each round will be evaluated once. The whole template is shown in Appendix A.

2.3.5 Results

Main Results The performance of LLM and the fine-tuned model was assessed in the DST task, and the corresponding experimental results are illustrated in Table 5. LLM's performance in the zero-shot DST task is worse than the fine-tuned model. Multi-domain settings present greater complexity, leading to poorer performance for all models. ChatGPT performs better than text-davinci-003, likely due to improved fine-tuning on chat-based instructions for better contextual understanding.

Case Study we sampled error examples and categorized the types of errors, as presented in Appendix Table 9. We argue that LLM's subpar performance can be attributed to three main reasons. Firstly, there is a conflict between the general knowledge and domain-specific knowledge of LLM, resulting in errors such as "hallucination". Secondly, The context being too long makes it challenging for LLM to capture the key points, resulting in errors such as "modifications-error" and "fill-less". Finally, the incorrect output format cannot be ignored. It results in "can-but-wrong" type errors, which account for a high proportion of 13.68%. These issues hamper LLM's comprehension, retention, and practical applicability.

Future Directions Based on the previous summary of the shortcomings of LLM, we believe that improvements can be made in the following aspects: First, a new architecture that can better incorporate domain-specific knowledge into LLM needs to be explored which improves its access to external knowledge bases, addressing issues like hallucinations. Secondly, we need a mechanism to compress lengthy contexts, extract key information, or guide LLM to focus on certain information. Finally, accessing databases can not be limited to traditional methods such as using SQL language. Vectorizing the database or using fuzzy matching methods can enhance the system's fault tolerance to model output formats.

2.4 Response Generation

2.4.1 Task Description

We evaluated the LLM's ability to interact with users using natural language in response generation tasks. This task aims to predict dialogue responses based on the given dialogue contexts. To conduct this experiment, we used the policy optimization setting introduced by Yang et al. (2021). The model takes the dialogue history and the database search results retrieved by the ground truth belief state as input, and generates responses according to the system act determined by the model itself. It should be noted that our response generation setting implicitly includes the prediction of dialogue policy. So, we did not evaluate LLMs' performance in policy learning separately during the pipeline evaluation.

2.4.2 Related Work

Pre-trained language models (PLMs) have been used to generate fluent and relevant responses based on dialogue history. One example is DialoGPT (Zhang et al., 2019), which is pre-trained on numerous conversation-like exchanges extracted from Reddit. S2KG (Zeng et al., 2022b) enhances the model's ability to select knowledge for generating responses by introducing semi-supervised pre-training based on task-oriented dialogues. Large language models (LLMs) have also been introduced to improve the quality of responses. For instance, LaMDA (Thoppilan et al., 2022) suggests that increasing the model's scale can improve safety and factual grounding. BlenderBot 3 (Shuster et al., 2022) enables large models to store information in long-term memory and search the internet for information. In this section, we will investigate the effectiveness of LLMs in response generation tasks.

2.4.3 Experiment Setup

We tested how well models performed in generating responses by analyzing 100 dialogues from different domains in MultiWOZ 2.1. Table 8 shows clear differences in the average number of turns and length of responses across various domains. Multi-domain dialogues tend to have more domains and longer turns than single-domain dialogues, like Train and Attraction, which can challenge models' ability to maintain long-term memory and reason in multi-turn dialogues. We compared ChatGPT's zero-shot response generation with text-davinci-003 and a strong fine-tuned baseline, Galaxy (He et al., 2022c). We utilize automatic evaluation metrics, including **BLEU** (Papineni et al., 2002), **Inform**, **Success**, and **Comb**, to measure task completion and response quality. For more information about these metrics, refer to Appendix C.

