KAPA: A Deliberative Agent Framework with Tree-Structured Knowledge Base for Multi-Domain User Intent Understanding

Jiakai Tang[♣♢♠*], **Shiqi Shen**[♡], **Zhipeng Wang**[♡], **Zhi Gong**[♡],
Xueyang Feng[♣♢♠], **Zexu Sun**[♣♢♠], **Haoran Tan**[♣♢♠], **Xu Chen** [♣♢♠†],
[♣]Gaoling School of Artificial, Renmin University of China, Beijing, China
[♢]Beijing Key Laboratory of Research on Large Models and Intelligent Governance
[♠]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE
[♡]Wechat, Tencent, Beijing, China
{tangjiakai5704, xu.chen}@ruc.edu.cn

Abstract

Dialogue assistants have become ubiquitous in modern applications, fundamentally reshaping human daily communication patterns and information access behaviors. In real-world conversational interactions, however, user queries are often volatile, ambiguous, and diverse, making it difficult accurately and efficiently grasp the user's underlying intentions. To address this challenge, we propose a simple yet effective deliberative agent framework that leverages human thought process to build high-level domain knowledge. To further achieve efficient knowledge accumulation and retrieval, we design a tree-structured knowledge base to store refined experience and data. Moreover, we construct a new benchmark, User-Intent-Understanding (UIU), which covers multi-domain, multi-tone, and sequential multi-turn personalized user queries. Extensive experiments demonstrate the effectiveness of our proposed method across multi-step evaluations.

1 Introduction

With the rapid development of Large Language Models (LLMs), the collaboration between LLMs and users for completing complex tasks is becoming more and more prevailing (He et al., 2025; Li et al., 2024; Feng et al., 2024). A major challenge of such collaboration lies in the accurate understanding of user intents, which are usually quite volatile, ambiguous, and diverse in real-world applications. To better understand user intents, people have proposed many effective strategies. For example, ProCoT (Deng et al., 2023b) uses a Chain-of-Thought (CoT) (Wei et al., 2022) approach to design its prompting scheme, applying dynamic reasoning and planning to decide the next step (*i.e.*, answering or asking clarification questions). Qian et al. (2024) propose to fine-tune the open-source LLMs using simulated dialogue data from user-assistant interactions, aiming to enhance their proactive interaction ability.

While the above methods have achieved many promising results, there are still several inherent limitations: (1) to begin with, existing studies usually leverage simple prompting strategies like CoT (Deng et al., 2023b) or construct labeled samples to train LLMs (Qian et al., 2024) for estimating user intents, which are either too far from real human thought processes or requiring too much computational resources and unexplainable to the users. (2) Secondly, some studies only focus on a single domain to build their conversation assistant. However, in practice, the user intents can be influenced by factors outside their domains. For example, given a user query like "I want advice on pursuing data science as a career", the assistant should not only have expertise in career planning but also a solid understanding of the technical field of data science. (3) In addition, existing works organize the external documentation into an unstructured list, which is less effective in retrieving related knowledge.

To solve the above challenges, in this paper, we design a deliberative agent framework, which is inspired by the human thought process including four stages, that is, perception, analysis, reflection, and experience accumulation. In specific, through a process analysis of the specific reasons behind an agent's success or failure in reasoning and actions during simulated user-assistant dialogues, our framework iteratively explores and reflects, ultimately distilling high-level insights into a knowledge base. To better incorporate and retrieve multi-domain knowledge, we equip the above agent framework with a tree-structured knowledge base. Specifically, we narrow the domain scope in a progressive and dynamic manner, transitioning from broad to refined levels to efficiently organize and re-

---

[*] This work was done while Jiakai Tang was visiting Tencent.

[†] Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 6150–6166
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

trieve query-related knowledge. Furthermore, considering that there are few public datasets available for multi-domain, multi-tone, sequential multi-turn user intent clarification tasks, we create a benchmark dataset for User-Intent-Understanding, called UIU, designed to evaluate our framework and support the advancement of research in this area.

In a summary, the main contributions of this paper are as follows: (1) We design a deliberative agent framework with tree-structured knowledge base to better estimate user intents under multi-domain settings. (2) We release a user-assistant interaction dataset, UIU, which can assess language model's capability on understanding user intents within complex multi-domain environments. (3) We conduct extensive experiments to evaluate the effectiveness of our proposed framework.

2 Related Works

This paper focuses on enhancing the proactive capabilities of LLM-based agents in human collaboration and addressing their upstream task in dialog systems, namely clarifying ambiguous user queries. Accordingly, we review related research on proactive agent and user intent understanding in the following sections.

2.1 Proactive Agent

Compared to traditional reactive agents (Ahn et al., 2022; Shen et al., 2024; Wang et al., 2023), proactive agents can autonomously decide when and how to act to better assist users (Deng et al., 2023a, 2025; Zhang et al., 2025), instead of directly responding to user's requests. Recently, proactive AI systems have attracted widespread attention across various fields. For example, Oh et al. (2024) introduce a proactive voice assistant for smart home scenario, guiding users to express dynamic preferences. Du et al. (2024) propose an in-vehicle intelligent assistant, empowering LLMs to automatically adjust vehicle controls to fulfill driver's potential needs. Another research direction focuses on developing proactive dialogue systems, such as task-oriented dialogue (Tang et al., 2019; Yang et al., 2022), prosocial dialogue (Ziems et al., 2022; Kim et al., 2022), non-collaborative dialogue (Jin et al., 2024; He et al., 2018), and user preference elicitation (Shang et al., 2023; Zhang et al., 2024b). In this paper, we mainly address proactive clarification task in the open-domain dialogue system.

2.2 User Intent Understanding

The goal of intent understanding is to predict whether user's query is ambiguous and, if so, how to proactively initiate clarifying questions for seeking task-related valuable information from user. This process ensures subsequent task execution is supported by precise and sufficient information, mitigating overly broad or erroneous assumptions about the current task in final responses. Most existing works adopt prompt engineering or fine-tuning methods to enhance the LLM's understanding ability. In prompt-based paradigm, CLAM (Kuhn et al., 2022) proposes domain-specific few-shot prompt mechanisms to guide models in generating accurate selective clarification responses. Zhang and Choi (2023) design a self-questioning approach that utilizes uncertainty modeling to decide when to proactively ask questions. ProCoT (Wei et al., 2022) introduces a proactive Chain-of-Thought (CoT) strategy within prompts to enhance LLMs' autonomous questioning capabilities. For tuning-based paradigm, recent studies (Qian et al., 2024; Andukuri et al., 2024; Sekulić et al., 2022; Chen et al., 2024c) utilize Supervised Fine-Tuning (SFT) through the next-token prediction loss with constructed agent action trajectories or dialogue data. For example, Qian et al. (2024) uses two GPT-4o models to build a User-Agent and an Assistant-Agent for automated dialogue generation, and then trains a LLM using the conversation logs to better comprehensive user intent. Additionally, reinforcement learning (RL) framework has been explored to optimize the proactive agent. For example, Zhang et al. (2024a) and Chen et al. (2024b) use different strategies to construct positive and negative response pairs for model tuning using Direct Preference Optimization (DPO) (Rafailov et al., 2024). The former is limited to double-turn conversations, while the latter designs a quasi-online DPO algorithm to extend to multi-turn dialogues.

3 Method

Previous works on intent understanding mainly follow either compute-intensive, non-transparent model fine-tuning or a simple pure prompt strategy that sacrifices accuracy, resulting an efficiency-effectiveness trade-off that remains unresolved in open-domain intent clarification scenarios. Furthermore, limiting intent estimation to single-domain settings is obviously inadequate for addressing the complex demands of real-world user requirements.

