From Exploration to Mastery: Enabling LLMs to Master Tools Via Self-Driven Interactions

ICLR’25

Changle Qu[1], Sunhao Dai[1], Xiaochi Wei[2], Hengyi Cai[3],
Shuaiqiang Wang[2], Dawei Yin[2], Jun Xu[1] * ,Ji-Rong Wen[1]

Presenter: Jinhan Xin
2025.05.09

Contents

- Background
- overview
- Methods
- Evaluation
- Thinking

Background

Tool learning ,which integrates external tools with LLMs, has significantly enhanced the capability of LLMs to address complex real-world tasks.

LangChain

Tool Alpaca

ToolBench

Background

By leveraging external tools, LLMs are able to access up-to-date information, interact with dynamic environments, and take actions beyond their original scope.

Background

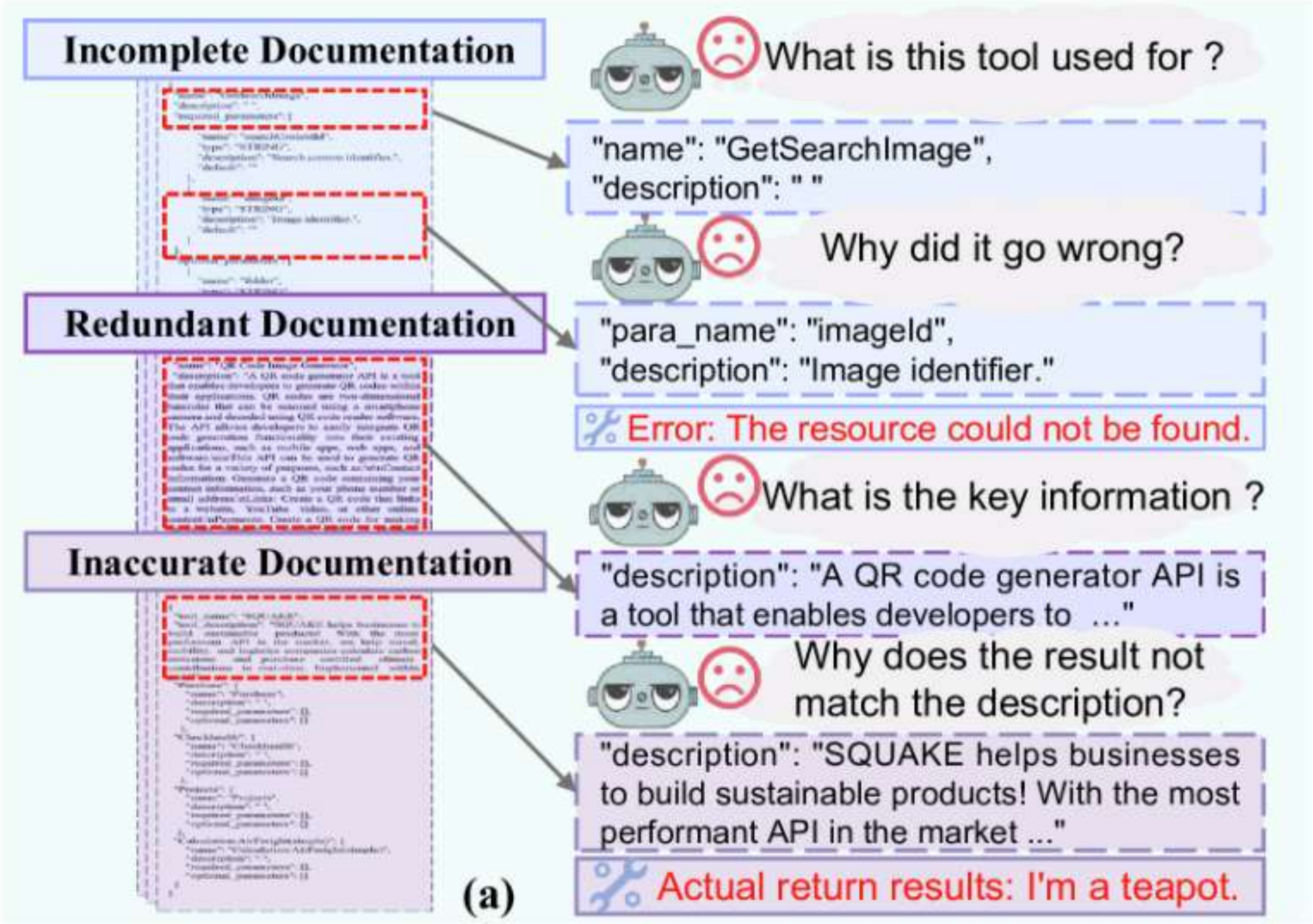
To effectively utilize these external tools, LLMs are typically provided with **tool documentation** as context.