2.4.4 Prompt Engineering

We designed a prompt for LLM to generate a system response based on dialogue history and ground truth database results. The prompt instructs LLM to act as a task-oriented dialogue system and only provide a system response without additional content. The complete template is in Appendix Figure 6. During the evaluation process, we fill in the placeholders in the prompt with the dialogue history and database results and use LLM's output as

Model	Domain	BLEU	Inform	Success	Comb
Galaxy	Train	11.31	90.00	90.00	101.31
text-davinci-003		3.41	90.00	40.00	68.41
ChatGPT		0.91	90.00	40.00	65.91
Galaxy	Taxi	20.97	100.00	100.00	120.97
text-davinci-003		2.99	100.00	0.00	52.99
ChatGPT		1.75	100.00	0.00	51.75
Galaxy	Restaurant	19.98	90.00	90.00	109.98
text-davinci-003		2.64	100.00	30.00	67.64
ChatGPT		4.20	90.00	20.00	59.20
Galaxy	Hotel	11.31	90.00	90.00	101.31
text-davinci-003		1.82	80.00	20.00	51.82
ChatGPT		2.54	90.00	20.00	57.54
Galaxy	Attraction	18.66	100.00	90.00	113.66
text-davinci-003		6.38	80.00	70.00	81.38
ChatGPT		5.44	90.00	70.00	85.44
Galaxy	Multi	21.43	88.00	70.00	100.43
text-davinci-003		2.40	76.00	24.00	52.40
ChatGPT		2.29	78.00	16.00	49.29

Table 6: Response Generation results on MultiWOZ.

the system response.

2.4.5 Results

Table 6 displays the results of Response Generation on MWOZ2.1. We observed that the performance of LLM models was significantly worse than that of the fine-tuned model Galaxy.

The LLM model did well on the Inform Rate, similar to Galaxy, but poorly on the Success Rate. For example, in the hotel domain, ChatGPT and text-davinci-003 scored 90, 80 on the Inform Rate respectively, but only 20, 20 on the Success Rate. We explained that LLMs understood user intent and integrated database results well, but due to AI safety limitations, they avoided actions such as booking and focused on providing information.

We found that LLM performance varies significantly across domains. For example, ChatGPT performs better in the Attraction domain with a BLEU score of 5.44 and 70% success rate. However, in the Hotel domain, the BLEU score drops to 2.54 and the success rate falls to 20%. We argue that simpler domains like Attraction require only simple information retrieval and integration, while more complex domains such as trains and hotels or multi-domains with complex scenarios require the model to have strong reasoning and long-term memory capabilities.

We found that ChatGPT did not perform significantly better than text-davinci-003, especially in multi-turn conversations. In fact, ChatGPT scored slightly lower than text-davinci-003 in terms of BLEU, Inform Rate, and Success Rate. Our analysis shows that ChatGPT's ability to understand and reason in multi-turn conversations is slightly inferior to that of text-davinci-003. For example, text-davinci-003 can accurately infer the departure and destination of a taxi based on the dialogue history, while ChatGPT needed to ask further questions to the user and was unable to extract relevant information from the dialogue history.

Overall, LLMs pre-trained on general corpus struggle to generate responses for task-oriented dialogues due to weak multi-turn conversation reasoning and long-term memory. Although LLMs are excellent in generating fluent responses based on existing information and understanding user goals, they may sometimes reject dialogue actions like booking due to AI safety concerns. To overcome these limitations, we recommend pre-training them on domain-specific data or using external models to augment them.

3 LLM for End-to-End TOD

3.1 Task Description

We explored the ability of LLM as a task-oriented dialogue system to interact with users in an end-to-end manner. In this task, the model should generate a belief state based on the dialogue history, query database results with the generated belief state, and finally generate responses.

3.2 Related Work

Most current work builds end-to-end systems by fine-tuning pre-trained language models. UBAR(Yang et al., 2021) trains the model on the entire dialog session sequence, which consists of the user's utterance, belief state, database result, system act, and system response. SPACE-3(He et al., 2022b) proposes maintaining task flow in TOD systems with a novel unified semi-supervised pre-trained conversation model. Some work has attempted to combine LLMs with end-to-end dialogue systems. Hudeček and Dušek (2023) introduces a pipeline for LLM-based TOD conversations to evaluate LLM performance.

3.3 Experiment Setup

We performed end-to-end modeling experiments on Zero-Shot LLM-based models, including ChatGPT and text-davinci-003, as well as strong fine-tuned models such as Galaxy. To evaluate the performance of end-to-end TOD systems, we report both automatic and human evaluation metrics. For automatic evaluation, we use the same metric as described in Section 2.4.3. For human evaluation, the details and results can be found in Appendix D.