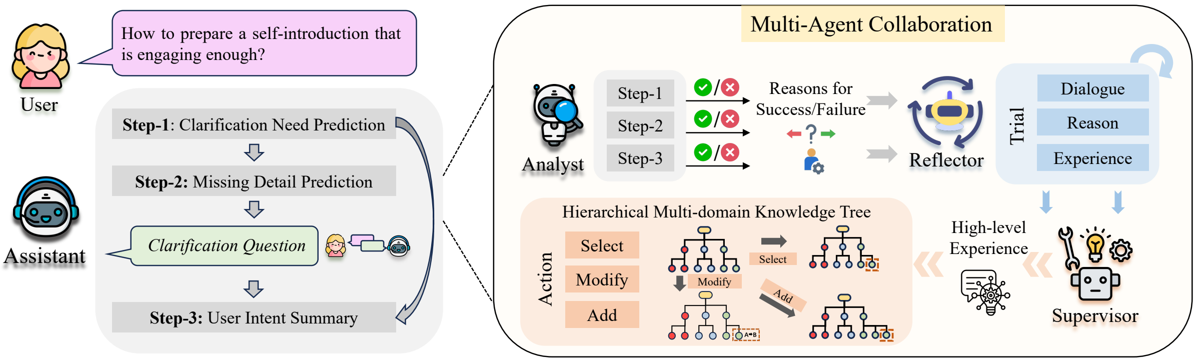

User
How to prepare a self-introduction that is engaging enough?
Assistant
Step-1: Clarification Need Prediction
Step-2: Missing Detail Prediction
Clarification Question
Step-3: User Intent Summary
Multi-Agent Collaboration
Analyst
Step-1
Step-2
Step-3
Reasons for Success/Failure
Reflector
Trial
Dialogue
Reason
Experience
Hierarchical Multi-domain Knowledge Tree
Action
Select
Modify
Add
Select
Modify
Add
High-level Experience
Supervisor

Figure 1: The overview of the proposed method.

To this end, we propose a multi-domain **K**nowledge-**A**ugmented **P**roactive **A**gent (**KAPA**, for short). Our core idea is to build a general multi-domain knowledge base that acts as an external extension for LLM-based agents, enabling efficient and cost-effective skill enhancement. The key challenge lies in how to acquire and integrate this knowledge so that the assistant can ask fewer clarification questions and think more comprehensively during user interactions, ultimately boosting both efficiency and intelligence in the agentic dialogue system. To address the above issues, we propose a deliberative agent framework that progressively extracts multi-domain knowledge through human-like cognitive processes, while maintaining and accessing this knowledge via a tree-structured hierarchical base.

3.1 Deliberative Agent Framework

Human knowledge can be naturally divided into explicit and tacit dimensions (Smith, 2001). While *explicit knowledge* is easily formalized and described in language, *tacit knowledge* is often more important but elusive, as it is deeply rooted in human actions and contexts and acquired through critical thinking. Inspired by (Schön, 2017; Revans, 1982), we enable the agent to engage in experiential learning within simulated authentic contexts. To be specific, by perceiving feedback from the environment (**Perception**), the agent can autonomously analyze the outcomes of its actions (**Analysis**) and iteratively refine its extracted experiences (**Reflection**). This process gradually accumulates high-level domain knowledge base (**Experience Accumulation**), thereby transforming tacit knowledge into explicit knowledge.

To realize the above four-stage process—"*perception* → *analysis* → *reflection* → *experience accumulation*"—we design a deliberate multi-agent collaborative framework, as shown in Fig. 1, with the function of each agent outlined as follows:

- **User:** Take on the role of initiating clear or vague query, while supplying necessary information to the assistant. We provide the hidden required information and the dialogue history to guide the user in reasonably answering clarification questions. We further randomize the user's speaking style (*e.g.*, succinct or enthusiastic) to better reflect real-world interactions.
- **Assistant:** The assistant is responsible for understanding and clarifying the user's implicit intent, performing the following three steps: (1) clarification need prediction, (2) missing detail prediction (if the current given query is deemed vague), and (3) user intent summary. For further details, please refer to Sec. 4.2.
- **Analyst:** By using the task-related metadata, the analyst agent evaluates whether each step carried out by the assistant achieves the expected goal and further analyzes the specific reasons behind any successes or failures.
- **Reflector:** Based on the detailed reasons provided by the analyst agent and the behavior trajectories in assistant-user interactions, the reflector can extract practical and high-level insights or lessons. In cases of repeated errors of the same type, the reflector agent should further refine earlier experience so that the assistant, guided by these adjustments, can more effectively avoid similar mistakes in the future.
- **Supervisor:** After receiving the insights distilled by the reflector, the supervisor agent is responsible for storing these experience and the corresponding dialogue instance in the knowledge base (refer to Sec. 3.2.1 for more details).

Next, we introduce that how the complete deliberate collection framework works. Specifically, in the interconnected four-stage procedure, the user and assistant agent engage in multi-turn dialogue. Initially, after the user initiates a query, the assistant needs to complete a predefined three-step action sequence before fulfilling the user's requirement (**Action**). Following this, the analyst evaluates the success of the action by determining whether the assistant accurately identified ambiguities in the user's query, fully predicted missing and inferable details by integrating the dialogue history, or comprehensively summarized the user's implicit intent at the corresponding step. After that, the analyst investigates the reasons behind the success or failure (**Analysis**) and relays this feedback to the reflector agent, which extracts more general insights from previous attempts (**Reflection**). In specific, if the assistant succeeds or reaches the maximum attempt times, these experiences are forwarded to the supervisor agent to update the knowledge base (**Knowledge Storage**). Otherwise, the refined lessons are filled into the assistant's prompt in the next trial, up to a maximum of three tries.

Under the collaborative efforts of multi-agents, we can establish a multi-domain general knowledge base to serve the assistant in understanding user intents across various areas. Compared to experts manually drafting multi-domain guiding rules (**Know What**), our deliberative learning process can adaptively discern correct behavioral principles in specific contexts (**Know How**), while leveraging the LLM's role-play and self-reflection abilities to explicate and enhance tacit knowledge. In the following section, we further discuss how the supervisor agent maintains this knowledge base.

3.2 Tree-Structured Knowledge Base

Instead of inefficiently organizing external data in an unstructured manner, we propose building a tree-structured knowledge base to hierarchically accumulate the collected experiences under the management of the supervisor agent. In the following, we introduce how to efficiently store and retrieve multi-domain knowledge in the base.

3.2.1 Knowledge Base Construction

To provide the assistant with multi-domain knowledge, we desgin to construct a **Hierarchical Multi-domain Knowledge Tree (HKTree)**. Specifically, in this hierarchical structure, domain ranges are refined progressively from top to bottom, with the top layer containing predefined domain names that cover broad areas such as sports, education, and career. When the supervisor agent receives new experiences, it firstly determines which top-level domain the current query falls under. For the lower-level tree nodes, the supervisor can select one of the following three actions:

- **[Choose]:** The query aligns with an existing subdomain and can be directly assigned to the corresponding tree node.
- **[Modify]:** A subdomain label requires minor adjustments to better cover the current dialogue topic (*e.g.*, renaming "AI industry analysis" to "market analysis and trends").
- **[Add]:** If the query does not clearly match any existing subdomain, a new subdomain node is created, and the data is assigned accordingly.

Through the above operations, the supervisor automatically achieves **dynamic clustering** of multi-domain knowledge. Besides, each node maintains a node representation $\mathbf{e}_n$, computed as follows:

$$\mathbf{e}_n = \frac{1}{|\Phi_n|} \sum_{(q,tht)\in\Phi_n} \text{Emb}(q \oplus tht),$$

where q and tht represent the user's query and the assistant's initial thought, respectively. Emb() denotes the BGE-M3 (Chen et al., 2024a) embedding function; Φ_n is the set of (q, tht) pairs passing through node n, and $\oplus$ indicates string concatenation. To prevent the unnecessary accumulation of redundant knowledge, the supervisor agent merges experiences linked to queries with a similarity score above 0.95, retaining no more than two instances.