Tool documentation should contain:

- How the tools work
- their potential uses
- How to be used to solve tasks

Compiling the ideal documentation for external tools that adapts to the specific needs of LLMs remains a challenging task.

Background



- **incomplete documentation :**difficult understand the purpose of a tool and when it should be invoked
- **redundant documentation :**containing irrelevant information, obscures key details, increases token consumption
- **inaccurate documentation :**leading to the potential misuse of the tool by LLMs

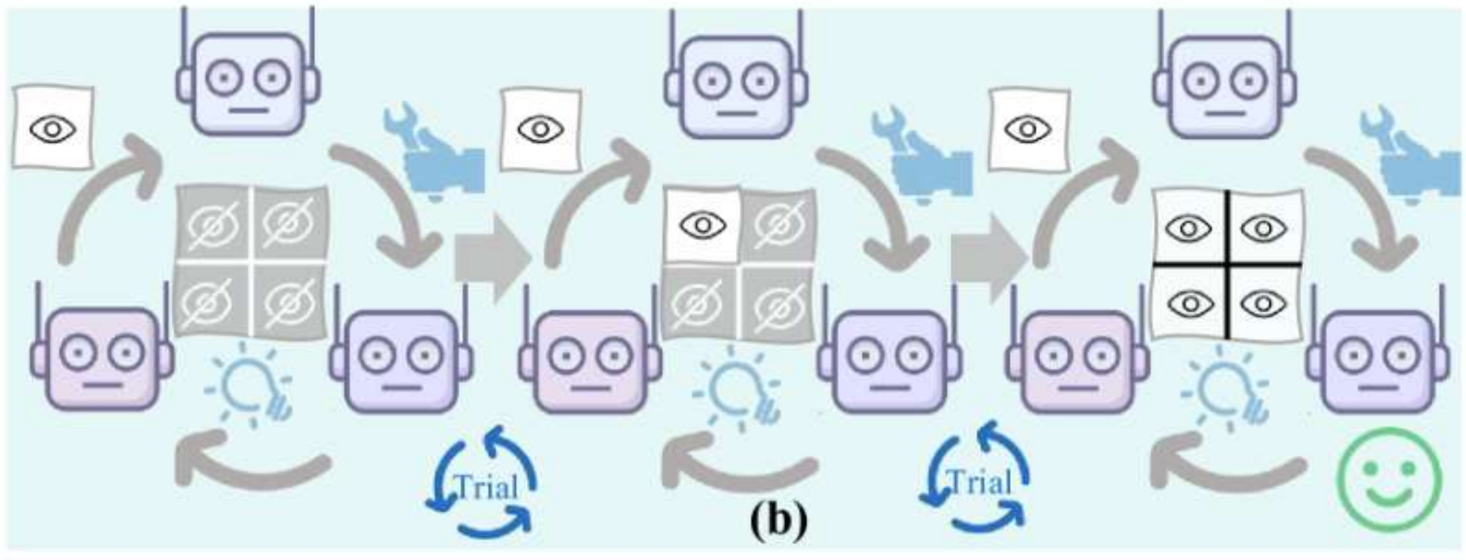
Background

Problems:

- Manual modification coverage is not comprehensive
- Manual modification time and labor cost are large
- tools are frequently updated, deprecated, or extended

These issues prevent LLMs from making effective use of the tool!