Model	Domain	BLEU	Inform	Success	Comb
Galaxy	Train	20.7	90	90	110.7
text-davinci-003		1.22	100	40	71.22
ChatGPT		0.36	100	40	70.36
Galaxy	Taxi	18.27	100	100	118.27
text-davinci-003		2.01	100	0	52.01
ChatGPT		1.57	100	0	51.57
Galaxy	Restaurant	17.54	90	90	107.54
text-davinci-003		3.91	70	20	48.91
ChatGPT		2.7	70	20	47.7
Galaxy	Hotel	14.6	100	100	114.6
text-davinci-003		0.99	70	20	45.99
ChatGPT		1.45	70	20	46.45
Galaxy	Attraction	16.47	100	80	106.47
text-davinci-003		3.79	90	70	83.79
ChatGPT		5.26	80	70	80.26
Galaxy	Multi	20.46	90	70	100.46
text-davinci-003		2.09	22	0	13.09
ChatGPT		2.32	68	10	41.32

Table 7: Automatic End2End results on MultiWOZ.

3.4 Prompt Engineering

Based on LLM, our pipeline includes two steps for generating the system response: 1) belief state generating and 2) system response generating.

Belief State Generating We created a prompt that uses dialogue history and a Belief State Template to generate belief states for LLM. The LLM is required to act as a task-oriented dialogue system using the provided dialogue history and belief state template, and return only the updated Belief State. The complete template is in Appendix Figure 7. During evaluation, we replace the placeholders in the prompt template with the dialogue history and belief state templates. The LLM generates a belief state that retrieves and returns results from the database. **System Response Generating** We use the same prompt as in section 2.4.4 to instruct LLMs to generate a system response based on dialogue history and retrieved database results. However, in the evaluation process, we replace the database result with the result retrieved by the generated belief states, rather than the ground truth database result.

3.5 Automatic Evaluation Results

Table 7 shows that the zero-shot LLM performed significantly worse than the fine-tuned model across all domains. Our analysis highlights a gap between LLMs' general knowledge and the domain-specific knowledge required by end-to-end dialogue systems. Therefore, fine-tuned models are still better at generating belief states for retrieval databases and responses than LLMs.

We found it difficult to achieve the user's goal using the LLM-based model. For example, in the restaurant domain, the success rate of text-davinci-003 and ChatGPT is only 20%. We identified two main reasons for this. (1) LLMs may not be able to actively use tools to acquire knowledge from external sources to enhance their abilities., leading to incorrect belief states and incorrect responses. (2) LLMs struggle with long-term memory and processing large amounts of information. In this scenario, LLMs lost most of the information, resulting in a decreased success rate.

We found that LLM-based models perform worse in multi-domain dialogues than in single-domain ones. For instance, ChatGPT scores 80.26 in the attraction domain but only 13.09 in the multi-domain. This is because models require diverse domain knowledge in multi-domain scenarios. For example, while generating a belief state, the model must master slot value information of all domains to produce correct values - a significant challenge for LLMs with only general knowledge.

We observed that ChatGPT performs similarly to text-davinci-003 in a single domain, but significantly outperforms it in multiple domains. For example, while the text-davinci-003 model only achieved a combined score of 13.09 in multi-domain tasks, ChatGPT achieved a score of 41.32. We argue that ChatGPT has a much stronger ability to follow instructions in complex scenarios than text-davinci-003.

Overall, there is still a significant gap between LLM and practical end-to-end task-oriented dialogue systems in terms of acquiring knowledge from external sources, handling long information, lacking diverse domain-specific knowledge, and weak reasoning abilities. Possible solutions include reinforcement learning(Qin et al., 2023), using external models to summarize long information, further tuning LLM on domain-specific dialogue, and using a chain-of-thought approach to enhance complex reasoning abilities(Wei et al., 2022).