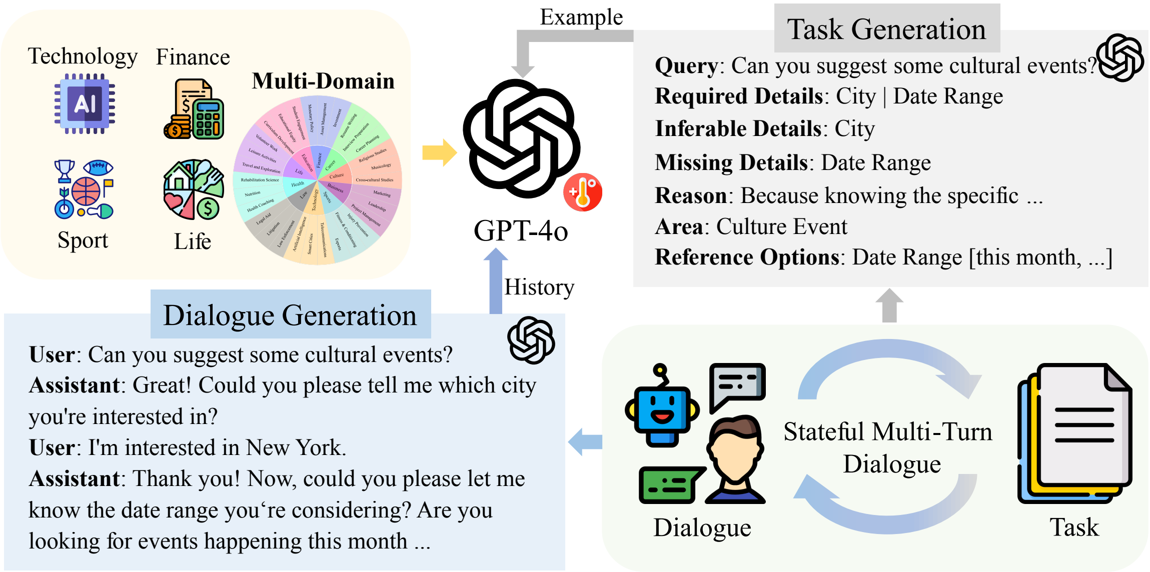
3.2.2 Knowledge Retrieval

During the inference stage, real-world user queries may target subdomains that do not precisely match the leaf nodes of the knowledge tree. To this end, we propose a knowledge retrieval strategy based on the soft allocation of high-level experience.

Formally, given a retrieval budget of M (M=8) experiences, our method initiates by generating the representation $\mathbf{e}_Q \in \mathbb{R}^d$ through concatenation embedding of the task query and the assistant's thought. Beginning at the top-level domain nodes, we collect the knowledge by executing the following layer-wise operations:

1. **Matching Score Computation.** We select the domain nodes with the top-K (K=3) highest cosine similarity, which is formulated as follows:

$$\tilde{\mathcal{N}_l} = \text{Top-}K(\{\cos(\mathbf{e}_Q, \mathbf{e}_n)\} \mid n \in \mathcal{N}_l),$$



Query: Can you suggest some cultural events?
Required Details: City | Date Range
Inferable Details: City
Missing Details: Date Range
Reason: Because knowing the specific ...
Area: Culture Event
Reference Options: Date Range [this month, ...]

User: Can you suggest some cultural events?
Assistant: Great! Could you please tell me which city you're interested in?
User: I'm interested in New York.
Assistant: Thank you! Now, could you please let me know the date range you‘re considering? Are you looking for events happening this month ...

Figure 2: The illustration of dataset generation process.

where $\mathcal{N}_l$ denotes the node set at l-th layer, and $\tilde{\mathcal{N}}_l$ represents the top-K matched nodes.

2. **Experience Budget Allocation.** The current layer's budget, M_l, is distributed among $\tilde{\mathcal{N}}_l$ in proportion to node's matching scores.
3. **Experience Aggregation.** We iterate the two preceding steps until M experiences are obtained, serving as the retrieved knowledge.

Overall, our HKTree-based approach provides three benefits over traditional methods: **(1)** compared to flattened knowledge storage, it significantly reduces computational demands (from $O(N)$ to $O(\log N)$) while avoiding homogenized retrieval outcomes; **(2)** it demonstrates scalable adaptability to larger data and diverse domains; and **(3)** in contrast to fine-tuning methods, our knowledge-augmented mechanism provides better transparency and interpretability, enabling the assistant to better understand user intents and make more informed decisions.

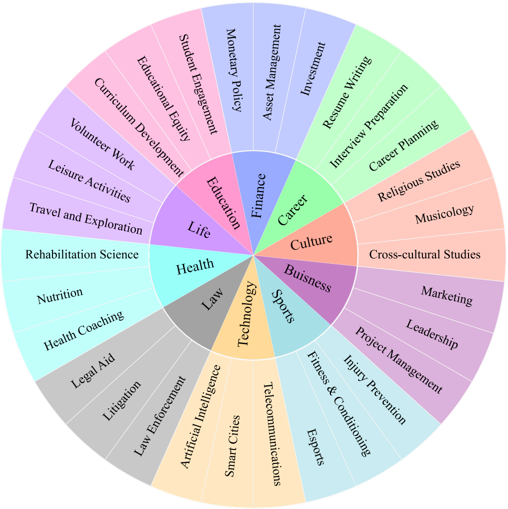
4 User-Intent-Understanding Benchmark

In real-world applications, user queries usually exhibit various styles and contents. **First**, an intelligent assistant should be able to accurately understand multi-turn successive queries. When faced with unclear information in vague queries, it can infer some missing details from history dialogue instead of only turning to redundant clarification questions. **Second**, in open-domain dialogue systems, user requirements may span diverse fields such as finance, technology, career, etc. **Third**, user tones can range from succinct to enthusiastic, so the proactive agent need adopt robust conversational strategy to foster both system friendliness and user satisfaction.

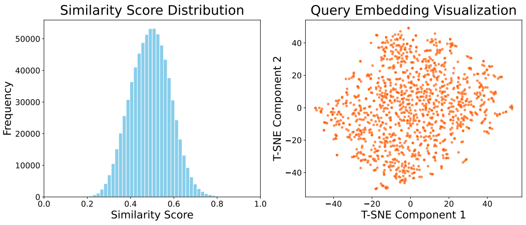
To create a benchmark that meets the above requirements for user intent understanding, we use a "task-dialogue" iterative generation method to produce sequential personalized tasks, reducing the complexity of simulating multi-task, multi-domain interactions. The illustration of the dataset generation process is shown in Fig. 2. In the following sections, we will elaborate on the task generation and dialogue generation, respectively.

4.1 Task Generation

Given the user's conversation history $\mathcal{H} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{t-1}\}$, then we need to generate a new task description $\mathcal{T}_t$ that includes the following metadata: (1) *Query*: the user's task requirement; (2) *Required Details*: essential information absent from the current query, including both inferable and missing details; (3) *Inferable Details*: details that can be deduced from the conversation history; (4) *Missing Details*: details that need be clarified from the user; (5) *Reason*: specific explanation of why each required detail is necessary for solving the task; (6) *Area*: the domain(s) involved in the query; (7) *Reference Options*: relevant options for each missing detail. Considering that real user queries may be vague or clear, we contain all metadata for vague query generation, while only query and area fields are used for clear tasks. We adopt 8:2 sampling probability ratio to create vague and



(a) Domain distribution of conversation topics.



(b) The visualization analysis of training data.

educational technology
operations management
financial planning
mental health
human resources
investment
education
strategic management
management
sports
health
business analytics
data science
personal growth
corporate finance
career counseling
career planning
personal finance
software engineering
educational psychology
career development
technology
public health
healthcare
cybersecurity
child health
nutrition
fitness
Travel

(c) Word cloud visualization of training data queries.

Figure 3: Visualization analysis of the UIU benchmark.

clear queries, respectively.

To enhance diversity in the generated tasks, we utilize multiple mechanisms to mitigate the issue of homogeneous outputs. **First**, we manually define 10 top-level domains and, with GPT-4o's assistance, create 30 subdomains for each (as shown in Fig. 3a). We then randomly select 10 subdomains from the domain pool to prompt GPT-4o to produce user queries for a wide range of scenarios. **Second**, we store newly generated tasks in a data repository as one-shot context for subsequent data generation, thereby further enhancing output variety. Note that we store clear and vague queries separately to prevent misguided induction. **Moreover**, we use a high temperature coefficient (*e.g.*, 0.9) to further increase randomness.

4.2 Dialogue Generation

Instead of directly constructing personalized sequential multi-task, we adopt an iterative procedure that alternates between generating the preceding task and producing the corresponding user-assistant dialogue. Specifically, for the dialogue generation process, we follow the work (Qian et al., 2024) to simulate the interactions between the user and the assistant with the following three steps:

1. **Clarification Need Prediction (CNP)**: For a given user query, the assistant analyzes whether the query is clear or vague by referencing the previous dialogue history. In other words, it determines whether further clarification questions are needed to gather missing details.
2. **Missing Detail Prediction (MDP)**: If the query is considered as ambiguous, the assistant is required to identify any missing information and then proactively ask clarification questions step by step to gather the necessary details. The assistant can utilize the reference options in the given task metadata for providing suitable choices to the user. Note that if the assistant determines in the previous step that the user query is clear, the MDP step should be skipped.
3. **User Intent Summary (UIS)**: In the final step, the assistant needs to summarize the user's intended goal, as well as any missing and inferred details required to complete the task, thereby providing comprehensive information for subsequent downstream task execution.

