ProTOD: Proactive Task-oriented Dialogue System Based on Large Language Model

Wenjie Dong[1], **Sirong Chen**[3], **Yan Yang**[12*]
[1]School of Computer Science and Technology, East China Normal University, China
[2]NPPA Key Laboratory of Publishing Integration Development, China
[3]School of Computing and Communications, Lancaster University, UK
Correspondence: yanyang@cs.ecnu.edu.cn

Abstract

Large Language Model (LLM)-based Task-Oriented Dialogue (TOD) systems show promising performance in helping users achieve specific goals in a zero-shot setting. However, existing systems engage with users in a reactive manner, relying on a basic single-query mechanism with the knowledge base and employing passive policy planning. The proactive TOD systems, which can provide potentially helpful information and plan cross-domain multi-task dialogue policies, have not been well studied. In addition, effective evaluation methods are also lacking. To address these issues, we propose ProTOD, a novel LLM-based proactive TOD framework designed to improve system proactivity and goal completion. First, we design an adaptive exploratory retrieval mechanism to dynamically navigate domain knowledge. Second, we introduce a two-stage passive-to-proactive policy planner that effectively organizes knowledge and actions relationship. Finally, we develop two distinct user simulators with different personalities to simulate real-world interactions and propose a new error measure called Human-targeted Policy Edit Rate (HPER) for evaluation. Experimental results show that ProTOD achieves state-of-the-art (SOTA) performance, improving goal completion rates by 10% while significantly enhancing the proactive engagement.[1]

1 Introduction

Task-oriented dialogue (TOD) systems are designed to assist users accomplish specific tasks (Wei et al., 2018; Valizadeh et al., 2023), such as querying flight tickets and hotel reservation (Rastogi et al., 2020). Traditional TOD systems are built mainly using a modularized pipeline, including natural language understanding, dialogue state tracking (Lee et al., 2019; Wu et al., 2019), dialogue policy planning (Takanobu et al., 2019), and natural language generation (Zhang et al., 2020). Later, end-to-end TOD systems (Sun et al., 2023a) integrate all the necessary functionalities for dialogue into a single model. Despite their great success, both systems require a large amount of annotated data. Recently, large language models (LLMs), such as ChatGPT[2] and GPT-4 (OpenAI, 2023), have revolutionized natural language processing with exceptional conversational skills, instruction-following abilities, and zero-shot generalization capabilities.

Figure 1: A demonstration of proactive and non-proactive systems in three typical TOD scenarios. The proactive system conducts precise information processing (a) to ensure informative responses and adopt proactive actions when appropriate, such as offering relevant information (a), requesting cross-domain service (b), and asking clarification questions (c).

Although existing LLM-based systems effectively help the user achieve goals (Xu et al., 2024; Zhang et al., 2023), the proactivity of TOD remains

[*]Corresponding author.
[1]https://github.com/melonhh/ProTOD
[2]https://chat.openai.com/

Proceedings of the 31st International Conference on Computational Linguistics, pages 9147–9164
January 19–24, 2025. ©2025 Association for Computational Linguistics

underexplored. As illustrated in Figure 1, a non-proactive system passively responds to user request, relying on single-turn retrieval via belief states or generated SQL queries, often struggles to offer useful alternative information or suggestions in failure scenarios (i.e., no match with the user needs). In the proactive version, the system offers potentially useful domain information through multi-turn precise retrieval by adjusting query conditions. Provides cross-domain services that may be interdependent and verifies information with the user to prevent misunderstandings. This proactive approach not only enhances the user experience, but also improves goal completion efficiency.

However, building LLM-based proactive TOD systems faces three main challenges: (1) Limited retrieval mechanism. Existing systems rely on single-turn retrieval methods, which do not offer comprehensive or exploratory options in failure scenarios (Qin et al., 2023). In addition, they struggle to handle complex user information needs. (2) Simple dialogue policy planning. Most LLM-based TOD systems primarily leverage the inherent conversational skills of the language model to generate response (Rohmatillah et al., 2023). However, proactive dialogue requires well-defined policies that can effectively plan and manage diverse external knowledge and handle task dependencies across multiple domain tasks within a dialogue. (3) Inadequate evaluation methods for proactive TOD. Current evaluation methodologies typically employ LLM-based user simulators to assess goal achievement (Sun et al., 2023b), but these simulators tend to be more cooperative and patient than real people, which has a significant impact on goal achievement and can lead to biased evaluation results. In addition, existing evaluation metrics make it difficult to fully evaluate the system's proactivity.

To address the above issues, in this paper, we propose a LLM-based framework (ProTOD), which is designed to improve the abilities of goal completion and proactive engagement in multi-domain task-oriented dialogues. Specifically, we first employ a dialogue state tracker in a chain-of-thought manner to extract and update the dialogue state, ensuring that the system has a clear and updated understanding of the context and the user requirement. Next, we design an exploratory retrieval mechanism for knowledge retriever equipped with a set of tools, offering comprehensive information and explorable options to users through proactively exploring domain knowledge under multi-turn plan-execution reasoning. Furthermore, we propose a dependency-enhanced policy planner that guides LLM through a passive to proactive two-stage planning process to effectively organize knowledge and actions relationships. Finally, to better evaluate the performance of ProTOD, we present two types of user simulators with different personalities for comparison: proactive and non-proactive. Additionally, to quantitatively evaluate the performance of policy planning, we propose a new error measure called the Human-targeted Policy Edit Rate (HPER). Experiments on the MultiWOZ and SGD datasets show that ProTOD significantly improves the effectiveness of goal completion and proactive engagement.

Our main contributions are summarized as follows:

- We propose a LLM-based proactive task-oriented dialogue system, comprising a dialogue state tracker, an exploratory knowledge retriever, and a two-stage dialogue policy planner, designed to enhance goal completion and proactive engagement in multi-domain task-oriented dialogues.
- We propose a novel exploratory retrieval mechanism that utilizes a chain-of-plan-execution retrieval strategy to adaptively explore domain knowledge with a suit of retrieval tools.
- We introduce a dependency-enhanced policy planner that manages sequential dependencies of actions for complex knowledge and cross-domain tasks in dialogue flows using a two-stage reactive-to-proactive planning method.
- Experiments results show that our method achieves a state-of-the-art (SOTA) performance with a 10% increase in goal completion.