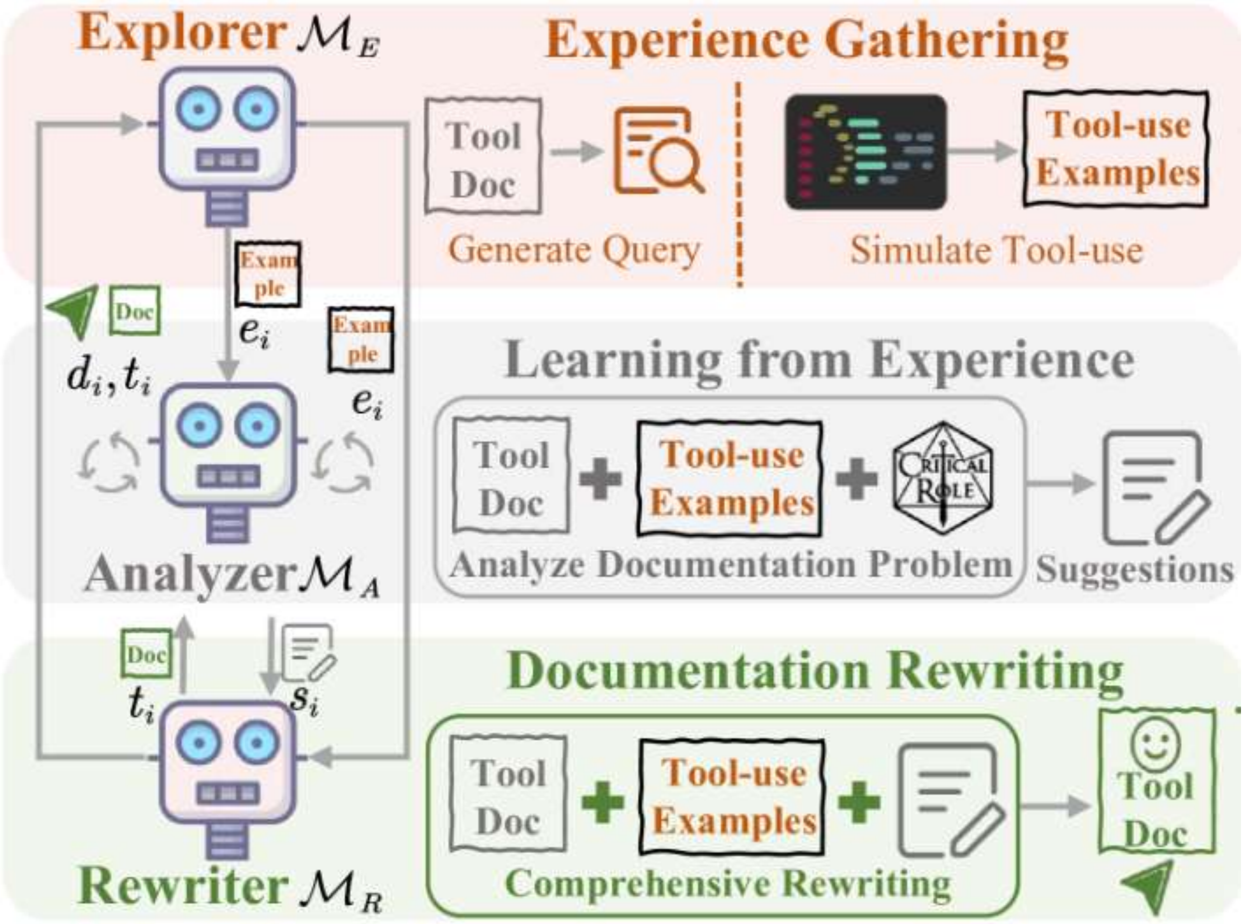
Background



Core Idea:trial and error

Humans acquire tool proficiency through **repeated interactions** and **hands-on experiences**. In light of this, this paper proposes **DRAFT**, conceptualized to automate the adjustment and optimization of tool documentation based on the **feedback** derived from the LLM’s **interaction**.

Overview



Experience gathering

It first undertakes the simulation of potential tool application scenarios, crafting explorative instances and capturing tool execution outcomes through a designed explorer.