We use GPT-4o to play both the user and the intelligent agent roles for simulating the dialogue interaction described above, with the user's utterance tone randomly set to be either succinct or passionate. We then add the current dialogue $\mathcal{D}_t$ to the history $\mathcal{H}$ as context for the next task generation, continuing this process until the preset maximum dialogue round number is reached. In this experiment, that maximum interaction round is randomly selected within the range of [2,5]. To ensure the data quality, we conduct manual reviews to filter out tasks that do not meet the predefined format and requirements, while applying minor adjustments to retain those that only require slight modifications.

We randomly split the generated user task sequences into training and test sets, as shown in

Level	Split	Training	Test
Overall	# Seq	350	50
	# Area	443	180
	# Task	1,218	116
Task	# Clear Task	308	45
	# Vague Task	910	135
	# Inferable Task	221	27
	# Missing Details	3.40	3.65
User	# Task	3.48	3.60
	# Missing Details	8.83	9.86

Table 1: The overview of data statistics.

Table 1. We analyze the details at the overall, task, and user levels. Note that an "Inferable Task" can fall under either the Clear or Vague categories. If all necessary information can be inferred or the user has provided sufficient required details, it is clarified "Clear Task"; otherwise, if further clarification is needed, it is considered as "Vague Task". More analysis are provided in the Appendix A.

5 Experiments

Through extensive experiments on UIU benchmark, we demonstrate that our proposed KAPA effectively enhances both open-source and closed-source LLMs, achieving performance comparable to or even better than fine-tuned models.

5.1 Experimental Settings

Baselines. We select three mainstream LLMs as our base models: the closed-source GPT-4o (Hurst et al., 2024) and two open-source language models, Qwen2.5-14B (Yang et al., 2024) and LLaMA3.1-8B (Dubey et al., 2024). For **prompt-based methods**, we adopt a CoT-based Proactive Agent following the work Deng et al. (2023b) (*abbr.* ***origin.***). For **fine-tuning methods**, we follow Qian et al. (2024) to perform Supervised Fine-Tuning using generated multi-turn dialogues in Sec. 4.2 (*abbr.* ***sft***). For our **knowledge-augmented methods**, three strategies are designed: **(1)** ***exa*****:** retrieve the most semantically relevant dialogue instance from the HKTree in a greedy manner, serving as a one-shot context in the prompt, **(2)** ***exp*****:** use multi-domain knowledge soft retrieval to acquire experience knowledge as query-related guidelines, and **(3)** ***both*****:** integrate *exa* and *exp* to form composite knowledge. For more comprehensive evaluations, we also apply our knowledge-augmented strategies to the SFT-trained Qwen2.5 and LLaMA3.1 models.

Evaluation. In light of the effectiveness of *LLM-as-a-judge* paradigm (Zheng et al., 2023; Dong et al., 2024; Raina et al., 2024), we transition from the traditional, inefficient human-in-the-loop evaluation systems to an automated assessment pipeline by reusing process supervision from analyst agent in Sec. 3.1. Additionally, we also conduct extra manual verification and corrections to ensure the reliability of our experimental results.

Metrics. We evaluate the user intent understanding task in four key dimensions (for more detailed metric measurements, please refer to Appendix B.2): **(1) Vagueness Judgement:** Assess whether assistants accurately predict query ambiguity through three metrics: Accuracy, Clearness-F1 (C-F1), and Vagueness-F1 (V-F1). **(2) Clarification Question:** Measure the effectiveness of clarification questions via Opinion Presentation Rate (OPR), Average Opinion Count (AOC), Redundant Question Count (RQC), Valid Question Count (VQC), and Valid Question Ratio (VQR). **(3) Intent Summary:** Quantify the assistant's coverage of details omitted in the current query (ODRR), missing details (MDRR), and inferable details (IDRR) within its final summary. **(4) Overall Evaluation:** Measure task completion Success Rate (SR) and Efficiency, defined as the ratio of clarified details per question.

5.2 Overall Performance

From the experimental results in Table 2, we derive the following observations:

(1) Our knowledge-augmented methods based on *exp* and *exa* significantly outperform baseline models. Furthermore, the experience-based augmentation more effectively guides models master domain-specific knowledge than example-based ones, achieving performance comparable to the hybrid strategy (*i.e.*, *both* method). This suggests that our deliberative agent framework, by leveraging human process to extract high-level domain insights, achieves superior generalization ability.

(2) While our methods generate some redundant clarification questions compared with ProCoT, they effectively capture more valuable user's implicit information, ultimately improving overall success rate. For example, compared to the ProCoT, the *both* strategy increases the SR metric by 11.17% for GPT-4o and 9.5% for Qwen2.5. Moreover, from an overall perspective, our method still achieves higher clarification efficiency, improving both dialogue quality and user satisfaction.

Model	Method	Vagueness Judgment			Clarification Question				
		Accuracy↑	C-F1↑	V-F1↑	OPR	AOC	RQC↓	VQC↑	VQR↑
GPT-4o	**orig.**	0.5946	0.5946	0.5946	91.48%	3.0500	**0.7165**	2.6535	**80.10%**
	+ exa	0.6806	**0.6849**	0.6761	93.16%	3.3039	0.8397	2.4962	75.09%
	+ exp	0.6848	0.5672	0.7521	90.04%	3.0310	0.8168	**2.8092**	79.29%
	+ both	**0.6957**	0.5882	**0.7586**	93.61%	3.3393	1.0763	2.4198	70.51%
Qwen2.5	**orig.**	0.5974	0.5974	0.5974	79.20%	2.4965	**0.7360**	2.3120	76.60%
	+ sft	0.6835	**0.6667**	0.6988	83.07%	2.7408	1.1395	2.2791	69.01%
	+ exa	0.6854	0.5974	0.6265	90.50%	3.0333	0.8320	**2.5280**	77.10%
	+ exp	0.6125	0.6111	**0.7358**	84.86%	2.5391	0.7364	2.3953	**78.29%**
	+ both	**0.6923**	0.6500	0.7255	86.00%	3.0231	0.8880	2.2080	71.60%
Qwen2.5-SFT	**+ exa**	0.6585	0.6410	0.6744	87.06%	3.2927	**1.0000**	2.1905	70.17%
	+ exp	**0.7159**	0.6575	**0.7573**	81.76%	2.7904	1.0077	**2.3308**	**72.92%**
	+ both	0.7079	**0.6579**	0.7451	85.22%	3.2811	1.1328	2.2266	67.11%
LLaMA3.1	**orig.**	0.2632	0.2222	0.3000	86.40%	2.7119	**0.8667**	2.6667	**76.20%**
	+ sft	**0.5476**	**0.5682**	**0.5250**	81.40%	2.7892	1.2155	2.2759	67.00%
	+ exa	0.3594	0.3279	0.3881	93.25%	3.2645	1.0300	2.5700	73.00%
	+ exp	0.3158	0.2353	0.3810	86.83%	2.7152	1.0857	**2.6952**	73.50%
	+ both	0.4478	0.3509	0.5195	85.77%	3.2693	1.0189	2.2170	67.09%
LLaMA3.1-SFT	**+ exa**	0.5405	0.5526	0.5278	87.33%	3.3699	1.3306	**2.4113**	67.04%
	+ exp	**0.6184**	**0.6506**	**0.5797**	85.70%	2.8789	**0.9837**	2.2764	**72.00%**
	+ both	0.5455	0.5455	0.5455	84.90%	3.3592	1.2764	2.1951	64.90%

(a) The performance comparison of the different methods in vagueness judgment and clarification questions.