2 Methodology

2.1 The Overall Framework

The framework of the ProTOD system is depicted in Figure 2. We decompose ProTOD into three key components: (1) **Dialogue State Tracker** is responsible for extracting and updating the state of the dialogue at each turn, ensuring that the system has a clear and updated understanding of the context and user needs; (2) **Knowledge Retriever** that utilizes a chain-of-plan-execution retrieval strategy to adaptively explore domain knowledge with a suit of retrieval tools. (3) **Policy Planner** that delivers dialogue policy and system response through

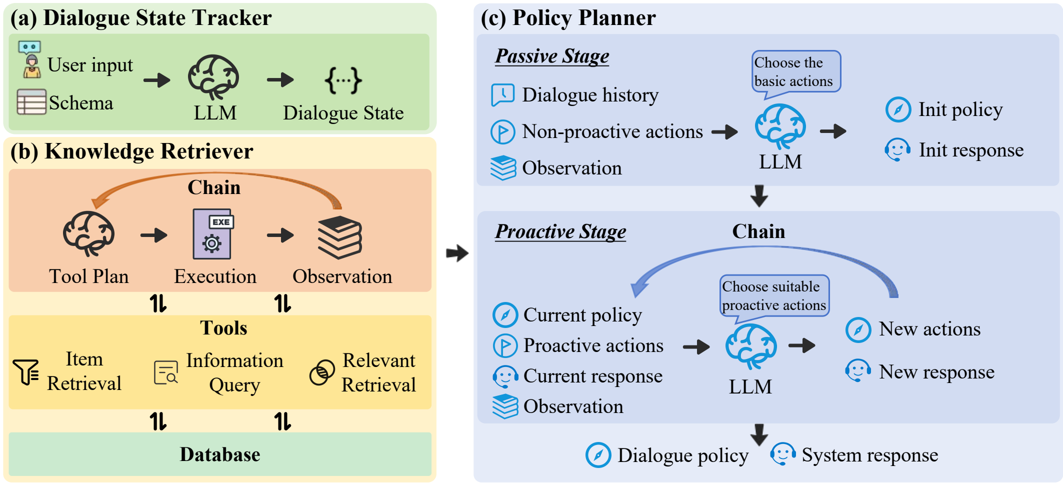

Figure 2: ProTOD framework. (a) Dialogue state tracker; (b) Knowledge retriever, consisting of a suit of tools, the chain-of-plan-execution strategy and domain database; (c) Policy planner, involving a two-stage planning process that transitions from passive to proactive.

a two-stage policy planning scheme that effectively organizes the knowledge and actions relationship from passive to proactive stage.

2.2 Dialogue State Tracker

During turn t, the Dialogue State Tracker accepts the user current input u_t, dialogue context C_{t-1}, and history belief state B_{t-1} to guide the LLM (parameter θ) in predicting the belief states B_t,

$$B_t = LLM_\theta(u_t, C_{t-1}, B_{t-1}) \tag{1}$$

which is a list of triplets recording values for slots in a particular domain: $(domain, slot, value)$. Since we need to handle conversations spanning multiple domains, we prompt the LLM to perform state tracking in a chain-of-thought manner: first, to detect the active domain-intent pairs, next to output the new state under each domain. We then use the outputs to update the accumulated global belief states.

2.3 Knowledge Retriever

Ensure that the system can handle various user inquiries and retrieve potentially useful information in case of failures (i.e., nothing matches with user needs), we use a chain-of-plan-execution strategy to adaptively explore domain knowledge using following tools:

(1) Item Retrieval. Item retrieval tool aims to propose a list item candidates that satisfy user demand from the entire item pool. We utilize a Structured Query Language (SQL) tool to filter candidates from the item database.

(2) Information Query. During the conversation, the system not only guides the user to complete tasks but also frequently addresses users' inquiries. For example, within the train domain, users may ask, *"What time is the earliest train to Cambridge on Saturday?"* To satisfy such inquiries, we design an information query tool that uses SQL expressions to retrieve detailed information from a backend database.

(3) Relevant Retrieval. To enhance user experience, it is essential for the system to proactively provide additional relevant information in case of failures to meet the core needs of the user. The tool will adjust query constraints based on previous query results and then generate new SQL statements. Based on the relevant information, we can respond to the user as follows: *"Unfortunately, there are no guesthouses that meet your needs. But we have 2 hotels if you are interested."*.

By incorporating these tools, LLMs can effectively handle a wide range of user inquiries. For instance, as shown in the Figure 1, the system sequentially invokes tool execution three times, adjusting the input according to the previous execution results. Inspired by ReAct (Yao et al., 2023) We utilize a chain-of-plan-execution retrieval strategy to adaptively explore external databases with these tools , where LLMs generate reasoning traces, actions, and observations in an interleaved manner. Our plan-execution process consists of the following two phases:

- **Plan**: At the k-th step, given the user input u_t, dialogue context C_{t-1}, belief state B_t, and descriptions of various tools F, LLM formulates the next tool f_k and the input i_k for that tool.

- **Execution**: The output feedback of each tool f_k is defined as $o_k = exec(f_k)$.

The plan-execution trajectory $\mathbf{K}_t = \{(f_1, o_1), \cdots, (f_n, o_n)\}$ serves as Knowledge Retriever's observation to support the next step of policy planning. Since item retrieval is not necessary at every turn, to maintain dialogue coherence, we use the most recently retrieved item information as the background knowledge.

2.4 Proactive Policy Planner

In multi-domain TOD, a dialogue policy is typically composed of a list of dialogue actions *(domain, action_type, slot_value_pairs)*. We find that the action combinations in the proactive policy not only depend on domain knowledge, dialogue context, and query result processing, but also on the dependencies between these actions. For example, as shown in the Figure 1, the action "*(restaurant, RequestSelect, food)*" depends on the action "*(restaurant, OfferRelevant, food)*" and the relevant domain knowledge, while the action "*(taxi, RequestCrossDomain)*" depends on the action "*(restaurant, Inform, book)*". Previous works have neglected the modeling of such action dependency relationships.

Therefore, the key to enhancing the proactivity of the conversation lies in organizing the knowledge and behavioral relationships. In our study, as depicted in the Figure 2 (c), we divide policy planning into two stages, passive and proactive, so that the system can take the reasonable combination of proactive actions based on domain knowledge.

In the first stage, the passive stage, the system adopts a necessary set of actions to respond to the current user needs, such as greeting, informing the results when the user is inquiring, and asking for essential information for booking when the user is making a reservation.

$$P_t^0, s_t^0 = LLM_\theta(u_t, C_{t-1}, B_{t-1}, K_t) \qquad (2)$$

where P_t^0 refers to the generated initial policy, and s_t^0 refers to the corresponding system response in turn t.