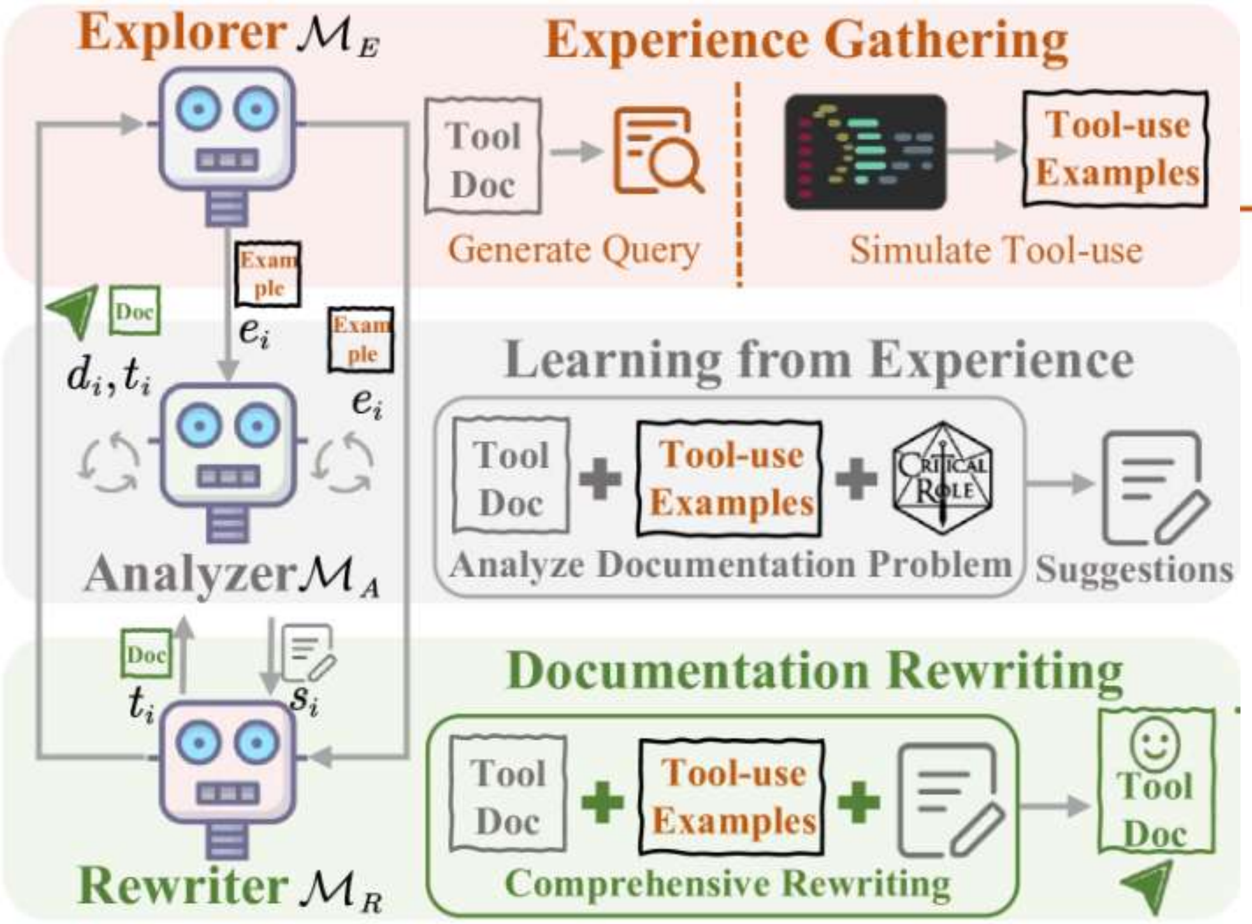
Learning from experience

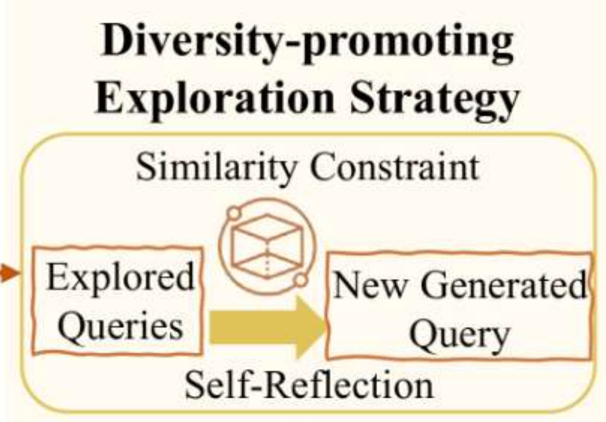
The analyzer dissects the prevailing documentation, amalgamating insights from the explorer's findings and feedback to moot documentation modification propositions.

Documentation rewriting

Finally, the rewriter amalgamates these insights, refining the tool documentation while simultaneously guiding further explorative pursuits by the explorer.