Model	Method	Intent Summary			Overall		
		ODRR↑	MDRR↑	IDRR↑	SR↑	Efficiency↑	Error↓
GPT-4o	**orig.**	57.10%	55.60%	85.00%	24.58%	0.6000	0.00%
	+ exa	60.33%	58.87%	78.79%	27.37%	0.5739	0.00%
	+ exp	**70.84%**	69.57%	**86.36%**	35.20%	0.6710	0.00%
	+ both	70.27%	**70.52%**	59.09%	**35.75%**	**0.7276**	0.00%
Qwen2.5	**orig.**	57.50%	55.30%	90.90%	25.70%	0.6993	0.00%
	+ sft	57.92%	56.37%	81.82%	30.17%	0.5406	0.00%
	+ exa	58.70%	58.20%	81.70%	27.37%	0.6402	0.00%
	+ exp	**65.53%**	**63.98%**	**95.45%**	34.08%	0.7956	0.00%
	+ both	62.90%	61.50%	83.30%	**35.20%**	**0.8028**	0.00%
Qwen2.5-SFT	**+ exa**	62.60%	**63.06%**	51.52%	**35.20%**	0.6424	0.00%
	+ exp	62.92%	62.99%	71.67%	30.17%	**0.6726**	0.00%
	+ both	**64.13%**	62.40%	**95.45%**	**35.20%**	0.6328	0.00%
LLaMA3.1	**orig.**	47.20%	45.70%	74.10%	8.38%	0.4552	22.91%
	+ sft	53.80%	52.40%	78.60%	**25.70%**	0.5154	0.00%
	+ exa	52.32%	51.73%	41.67%	12.85%	0.4918	15.64%
	+ exp	**59.02%**	57.27%	**78.79%**	10.06%	0.5794	20.67%
	+ both	57.42%	**57.66%**	58.33%	16.76%	**0.6809**	14.53%
LLaMA3.1-SFT	**+ exa**	51.20%	50.79%	51.52%	22.35%	0.4818	0.00%
	+ exp	**57.90%**	**56.00%**	**78.80%**	**26.26%**	**0.5782**	0.00%
	+ both	53.50%	53.60%	51.70%	23.46%	0.5347	0.00%

(b) The performance comparison of the different methods in intent summary and overall evaluation.

Table 2: Experimental results of various metrics for different methods with GPT-4o and Qwen2.5-14B. The notation '↑' indicates better performance with higher values, while '↓' indicates better performance with lower values.

(3) For large-scale language models (*e.g.*, 14B+ parameters), SFT-trained models demonstrate marginal performance gains. This likely stems from their inherent robustness in instruction following and advanced contextual comprehension. In open-domain tasks, the plugged high-level insights can more effectively and adaptively activate LLMs' specific-domain capability. Notably, we observe

Method	VQC	VQR	ODRR	SR	Efficiency
Random	2.4034	76.70%	55.70%	32.40%	0.6448
Greedy	2.6641	75.30%	62.50%	33.52%	0.5703
Ours	**2.8092**	**79.29%**	**70.84%**	**35.20%**	**0.6710**

Table 3: Comparison among different retrieval methods.

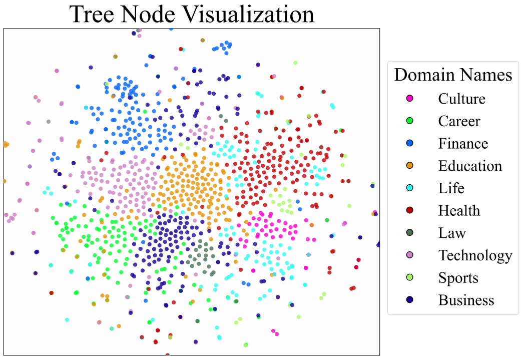


Figure 4: The visualization of query representation.

a subtle decline in the capacity to utilize external knowledge after fine-tuning, which we attribute to the force fitting of a small amount of dialogue data may hinder the model's generalization ability to integrate more diverse external information.

(4) Compared with strong LLMs (*e.g.*, Qwen2.5-14B and GPT-4o in this paper), the base LLaMA3.1-8B LLM cannot always precisely follow the predefined instruction output format in our tasks. Specifically, we can observe that original LLaMA-based models have a relatively high error rate (*i.e.*, 22.91%). Although one-shot in-context learning can somewhat reduce formatting errors and improve performance in different stages, fine-tuned LLMs demonstrate more substantially advancements in instruction-following abilities. Specifically, the sft-based approach increases the original 8.38% success rate to 25.70%, with further performance gains achievable through the integration of experiential knowledge.

5.3 Ablation Study

To validate the effectiveness of our knowledge retrieval strategy, we compare with the following two variants: (1) **Random retrieval**, which selects experiences randomly, and (2) **Greedy retrieval**, which samples experiences from the leaf node most similar to the current user query. We use GPT-4o as the base model for our experiments. To ensure a fair comparison, we limit the assistant's prompt to only the retrieved experiences, without using one-shot in-context learning.

From the results in Table 3, we see that our method significantly outperforms the two retrieval variants on the main metrics. This shows that our dynamic allocation mechanism could retrieve more diverse and informative query-related experiences, thereby achieving higher efficiency and success rate. It is worth noting that the random retrieval method also demonstrates competitive performance, which validates the robustness and generalization of the extracted high-level knowledge.

5.4 Further Analysis

In this section, we conduct detailed analysis to verify the rationality of the supervisor agent's automatic knowledge allocation on the HKTree. As illustrated in Fig. 4, queries across different domains exhibit concentrated distribution patterns, indicating that the supervisor agent successfully achieves the expected dynamic clustering. Interestingly, and in line with human intuition, topics sharing latent semantic connections such as daily life, health, and fitness are positioned closer together, while financial topic appears relatively distant from fitness clusters. This result shows that our online dynamic construction approach can reasonably supply domain-relevant knowledge for query retrieval.

For more detailed experiments, please refer to the Appendix C.

5.5 Conclusion

In this paper, inspired by human thought process, we propose a deliberate agent framework to extract high-level knowledge from simulated experiential learning. Specifically, we introduce five agents to execute a four-stage process consisting of perception, analysis, reflection, and knowledge accumulation. Additionally, we meticulously design a tree-structured knowledge base to enhance the efficient multi-domain knowledge storage and retrieval. To systematically study intent understanding tasks, we develop the UIU benchmark, which contains multi-domain, multi-tone, and sequential multi-turn interactions. Extended experiments demonstrate the effectiveness of our knowledge-enhanced approach.

Limitations

Although our approach shows promising results, there are still several limitations. First, our work focuses on the upstream part of the dialogue assistant workflow, leaving the final performance of downstream task execution to be further verified. Second, although our experiments confirm the feasibility of transferring experience from strong models (*i.e.*, GPT-4o) to weak models, the effectiveness of transferring experience refined by weak models to strong models still requires further validation. Finally, our experiments reveal that even the state-of-the-art large language models, when serving as verifies, still underperform human performance in complex contextual scenarios. Nonetheless, by harnessing the advantages and of llm-as-a-judge paradigm, the reasons and decisions provided by the LLMs significantly reduce human workload.

Ethical Statement

This paper studies how to build a multi-domain general knowledge base that helps intelligent agents better understand user intentions. Here, we discuss the related ethical considerations. First, we utilize the GPT-4o API to build the benchmark and automate the evaluation of subsequent experiments, without infringing upon any existing commercial software. Our released benchmark is only used for research purposes. Secondly, in terms of safety and privacy, it is essential to encrypt or mask users' personal data when applying the knowledge base in practice. This measure prevents the data from being used as context for other users, reducing the risk of privacy breaches.

Acknowledgments

This work is supported in part by National Natural Science Foundation of China (No. 62422215 and No. 62472427), Tencent, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, Public Computing Cloud, Renmin University of China, fund for building world-class universities (disciplines) of Renmin University of China.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. 2024. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Maximillian Chen, Ruoxi Sun, Sercan Ö Arık, and Tomas Pfister. 2024b. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. *arXiv preprint arXiv:2406.00222*.

Yue Chen, Chen Huang, Yang Deng, Wenqiang Lei, Dingnan Jin, Jia Liu, and Tat-Seng Chua. 2024c. Style: Improving domain transferability of asking clarification questions in large language model powered conversational agents. *arXiv preprint arXiv:2405.12059*.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023a. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*.

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. *arXiv preprint arXiv:2305.13626*.

Yang Deng, Lizi Liao, Wenqiang Lei, Grace Yang, Wai Lam, and Tat-Seng Chua. 2025. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*.

Yijiang River Dong, Tiancheng Hu, and Nigel Collier. 2024. Can llm be a personalized judge? *arXiv preprint arXiv:2406.11657*.