4 Conclusion

We have empirically studied the effect of ChatGPT on task-oriented dialogue systems. We find that ChatGPT performs well on dialogue understanding tasks such as intent detection and slot filling, but fails to understand complex multi-turn conversations and interact with KB in dialogue state tracking and response generation. Our experiments show that there is still room for improvement to ChatGPT on these TOD tasks. We hope that this study can inspire future works, such as incorporating domain knowledge, understanding complex instructions, modeling long-term memory and interacting with external knowledge bases.

Limitations

This work is a preliminary empirical study on the effect of ChatGPT on TOD, and it has several limitations. (1) Considering that MultiWOZ is the most classic task-oriented dialogue dataset and includes labels for almost all the tasks we need to evaluate, we primarily conduct experiments on this dataset. In the future, we will evaluate additional datasets to ensure more solid experimental settings. (2) Due to the API cost, this work uses a small scale of test samples and limited prompt templates, which may result in biased results. We only conduct analysis on some tasks regarding the impact of more advanced prompt strategies and different prompt templates. (3) We conduct our experiments at the beginning of March. This ChatGPT version is not consistent with the current one. Therefore, new results are possibly higher than those in the paper. (4) We select ChatGPT as a representative of LLMs but there exist many other LLMs like Claude, PaLM 2, etc. Since these works are not publicly available until our paper, we leave more comparisons to future work.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners.

Paweł 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 *Conference on Empirical Methods in Natural Language Processing*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Jonathan H. Choi, Kristin E. Hickman, Amy B. Monahan, and Daniel B. Schwarcz. 2023. Chatgpt goes to law school. *SSRN Electronic Journal*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Guanting Dong, Daichi Guo, Liwen Wang, Xuefeng Li, Zechen Wang, Chen Zeng, Keqing He, Jinzheng Zhao, Hao Lei, Xinyue Cui, Yi Huang, Junlan Feng, and Weiran Xu. 2022. PSSAT: A perturbed semantic structure awareness transferring method for perturbation-robust slot filling. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5327–5334, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur.

2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Y Fujii, F Shiota, Y Miki, and T Morokuma. 1998. Vertical displacement determination in the nrlm superconducting magnetic levitation system. In *1998 Conference on Precision Electromagnetic Measurements Digest (Cat. No. 98CH36254)*, pages 233–234. IEEE.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 564–569.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Z. Hakkani-Tür. 2019. Dialog state tracking: A neural reading comprehension approach. In *SIGDIAL Conferences*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018a. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Chih-Wen Goo, Guang-Lai Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung (Vivian) Chen. 2018b. Slot-gated modeling for joint slot filling and intent prediction. In *North American Chapter of the Association for Computational Linguistics*.

Keqing He, Shuyu Lei, Yushu Yang, Huixing Jiang, and Zhongyuan Wang. 2020a. Syntactic graph convolutional network for spoken language understanding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2728–2738, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Keqing He, Jingang Wang, Chaobo Sun, and Wei Wu. 2022a. Unified knowledge prompt pre-training for customer service dialogues. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

Keqing He, Yuanmeng Yan, and Weiran Xu. 2020b. Learning to tag oov tokens by integrating contextual representation and background knowledge. In *Annual Meeting of the Association for Computational Linguistics*.

Keqing He, Jinchao Zhang, Yuanmeng Yan, Weiran Xu, Cheng Niu, and Jie Zhou. 2020c. Contrastive zero-shot learning for cross-domain slot filling with adversarial attack. In *International Conference on Computational Linguistics*.

Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022b. Space-3: Unified dialog model pre-training for task-oriented dialog understanding and generation. *arXiv preprint arXiv:2209.06664*.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022c. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Vojtěch Hudeček and Ondřej Dušek. 2023. Are llms all you need for task-oriented dialogue? *arXiv preprint arXiv:2304.06556*.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76:11377–11390.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Bing Liu and Ian Lane. 2016. Joint online spoken language understanding and language modeling with recurrent neural networks. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 22–30, Los Angeles. Association for Computational Linguistics.