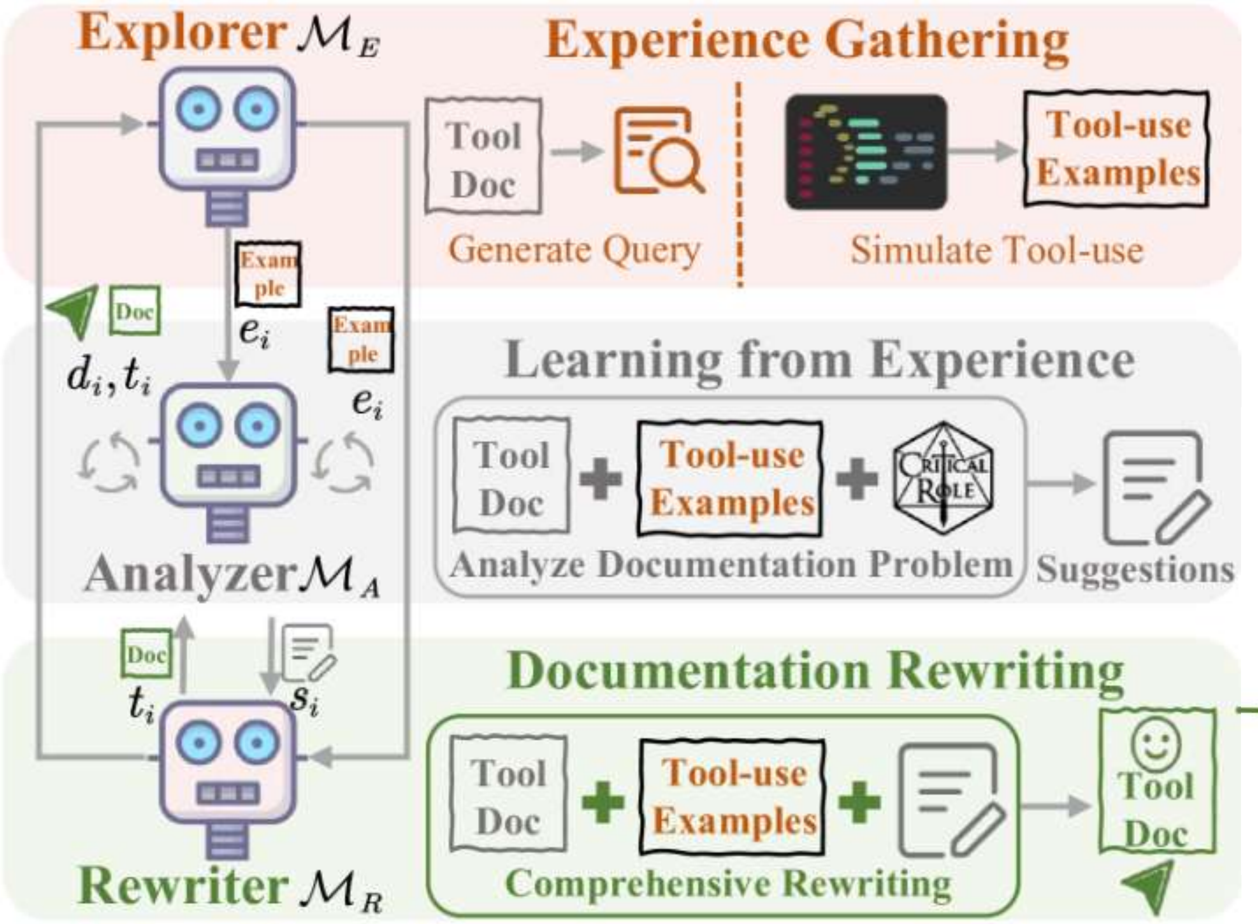
Overview



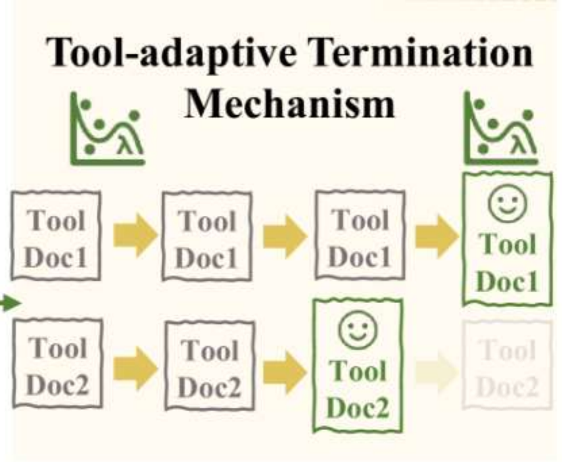


To optimize the process,This paper design a diversity-promoting exploration strategy to ensure diversity in exploration, thus providing a wider range of samples for subsequent rewriting.

Overview



Halting the iterative process once the documentation aligns with the comprehension of LLMs, thereby saving time and resources while preventing overfitting.