Huifang Du, Xuejing Feng, Jun Ma, Meng Wang, Shiyu Tao, Yijie Zhong, Yuan-Fang Li, and Haofen Wang. 2024. Towards proactive interactions for in-vehicle conversational assistants utilizing large language models. *arXiv preprint arXiv:2403.09135*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, Yankai Lin, Xu Chen, Zhiyuan Liu, and Ji-Rong Wen. 2024. Large language model-based human-agent collaboration for complex task solving. *arXiv preprint arXiv:2402.12914*.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. *arXiv preprint arXiv:1808.09637*.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, et al. 2025. Pasa: An llm agent for comprehensive academic paper search. *arXiv preprint arXiv:2501.10120*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Chuhao Jin, Kening Ren, Lingzhen Kong, Xiting Wang, Ruihua Song, and Huan Chen. 2024. Persuading across diverse domains: a dataset and persuasion large language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1678–1706.

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. *arXiv preprint arXiv:2205.12688*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.

Junkai Li, Siyu Wang, Meng Zhang, Weitao Li, Yunghwei Lai, Xinhui Kang, Weizhi Ma, and Yang Liu. 2024. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957*.

Jeesun Oh, Wooseok Kim, Sungbae Kim, Hyeonjeong Im, and Sangsu Lee. 2024. Better to ask than assume: Proactive voice assistants' communication strategies that respect user agency in a smart home environment. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *arXiv preprint arXiv:2402.14016*.

R.W. Revans. 1982. *The Origins and Growth of Action Learning*. Studentlitteratur.

Donald A Schön. 2017. *The reflective practitioner: How professionals think in action*. Routledge.

Ivan Sekulić, Mohammad Aliannejadi, and Fabio Crestani. 2022. Evaluating mixed-initiative conversational search systems via user simulation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 888–896.

Chenzhan Shang, Yupeng Hou, Wayne Xin Zhao, Yaliang Li, and Jing Zhang. 2023. Multi-grained hypergraph interest modeling for conversational recommendation. *AI Open*, 4:154–164.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Elizabeth A Smith. 2001. The role of tacit and explicit knowledge in the workplace. *Journal of knowledge Management*, 5(4):311–321.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric P Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. *arXiv preprint arXiv:1905.11553*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

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.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhitong Yang, Bo Wang, Jinfeng Zhou, Yue Tan, Dongming Zhao, Kun Huang, Ruifang He, and Yuexian Hou. 2022. Topkg: Target-oriented dialog via global planning on knowledge graph. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 745–755.

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025. A survey on multi-turn interaction capabilities of large language models. *arXiv preprint arXiv:2501.09959*.

Michael JQ Zhang and Eunsol Choi. 2023. Clarify when necessary: Resolving ambiguity through interaction with lms. *arXiv preprint arXiv:2311.09469*.

Michael JQ Zhang, W Bradley Knox, and Eunsol Choi. 2024a. Modeling future conversation turns to teach llms to ask clarifying questions. *arXiv preprint arXiv:2410.13788*.

Xiaoyu Zhang, Ruobing Xie, Yougang Lyu, Xin Xin, Pengjie Ren, Mingfei Liang, Bo Zhang, Zhanhui Kang, Maarten de Rijke, and Zhaochun Ren. 2024b. Towards empathetic conversational recommender systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 84–93.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Caleb Ziems, Jane A Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. 2022. The moral integrity corpus: A benchmark for ethical dialogue systems. *arXiv preprint arXiv:2204.03021*.

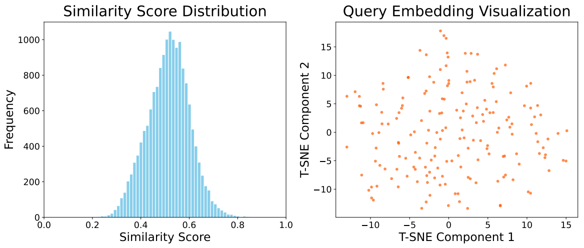


Figure 5: The visualization analysis of the test data.

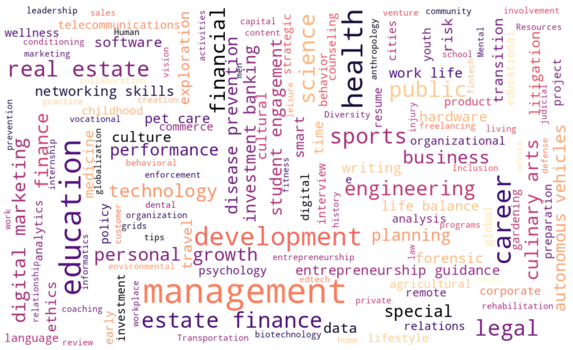


Figure 6: Word cloud visualization of queries in the test data.

A UIU Benchmark Analysis

We examine the diversity performance of the generated tasks. In specific, we first use the BGE-M3 embedding model to convert all queries into 1024-dimensional vectors, then calculate their pairwise similarity scores. As shown in Fig. 3b, the average similarity hovers around 0.4917 with a standard deviation of 0.0919, indicating moderate overall similarity across queries and demonstrating the effectiveness of our method in enhancing data diversity. Besides, we visualize the queries from the training data using t-SNE and present a corresponding word cloud in Fig. 3b and Fig. 3c, respectively, presenting the broad topic distribution of UIU benchmark. The data distribution of the test set shows similar patterns, with the corresponding visual analysis and word cloud presented in Fig. 5 and Fig. 6, respectively.

B Experimental Setup

B.1 Experimental Details

For SFT-based models, we perform fine-tuning with LoRA (Hu et al., 2021), setting the rank and alpha to 32 and 64, respectively, over a maximum of five epochs. During inference, we fix the temperature parameter at 0.6. Moreover, the HKtree is constrained to a maximum depth of 2, with each level accommodating at most 10 child domain nodes. For each action (*i.e.*, CNP, MDP, and UIS), the assistant is allowed a maximum of three retry attempts, where the detailed retry success rate is illustrated in Fig 8. The open-code implementation of KAPA is available at `https://anonymous.4open.science/r/KAPA-F32D`.

B.2 Detailed Metric Calculation

Notation Definition: Given a user u ($u \in \mathcal{U}$) with a set of queries $\mathcal{Q}_u = \{q_1^u, q_2^u, \dots\}$, each query q has a ground-truth label $y_q \in \{\text{vague}, \text{clear}\}$ while the assistant's predicted label is denoted as $\hat{y}_q$. The omitted but required details, missing details, and inferable details for q are defined as $\mathcal{F}_q^{\text{req}}, \mathcal{F}_q^{\text{mis}}, \mathcal{F}_q^{\text{inf}}$, respectively, and the corresponding assistant's predictions are denoted as $\hat{\mathcal{F}}_q^{\text{req}}, \hat{\mathcal{F}}_q^{\text{mis}}, \hat{\mathcal{F}}_q^{\text{inf}}$, respectively. Note that the set of required details comprises both missing and inferable details, *i.e.*, $F_q^{req} = F_q^{\text{mis}} \cup F_q^{\text{inf}}$ and $\hat{F}_q^{\text{req}} = \hat{F}_q^{\text{mis}} \cup \hat{F}_q^{\text{inf}}$. For vague queries, the assistant proactively initiates a series of clarification questions ($\mathcal{CQ}$), each accompanied by some reference options $\mathcal{O}$. Among these questions, only some pertain to missing details, denoted as $\mathcal{CQ}^{\text{val}}$. Next, we present the formal definitions of the metrics introduced in Sec. 5.1 are calculated.

1. **Vagueness Judgment**

- **Accuracy**: The proportion of correctly predicted vague or clear queries.

$$\text{Accuracy} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}(\hat{y} = y)}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}.$$

- **Clearness-F1 (C-F1)**: The F1 score of correctly predicted clear queries.

$$\text{C-P} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(\hat{y} = y \wedge \hat{y} = \text{clear}\big)}{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(\hat{y} = \text{clear}\big)},$$

$$\text{C-R} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(\hat{y} = y \wedge \hat{y} = \text{clear}\big)}{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(y = \text{clear}\big)},$$

$$\text{C-F1} = \frac{2 \times \text{C-Precision} \times \text{C-Recall}}{\text{C-Precision} + \text{C-Recall}}.$$

- **Vagueness-F1 (V-F1)**: The F1 score of correctly predicted vague queries.

$$\text{V-P} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(\hat{y} = y \wedge \hat{y} = \text{vague}\big)}{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(\hat{y} = \text{vague}\big)},$$

$$\text{V-R} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(\hat{y} = y \wedge \hat{y} = \text{vague}\big)}{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}\big(y = \text{vague}\big)},$$

$$\text{V-F1} = \frac{2 \times \text{V-Precision} \times \text{V-Recall}}{\text{V-Precision} + \text{V-Recall}}.$$

2. Clarification Question

The following metrics are calculated only in cases where the assistant **correctly** predicts the user's **vague** intent.