In the second stage, the proactive stage, the system augments the initial policy with reasonable proactive actions, such as collaboratively providing additional information, asking clarification questions, and offering cross-domain services, etc. In order to simulate the dependency between actions, we perceive the augmentation as a multi-step decision-making process. At each step i, the system assesses the necessity of augmenting the existing policy with additional dialogue actions from a predefined set A. If an enhancement is required, generate supplementary policy and corresponding response.

$$P_t^i, s_t^i = LLM_\theta(u_t, K_t, P_t^{i-1}, s_t^{i-1}, A^i) \qquad (3)$$

The final system response s_t , is derived from the output of the last augmentation step.

You are a non-proactive user chatting with an Assistant to complete some tasks. You should carefully read and understand the User Goals below, then talk with the Assistant and gradually express the intents in the goals.
Important:
1. If the Assistant makes a mistake, do not correct it.
2. Do not provide new demand unless the Assistant asks.
3. If the information you want to know has been provided in the conversation history, do not ask again.
4. The expression of your needs should follow the order provided by the User Goals, and avoid expressing too much at once.
5. You are simulating a user, not the Assistant.
6. Do not provide information or ask questions outside of the User Goals.
7. End the conversation with "<END>" when the dialogue goal are completed or the Assistant does not actively asks or assists.
User Goals: {user_goal}
The conversation you have completed so far: {history}

Figure 3: Prompt for non-proactive user simulator. Texts in bracket represent the placeholders for variables.

3 Experiments

3.1 Experimental Setup

Evaluation Strategies. To quantitatively assess ProTOD, we designed the following two evaluation strategies:

(1) End-to-end Evaluation. Depending on the dialogue flow, models for generating responses can be categorized into two groups: the end-to-end setting and the policy-optimization setting. End-to-end models use only the dialogue context as input to generate responses, while policy-optimization models use also the ground-truth dialog states to generate response. We evaluate the end-to-end dialogue generation performance.

(2) User Simulator. Thanks to the powerful understanding and generation abilities of current LLMs, the user simulator can be built only by an instruction-following language model. We manually tune a role-playing prompt to facilitate GPT-4

Model	Inform	Success	BLEU	Comb	CBE	#Uni	#Tri
SimpleTOD (Hosseini-Asl et al., 2020)	84.4	70.1	15.0	92.3	-	-	-
UBAR (Yang et al., 2020)	83.4	70.3	17.6	94.5	2.10	478	5238
GALAXY (He et al., 2022)	85.4	75.7	19.6	100.2	1.75	295	2275
Mars (Sun et al., 2023a)	88.9	78.0	**19.9**	**103.4**	1.65	288	2264
TOATOD (Bang et al., 2023)	90.0	79.8	17.04	101.94	-	-	-
SGP-TOD (Zhang et al., 2023)	83.9	69.9	9.1	86.0	-	-	-
AutoTOD (Xu et al., 2024)	87.2	82.8	9.3	94.3	2.62	1722	10188
ProTOD	**91.7**	**83.3**	8.9	96.4	**3.26**	**1951**	**14345**

Table 1: End-to-end evaluation results on MultiWOZ 2.0. All the models are evaluated by only feeding with dialogue history. Comb, CBE, #Uni and #Tri stand for Combine, Conditional bigram Entropy, the number of unigrams and the number of tri-grams.

in emulating real-world users with varying goals. The simulated user engages with TOD system to complete all the goals. However, we observe that simulated users tend to be more collaborative and patient than actual human users. This discrepancy can lead to biased evaluation metrics results. Therefore, we design two different types of user simulators for comparison: proactive, and non-proactive. The prompt of non-proactive user simulator is shown in the Figure 3.

Dataset. We experiment with two of the currently most prominent benchmark datasets for multi-domain task-oriented dialogue:

- MultiWOZ (Budzianowski et al., 2018) is a well-known dataset of human-human conversations, spanning seven domains: restaurant, attraction, train, hotel, taxi, police, and hospital. To the best of our knowledge, most of prior work on policy and response generation has evaluated on MultiWOZ 2.0, so we take the 2.0 version for wide baseline models.

- SGD (Rastogi et al., 2020) is a schema-guided TOD dataset spanning over 26 services. Database interaction is considered in the dataset, but no real database is provided. Therefore, we implement the DB API using the database results provided in the schema and we collect the user actions in each dialogue to form the user goals for user simulator.

Baseline. We evaluate the zero-shot performance of the proposed ProTOD by comparing it with two groups of methods: full-shot training models and zero-shot prompting methods. The full-shot training methods are SimpleTOD (Hosseini-Asl et al., 2020), UBAR (Yang et al., 2020), GALAXY (He et al., 2022), Mars (Sun et al., 2023a), and TOATOD (Bang et al., 2023). The two prompting methods: SGP-TOD (Zhang et al., 2023) is a schema-guided prompting method that builds TOD systems effortlessly using LLMs, AutoTOD (Xu et al., 2024) gives up the traditional pipelined modular components and requires only a language model that follows instructions so that tasks can be processed with a simple instruction pattern and external APIs.

Metrics. In the end-to-end evaluation strategy, we evaluate the dialog generation performance using the same metrics as those listed in (Budzianowski et al., 2018): *Inform* measures whether the system offers the right entity for users. *Success* measures whether the system is able to answer all attributes requested by users. *BLEU* (Papineni et al., 2002) measures the word overlap of the generated response against the reference in the corpus. *Combined* judges the overall quality, which is computed as $(BLEU + 0.5 * (Inform + Success))$. In addition, we use some language diversity metrics to evaluate the quality of dialogue responses: number of unique output words (*#Uni*) and trigrams (*#Tri*), and bigram conditional entropy (*CBE*). In the user simulator evaluation strategy, we measure whether the system makes the reservation successfully for the user using *Book*, which is a particular metric defined in Xu et al. (2024). And the *Combined* is computed as $(0.5 * Inform + 0.25 * (Success + Book))$.