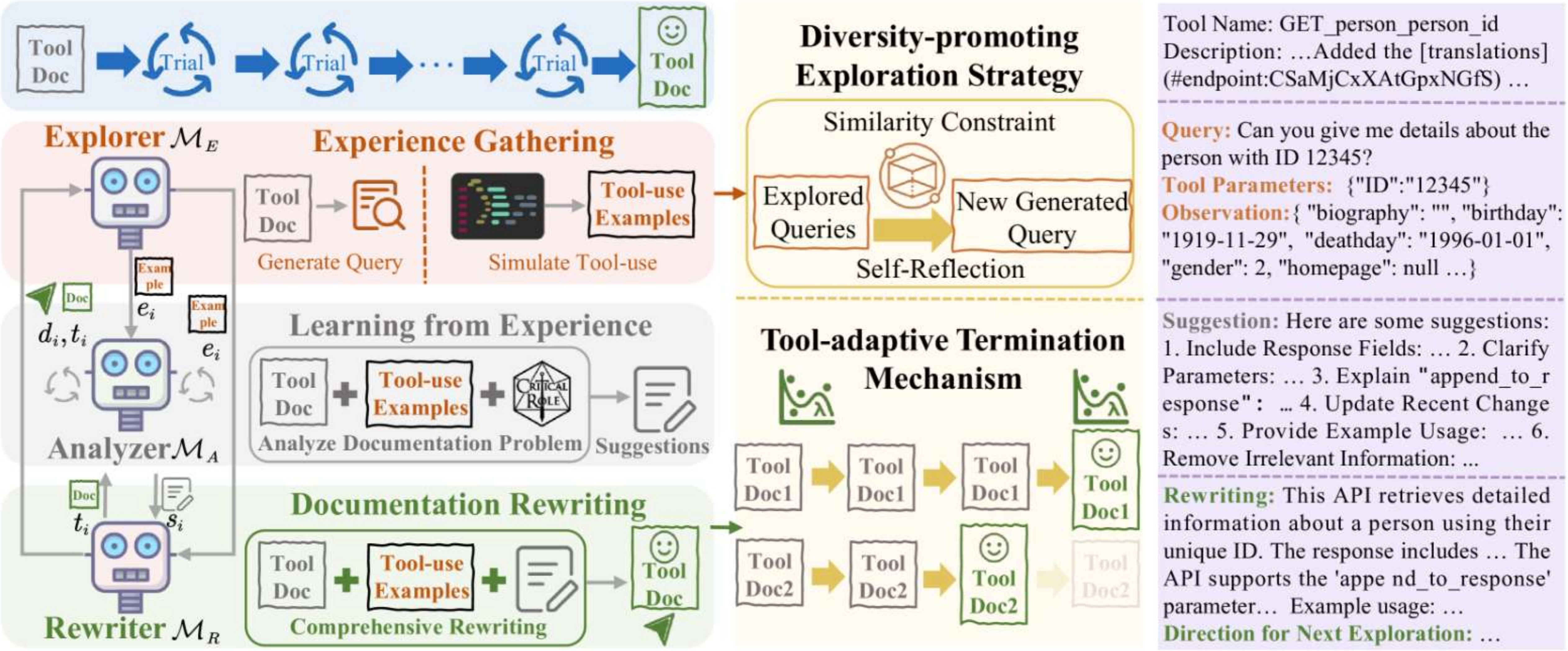
Overview

Core Idea： trial and error

- inefficiencies and inaccuracies within existing tool documentation hamper the effective utilization of tools by LLMs.

- DRAFT, designed to dynamically adjust and optimize tool documentation based on the interaction feedback

Overview



Tool Doc
Trial
Trial
Trial
Tool Doc
Explorer $\mathcal{M}_E$
Experience Gathering
Tool Doc
Generate Query
Tool-use Examples
Simulate Tool-use
Example
Doc
e_i
d_i, t_i
Example
e_i
Learning from Experience
Tool Doc
Tool-use Examples
Analyze Documentation Problem
Suggestions
Analyzer $\mathcal{M}_A$
Doc
t_i
s_i
Documentation Rewriting
Tool Doc
Tool-use Examples
Comprehensive Rewriting
Tool Doc
Rewriter $\mathcal{M}_R$
Diversity-promoting Exploration Strategy
Similarity Constraint
Explored Queries
New Generated Query
Self-Reflection
Tool-adaptive Termination Mechanism
Tool Doc1
Tool Doc1
Tool Doc1
Tool Doc1
Tool Doc2
Tool Doc2
Tool Doc2
Tool Doc2
Tool Name: GET_person_person_id
Description: …Added the [translations] (#endpoint:CSaMjCxXAtGpxNGfS) …
Query: Can you give me details about the person with ID 12345?
Tool Parameters: {"ID":"12345"}
Observation: { "biography": "", "birthday": "1919-11-29", "deathday": "1996-01-01", "gender": 2, "homepage": null …}
Suggestion: Here are some suggestions: 1. Include Response Fields: … 2. Clarify Parameters: … 3. Explain "append_to_response": … 4. Update Recent Changes: … 5. Provide Example Usage: … 6. Remove Irrelevant Information: …
Rewriting: This API retrieves detailed information about a person using their unique ID. The response includes … The API supports the 'appe nd_to_response' parameter… Example usage: …
Direction for Next Exploration: …

Methods

Experience Gathering

Design an Explorer $\mathcal{M}_\mathrm{E}$ to simulate plausible scenarios in which the tool may be utilized.

$$e_i = \mathcal{M}_E(t_{i-1}, d_{i-1}, \mathcal{H}_i), \quad (1)$$

At the i-th iteration, the Explorer $\mathcal{M}_\mathrm{E}$ generates an exploration instance e_i based on the **current tool documentation $\boldsymbol{t_{i-1}}$**, **next-step exploration direction $\boldsymbol{d_{i-1}}$** from the Rewriter $\mathcal{M}_\mathrm{R}$, and **the previous history $\boldsymbol{\mathcal{H}_i = \{(e_j, r_j) | j < i\}}$.**

Methods

Experience Gathering

$$e_i = \mathcal{M}_E(t_{i-1}, d_{i-1}, \mathcal{H}_i), \quad (1)$$

Where e_i consists of a **user query** $\boldsymbol{e_i^q}$ related to the tool and **the necessary parameters** $\boldsymbol{e_i^p}$. The initial tool documentation is denoted as t_0, representing the raw documentation provided in the dataset. After generating e_i, the Explorer invokes the tool to obtain the result r_i returned by the tool.