- **Option Presentation Rate (OPR)**: The proportion of questions that present options.

$$\text{OPR} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \sum_{\mathcal{CQ}_q} \mathbb{1}(|\mathcal{O}| > 0)}{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} |\mathcal{CQ}_q|}$$

- **Average Opinion Count (AOC)**: The average number of options presented in each clarification question.

$$\text{AOC} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \sum_{\mathcal{CQ}_q} |\mathcal{O}|}{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \sum_{\mathcal{CQ}_q} \mathbb{1}(|\mathcal{O}| > 0)}$$

- **Redundant Question Count (RQC)**: The number of redundant questions asked by the assistant.

$$\text{RQC} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \sum_{cq \in \mathcal{CQ}_q} \mathbb{1}(cq \notin \mathcal{CQ}_q^{\text{val}})}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

- **Valid Question Count (VQC)**: The number of valid questions asked by the assistant.

$$\text{VQC} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \sum_{cq \in \mathcal{CQ}_q} \mathbb{1}(cq \in \mathcal{CQ}_q^{\text{val}})}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

- **Valid Question Ratio (VQR)**: The proportion of valid questions asked by the assistant.

$$\text{VQR} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \frac{\sum_{cq \in \mathcal{CQ}_q} \mathbb{1}(cq \in \mathcal{CQ}_q^{\text{val}})}{|\mathcal{CQ}_q|}}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

3. Intent Summary

- **Omitted Details Recover Rate (ODRR)**: The proportion of omitted details recovered by the assistant.

$$\text{ODRR} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \frac{|\hat{\mathcal{F}}_q^{\text{req}} \cap \mathcal{F}_q^{\text{req}}|}{|\mathcal{F}_q^{\text{req}}|}}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

- **Missing Details Recovery Rate (MDRR)**: The proportion of missing details recovered by the assistant.

$$\text{MDRR} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \frac{|\hat{\mathcal{F}}_q^{\text{mis}} \cap \mathcal{F}_q^{\text{mis}}|}{|\mathcal{F}_q^{\text{mis}}|}}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

- **Inferable Details Recovery Rate (IDRR)**: The proportion of inferable details recovered by the assistant.

$$\text{IDRR} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \frac{|\hat{\mathcal{F}}_q^{\text{inf}} \cap \mathcal{F}_q^{\text{inf}}|}{|\mathcal{F}_q^{\text{inf}}|}}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

4. Overall Evaluation

- **Success Rate (SR)**: The proportion of successful task completions.

$$\text{SR} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \mathbb{1}(\text{task completion})}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

  where "task completion" means the assistant accurately predicts the ambiguity in the user's query (in particular, for ambiguous query, the assistant can completely recover all required details in the final summary).

- **Efficiency**: The ratio of clarified details to the number of clarification questions asked. Higher efficiency indicates that the assistant can gather more useful information with fewer clarification questions.

$$\text{Efficiency} = \frac{\sum_{u \in \mathcal{U}} \sum_{q \in \mathcal{Q}_u} \frac{|\hat{\mathcal{F}}_q^{\text{mis}} \cap \mathcal{F}_q^{\text{mis}}|}{|\mathcal{CQ}_q|}}{\sum_{u \in \mathcal{U}} |\mathcal{Q}_u|}$$

Top-Level Domain: Buisness

Human Resources and Employee Management

- Prioritize asking about the most fundamental missing detail first, such as the specific market sector, to provide relevant and tailored information efficiently.
- When users request advice on improving team efficiency, always ask about potential underlying issues such as productivity and motivation, even if not explicitly mentioned.
- Always include a reference to the current phase of the project when asking for clarifications to better understand the user's request for optimization.

Supplier and Supply Chain Management

- Enhance skills in accurately determining the clarity of user queries by leveraging detailed conversation history to avoid unnecessary clarification requests.
- For each clarification step, ensure to ask targeted questions that progressively narrow down the user's needs and goals, leading to a clear and actionable summary.
- For queries related to vendor selection or similar tasks, prioritize clarifying the criteria for evaluation first, followed by timeframe, budget, and desired outcome to streamline the information-gathering process.

Figure 7: Randomly selected Business subdomains and their experiences.

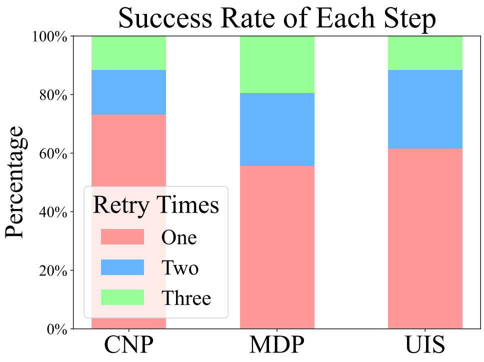


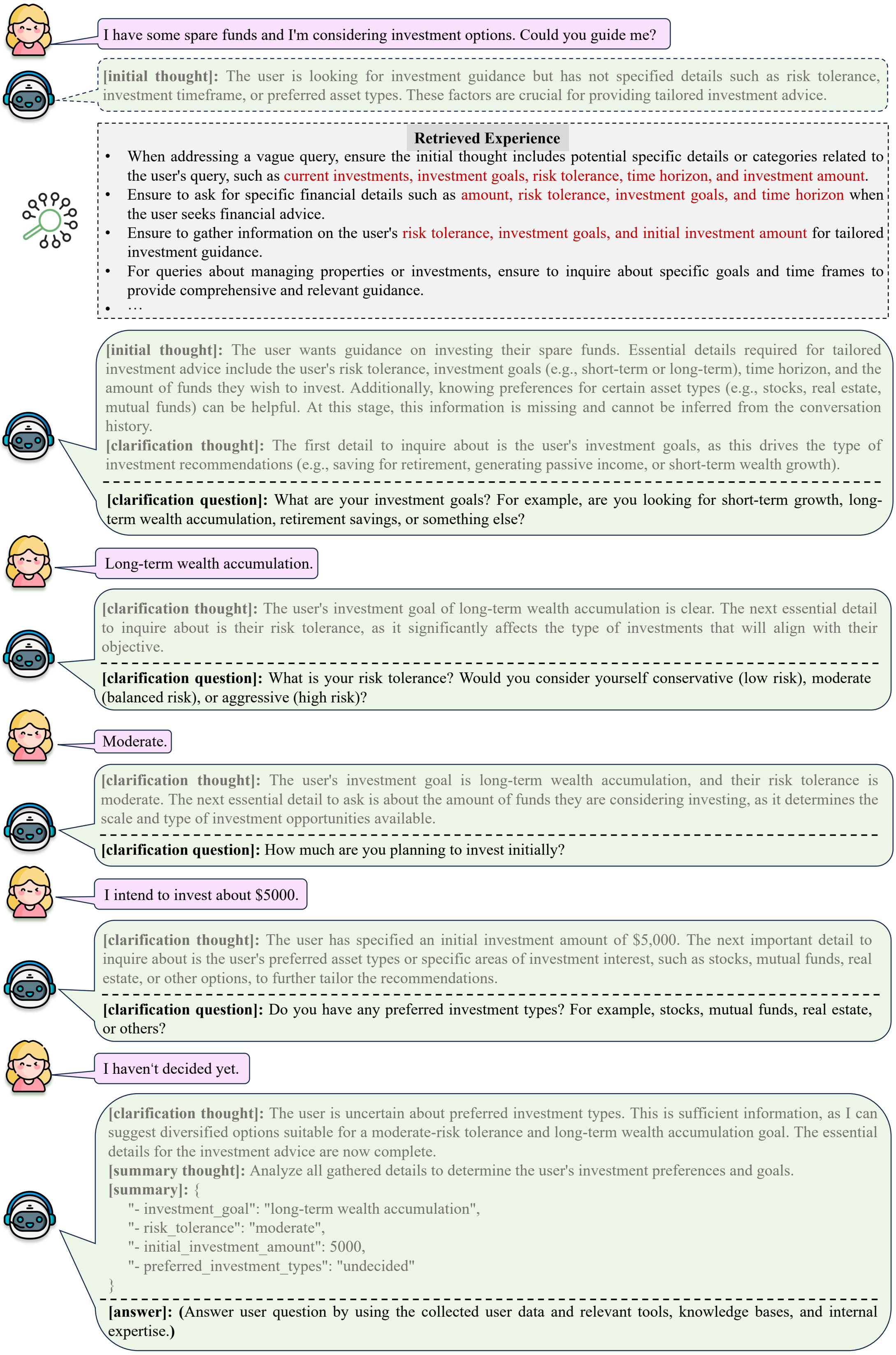
Figure 8: Success rate of retries at each step.

