Self-Explanation Prompting Improves Dialogue Understanding in Large Language Models

Haoyu Gao[123†], **Ting-En Lin**[2], **Hangyu Li**[2], **Min Yang**[3*],
Yuchuan Wu[2], **Wentao Ma**[2], **Fei Huang**[2], **Yongbin Li**[2*]
[1] University of Science and Technology of China [2] Alibaba Group
[3]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
{hy.gao, min.yang}@siat.ac.cn
shuide.lyb@alibaba-inc.com

Abstract

Task-oriented dialogue (TOD) systems facilitate users in executing various activities via multi-turn dialogues, but Large Language Models (LLMs) often struggle to comprehend these intricate contexts. In this study, we propose a novel ”Self-Explanation” prompting strategy to enhance the comprehension abilities of LLMs in multi-turn dialogues. This task-agnostic approach prompts the model to analyze each dialogue utterance before task execution, thereby improving performance across various dialogue-based tasks. Experimental results from six benchmark datasets confirm that our method consistently outperforms other zero-shot prompts and matches or exceeds the efficacy of few-shot prompts, demonstrating its potential as a powerful tool in enhancing LLMs’ comprehension in complex dialogue tasks.

Keywords: large language model, prompting, dialog understanding

1. Introduction

Recent advancements in large language models (LLMs) have achieved great success in various NLP tasks (Brown et al., 2020; Touvron et al., 2023; Chowdhery et al., 2022). However, the vast model parameters pose challenges in downstream fine-tuning. To circumvent these challenges, diverse prompting strategies have been researched to enhance LLM performance (Brown et al., 2020; Liu et al., 2021; Sorensen et al., 2022; Huang et al., 2023; Bhattacharjee et al., 2023; Chae et al., 2023; Si et al., 2023; Zhao et al., 2023). *In-context learning* emerges as a viable alternative to fine-tuning, leveraging examples to augment language processing abilities. To elicit the reasoning ability of LLMs, Chain-of-Thought has been integrated within the prompting framework, showing remarkable performance in tasks requiring complex reasoning (Wei et al., 2022; Miao et al., 2021; Talmor et al., 2018; Yao et al., 2024; Besta et al., 2023; Zhou et al., 2022). Stemming from CoT prompting, numerous studies have delved into refining CoT via prompt design modifications (Li et al., 2022; Fu et al., 2022; Zhang et al., 2022) and optimizing reasoning paths (Wang et al., 2022a,b; Zelikman et al., 2022). In contrast, to reduce dependency on human demonstrations, the Zero-shot CoT (Kojima et al., 2022) employs the post-append instruction, “Let’s think step by step” to let Large Language Models (LLMs) automatically generate reasoning steps.

Context: James decides to run 3 sprints 3 times a week.
He runs 60 meters each sprint.
Question: How many total meters does he run a week? **Reasoning task**

Context: **Task-oriented Dialogue task**
USER:I'd like to know what's available as cheap Asian Oriental food.
SYSTEM:You must try dojo noodle bar in the centre of town!
USER:I need to get the address please.
...
SYSTEM:The TR1992 leaves at 21:35. would that work for you?
USER:Yes, please book 5 seats for me.
Question:
When does the user need train to arrive?
How many train tickets needed by the user for the train?
What the cuisine that the user explicitly requested of the restaurant?
What the name that the user explicitly requested of the restaurant?
Which day that the user explicitly requested of the restaurant booking?

Figure 1: The input examples for the reasoning task and the task-oriented dialogue are structured into two components: Context and Question.

Despite the effectiveness of CoT prompting, most existing prompting methods focus on eliciting the reasoning ability inherent in large language models. However, these techniques might need to be revised when applied to tasks that require contextual comprehension rather than reasoning steps. Specifically, dialogue-based tasks (Lin et al., 2022; Hu et al., 2022a; Li et al., 2023; Cai et al., 2023) serve as typical examples that require strong comprehension ability rather than reasoning ability. The task-oriented dialogue (TOD) (He et al., 2022a,b,c) is one of the most representative tasks that facilitates users in executing various activities, including but not limited to hotel and restaurant reservations, by engaging in multi-turn dialogues (Gao et al., 2023; Qian et al., 2023; Yu

[*] Corresponding authors.
[†] Work done while interning at Alibaba.

LREC-COLING 2024, pages 14567–14578
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

Task	Dataset	Avg. #Tokens	Context	Answer Search Space	Prompting Method	Focus on
Reasoning	MultiArith GSM8K	16.6 33.6	Short	Internal	Chain-of-Thought Plan-and-Solve	Reasoning Step
Dialog Understanding	SGD MultiWOZ	940.9 1229.7	Long	External	Self-Explanation	Context

Table 1: Comparative analysis of reasoning and dialogue understanding tasks, highlighting the distinctive application of the proposed Self-Explanation method.

et al., 2023). An illustrative example of the reasoning task and the TOD task can be seen in Figure 1. Contrary to the reasoning task, which typically consists of concise context, the TOD mainly involves multi-turn dialogues with long contexts. Not only do these tasks differ in terms of context length, but they also exhibit differences across numerous other dimensions. As delineated in Table 1, the reasoning task predominantly emphasizes intricate problem-solving steps that entail extensive computations and conversions. This underscores the model's inherent ability to reason. Consequently, the scope of searching for an answer predominantly resides within the model.