Methods

Experience Gathering

Diversity-promoting exploration strategy:

Similarity Constraint. When generating a new instance, the `Explorer` calculates the cosine similarity between the new generated query e_i^q and all prior queries e_j^q for $j < i$, using embedding vectors obtained from OpenAI's *text-embedding-ada-002*[1]. The similarity is computed as:

$$\max_{j<i} \mathrm{sim}(\mathbf{e}_i^q, \mathbf{e}_j^q) < \phi, \quad (2)$$

Self-Reflection. If the similarity constraint is not satisfied, It discards the current instance and analyzes the reasons for the overlap, adjusting its approach to generate a new query that explores different aspects of the tool.

Methods

Experience Gathering

```
// Experience Gathering (§ 2.2)
Instruct Explorer to generate an exploratory instance e_i using Eq. (1)
while max_{j<i} sim(e_i^q, e_j^q) > φ do
|   Instruct Explorer to generate a new exploratory instance e_i
end
Instruct Explorer to capture the outcomes of tool execution r_i
```

Methods

Experience Gathering

Explorer

Task Prompt:

Your task is to answer the user's query as best you can. You have access to the following tools, which you can use via tool calling to help with your response: **{Tool Documentation}**

Now you have the chance to explore the available tools. You can do this by ...

Here is an example: **{"User Query": " ", "Parameters"::{ }}**

Memory Mechanisms:

Below are queries you have already explored: **{Explored queries}**

Based on these, try to explore queries that ...

Here are some suggestions to explore the tool: **{Suggestions}** ...

prompt template for `Explorer`.

Methods

Learning from Experience

In this phase, we introduce an Analyzer $\mathcal{M}_{\mathrm{A}}$.

$$s_i = \mathcal{M}_A(t_{i-1}, e_i, r_i, \mathcal{T}_i). \quad (3)$$

Formally, at the i-th iteration, the Analyzer $\mathcal{M}_A$ takes the following inputs: **current tool documentation $\boldsymbol{t_{i-1}}$**, **exploration instance $\boldsymbol{e_i}$**, **tool feedback $\boldsymbol{r_i}$** provided by the Explorer $\mathcal{M}_E$, and **the history of documentation revisions $\mathcal{T}_i = \{\boldsymbol{t_j} | \boldsymbol{j} < \boldsymbol{i}\}$.** Then the Analyzer $\mathcal{M}_A$ analyzes these inputs to identify issues and generate revision suggestions s_i

Methods

Learning from Experience

```
// Learning from Experience (§ 2.3)
Instruct Analyzer to learn from experience and provide suggestions s_i for modifications using Eq. (3)
```

Analyzer

Task Prompt:

You task is to provide suggestions for modifying the tool documentation based on the current tool documentation, the explored queries and parameters, and the results returned by the tool. You have access to the following tools: **{Tool Documentation}** ...
Below are explored queries, the required parameters and the outputs of the tool: **{Explored_examples}**
Here is an example: **{"Suggestions": " "}**

Memory Mechanisms:

The following is the history of you modifying the tool description: **{History}**
Based on the above information, provide more constructive suggestions.

1. **一致性**：当前描述是否与工具实际返回结果一致?
2. **全面性**：文档是否覆盖所有关键功能、边界条件和错误处理?
3. **简洁性**：是否存在冗余或无关信息?

Methods

Documentation Rewriting

$$d_i, t_i = \mathcal{M}_R(t_{i-1}, e_i, r_i, s_i, \mathcal{T}_i). \quad (5)$$

exploration instances e_i **corresponding tool return results** r_i **revision suggestions** s_i **rewrite history** $\mathcal{T}_i$

By integrating these inputs, the Rewriter $\mathcal{M}_R$ produces an updated version of the tool documentation t_i and provides suggestions for the next round of exploration directions d_i.

Methods

Documentation Rewriting

Tool-adaptive termination mechanism:when there is minimal change between two consecutive versions of the documentation, indicating that the Rewriter has sufficiently aligned the documentation with the LLM’s understanding.
Measure the degree of change Δ between iterations by calculating both the **word-match metric** and **the semantic-match metric.**

$$\Delta = \frac{\mathrm{sim}(\mathbf{e}_i^t, \mathbf{e}_{i-1}^t) + \mathrm{BLEU}(t_i, t_{i-1})}{2}, \qquad (5)$$

where e_i^t and e_{i-1}^t are the embedding vectors of t_i and t_{i-1} obtained using OpenAI’s *text-embedding-ada-002*[2]. The function sim $(\cdot,\cdot)$ calculates the cosine similarity between the semantic embedding vectors of two documentation versions, and BLEU$(\cdot,\cdot)$measures the n-gram overlap between them. If Δ exceeds a predefined termination threshold τ, we stop the iterative modifications.