Sihong Liu, Jinchao Zhang, Keqing He, Weiran Xu, and Jie Zhou. 2021. Scheduled dialog policy learning: An automatic curriculum learning framework for task-oriented dialog system. In *Findings*.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, Vinay Vishnumurthy Adiga, and E. Cambria. 2021. Recent advances in deep learning based dialogue systems: a systematic survey. *Artificial Intelligence Review*, 56:3055–3155.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Lidén, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Integrating planning for task-completion dialogue policy learning. *ArXiv*, abs/1801.06176.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *ArXiv*, abs/2002.12328.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. Td-gin: Token-level dynamic graph-interactive network for joint multiple intent detection and slot filling. *arXiv preprint arXiv:2004.10087*.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*.

Yiqiu Shen, Laura Heacock, Jonathan Elias, Keith D Hentel, Beatriu Reig, George Shih, and Linda Moy. 2023. Chatgpt and other large language models are double-edged swords. *Radiology*, page 230163.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Annual Meeting of the Association for Computational Linguistics*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Weirong Ye, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xingxu Xie. 2023a. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *ArXiv*, abs/2302.12095.

Liwen Wang, Xuefeng Li, Jiachi Liu, Keqing He, Yuanmeng Yan, and Weiran Xu. 2021. Bridge to target domain by prototypical contrastive learning and label confusion: Re-explore zero-shot learning for slot filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9474–9480, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. *arXiv preprint arXiv:2004.06871*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Annual Meeting of the Association for Computational Linguistics*.

Derek Xu, Shuyan Dong, Changhan Wang, Suyoun Kim, Zhaojiang Lin, Akshat Shrivastava, Shang-Wen Li, Liang-Hsuan Tseng, Alexei Baevski, Guan-Ting Lin, et al. 2022. Introducing semantics into speech encoders. *arXiv preprint arXiv:2211.08402*.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *International Conference on Computational Linguistics*.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.

Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. 2014. Recurrent conditional random field for language understanding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4077–4081. IEEE.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks? *ArXiv*, abs/2304.02015.

Weihao Zeng, Keqing He, Zechen Wang, Dayuan Fu, Guanting Dong, Ruotong Geng, Pei Wang, Jingang Wang, Chaobo Sun, Wei Wu, and Weiran Xu. 2022a. Semi-supervised knowledge-grounded pre-training for task-oriented dialog systems.

Weihao Zeng, Keqing He, Zechen Wang, Dayuan Fu, Guanting Dong, Ruotong Geng, Pei Wang, Jingang Wang, Chaobo Sun, Wei Wu, and Weiran Xu. 2022b. Semi-supervised knowledge-grounded pre-training for task-oriented dialog systems. In *Proceedings of the Towards Semi-Supervised and Reinforced Task-Oriented Dialog Systems (SereTOD)*, pages 39–47, Abu Dhabi, Beijing (Hybrid). Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

A DST Prompt

We constructed a prompt for the ChatGPT to complete the DST task. We take instructions, dialogue history, and belief state templates as inputs, and ChatGPT outputs the current turn's belief state. For multi-round conversations, they will be divided into rounds, and each round will be evaluated once. The whole template is shown in Figure 5. The Belief State template standardizes the output format of LLM for our subsequent parsing, while also providing slot information to LLM. For single-domain dialogues, we only provide the corresponding domain's slots, while for multi-domain dialogues, we provide all slots. This is because we have found that there is a high possibility of domain confusion errors when providing slots for multiple domains, which can obscure other errors. Therefore, when testing the effectiveness of LLM on a single domain, we only provide slot information for a single domain. We also add the following sentence additionally in the prompt when testing on multiple domains.

<Task description> I need you to help me to detect the intent of user's query in a dialog. So I will give you a utterance and its dialog history. You need to tell me the intent of this utterance. The supported intents include [intent1], [intent2] ,... [intentN]... You can only classify the utterance using the above intents and one utterance may include more than one intent.
<User query> Please tell me the intent of this text according its dialog history: [Here is the text]
<Dialog history> [Here is the dialog history]
<Output format> Please respond to me with the format of "Intent: xx"

Figure 3: The ChatGPT prompt for the intent detection task.