C Additional Experimental Results

C.1 Case Study

To further demonstrate the benefits of our knowledge-augmented approach, we present a case of interactions between an assistant and a user using GPT-4o. As shown in Fig. 9, the user initially poses an investment-related query in the financial domain. In its initial thought, the assistant mentions "risk tolerance", "inventment timeframe", and "preferred asset types", but it fails to inquire about the user's specific investment amount, which is essential for making a comprehensive investment suggestions. Based on the user's query and the assistant's initial thought, our method successfully retrieves relevant experiential knowledge in the finance domain, including several pieces of advice emphasizing the importance of the "investment amount" (marked in red). The extracted knowledge further guides the assistant to reevaluate which missing details are required to clarification, thereby improving the overall dialogue efficiency and boosting user satisfaction.

Moreover, we randomly select five top-level domains along with their corresponding subdomains. Then for each domain, the randomly selected experiences are presented in Fig. 7 (business), 10 (health), 11 (daily life), 12 (career), and 13 (technology), respectively. Based on these experiences, we observe that the extracted high-level insights contain both domain-specific knowledge and general applicable principles for understanding user intent, further validating the effectiveness of our multi-domain deliberate agent framework to extract high-level tacit experience.



I have some spare funds and I'm considering investment options. Could you guide me?

[initial thought]: The user is looking for investment guidance but has not specified details such as risk tolerance, investment timeframe, or preferred asset types. These factors are crucial for providing tailored investment advice.

Retrieved Experience

- When addressing a vague query, ensure the initial thought includes potential specific details or categories related to the user's query, such as current investments, investment goals, risk tolerance, time horizon, and investment amount.
- Ensure to ask for specific financial details such as amount, risk tolerance, investment goals, and time horizon when the user seeks financial advice.
- Ensure to gather information on the user's risk tolerance, investment goals, and initial investment amount for tailored investment guidance.
- For queries about managing properties or investments, ensure to inquire about specific goals and time frames to provide comprehensive and relevant guidance.
- …

[initial thought]: The user wants guidance on investing their spare funds. Essential details required for tailored investment advice include the user's risk tolerance, investment goals (e.g., short-term or long-term), time horizon, and the amount of funds they wish to invest. Additionally, knowing preferences for certain asset types (e.g., stocks, real estate, mutual funds) can be helpful. At this stage, this information is missing and cannot be inferred from the conversation history.
[clarification thought]: The first detail to inquire about is the user's investment goals, as this drives the type of investment recommendations (e.g., saving for retirement, generating passive income, or short-term wealth growth).

[clarification question]: What are your investment goals? For example, are you looking for short-term growth, long-term wealth accumulation, retirement savings, or something else?

Long-term wealth accumulation.

[clarification thought]: The user's investment goal of long-term wealth accumulation is clear. The next essential detail to inquire about is their risk tolerance, as it significantly affects the type of investments that will align with their objective.

[clarification question]: What is your risk tolerance? Would you consider yourself conservative (low risk), moderate (balanced risk), or aggressive (high risk)?

Moderate.

[clarification thought]: The user's investment goal is long-term wealth accumulation, and their risk tolerance is moderate. The next essential detail to ask is about the amount of funds they are considering investing, as it determines the scale and type of investment opportunities available.

[clarification question]: How much are you planning to invest initially?

I intend to invest about $5000.

[clarification thought]: The user has specified an initial investment amount of $5,000. The next important detail to inquire about is the user's preferred asset types or specific areas of investment interest, such as stocks, mutual funds, real estate, or other options, to further tailor the recommendations.

[clarification question]: Do you have any preferred investment types? For example, stocks, mutual funds, real estate, or others?

I haven‘t decided yet.

[clarification thought]: The user is uncertain about preferred investment types. This is sufficient information, as I can suggest diversified options suitable for a moderate-risk tolerance and long-term wealth accumulation goal. The essential details for the investment advice are now complete.
[summary thought]: Analyze all gathered details to determine the user's investment preferences and goals.
[summary]: {
"- investment_goal": "long-term wealth accumulation",
"- risk_tolerance": "moderate",
"- initial_investment_amount": 5000,
"- preferred_investment_types": "undecided"
}

[answer]: (Answer user question by using the collected user data and relevant tools, knowledge bases, and internal expertise.)

Figure 9: A case study of the assistant’s interaction with the user.

Top-Level Domain: Health

Behavioral Health and Therapy Services

- Always ask for the user's current location when the query involves finding a nearby service or place.
- Always identify and address the most critical missing information first when a query is vague.
- When identifying a behavioral issue, systematically gather information on the type of issue, age of the individual, specific symptoms, duration of symptoms, and any previous treatments to provide tailored advice.

Healthcare Skills and Professional Development

- Ensure to ask for the type of infectious disease first when the query involves planning for an outbreak.
- When summarizing complex queries, break down each requirement into distinct, manageable parts to ensure comprehensive coverage and clarity.
- Ensure that each clarification question is directly related to the previous user response to maintain a logical flow and avoid redundant questions.

Figure 10: Randomly selected Health subdomains and their experiences.

Top-Level Domain: Life

Travel Planning and Experiences

- Identify and prioritize asking for the most critical missing information first, using conversation history to infer details and minimize unnecessary clarifications.
- When summarizing user preferences, include both the specific activities and the broader categories (e.g., relaxing or entertaining) to capture the full scope of the user's interests.
- Leverage historical context and user preferences to infer and provide concise, actionable information without unnecessary clarifications.

Sustainable Living and Home Organization

- Always consider financial and time-related constraints when clarifying vague project-related queries.
- Always include any hidden or inferred information, such as specific budget ranges, in the summary to ensure completeness.
- Cross-reference the current query with previous conversation history to validate the completeness of the information provided.

Figure 11: Randomly selected Life subdomains and their experiences.

Top-Level Domain: Career

Interview Preparation and Professional Skills

- When summarizing user queries, clearly list all key details to confirm the query's completeness and avoid overlooking any critical information.
- Include considerations of long-term career goals and aspirations when discussing significant career changes to provide a comprehensive response.
- For queries related to improvement or enhancement, always inquire about the desired outcome or specific goals to tailor the response effectively.

Job Search and Resume Writing

- When a query includes specific details about the user's background and requirements, consider it clear and proceed with summarizing the provided information.
- Enhance the ability to consistently apply criteria for query clarity by leveraging detailed historical context to make informed assumptions about user intent.
- When summarizing, ensure that all aspects of the user's query, including their specific needs or challenges, are clearly articulated to avoid missing critical information.

Figure 12: Randomly selected Career subdomains and their experiences.

Top-Level Domain: Technology

Software Performance and Optimization

- Enhance the assistant's ability to utilize historical context to infer and address user queries confidently without unnecessary clarifications.
- When handling clear queries, the assistant should summarize the task, input, output, and functionality concisely to confirm understanding.
- Always inquire about the target performance metrics or goals when the user seeks advice on improving performance.

Data Management and Preprocessing

- Prioritize identifying the project type early in the conversation to tailor subsequent questions effectively.
- When addressing vague queries, suggest specific techniques or methods related to the task to guide the user more effectively.
- When the query involves security or privacy concerns, explicitly confirm the type of data, security measures, and access permissions to ensure comprehensive understanding and accurate guidance.

Figure 13: Randomly selected Technology subdomains and their experiences.