We conduct a multidimensional human evaluation to supplement the limitations of automated metrics at three levels: turn, dialogue, and policy. Specifically, we evaluate understandability (Und), context relevance (*Rel*), fluency (*Flu*), and reasonability (*Rea*) at the turn level. For dialogue level evaluation, we assess coherence (*Coh*), informativeness (*Inf*), usefulness (*Use*), and proactiv-

Model	Proactive Simulator				Non-Proactive Simulator			
	Inform	**Success**	**Book**	**Comb**	**Inform**	**Success**	**Book**	**Comb**
TOATOD (Bang et al., 2023)	41.8	34.4	-	29.5	28.4	26.0	-	20.7
AutoTOD (Xu et al., 2024)	80.3	65.2	81.4	76.6	61.5	50.7	55.2	57.2
ProTOD	**89.5**	**80.4**	**87.0**	**86.6**	**85.7**	**76.5**	**82.6**	**82.6**
- w/o KR	80.7	69.5	82.6	78.4	67.5	61.2	72.8	67.3
- w/o Policy	84.6	74.0	83.2	81.7	62.0	55.2	69.4	62.2

Table 2: Evaluation results with user simulator on MultiWOZ 2.0. "w/o KR" denotes the removal of Knowledge Retriever, using the dialogue state to directly match the database. "w/o Policy" refers to removing the two-stage policy planner and adopting a single-stage approach instead.

ity (*Pro*). Regarding policy level, the subjective nature of proactive policy planning means there is often no single correct answer. Inspired by a widely used machine-translation metric named HTER (Human-targeted Translation Edit Rate) (Snover et al., 2006), a human-in-the-loop evaluation that involves a procedure for creating targeted references, we propose a new error measure called Human-targeted Policy Edit Rate (HPER). In simple terms, HPER refers to the necessary edit (add, remove, change) rate required to modify an output policy into a reasonable one. Specifically:

$$HPER = \frac{\#\ of\ edits}{\#\ of\ actions\ and\ slots} \tag{4}$$

Implementation Details. We employ GPT-3.5 (gpt-3.5-turbo-0125) as the fixed LLM to build ProTOD. The model used for the user simulator is GPT-4 (gpt-4-0613). We use OpenAI API for using the OpenAI series models.The greedy decoding strategy is used for all models. The SQL executor is implemented with SQLite integrated in pandasql[3]. Details on dialogue actions are provided in the Appendix A.1.

3.2 Results on MultiWOZ

End-to-end Evaluation. The evaluations results of end-to-end strategy on MultiWOZ 2.0 are shown in Table 1. ProTOD shows competitive performance in both Inform rate and Success rate compared to all baseline methods, demonstrating the robust language understanding and task completion capabilities of ProTOD. This excellent performance can be partially credited to our exploratory retrieval mechanism. In the end-to-end setting, TOD systems are only fed with golden dialogue history, meaning the dialogue policy has no impact on the subsequent conversation flow. Thus, the performance gap between our approach and baseline models lies in the retrieval mechanism while they employ a single-turn retrieval method. The improvement of the Inform rate further demonstrates the advantage of our retrieval mechanism in delivering more precise entity information. In terms of language diversity, we can see that ProTOD gains the highest scores across all three diversity metrics, which also suggests our exploratory retrieval mechanism is effective in helping the system generate more informative responses. All prompting methods underperform on the BLEU metric, this is expected since they are not exposed to any grounding utterances in the dataset.

	Turn-Level			
	Und	**Rea**	**Rel**	**Flu**
TOATOD	1.55	1.60	1.50	1.65
AutoTOD	1.75	1.73	1.78	1.85
ProTOD	**1.82**	**1.75**	**1.79**	**1.86**
- w/o KR	1.80	1.70	1.72	1.80
- w/o Policy	1.79	1.68	1.70	1.85
	Dialogue-Level			
	Coh	**Inf**	**Use**	**Pro**
TOATOD	1.55	1.45	1.40	1.48
AutoTOD	1.70	1.69	1.72	1.68
ProTOD	**1.78**	**1.82**	**1.79**	**1.81**
- w/o KR	1.75	1.70	1.72	1.74
- w/o Policy	1.72	1.75	1.60	1.70

Table 3: Human evaluation results at turn-level and dialogue-level on MultiWOZ dialogues. Each aspect with a range of [0, 2].

Evaluation with User Simulator. We evaluate the goal achievement of models in multi-turn conversations with two types of user simulators: proactive and non-proactive. The evaluation results are shown in Table 2. We observe that LLM-based models significantly outperform the full-shot training model. This indicates, on the one hand, that previous TOD systems cannot work well in real

[3]https://github.com/yhat/pandasql/

dialogue scenarios, and on the other hand, it highlights the necessity of user simulator evaluation. Additionally, ProTOD and its ablation variants outperform baseline methods across all metrics, particularly in Success rate, under both simulators, showcasing ProTOD's strong goal completion abilities. Notably, ProTOD experiences less performance degradation with the non-proactive user simulator, demonstrating the excellent robustness and abilities of proactive policy planning. The ablation study shows that the removal of the exploratory retrieval mechanism and the policy planning significantly impact the model's performance, particularly under the non-proactive user simulator. This further suggests that our exploratory retrieval mechanism, policy planning, and their collaboration are essential for goal completion and proactive engagement.

3.3 Human Evaluation

Turn-level and Dialogue-level. We randomly sample 100 dialogues from the test set and ask for 3 graduate students to rate from turn and dialogue levels, and edit the policy when it is deemed unreasonable. One full-shot baseline model (TOATOD), one zero-shot baseline model (AutoTOD) and three ProTOD variants are taken into account. The evaluation results at the turn level and the dialogue level are shown in Table 3. Across both levels, ProTOD achieves comparable results compared to baseline models. Compared to AutoTOD, ProTOD shows minimal differences in the four turn-level metrics, but it significantly outperforms AutoTOD in the four dialogue-level metrics, especially in the informativeness score and the proactivity score.

Policy-level. Evaluation results at policy level are shown in Table 4, providing insights into the advantages of using our two-stage policy planning and exploratory retrieval mechanism. Although removal of the knowledge retriever only slightly increases the error rate (HPER increases by 0.5%), it significantly reduces the overall proactivity of the model, with the frequency of all proactive actions declining, especially in the actions of *InformAddition* and *NoOfferRelevant*. The removal of policy planning appears to introduce significantly more errors on some actions like *RequestCrossDomian* and *RequestVerify*, both of which rely on the domain information and the relationships of actions. This further confirms the appropriateness of employing a two-stage approach in our policy planning, as it efficiently organizes domain knowledge and manages action dependencies.