B DST Error Descriptions

We have summarized the error types of DST in Table 9. The meaning of each error is as follows: **"Slot Wrong"** means that the correct value has been extracted, but has been filled into the wrong slot. **"Modifications Error"** means that if the user modifies a certain slot value multiple times, ChatGPT may not be able to recognize the modified slot value. **"Ignore Error"** means that when the user can accept all the possible values for a certain slot, this slot should be filled with ignore, but ChatGPT does not tend to do so. The meaning of **"Fill Less"** is that some slot values have been

Domain	Doma.num	Dial.num	Turn.num	Intent.num	Slot.num	Slot-turn.num	Slot-label.num	Belief-State.len	Turn.len
Train	1.0	10	4.5	16	6	1.3	23	5.2	22.8
Taxi	1.0	10	3.8	10	4	1.3	19	5.2	22.8
Hotel	1.0	10	6.2	21	10	1.3	49	6.9	27.7
Restaurant	1.0	10	4.5	17	7	1.6	17	5.3	26.4
Attraction	1.0	10	3.5	12	3	1.3	33	1.9	27.0
Multi	2.3	50	7.8	64	30	1.5	141	10.1	27.1

Table 8: MultiWOZ Dataset statistics. "Doma.num" represents the number of involved domains. "Dial.num" represents the number of dialogues per domain. We randomly select 10 dialogues for each domain from the original test set. "Turn.num" represents the average number of turns per dialogue. "Intent.num" represents the number of involved intents per domain. "Slot.num" represents the total slot number per domain. "Slot-turn.num" represents the average slot number per turn. "Slot-label.num" represents the total values number per domain. "Belief-State.num" represents the average slot number of the final belief state. "Turn.len" represents the average length each turn.

<Task description> I need you to identify the slots of a user's query in a dialog. I will give you an utterance and its dialog history. The categories of slots can only come from a predefined set of categories. Note that each sentence may have multiple slots. I will give you a predefined set of slot categories. Predefined slot categories include: [type1], [type2] ,... [typeN]
<User query> Please tell me the slots and their categories in the following text: [Here is the text]
<Dialog history> [Here is the dialog history]
<Output format> Please respond to me in the format of "slot : category "

Figure 4: The ChatGPT prompt for the slot filling task.

<Task description> Do the task of dialogue state tracking! I'll give you a dialogue history and a template that describes the belief state. Based on your understanding of the slots, you need to ccurately fill in the slot values. For slots that are not mentioned in the dialogue history, leave them as "". You must strictly follow the template utput, without any extra words. The template will be given to you in json format, so you also need to output in json format.
<Additional prompt for multi> Pay attention to that each slot belongs to one domain and there are 5 domains : taxi, hotel, restaurant, train and attraction. You must carefully fill the slot which has similar slot but not in the same domain, such as hotel-area and restaurant-area. You should carefully tell which domain user talked about and fill the slot in that domain!
<Belief State Template> [Here is belief state template in json format]
<Dialogue History> [Here is the Dialogue history]

Figure 5: The prompt we design to assist ChatGPT in performing DST.

missed. The meaning of **"Hallucination"** is that ChatGPT will fill in some slot values that have not appeared in the conversation history based on its own world knowledge. "boundary-error" means that ChatGPT tends to confuse two slots with the same name but different domains. **"Unconfirmed Error"** means that ChatGPT tends to fill in the slot values that have been suggested by the system but have not been confirmed by the user. The meaning of **"Over Inference"** is that ChatGPT is not careful enough when filling in slot values. When it sees that the user uses "I", it likes to default the number of people to 1. When the user wants to find a restaurant with the word "curry" in its name, it will assume that the user only wants to eat curry, and it also likes to default the time to today. The meaning of **"Can But Wrong"** is that ChatGPT has extracted some slot values that are correct but have small errors compared to the ground truth, such as missing an article or being too specific. The meaning of **"Ground Truth Wrong"** is that we have checked some of the errors made by ChatGPT and found that the generation of ChatGPT is reasonable, while, in contrast, the data labeling is wrong.