However, when performing dialogue-based tasks, success depends on a strong understanding of the dialogue context rather than complex reasoning. TOD tasks mainly obtain information directly from the external contexts to which the search space for answers is strongly related. The different emphases of the two tasks resulted in the underperformance of CoT prompts in dialogue-based tasks. Judging from the results of existing evaluation studies (Heck et al., 2023; Hudeček and Dušek, 2023; Bang et al., 2023), the current LLMs with unoptimized prompting perform significantly worse than specialized small models on some dialogue-based tasks. Hu et al. (2022b) have reformulated the dialogue state tracking task into a few-shot text-to-SQL paradigm, utilizing the robust code capabilities of Codex. While this represents an intriguing approach for DST tasks, the text-to-SQL may not be universally applicable, particularly in procedural TOD tasks such as next-action prediction. Additionally, the example retriever needs to be retrained for each new dataset, which imposes limitations on this approach.

To address the above issues, we explore several ways to enhance the comprehension capabilities of LLMs by mimicking the way humans solve conversational problems (Chi et al., 1989). We introduce the Self-Explanation prompt strategy, prompt the model to explain every utterance in the dialogue first, and then complete the task based on the generated explanation. Despite its simplicity, the proposed method enhances the performance of dialogue comprehension of LLMs in various dialogue-based tasks. More importantly, our prompt is task-agnostic and can be easily applied to a variety of dialogue-based tasks. We evaluate the proposed method across six dialogue-based datasets. The results show that our prompt consistently surpasses other zero-shot prompts and is on par with or surpasses few-shot prompts. In summary, our contributions include:

- We conduct a comprehensive comparison between reasoning tasks and dialogue understanding tasks, identifying the limitations of current prompting methods.
- We propose a simple yet effective prompting strategy, Self-Explanation, that significantly enhances the dialogue comprehension capacities of large language models.
- Extensive experiments on six dialogue-based datasets have demonstrated that the proposed method surpasses existing prompting approaches in performance.

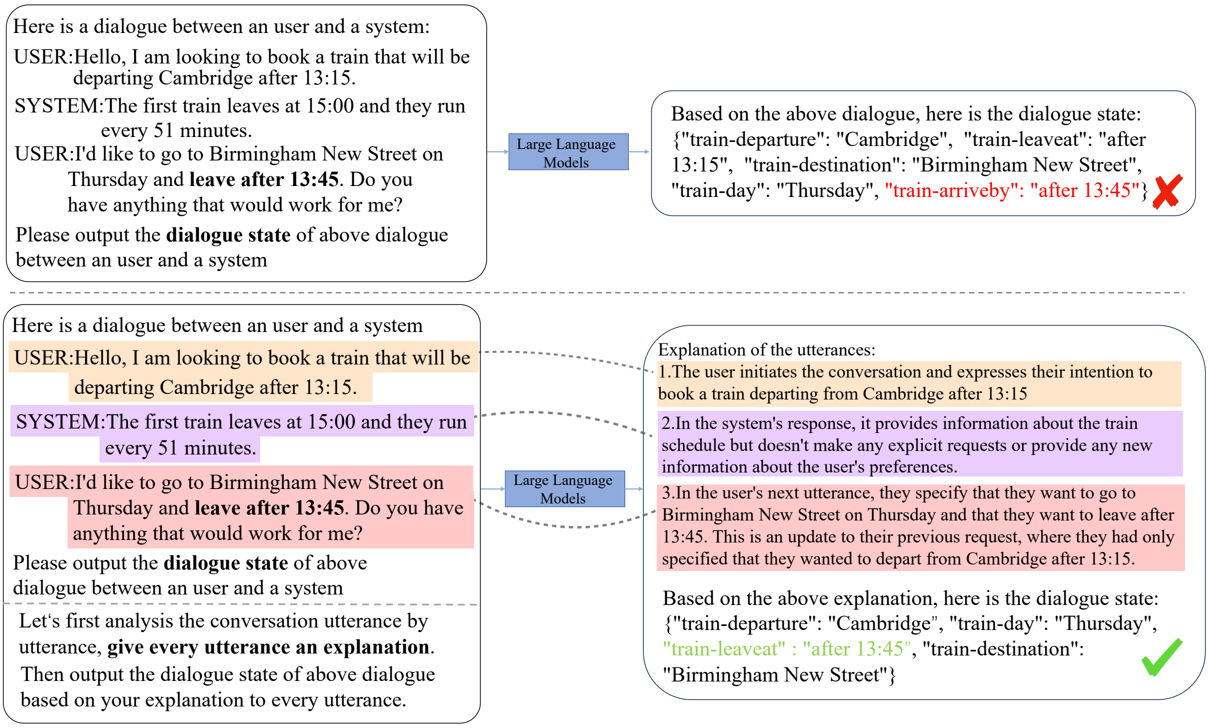
2. Method

2.1. Formalization

As illustrated in Figure 3, task completion can be deconstructed into four parts: context, question, intermediate steps, and final answer. The former two components belong to the input, while the latter two belong to the output generated by LLMs.

The context, denoted as $\mathcal{C}$, provides a descriptive framework that outlines the problem setting and background. For reasoning tasks, this context describes a specific situation. An example of this can be observed in Figure 1, where $\mathcal{C}$ contains the activities of James. Meanwhile, in the context of TOD tasks, $\mathcal{C}$ typically is a multi-turn dialogue.