3.4 Results on SGD

The evaluation results for the SGD dataset are presented in Table 5. We can see that ProTOD outperforms the baseline models in terms of both goal completion and language diversity. Notably, in the ablation experiments, ProTOD seems to be more sensitive to the removal of the knowledge retriever than the removal of policy planner, which is not observed in MultiWOZ. This is because the difference in user goal formats between the two datasets. The user goals in MultiWOZ comprise of natural language instructions, while SGD uses a structured format. With structured user goals, the use simulator tends to present multiple requests at once, increasing the complexity of retrieval. This further underscores the strengths of our exploratory retrieval mechanism.

4 Related Work

4.1 Task-oriented Dialogue System

Task-oriented dialogue (TOD) systems have been studied for decades. Traditional TOD systems rely on a pipeline architecture, with components such as natural language understanding, dialogue state tracking, dialogue policy, and natural language generation being optimized independently (Zhang et al., 2020). Although pipeline approach makes it easier to manage and update individual components, it suffers from limitations such as poor adaptability to new dialogue scenarios and weak fault tolerance. In response, end-to-end TOD systems were developed (Yang et al., 2020; Hosseini-Asl et al., 2020), combining the entire pipeline into a single model for joint training (He et al., 2022; (Sun et al., 2023a)), which improves optimization and response quality. Recently, LLMs have been applied to TOD systems (Hudecek and Dusek, 2023), enhancing their ability to handle diverse contexts and improving fault tolerance and adaptability (Zhao et al., 2023). Research in this area (Zhang et al., 2023; Xu et al., 2024) highlights the increasing flexibility and performance of LLM-based approaches, marking a significant advancement in TOD systems. In this paper, we focus on the proactivity of TOD systems, improving user experience and goal completion efficiency through exploratory retrieval and proactive dialogue policy planning.

4.2 Proactive Dialogue

Recent advancements in conversational system design have focused on enhancing proactivity to ad-

Model	HPER(%)	Frequency(%)/ER(%)				
		Inform Addition	Request Select	Request Verify	Request CrossDomain	NoOffer Relevant
ProTOD	**15.8**	14.4/**10.2**	6.8/4.0	2.4/**5.2**	6.9/7.4	6.0/**2.0**
- w/o KR	16.3	13.2/11.5	4.1/**3.5**	2.0/6.8	6.1/**7.2**	0.1/100
- w/o Policy	26.6	13.3/11.0	3.9/3.8	5.0/39.5	5.8/27.0	1.1/30.5

Table 4: Human evaluation results at policy level on MultiWOZ 2.0. Frequency and ER represent the frequency and edit rate for each type of action, respectively.

Model	Inform	Success	CBE	#Uni	#Tri
SimpleTOD	12.7	9.8	2.01	573	3011
AutoTOD	45.1	23.0	2.81	1792	12263
ProTOD	**50.4**	**24.9**	**3.26**	**2021**	**15149**
- w/o KR	35.5	20.2	2.88	1720	12100
- w/o Policy	46.0	22.4	3.02	1904	14003

Table 5: Evaluation results with non-proactive user simulator on SGD.

dress challenges across three primary dialogue types (Deng et al., 2023a): In open-domain dialogues, proactive systems are designed to lead conversations, with approaches such as target-guided dialogues (Tang et al., 2019; Wang et al., 2023), where the system directs discussions towards specific topics (Tang et al., 2019), and prosocial dialogues (Kim et al., 2022; Chen et al., 2023), which involve guiding conversations constructively in response to problematic user behavior. In task-oriented dialogues, two types of proactivity are emphasized: non-collaborative dialogues (Li et al., 2020), where the system and user may have conflicting goals, and enriched dialogues (Balaraman and Magnini, 2020; Yan et al., 2023), where the system provides supplementary information not explicitly requested. For information-seeking dialogues, proactivity is crucial in asking clarification questions (Aliannejadi et al., 2019) and eliciting user preferences (Zhang et al., 2018), both of which enhance system performance. To trigger the proactivity of LLMs, several attempts have been made on chain-of-thought prompting scheme (Deng et al., 2023b; He et al., 2024). Differing from their methods, we model both the dependencies of actions on knowledge and the interrelationships between actions through a novel two-stage planning scheme to enhance the effectiveness of policy planning.

5 Conclusion

In this paper, we propose ProTOD, a LLM-based framework, consisting of a dialogue state tracker, an exploratory knowledge retriever, and a two-stage policy planner to enhance goal completion and proactivity in multi-domain dialogues. Our system features a novel exploratory retrieval mechanism that uses a chain-of-plan-execution strategy to adaptively explore domain knowledge, and a dependency-enhanced two-stage policy planner that improves proactivity. To better evaluate proactive dialogues, we propose a new error measure, Human-targeted Policy Edit Rate. Experiments on MultiWOZ and SGD datasets demonstrate significant improvements in goal completion and proactive engagement with ProTOD. In the future, we plan to generalize our method to other tasks and domains.

6 Limitations

Although ProTOD shows significant advancements in goal completion and proactive engagement, it still exhibits certain limitations that warrant future improvements. (1) ProTOD has only been implemented with one LLM (GPT-3.5) due to the API cost. But a broader comparison with other well-known LLMs like GPT-4, Claude (Anthropic, 2023), and PaLM (Chowdhery et al., 2023) is essential. (2) Our user simulators, although designed to mimic different personalities, may not fully capture the complexity and unpredictability of real human interactions (Luo et al., 2024). Further studies involving user-centered evaluation (Abolghasemi et al., 2024) are necessary to validate the system's effectiveness in practical applications. (3) Proactivity is a crucial property in intelligent conversation. However, without thoughtful design, proactive systems risk being perceived as intrusive by human users (Deng et al., 2024). Therefore, we need to consider the ethical and social impacts of the agent, not just its technical capabilities.

References

Amin Abolghasemi, Zhaochun Ren, Arian Askari, Mohammad Aliannejadi, Maarten Rijke, and Suzan Verberne. 2024. CAUSE: Counterfactual assessment of user satisfaction estimation in task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14623–14635, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 475–484. ACM.

Anthropic. 2023. Introducing claude. Accessed: 2024-06-06.

Vevake Balaraman and Bernardo Magnini. 2020. Proactive systems and influenceable users: Simulating proactivity in task-oriented dialogues. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue-Full Papers, Virually at Brandeis, Waltham, New Jersey, July. SEMDIAL.*

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369, Toronto, Canada. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Maximillian Chen, Xiao Yu, Weiyan Shi, Urvi Awasthi, and Zhou Yu. 2023. Controllable mixed-initiative dialogue generation through prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 951–966, Toronto, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023a. A survey on proactive dialogue systems: Problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 6583–6591. ijcai.org.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023b. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10602–10621, Singapore. Association for Computational Linguistics.