C Automatic Evaluation Metrics

To measure task completion and response quality, we report the following automatic evaluation metrics: (1) **BLEU**(Papineni et al., 2002) measures the quality of the generated response. (2) **Inform** measures whether the system has provided the correct entity. (3) **Success** measures whether the system has answered all the requested information. (4) **Comb**(Mehri et al., 2019) measures the overall quality of the system, computed as (Inform + Success) x 0.5 + BLEU.

<Task description> You should act as a task-oriented dialogue system. I will give your dialogue history, database results. You should give the response according to them. You should only return the system response. Do not provide other content!
<Dialogue History> [Here is dialogue history]
<DataBase Result> [Here is the database result]

System Response:

Figure 6: The prompt we design to assist ChatGPT in performing response generation.

<Task description> You should act as a task-oriented dialogue system. I will give your dialogue history and belief state template. You should fill each of the states with slot value in provided Belief State. You should only return the Updated Belief State Template. Do not provide other content!
<Dialogue History> [Here is dialogue history]
Belief State Template:
<DataBase Result> [Here is the belief state template]

Update Belief State:

Figure 7: The prompt we design for ChatGPT to generate belief states.

Error Type	**Ratio**
Unconfirmed Error	34.74%
Fill Less	17.89%
Can But Wrong	13.68%
Slot Wrong	7.37%
Ignore Error	7.37%
Over Inference	6.32%
Hallucination	4.21%
Ground Truth Wrong	4.21%
Modifications Error	3.16%
Boundary Error	1.05%

Table 9: Error types of Dialog State Tracking for ChatGPT. For the explanation of each error type, please refer to Appendix B.

Model	**Domain**	**Success**	**Coherency**	**Fluency**
Galaxy		2.7	2.6	2.7
text-davinci-003	Train	1.6	2	2.6
ChatGPT		1.7	2.3	3
Galaxy		3	2.6	3
text-davinci-003	Taxi	1.7	2.3	2.6
ChatGPT		1.3	2	3
Galaxy		2.7	2.4	3
text-davinci-003	Restaurant	2.1	2	2.3
ChatGPT		2.7	2	2.6
Galaxy		2.7	3	2.7
text-davinci-003	Hotel	2.1	2.3	2.6
ChatGPT		2.3	2	2.7
Galaxy		3	3	2.6
text-davinci-003	Attraction	1.9	2.6	2.7
ChatGPT		2.7	2.6	3
Galaxy		2.54	2.84	2.76
text-davinci-003	Multi	1	1.54	2.24
ChatGPT		2.24	2.42	2.78

Table 10: Human Evaluation End2End results on MultiWOZ.

D Human Evaluation

D.1 Human Evaluation Details

We manually evaluated the end-to-end modeling performance of the model. To do this, we randomly selected 100 dialogue samples from different domains and collected the corresponding responses generated by ChatGPT, text-davinci-003, and Galaxy. We asked five professional linguistic evaluators to rate the quality of the generated dialogue based on three metrics: (1) **Success** measures whether the system achieved the user's goal by interacting with them. (2) **Coherency** measures whether the system's response is logically coherent with the dialogue context. (3) **Fluency** measures the fluency of the system's response. Each metric was rated on a scale of 1 (worst) to 3 (best). The inter-annotator agreement for Success, Coherency, and Fluency was 0.61, 0.63, and 0.60, respectively. The final score for each metric was the average score of the 5 annotators.

D.2 Human Evaluation Results

Table 10 presents the results of human evaluation on the MWOZ2.1 dataset. We observe a relatively consistent correlation between human evaluation and automatic evaluation. According to the human evaluation, **LLMs score higher in fluency but lower in coherency**. Our analysis indicates that LLM's long dialogue comprehension and reasoning abilities are weak, while its ability to generate fluent text is strong. In the cases examined, we found that as the dialogue becomes longer, LLM starts to repeat its generated responses and lacks a proper understanding of new user queries.