As for the question component, represented by $\mathcal{Q}$ is an inquiry for specific information related to $\mathcal{C}$. In the realm of reasoning tasks, $\mathcal{Q}$ typically asks for a value derived from multi-step computations. This implies that the solution isn't readily available within $\mathcal{C}$. To illustrate, refer to Figure 1 where $\mathcal{Q}$



Here is a dialogue between an user and a system:

USER:Hello, I am looking to book a train that will be departing Cambridge after 13:15.

SYSTEM:The first train leaves at 15:00 and they run every 51 minutes.

USER:I'd like to go to Birmingham New Street on Thursday and **leave after 13:45**. Do you have anything that would work for me?

Please output the **dialogue state** of above dialogue between an user and a system

Based on the above dialogue, here is the dialogue state: {"train-departure": "Cambridge", "train-leaveat": "after 13:15", "train-destination": "Birmingham New Street", "train-day": "Thursday", "train-arriveby": "after 13:45"} ✗

Here is a dialogue between an user and a system

USER:Hello, I am looking to book a train that will be departing Cambridge after 13:15.

SYSTEM:The first train leaves at 15:00 and they run every 51 minutes.

USER:I'd like to go to Birmingham New Street on Thursday and **leave after 13:45**. Do you have anything that would work for me?

Please output the **dialogue state** of above dialogue between an user and a system

Let's first analysis the conversation utterance by utterance, **give every utterance an explanation**. Then output the dialogue state of above dialogue based on your explanation to every utterance.

Explanation of the utterances:

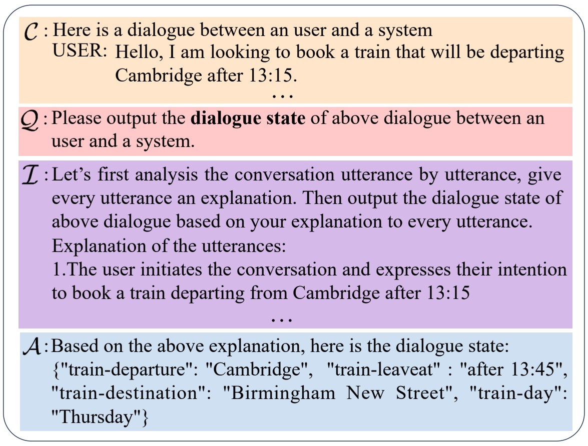
1.The user initiates the conversation and expresses their intention to book a train departing from Cambridge after 13:15

2.In the system's response, it provides information about the train schedule but doesn't make any explicit requests or provide any new information about the user's preferences.

3.In the user's next utterance, they specify that they want to go to Birmingham New Street on Thursday and that they want to leave after 13:45. This is an update to their previous request, where they had only specified that they wanted to depart from Cambridge after 13:15.

Based on the above explanation, here is the dialogue state: {"train-departure": "Cambridge", "train-day": "Thursday", "train-leaveat" : "after 13:45", "train-destination": "Birmingham New Street"} ✓

Figure 2: Example inputs and outputs of GPT-3 with No explanation ahead (upper) and Explain before answer (lower). Explanation greatly improves the understanding of the dialogue.



$\mathcal{C}$: Here is a dialogue between an user and a system
USER: Hello, I am looking to book a train that will be departing Cambridge after 13:15.
…

$\mathcal{Q}$: Please output the **dialogue state** of above dialogue between an user and a system.

$\mathcal{I}$: Let's first analysis the conversation utterance by utterance, give every utterance an explanation. Then output the dialogue state of above dialogue based on your explanation to every utterance.
Explanation of the utterances:
1.The user initiates the conversation and expresses their intention to book a train departing from Cambridge after 13:15
…

$\mathcal{A}$: Based on the above explanation, here is the dialogue state: {"train-departure": "Cambridge", "train-leaveat" : "after 13:45", "train-destination": "Birmingham New Street", "train-day": "Thursday"}

Figure 3: The structure of inputs and outputs with four parts.

probes for the aggregate distance James runs in a week. Addressing this necessitates discerning the frequency of James' sprints per week and the distance of each sprint. Subsequent multiplication of these two values yields the final result.

On the other hand, in a TOD task, the $\mathcal{Q}$ is more straightforward, often inquiring about the existence of specific information. Using Figure 1 as a reference, the question might ask for the scheduled departure time of a reserved train or the type of cuisine a user seeks. Responses to these types of inquiries are readily extractable from $\mathcal{C}$, obviating the need for additional computation.

In general, when presented with context $\mathcal{C}$ and a query $\mathcal{Q}$, LLMs is tasked with generating the corresponding answer $\mathcal{A}$, following a probabilistic distribution as denoted by:

$$\mathcal{A} \sim P(a|\mathcal{Q},\mathcal{C}) \tag{1}$$

However, a more refined prompting to guide the LLMs in their response generation entails the utilization of some intermediate problem-solving steps in order to enhance LLMs performance. Instead of immediately generating the final answer $\mathcal{A}$, the model is prompted to generate a series of intermediate problem-solving steps denoted as $\mathcal{I}$ based on the $\mathcal{C}$ and $\mathcal{Q}$:

$$\mathcal{I} \sim P(i|\mathcal{Q},\mathcal{C}) \tag{2}$$

The format or content of $\mathcal{I}$ can vary significantly across different tasks. For reasoning tasks, $\mathcal{I}$ might encompass multiple reasoning steps, while for understanding tasks, it could involve interpretations of $\mathcal{C}$. Comparatively, let the LLM to first generate these $\mathcal{I}$, before arriving at the $\mathcal{A}$, is a more logical approach:

$$\mathcal{A} \sim P(a|\mathcal{I},\mathcal{Q},\mathcal{C}) \tag{3}$$

This paradigm shift not only enables the model to demonstrate a deeper understanding of the problem but also provides greater transparency into its problem-solving process, potentially enhancing the model's interpretability and performance across a wide range of tasks.