Yang Deng, Lizi Liao, Zhonghua Zheng, Grace Hui Yang, and Tat-Seng Chua. 2024. Towards human-centered proactive conversational agents. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 807–818, New York, NY, USA. Association for Computing Machinery.

Tao He, Lizi Liao, Yixin Cao, Yuanxing Liu, Ming Liu, Zerui Chen, and Bing Qin. 2024. Planning like human: A dual-process framework for dialogue planning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4768–4791, Bangkok, Thailand. Association for Computational Linguistics.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. GALAXY: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10749–10757. AAAI Press.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

Vojtech Hudecek and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2023, Prague, Czechia, September 11 - 15, 2023*, pages 216–228. Association for Computational Linguistics.

Hyunwoo Kim, Youngjae Yu, Liwei Jiang, Ximing Lu, Daniel Khashabi, Gunhee Kim, Yejin Choi, and Maarten Sap. 2022. ProsocialDialog: A prosocial backbone for conversational agents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4005–4029, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Yu Li, Kun Qian, Weiyan Shi, and Zhou Yu. 2020. End-to-end trainable non-collaborative dialog system. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8293–8302. AAAI Press.

Xiang Luo, Zhiwen Tang, Jin Wang, and Xuejie Zhang. 2024. DuetSim: Building user simulator with dual large language models for task-oriented dialogues. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5414–5424, Torino, Italia. ELRA and ICCL.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Libo Qin, Wenbo Pan, Qiguang Chen, Lizi Liao, Zhou Yu, Yue Zhang, Wanxiang Che, and Min Li. 2023. End-to-end task-oriented dialogue: A survey of tasks, methods, and future directions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5925–5941, Singapore. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.

Mahdin Rohmatillah, Jen-Tzung Chien, et al. 2023. Advances and challenges in multi-domain task-oriented dialogue policy optimization. *APSIPA Transactions on Signal and Information Processing*, 12(1).

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023a. Mars: Modeling context & state representations with contrastive learning for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11139–11160, Toronto, Canada. Association for Computational Linguistics.

Weiwei Sun, Shuyu Guo, Shuo Zhang, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2023b. Metaphorical user simulators for evaluating task-oriented dialogue systems. *ACM Transactions on Information Systems*, 42(1):1–29.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5624–5634, Florence, Italy. Association for Computational Linguistics.

Mina Valizadeh, Xing Qian, Pardis Ranjbar-Noiey, Cornelia Caragea, and Natalie Parde. 2023. What clued the AI doctor in? on the influence of data source and quality for transformer-based medical self-disclosure detection. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1201–1216, Dubrovnik, Croatia. Association for Computational Linguistics.

Jian Wang, Yi Cheng, Dongding Lin, Chak Leong, and Wenjie Li. 2023. Target-oriented proactive dialogue systems with personalization: Problem formulation and dataset curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1143, Singapore. Association for Computational Linguistics.

Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuanjing Huang, Kam-fai Wong, and Xiangying Dai. 2018. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–207, Melbourne, Australia. Association for Computational Linguistics.

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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand. Association for Computational Linguistics.

Sitong Yan, Shengli Song, Jingyang Li, Shiqi Meng, and Guangneng Hu. 2023. TITAN : Task-oriented dialogues with mixed-initiative interactions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 5251–5259. ijcai.org.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2020. UBAR: towards fully end-to-end task-oriented dialog systems with GPT-2. *CoRR*, abs/2012.03539.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023. SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13348–13369, Singapore. Association for Computational Linguistics.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 177–186. ACM.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Jeffrey Zhao, Yuan Cao, Raghav Gupta, Harrison Lee, Abhinav Rastogi, Mingqiu Wang, Hagen Soltau, Izhak Shafran, and Yonghui Wu. 2023. AnyTOD: A programmable task-oriented dialog system. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16189–16204, Singapore. Association for Computational Linguistics.

A Appendix

A.1 Dialogue Action

Based on the coarse-grained dialogue actions in MultiWOZ dataset, TITAN (Yan et al., 2023)redesigned five proactive actions *RequestSelect*, *RequestVerfify*, *InformAddition*, *NoOfferRelevant*, *RequestCrossDomain* and two non-proactive dialogue actions *RequestSpecify*, *InformSpecific*. We find the seven dialogue actions to be well-suited for most multi-domain task-oriented dialogue tasks. These dialogue actions have been applied in our action space, and all actions' definitions are provided in the Figure 8 and Figure 9.

A.2 Discussion

A.2.1 Ablation Study

In Table 6, we study the impact of the three components of ProTOD (Dialogue State Tracker, Knowledge Retriever, Policy Planner) on MultiWOZ 2.0 under non-proactive simulator. Removing any of these components leads to consistent declines in all evaluation metrics, indicating each of the three components contributes significantly to the overall performance of ProTOD. Specifically, the performance degradation caused by the removal of dialogue state tracker is greater than that caused by the removal of knowledge retriever, suggesting that the performance of knowledge retriever is more dependent on the belief state provided by dialogue state tracker. Similarly, the absence of the Policy Planner leads to a significant decline, showing its crucial role under the proactive dialogue setting.

Model	Inform	Success	Book	Combine
ProTOD	**85.7**	**76.5**	**82.6**	**82.6**
- w/o DST	63.1	49.3	60.1	58.9
- w/o KR	67.5	61.2	72.8	67.3
- w/o Policy	62.0	55.2	69.4	62.2
- w/o KR & Policy	51.0	42.5	50.1	48.65

Table 6: Ablation study on the impact of the three components in the proposed ProTOD on MultiWOZ 2.0 under non-proactive user simulator.

A.2.2 Case Study

We present a dialogue fragment between ProTOD and the proactive user simulator in Table 7. The user simulator is fed with the user goals and engages in dialogue with ProTOD, achieving the goals step by step with the assistance and guidance of ProTOD.

A.3 Prompt

In this section, we will share our prompts used in different components.

A.3.1 Proactive User Simulator

The prompt to instruct LLM to play as a proactive user is illustrated in Figure 6.