Model	Domain	BLEU	Inform	Success	Comb
Zero-Shot		0.36	100	40	70.36
CoT	Train	1.31	100	40	71.31
Few-Shot		1.15	100	40	71.15
Zero-Shot		1.57	100	0	51.57
CoT	Taxi	0.34	100	0	50.34
Few-Shot		1.51	100	0	51.51
Zero-Shot		2.7	70	20	47.7
CoT	Restaurant	1.67	80	20	51.67
Few-Shot		1.06	80	20	51.06
Zero-Shot		1.45	70	20	46.45
CoT	Hotel	2.10	80.0	20	52.10
Few-Shot		2.11	60	10	37.11
Zero-Shot		5.26	80	70	80.26
Cot	Attraction	3.39	70	70	73.39
Few-Shot		6.11	70	70	76.11
Zero-Shot		2.32	68	10	41.32
CoT	Multi	1.95	62.0	12.0	38.95
Few-Shot		2.81	54	8.0	33.81

Table 11: Automatic End2End result of Different Prompt Strategies on MultiWOZ. Zero-Shot, CoT, and Few-Shot represent the default setting we use, the Zero-CoT setting, and the setting where Few-Shot examples are added.

Model	Domain	BLEU	Inform	Success	Comb
Origin		0.36	100	40	70.36
Template 1	Train	0.45	100	40	70.45
Template 2		0.86	100	40	70.86
Origin		1.57	100	0	51.57
Template 1	Taxi	1.96	100	0	51.96
Template 2		2.30	100	0	52.30
Origin		2.7	70	20	47.7
Template 1	Restaurant	1.32	80	20	51.32
Template 2		1.15	80	20	51.15
Origin		1.45	70	20	46.45
Template 1	Hotel	2.06	70	20	47.06
Template 2		1.85	80	20	51.85
Origin		5.26	80	70	80.26
Template 1	Attraction	5.86	70	70	75.86
Template 2		5.36	80	70	80.36
Origin		2.32	68	10	41.32
Template 1	Multi	1.27	50.0	10.0	31.27
Template 2		1.86	56.0	10.0	34.96

Table 12: Automatic End2End result of Different Prompt Templates on MultiWOZ. The origin represents the default prompt that we use. Template 1 and Template 2 are the other prompt templates that we design.

E Different Prompt Strategies

We typically evaluate most tasks using the zero-shot setting. To investigate the impact of more advanced prompt strategies on model performance, we conducted tests using Zero-CoT (Kojima et al., 2022) and Few-Shot approaches for end-to-end dialogue tasks. In the Zero-CoT approach, we added the phrase "Let's think step by step" after generating the belief state and response prompts. In the Few-shot setting, we included examples in the prompt for generating belief states and response prompts. Table 11 results indicate that the Few-Shot setting can slightly improve the BLEU score, but there is no significant improvement in other metrics such as inform rate and success rate. Furthermore, CoT does not enhance the end-to-end performance at all. These findings suggest that there is still a considerable gap between current LLMs and practical end-to-end task-oriented dialogue systems. Therefore, it is necessary to develop more effective strategies to enhance the ability of LLMs.

F Bias of the Prompt Template

To reduce the bias introduced by the Prompt Template and improve the reliability of automatic evaluation, we develop several prompt templates (as shown in Table 13) and evaluated their effectiveness on end-to-end tasks. Table 12 indicates that the biases resulting from different Prompt Templates are relatively minor, which further validates the relative reliability of our automatic evaluation approach.

Method	Belief State	Response
Prompt Template 1	Your role is to act as a task-oriented dialogue system. I will provide you with a dialogue history and a template for your belief state. Your task is to fill each state with the appropriate slot value from the provided Belief State. Your output should only consist of an updated Belief State Template. Please refrain from including any other content.	Your role is to act as a task-oriented dialogue system. You will receive the dialogue history and database results, and provide a response based on them. Your response should only be the system's response and should not include any additional content.
Prompt Template 2	As a task-oriented dialogue system, your goal is to fill in each state of the provided belief state template with the corresponding slot value based on the dialogue history. Your output should only consist of the updated belief state template. Please refrain from including any additional information.	As a task-oriented dialogue system, your role is to respond based on dialogue history and database results. Your responses should be limited to system responses and should not include any additional content.

Table 13: Different Prompt Templates.