2.2. Self-Explanation

Humans often find it challenging to respond to questions grounded in extensive new information. One strategy that has been empirically shown to enhance comprehension of new material is self-explanation. The concept of self-explanation, originating from psychological research (Chi et al., 1989), involves learners generating explanations for themselves while processing unfamiliar content. Notably, this study demonstrated that learners engaging in self-explanation were better able to grasp core concepts and principles than their counterparts who did not employ this strategy.

Drawing inspiration from human cognitive processes and this psychological paradigm, we introduce the Self-Explanation prompting method, a zero-shot prompting technique designed to enhance multi-turn dialogue comprehension. Within the process, models initially provide explanations for each utterance in a multi-turn dialogue. Subsequently, these models execute the specified task, relying on their previously generated explanations. In the process of explaining, the large language models (LLMs) transform low-level natural language inputs into more abstract, high-level information, such as the intent or action of the speaker.

The framework is structured without the need for demonstration examples. Following the problem formalization in section 2.1, we organize the inputs using the template "$\mathcal{C}$:[C]. $\mathcal{Q}$:[Q]. $\mathcal{A}$:[A]", wherein [C] and [Q] represents the input slot designated for the context and question, respectively. As for the last part, [A] is populated by manually curated instructions prompting the model to generate intermediate steps $\mathcal{I}$. Central to our method is the instruction: *"Provide explanations for each utterance and then respond based on these explanations."* For the decoding strategy, we opt for the straightforward greedy decoding method, though beam search decoding could be employed to produce a broader range of explanations.

3. Experiments

3.1. Experimental Setup

3.1.1. Datasets and task

We evaluate our self-explanation on six datasets from three categories of dialogue understanding tasks: Task-oriented dialogue (TOD), Emotion Recognition in Conversations (ERC) task and Response Selection (RS) task. For TOD task, the datasets can be divided into two types based on the dialogue schema: Procedural and Declarative (Mannekote et al., 2023). In the context of task-oriented dialogue, a dialogue schema refers to a structured representation of the conversational flow or the pivotal entities, commonly referred to as "slots," that must be identified and captured. The Procedural schema, derived from the STAR dataset (Mosig et al., 2020), represents a dialogue domain as a directed graph closely resembling a flowchart. Within this schema, discrete nodes correspond to various elements such as user utterances, system responses, or backend service interactions. The central objective of the procedural schema is to rigorously adhere to the prescribed task flow. For Procedural schema, we choose the STARv2 (Zhao et al., 2022) dataset.

STARv2 dataset, which is an upgraded version of STAR (Mosig et al., 2020) with new ground truth belief state and new natural language action descriptions. STAR is a schema-guided task-oriented dialogue dataset consisting of 24 tasks across 13 domains. We evaluate the next action prediction task, which is to predict the next system action conditioned on the dialogue history and take the weighted F-1 score as the metric.

The Declarative format, based on the Schema-Guided Dialogue (SGD) dataset (Rastogi et al., 2020) and MultiWOZ dataset (Budzianowski et al., 2018), aims to capture the slots defined in dataset ontology. For the declarative format schema, we select MultiWOZ 2.1, SGD, and SpokenWOZ (Si et al., 2024) dataset and evaluate the dialogue state tracking task, using Joint Goal Accuracy (JGA) as the metric.

MultiWOZ2.1 is a fully-labeled collection of human-human written conversations spanning multiple domains and topics. It contains 7 domains, 35 slots, and over 10k dialogues.

SGD is another declarative format dataset containing over 16k multi-domain conversations spanning 16 domains with more slots and possible values compared to MultiWOZ.

SpokenWOZ is a new multi-modal spoken TOD dataset containing 8 domains, 5.7k dialogues, and 35 slots. It introduces the unique challenges in spoken conversation.

Besides the task-oriented dialogue, we also choose two datasets: **MELD** (Poria et al., 2018) and **MuTual** (Cui et al., 2020) from the Emotion Recognition in Conversations (ERC) task and response selection task, respectively. MELD contains over 10k utterances from the TV series Friends, and each utterance is annotated with emotion and sentiment labels. MuTual consists of 8k manually annotated dialogues based on Chinese students English listening comprehension exams. For both datasets, we use accuracy as the metric.

Model	Method	TOD				ERC	RS
		MultiWOZ2.1	STARv2	SGD	SpokenWOZ	MELD	MuTual
Llama2-7B -Chat	Vanilla	1.15	29.51	1.01	1.15	45.9	24.04
	Zero-shot CoT	1.87	26.4	1.09	0.71	46.4	25.66
	Plan-and-Solve	1.66	33.5	4.03	2.57	46.1	26.64
	Self-Explanation	**2.7**	**48.26**	**5.62**	**4.82**	**46.74**	**31.55**
Llama2-70B -Chat	Vanilla	11.28	51.24	8.76	3.68	58.83	**56.09**
	Zero-shot CoT	8.93	45.09	7.69	6.75	59.92	55.19
	Plan-and-Solve	10.19	50.6	8.06	**7.13**	59.33	54.74
	Self-Explanation	**12.1**	**56.35**	**11.18**	6.67	**60.98**	45.26
ChatGPT	Vanilla	35.93	51.88	18.96	13.75	59.14	68.97
	Zero-shot CoT	27.45	51.85	19.69	13.26	61.48	70.61
	Plan-and-Solve	38.33	56.74	21.11	14.5	58.38	69.77
	Self-Explanation	**44.44**	**63.66**	**21.81**	**14.89**	**61.71**	**71.58**
GPT4	Self-Explanation	50.97	70.27	25.75	25.94	63.51	91.87