A.3.2 Dialogue State Tracker

As show in the Figure 4, the proposed Dialogue State Tracker contains four parts: 1. a task instruction that offers general guidance on belief state prediction; 2. belief instruction of all domains; 3.formatting example illustrating the anticipated output format to direct the LLM. 4.test input, i.e., the given dialog history.

A.3.3 Knowledge Retriever

We implement tree distinct tools: information query tool, item retrieval tool, and relevant retrieval tool in knowledge retriever. The description of information query tool, item retrieval tool, relevant retrieval tool are illustrated in Figure 5. As described in Section 3.3, we generate tool plan through plan-execution reasoning. The prompt for knowledge retriever is illustrated in Figure 7.

A.3.4 Policy Planner

As described in Section 3.4. The policy planner comprises two stages: passive stage and proactive stage. The prompt of passive stage and proactive stage are illustrated in Figure 8 and Figure 9 respectively.

Goals	1. You are looking for places to go in town. The attraction should be in the west and should be in the type of cinema; 2. If there is no such attraction, how about one that is in the type of museum; 3. Make sure you get phone number, postcode, and address; 4. You are also looking for a place to stay. The hotel should include free wifi and should be in the south; 5. The hotel should be in the type of guesthouse;
User	Can you tell me if there are any cinemas in the west part of town?
ProTOD	**DST**: Active domains: ["attraction-find"] Dialogue state: {"type": "cinema", "area": "west"} **KR**: Item(SELECT * FROM attraction WHERE type='cinema' AND area='west';) **Tool Result**: None **KR**: Relevant(remove: area, SELECT * FROM attraction WHERE type='cinema') **Tool Result**: [list of 2 attractions] **Stage 1**: Policy: {"attraction-NoOffer": {"area": "west"}} **Stage 2**: Policy: {"attraction-NoOfferRelevant": {"type": "cinema"}", "reqmore": {}} **Response**: I am sorry, there are no cinemas in the west part of town. However, here are some cinemas in other areas: Cineworld Cinema in the south part of town and Vue Cinema in the center. Would you like more information on these options?
User	That's unfortunate. How about museums? Are there any in the west part of town?
ProTOD	**DST**: Active domains: ["attraction-find"] Dialogue state: {"type": "museum", "west"} **KR**: Item(SELECT * FROM attraction WHERE type='museum' AND area='west';) **Tool Result**: [list of 7 attractions] **Stage 1**: Policy: {"attraction-InformSpecific": {"type": "museum", "area": "west"}} **Stage 2**: Policy: {"attraction-InformAddition": "choice": "7", "attraction-RequestSelect": {}"} **Response**: Yes, there are museums in the west part of town. There are 7 museums in that area. Would you like more details on any of them?
User	Yes, please give me one of them. I would like to know the phone number, postcode, and address.
ProTOD	**DST**: Active domains: ["attraction-find"] Dialogue state: {"type": "museum", "west"} **KR**: Information(SELECT address, phone, postcode FROM attraction WHERE name='Cafe Jello Gallery';) **Tool Result**: {"address": "13 Magdalene Street", "phone", "1223312112", "postcode": "cb30af"} **Stage 1**: Policy: {"attraction-InformSpecific": {"name": "Cafe Jello Gallery", "address": "13 Magdalene Street", "phone": "1223312112", "postcode": "cb30af"}} **Stage 2**: Policy: {"reqmore": {}} **Response**: You'll find Cafe Jello Gallery at 13 Magdalene Street, cb30af, or contact them at 1223312112. Is there anything else you would like to know?
User	Can you help me find a guesthouse in the south part of town that includes free wifi?
ProTOD	**DST**: Active domains: ["hotel-find"] Dialogue state: "area": "south", "internet": "yes", "type": "guesthouse" **KR**: Item(SELECT * FROM hotel WHERE area='south' AND internet='yes' AND type='guesthouse';) **Tool Result**: [list of 3 hotels] **Stage 1**: Policy: {"hotel-InformSpecific": {"name": "Bridge Guest House"}} **Stage 2**: Policy: {"hotel-InformAddition": {"pricerange": "moderate", "stars": "4"}, "book-RequestCrossDomain": "} **Response**: Sure, the Bridge Guest House meets your requirements. It is a 4-star guest house with moderate price. Would you like me to help you make a reservation?

Table 7: A interactive example between a user simulator and ProTOD.

As a dialogue state tracker, your task is to keep track of the dialogue state throughout the conversation. The dialogue state represents the user's goal and the necessary information extracted from their inputs. The system should accurately extract and update the dialogue state to reflect the current context of the conversation. This includes tracking the domain-intent pairs and extracting relevant information.

The background area of the conversation is Cambridge. The table below shows the intents defined for each domain:

Domain	Intents
restaurant	find, book
attraction	find
hotel	find, book
taxi	book
train	find, book

The following slots are tracked in the dialogue state(Not modify the schema of each domain!):

```
"hotel": {
   "book": {
      "booked": [],
      "day": "day of the hotel booking. Related values: [monday, tuesday, wednesday, thursday, friday, saturday, sunday]",
      "people": "number of people for the hotel booking",
      "stay": "length of stay at the hotel"
   },
   "semi": {
      "name": "name of the hotel",
      "area": "area or place of the hotel. Related values: [centre, east, north, south, west]",
      "parking": "whether the hotel has parking. Related values: [no, yes]",
      "pricerange": "price budget of the hotel. Related values: [cheap, moderate, expensive]",
      "stars": "star rating of the hotel. Related values: [0, 2, 3, 4]",
      "internet": "whether the hotel has internet. Related values: [no, yes]",
      "type": "what is the type of the hotel. Related values: [guesthouse, hotel]"
   }
}
```

Task Logic
To complete this task, you need to perform two steps of reasoning:
First, determine the list of domain-intent pairs involved in the current dialogue turn;
Second, extract the dialogue state corresponding to those domains;

Output format you must follow:
Active domains: ["domain-intent"]
Dialogue state:
```
{
   "domain": {
      "book" {
         "booked": []
         "slot1": "value1"
      },
      "semi": {
         "slot2": "value2"
         "slot3": "value3"
      }
   }
}
```
Example
{example}

{test}

Figure 4: Prompt for Dialogue State Tracker. Schema in certain domains have been omitted due to space limitations.