Table 2: Comparing the performance of Vanilla, zero-shot CoT, Plan-and-Solve, and Self-Explanation prompting methods on six dialogue datasets using different models. For STARv2 and the rest of TOD datasets, we use the weighted F1 score and Joint Goal Accuracy(JGA), respectively. As for MELD and MuTual, the accuracy metric is applied.

Method	Model	TOD				ERC	RS
		MultiWOZ2.1	STARv2	SGD	SpokenWOZ	MELD	MuTual
Few-shot (Vanilla+4shots)	Llama2- 7B-Chat	16.99	31.59	5.95	4.81	45.7	32
	Llama2- 70B-Chat	29.1	50.4	5.81	7.7	58.23	56.21
	ChatGPT	40.38	52.39	17.34	14.13	55.09	**72.51**
Zero-shot (Self-Explanation)	ChatGPT	**44.44**	**63.66**	**21.81**	**14.89**	**61.71**	71.58

Table 3: Comparing the performance of Vanilla+4shots and Self-Explanation prompting methods on six dialogue datasets using different models.

3.1.2. Baselines

We compare our proposed zero-shot Explanation with two types of prompt baselines: Zero-shot baselines and Few-shot. For zero-shot baselines, we include Zero-shot-CoT (Kojima et al., 2022) and Plan-and-Solve Prompting (Wang et al., 2023). The former appends "Let's think step by step" to the prompt. The latter extends the Zero-shot CoT with a plan ahead and then carries out the plan. Besides the zero-shot baselines, we also evaluate the In-Context learning prompt performance on TOD task. Considering the sample of TOD task consists of a multi-turn dialogue and the slot list, we only use 4 examples that for not exceed the context window size. As for example selection, we randomly selected 4 examples with the same domain as the test sample.

3.1.3. Model

We use two types of LLM: Closed-source LLM ChatGPT and Open-source LLM LLAMA 2. For closed-source LLM, we choose the most widely-used LLMs ChatGPT with public APIs. We set the temperature to 0 for prompting without explanation and 1 for our self-explanation prompting for a variety of explanations. For easy comparison, we also incorporate the open-source LLM, LLAMA 2 (Touvron et al., 2023). We use the fine-tuned version LLAMA 2-7B-Chat and LLAMA 2-70B-Chat, which is optimized for dialogue use cases.

3.2. Main Results

Table 2 and 3 present the performance of our method compared to baseline approaches across six distinct datasets under zero-shot and few-shot settings. In the zero-shot scenario, our prompting method consistently surpasses the baselines

Method	Prompt	MultiWOZ 2.1 (JGA)
Vanilla	Answer the questions based on the above dialogue	35.93
Understand	Before you answer, **first understand the dialogue**, then answer the questions based on your understanding and original dialogue	36.52
Summary	Before you answer, **first summarize the dialogue**, then answer the questions based on your summary and original dialogue	40.98
Self-Explanation	Before you answer, first analyze the dialogue utterance by utterance, **give every utterance an explanation**. Then answer the questions based on your explanation	44.44

Table 4: The effect of different intermediate steps measured on MultiWOZ2.1 with GPT3.5-turbo.

on all evaluation datasets with open-source and close-source models, except llama2-70b on SpokenWOZ and MuTual. While CoT prompting does not enhance performance on TOD tasks. This significant improvement in all sizes of the model underscores the effectiveness and generalization of self-explanation prompting.

Among all six datasets, our method improves the most on the STARv2 dataset. The procedural task format aligns well with our prompting method. Fine-grained sentence-by-sentence explanations play a pivotal role in comprehending the dialogue flow and following the given schema. The enhanced performances on MultiWOZ, SGD, and SpokenWOZ further affirm that the dialogue state tracking task greatly benefits from self-explanation prompts. By providing explanations for each utterance, the likelihood of overlooking dialogue states is diminished. In addition to the task-oriented dialogue tasks, we assessed the impact of self-explanation prompting on both the ERC and RS tasks. However, the gains here were relatively modest in comparison to the TOD tasks. Given that our explanations are rooted in semantic interpretations, they may not be as beneficial for tasks centered on emotion recognition.

Compared to the few-shot baseline, our zero-shot prompting either outperforms or matches performance across all six datasets on all models we tested, except for the open-source model on MultiWOZ2.1 dataset. This factor indicates that a comprehensive understanding of dialogue is more critical than merely having a set of examples. The improvement of in-context learning is largely attributed to showing model input-label pairing formats and label space. For TOD tasks, the input usually consists of multi-turn dialogues encompassing various topics, necessitating a deep understanding of the dialogue. The intricate nature of TOD tasks demands a high level of comprehension, which merely a few examples fail to deliver.

3.3. Analysis

3.3.1. Effect of Intermediate steps

Drawing from psychological research, specifically (Chi et al., 1989), it's evident that not all explanations confer the same benefits. Factors like content, quality, and depth of explanations all have an impact on the final result. To assess the impact of different content of intermediate steps $\mathcal{I}$ on dialogue comprehension, we carried out a comparative study using ChatGPT on the MultiWOZ2.1 dataset, testing four distinct prompting methods. The results of these tests can be found in Table 4.

In the **Vanilla** method, no additional instruction is given before the model provides its response. There are no intermediate steps $\mathcal{I}$ to condition on. In the **Understand** method, the model is simply prompted with ”Understand the dialogue first” prior to answering. However, there's no specified format for the $\mathcal{I}$. With the **Summary** method, the model is prompted to first summarize the dialogue. It then bases its answer on both the summary and the original dialogue. In this method, the $\mathcal{I}$ is a coarse-grained summarization of the dialogue. Our observations revealed that when comparing the self-explanation method with Vanilla, there was a notable decline in performance. This suggests that pre-processing or understanding the dialogue is essential for optimal performance. Merely prompting the model to understand the dialogue without detailed instruction for $\mathcal{I}$ also resulted in reduced performance. This demonstrates the importance of precise comprehension guidelines. The Summary method explicitly directs the model to use the summary as a means of comprehension, subsequently answering based on that summary. This approach enhanced performance by approximately 5% JGA in comparison to the Vanilla method. However, summarizing is a relatively broad-strokes approach and might overlook finer details essential for the TOD task. Finally, our Self-Explanation prompting

Method	Prompt	MultiWOZ2.1 (JGA)
Commentary	Deconstruct the dialogue, giving an interpretative commentary on each sentence.	36.23
Interpret	Review the dialogue carefully and interpret each sentence within the conversation.	38.46
Insights	Go over the conversational sentences one by one, offering insights into their meanings.	39.17
Elucidation	Break down the conversation and furnish an elucidation for every individual sentence.	40.41
Self-Explanation	Analyze the dialogue utterance by utterance, give every utterance an explanation	**44.44**

Table 5: The effect of variant of trigger sentence on MultiWOZ2.1 with GPT3.5-turbo.

demonstrates superior performance, yielding an approximately 9% improvement in the JGA metric when compared to the Vanilla prompting method. This enhancement underscores the efficacy of our fine-grained, sentence-by-sentence explanation in enhancing the dialogue comprehension of LLMs.

To further understand the effects of different trigger sentences similar for self-explanation. We conducted a comparative analysis of the impact of variations or analogous trigger sentences on Self-Explanation using ChatGPT on the MultiWOZ2.1. Our investigation involved a of methods such as ”Commentary,” ”Interpret,” ”Insights,” ”Elucidation,” and ”Self-Explanation. The results are presented in Table 5. The Self-Explanation method exhibits superior performance on the MultiWOZ 2.1 dataset, achieving a JGA score of 44.44. In contrast, methods such as Commentary and Interpretation demonstrate more superficial analyses, yielding lower JGA scores of 36.23 and 38.46, respectively. Insights and Elucidation offer a more detailed breakdown of the dialogue but still fall short of the personalized engagement achieved by Self-Explanation, as evidenced by their JGA scores of 39.17 and 40.41.

3.3.2. Error Analysis

We conducted a comprehensive analysis of errors in three distinct categories: Hallucination, Missed Information (Missing info), and Mismatch. The Hallucination errors occur when the model generates additional dialogue states that are not present in the dialogue. These errors suggest an over-generation of information by the model. The Missed Information errors manifest when the model omits or fails to include dialogue states that are clearly specified within the dialogue. Such errors indicate a deficiency in the model's capacity to capture essential information. As for mismatch errors, they are observed when the dialogue states generated by the model do not align with the information specified in the dialogue.

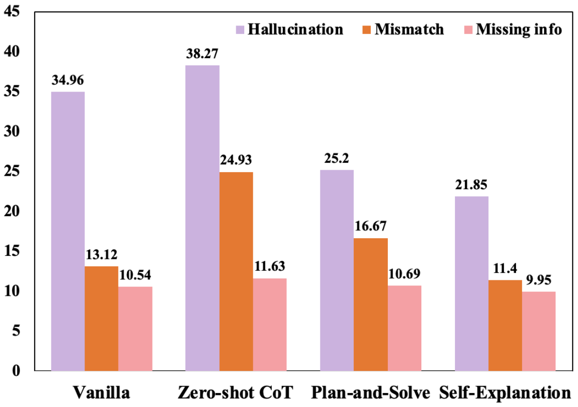


Figure 4: The error rate of three error types in MultiWOZ2.1 dataset with different prompting methods.

As shown in Figure 4, we demonstrate the effectiveness of our proposed prompting method in minimizing all three error types. Specifically, our method exhibits a substantial reduction in both hallucination and mismatch errors when compared to other prompting methods. This outcome underscores the significant contribution of our explanation-based approach in mitigating dialogue misunderstanding errors. Regarding miss information errors, the fine-grained sentence-by-sentence explanation helps ensure that the model does not overlook crucial information distributed across multiple turns of dialogue, which prevents the omission of essential details and thereby enhances the overall accuracy and completeness of the generated dialogue states.