Tool Name: Item Retrieval Tool
```
The tool is a filtering tool. The tool is useful when user want {domain}s with some conditions on {domain} properties.
The input of the tool should be a one-line SQL SELECT command converted from hard conditions. Here are some rules:
1. always use pattern match logic for columns with string type;
2. only one {domain} information table is allowed to appear in SQL command;
3. select all {domain}s that meet the conditions, do not use the LIMIT keyword;
4. use given related values for categorical columns instead of user's description.
```
Tool Name: Information Query Tool
```
The tool is used to look up {domain}'s detailed information in a {domain} information table (including statistical information), like number, address, phone and so on. \
The input of the tools should be a SQL command (in one line) converted from the search query, which would be used to search information in {domain} information table. \
You should try to select as less columns as you can to get the necessary information.
```
Tool Name: Relevant Retrieval Tool
```
The tool slightly modify the query conditions to provide related information that closely meet the user's core requirements when exact match is not available in the database.
The input of the tool should be a new one-line SQL modified from previous SQLs. Here are some rules:
1. reduce one query condition to retrieve more information;
2. change the query condition when cannot be reduced..
```

Figure 5: Description of tools.

You are a collaborative and patient user interacting with an Assistant to complete some tasks. You should carefully read and understand the User Goals below, then talk with the AI Assistant and gradually express the intents in the goals. Your purpose is to achieve the goals as much as possible.

Note that the Assistant is not perfect. It may make various mistakes, including ignoring the user's requests, executing the wrong instructions, forgetting early conversation content, etc. The user you play should remind him to correct when you find that the AI assistant made a mistake, and complete the task as much as possible.

Important:
1. The expression of your needs should follow the order provided by the User Goal, and avoid expressing too much at once.
2. You are simulating the User, not the Assistant.
3. Do not provide information or ask questions outside of the User Goals.
4. End the conversation with "<END>" when you achieved the goals.

User Goals: {user_goal}

The conversation you have completed so far: {history}

Figure 6: Prompt for proactive user simulator.

```
As an intelligent knowledge retrieval agent, your task is to fetch relevant information from predefined knowledge base
through communicating with external tools based on user inquiry and current dialogue state.

You have access to the following tools:
{tools}

If the user's intention is to make a reservation, no query tools are needed.
If the user is looking up information of some item, such as address, phone and so on, you should take the InformationQuery.
If the user provides conditions for filtering, you should take the ItemRetrieval.
If the previous query yielded no results, you should take the RelevantRetriveal to retrieval relevant information that close
to user's core requirements.

All SQL commands are used to search in the following information tables:
{table_info}

First you need to think whether to use tools. If no, use the format to output:
###
Question: Do I need to use tools to process user's input?
Thought: No, I do not need to use tools because I can answer based on the cached query;
###

If use tools, use the format:
###
Question: Do I need to use tools to process user's input?
Thought: Yes, since .., I need to take ..
Action: tool_name
Action Input: the input to tool
Observation: the result of tool execution.
###

Current user input: {input}

Tool execution trajectory and query cache: {query_cache}

Let's think step by step. Begin!
```

Figure 7: Prompt for Knowledge Retriever.

```
Your task is to decide on the best course of action based on the current dialogue state, dialogue history. Use the information tracked to determine the most appropriate action and generate a appropriate response.

# Basic system dialog actions

Actions need to take slots:
1.NoOffer: inform the user of the situation and the reason when their needs cannot be met;
2.InformSpecific: provide the explicit answer to the user's direct request or booking details;
3.RequestSpecify: ask the user for specific information, such as reservation details or requirements.

Actions cannot take slots:
1.welcome: acknowledge the user at the beginning of the interaction;
2.bye: end the interaction and say goodbye to the user;
3.greet: greet the user in a friendly manner;
4.thank: express gratitude towards the user for their input or cooperation;

There are Three types of values:
1) If a slot takes a binary value, e.g., 'internet' or 'parking', the value is either 'yes' or 'no'.
2) If a slot is under the act 'RequestSpecify', e.g., 'RequestSpecify' about 'area', the value is expressed as '?'.
3) The value that appears in the utterance e.g., the name of a restaurant.

# Instruction
1. Ensure actions are reasonable and avoid conflicts
2. Ensure not more than one questions in your response.
3. Generate concise and appropriate responses based on the selected dialogue actions.

# Examples
{example}

> Dialogue history
{history}

> Active domains
{active_domains}

> Dialogue state
{state}

> Knowledge retrieval observation
{query_cache}

> User input
Human: {input}

Use the format to output:
###
Question: What basic actions should I take to respond to the user?
Thought: Beacuse..., I have to...
Action: {"domain1-act1": {"slot1": "value1", "slot2": "value2"}, "domain2-act2": {"slot3": "value3", "slot4": "value4"}}
Response: [Generated response based on the actions]
###
```

Figure 8: Prompt for the passive stage of Policy Planner.

```
Your task is to choose appropriate proactive actions to supplement the Assistant's response.

## Proactive dialog actions as follows:
1.InformAddition: provide additional, implicit information that could be helpful to the user but was not explicitly requested;
2.OfferRelevant: provide relevant information to the user when their needs cannot be met;
3.RequestSelect: ask the user to make a selection if there are several options that meet the user's need;
4.RequestSpecify: ask the user for a specific preference or booking information;
5.RequestVerify: ask clarification question when copying similar slot across different domains or the user's request is unclear;
6.RequestCrossDomain: offer cross-domain services that might suit the user when current goal is completed.
7.reqmore: ask if further assistance is needed.

There are Three types of values:
1) If a slot takes a binary value, e.g., 'internet' or 'parking', the value is either 'yes' or 'no'.
2) If a slot is under the act 'RequestSpecify', e.g., 'RequestSpecify' about 'area', the value is expressed as '?'.
3) The value that appears in the utterance e.g., the name of a restaurant.

# Instruction
1. Ensure actions are reasonable and avoid conflicts
2. Ensure not more than one questions in your response.
3. Generate concise and appropriate responses based on the selected dialogue actions.

## Examples
{examples}

> Dialogue history
{history}

> Active domains
{active_domains}

> Dialogue state
{state}

> Knowledge retrieval observation
{query_cache}

> User input
User: {input}

> Assistant response
Assistant: {response}

Use the format to output:
###
Question: Is it still necessary to supplement additional proactive actions to the existing response?
Thought: Yes or No. If yes, analyze which action would be helpful from the perspectives of information quality and user experience.
Action: {"domain1-act1": {"slot1": "value1", "slot2": "value2"}}
Response: [Generate final Assistant Response]
###
```

Figure 9: Prompt for the proactive stage of Policy Planner.