3.3.3. Case Study

To have a straightforward understanding of how explanation affects task completion, We manually checked all the cases of the MultiWOZ2.1

Error Reason	Dialogue Content	Self-Explanation	Vanilla
Time involved	👤 I need to **get to** Michaelhouse cafe **by 12:45**.	taxi-arriveby: 12:45	taxi-leaveat: 12:45
Missing info.	👤 I am **leaving Cambridge** at 12:00 on Sunday, can you please tell me the travel time on that ride?	train-departure: cambridge	train-departure: None
Task unclear understand	👤 Please help me find the **attraction downing college**. 💻 Yes, it's on Regent Street **in the centre of town**. Would you like the phone number?	attraction-name: downing college	attraction-name: downing college attraction-area: centre

Table 6: The example of three typical error reasons in the MultiWOZ2.1 dataset where Self-Explanation gets the correct answer, and Vanilla gets the incorrect answer.

dataset where self-explanation led to correct answers, while the Vanilla approach resulted in incorrect responses. As shown in Table 6, we summarized three typical errors.

Broadly, these errors can be categorized into three primary reasons: temporal confusion, information omission, and task comprehension issues. For the first error reason, temporal confusion primarily manifests as a misunderstanding about the relation between times. For instance, consider a scenario where a user requests a taxi arrive at 12:45. In this case, the Vanilla prompting model incorrectly assigns 12:45 to the time of taxi departure. While sentence-level explanation rectified such temporal misinterpretations.

The second reason for error is missing information, which mostly happens in the long dialogue. The large amount of dialogue information may distract the model from correctly capturing all the information needed to complete the task. As the case of this error shows, the user expresses the place, time, and date of departure in a single sentence. The model output of vanilla prompting overlooks the place of departure, whereas the output of self-explanation prompting correctly captures all the information about the user request.

The last error reason is a task-specific error. In the dialogue state track task, the dialogue state should only include the information that the user explicitly requested while excluding the details that the system provides. In the illustrative example of the final type of error, the user explicitly requests an attraction called Downing College, to which the system subsequently provides relevant details. The model output of self-explanation prompting correctly distinguishes between the user's query and the system's response. Conversely, the output of Vanilla prompting mistakenly includes the system response within the dialogue state, leading to task failure.

3.3.4. Connection with CoT Prompting

We have explored self-explanation prompting as a simple way to enhance the understanding of multi-turn dialogue in large language models. In this section, we explore the concept of self-explanation prompting and its relationship with CoT prompting, shedding light on how these two techniques contribute to improving the performance of large language models.

From a broader perspective, OpenAI's documentation indicates that giving models a moment to ”think” is instrumental in improving their performance. This aligns with the human thinking process, where rushing to conclusions often results in errors. Referring to Section 2.1, the intermediate steps $\mathcal{I}$ can be viewed as the concrete expression of ”think”. Specifically, CoT prompting enforces a sequence of reasoning steps before accepting an answer, effectively granting models ”thinking” time. Similarly, our self-explanation prompting offers models a moment of reflection, but it steers them to interpret the intricate context, $\mathcal{C}$, as opposed to generating the reasoning paths.

From a micro perspective, CoT prompting guides the model toward a solution by gradually narrowing the scope of potential answers with multiple reasoning steps. In tasks requiring reasoning, the solution isn't straightforwardly derived from the $\mathcal{C}$. The response involves extensive calculations and transformations, heavily drawing on the model's innate reasoning ability. This suggests the solution space is largely related to the reasoning capabilities of the model. The intermediate reasoning steps elicited by CoT prompting put constraints on the solution space.

Conversely, in the TOD task, the query $\mathcal{Q}$ typically seeks details readily available within $\mathcal{C}$. Unlike reasoning assignments, these questions don't demand intricate computations. Thus, the solution space for TOD primarily resides within $\mathcal{C}$. Our self-explanation prompting method, designed to enhance dialogue comprehension, provides a different and more suitable dialogue-based tasks approach to narrowing down the solution space. Empowering the model to generate accurate responses by harnessing the information already present in the context.

In summary, while CoT prompting and self-explanation prompting differ in their specific objectives, they both share a fundamental goal of improving model performance by providing more time to think to narrow down the answer search space. CoT prompting guides models through complex reasoning paths, while self-explanation prompting encourages models to thoroughly grasp the context, ultimately leading to improving the performance on downstream tasks.

4. Related Work

Prompting Methods: The exploration of prompting methods for large language models has been vast. One of the conventional methods is in-context learning (ICL), as highlighted by GPT-3(Brown et al., 2020). In ICL, multiple demonstrations are provided before a test sample, and the model's performance significantly hinges on these demonstrations (Lu et al., 2021). Liu et al. (2021), endeavor to retrieve examples semantically similar to a test query sample, utilizing metrics like the L2 distance or cosine-similarity distance derived from sentence embeddings. In addition to these distance metrics, the concept of mutual information emerges as a potent example selection criterion (Sorensen et al., 2022). Here, the goal is to select a template that optimizes the mutual information between the input and the model's output. Taking this further, several studies, such as (Rubin et al., 2021), have shifted towards a supervised approach, training models to pick the most relevant demonstrations from a pool of candidates.

Reasoning Strategies: Beyond merely selecting examples, their format or ordering can significantly influence a model's performance. The Chain-of-Thought (CoT) strategy (Wei et al., 2022), a pioneering prompting approach designed to enhance the performance of large language models (LLMs) on intricate reasoning tasks. Unlike ICL, which relies on prepending input-output pairs, CoT integrates a sequence of intermediate reasoning steps into the demonstration, thereby amplifying the reasoning capabilities of LLMs. In order to empower the model planning capabilities□Tree of Thoughts(ToT)(Yao et al., 2024) was proposed to enhance LLM's capability for complex problem solving through tree search via a multi-round conversation. Recognizing the importance of diverse reasoning paths, the self-consistency strategy (Wang et al., 2022a) was introduced. It first creates multiple reasoning paths rather than just one and subsequently selects the most coherent answer by considering all the generated paths. Further automation in this domain is achieved with zero-shot CoT (Kojima et al., 2022). Instead of relying on human-annotated reasoning sequences, this method induces the model to generate reasoning steps by simply prompting it to "think step by step".

5. Conclusion

In this paper, we find CoT prompting is suboptimal for multi-turn dialogue tasks. To enhance the comprehension of LLM, we propose a new zero-shot prompting strategy called Self-Explanation prompting, which guides the LLM to first understand the multi-turn dialogue by explaining every utterance and then completing the task based on dialogue with its explanation. Extensive experiments show that explanation prompting can boost the LLMs contextual understanding of multi-turn dialogue and significantly outperform or perform on par with the previous zero-shot and few-shot baselines.

Acknowledgments

Min Yang was supported by National Key Research and Development Program of China (2022YFF0902100), National Natural Science Foundation of China (62376262), Shenzhen Science and Technology Innovation Program (KQTD20190929172835662), Shenzhen Basic Research Foundation (JCYJ20210324115614039). This work was supported by Alibaba Group through Alibaba Innovative Research Program.

Bibliographical References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. Llms as counterfactual explanation modules: Can chatgpt explain black-box text classifiers? *arXiv preprint arXiv:2309.13340*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Yucheng Cai, Wentao Ma, Yuchuan Wu, Shuzheng Si, Yuan Shao, Zhijian Ou, and Yongbin Li. 2023. Unipcm: Universal pre-trained conversation model with task-aware automatic prompt. *arXiv preprint arXiv:2309.11065*.

Hyungjoo Chae, Yongho Song, Kai Tzu-iunn Ong, Taeyoon Kwon, Minjin Kim, Youngjae Yu, Dongha Lee, Dongyeop Kang, and Jinyoung Yeo. 2023. Dialogue chain-of-thought distillation for commonsense-aware conversational agents. *arXiv preprint arXiv:2310.09343*.

Michelene TH Chi, Miriam Bassok, Matthew W Lewis, Peter Reimann, and Robert Glaser. 1989. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science*, 13(2):145–182.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. *arXiv preprint arXiv:2004.04494*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Haoyu Gao, Rui Wang, Ting-En Lin, Yuchuan Wu, Min Yang, Fei Huang, and Yongbin Li. 2023. Unsupervised dialogue topic segmentation with topic-aware contrastive learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2481–2485.

Wanwei He, Yinpei Dai, Binyuan Hui, Min Yang, Zheng Cao, Jianbo Dong, Fei Huang, Luo Si, and Yongbin Li. 2022a. Space-2: Tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 553–569.

Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022b. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 187–200.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, and Yongbin Li. 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*, pages 10749–10757.

Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-Chin Lin, Carel van Niekerk, and Milica Gašić. 2023. Chatgpt for zero-shot dialogue state tracking: A solution or an opportunity? *arXiv preprint arXiv:2306.01386*.

Guimin Hu, Ting-En Lin, Yi Zhao, Guangming Lu, Yuchuan Wu, and Yongbin Li. 2022a. Unimse: Towards unified multimodal sentiment analysis and emotion recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7837–7851.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. 2022b. In-context learning for few-shot dialogue state tracking. *arXiv preprint arXiv:2203.08568*.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, pages 90–93.

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

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.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*.

Zaijing Li, Ting-En Lin, Yuchuan Wu, Meng Liu, Fengxiao Tang, Ming Zhao, and Yongbin Li. 2023. Unisa: Unified generative framework for sentiment analysis. *arXiv preprint arXiv:2309.01339*.

Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3299–3308.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Amogh Mannekote, Mehmet Celepkolu, Joseph B Wiggins, and Kristy Elizabeth Boyer. 2023. Exploring usability issues in instruction-based and schema-based authoring of task-oriented dialogue agents. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, pages 1–6.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*.

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Yushan Qian, Bo Wang, Ting-En Lin, Yinhe Zheng, Ying Zhu, Dongming Zhao, Yuexian Hou, Yuchuan Wu, and Yongbin Li. 2023. Empathetic response generation via emotion cause transition graph. *arXiv:2302.11787*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2024. Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. *Advances in Neural Information Processing Systems*, 36.

Shuzheng Si, Shuang Zeng, and Baobao Chang. 2023. Mining clues from incomplete utterance: A query-enhanced network for incomplete utterance rewriting. *arXiv preprint arXiv:2307.00866*.

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022b. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Tianshu Yu, Haoyu Gao, Ting-En Lin, Min Yang, Yuchuan Wu, Wentao Ma, Chao Wang, Fei Huang, and Yongbin Li. 2023. Speech-text pre-training for spoken dialog understanding with explicit cross-modal alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7900–7913.

Eric Zelikman, Jesse Mu, Noah D Goodman, and Yuhuai Tony Wu. 2022. Star: Self-taught reasoner bootstrapping reasoning with reasoning.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. 2023. Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*.

Jeffrey Zhao, Yuan Cao, Raghav Gupta, Harrison Lee, Abhinav Rastogi, Mingqiu Wang, Hagen Soltau, Izhak Shafran, and Yonghui Wu. 2022. Anytod: A programmable task-oriented dialog system. *arXiv preprint arXiv:2212.09939*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.