Methods

Documentation Rewriting

```
// Documentation Rewriting (§ 2.4)
Instruct Rewriter to revise the documentation based on experience and suggestions to get
  revised tool documentation t_i and propose new exploration directions d_i using Eq. (4)
Calculate the similarities Δ between t_{i-1} and t_i using Eq. (5)
if Δ > τ then
  |  Break
end
```

DRAFT automates tool documentation updates through trial-and-error learning, reducing manual effort while improving alignment with LLMs' operational understanding. It dynamically adapts documentation to evolving tool features, ensuring accuracy and relevance.

Rewriter

Task Prompt:

Your task is to refine the description of the tool based on the queries and parameters you've explored, the outputs provided by the tool, as well as the suggestions provided. You have access to the following tools: **{Tool Documentation}** ...

Below are explored queries, the required parameters and the outputs of the tool: **{Explored_examples}**

Additionally, here are some suggestions for updating the tool description: **{Suggestions}** ...

Here is an example: { **"Rewritten description": "", "Suggestions for exploring": ""**}

Memory Mechanisms:

The following is the history of you modifying the tool description: **{History}**

Based on the information above, rewrite the tool description and provide more constructive suggestions for exploration.

Methods

Documentation Rewriting

Algorithm 1: The Learning Algorithm of DRAFT

Input: Raw tool documentation set $\mathcal{D}$, iteration round I, similarity threshold ϕ, termination threshold τ.

Output: Revised tool documentation set $\tilde{\mathcal{D}}$.

Initialize the revised tool documentation set $\tilde{\mathcal{D}} \leftarrow \emptyset$
for *raw tool documentation* $t \in \mathcal{D}$ **do**
    for $i = 1$ *to* I **do**
        `// Experience Gathering` (§ 2.2)
        Instruct `Explorer` to generate an exploratory instance e_i using Eq. (1)
        while $\max_{j<i} \mathrm{sim}(\mathbf{e}_i^q, \mathbf{e}_j^q) > \phi$ **do**
            Instruct `Explorer` to generate a new exploratory instance e_i
        end
        Instruct `Explorer` to capture the outcomes of tool execution r_i
        `// Learning from Experience` (§ 2.3)
        Instruct `Analyzer` to learn from experience and provide suggestions s_i for modifications using Eq. (3)
        `// Documentation Rewriting` (§ 2.4)
        Instruct `Rewriter` to revise the documentation based on experience and suggestions to get revised tool documentation t_i and propose new exploration directions d_i using Eq. (4)
        Calculate the similarities Δ between t_{i-1} and t_i using Eq. (5)
        if $\Delta > \tau$ **then**
            Break
        end
    end
    Updating the revised tool documentation set $\tilde{\mathcal{D}} \leftarrow \tilde{\mathcal{D}} \cup t_i$
end
return $\tilde{\mathcal{D}}$

Full pseudocode

Evaluation

Datasets: ToolBench RestBench

Evaluation Metrics:

- Correct Path Rate(CP%):测量模型生成的工具调用序列中包含真实工具路径（作为子序列）的实例比例（是否符合正确步骤）。
- Win Rate(Win%):通过基于ChatGPT的评估器进行成对比较来评估有效性，能够捕捉基于规则指标无法反映的细微性能差异。（该方法弥补了传统自动化指标（仅关注表面匹配）的不足，通过大模型模拟人类判断，识别生成结果中更复杂的质量差异（如语义合理性、上下文一致性）。

Baselines: ReAct DFSDT EasyTool（用于工具调用）

Implementation Details:use the *GPT-4o* as the backbone model for DRAFT, set the similarity threshold φ to 0.9, termination threshold τ to 0.75, maximum iteration count to 5 use closed-source models *GPT-4o* and *GPT-4o-mini* , as well as the open-source model *Llama-3-70B* to evaluate.

Evaluation

Model	Method 方法	RestBench-TMDB		RestBench-Spotify		ToolBench 工具台	
		CP%	Win%	CP%	Win%	CP%	Win%
GPT-4o-mini	ReAct	48.00	50.00	24.56	50.00	35.00	50.00
	DFSDT	50.00	68.00	35.08	61.40	37.00	84.00
	EasyTool	56.00	75.00	-	-	42.00	85.00
	DRAFT (Ours) 草稿（ours)	**62.00**	**82.00**	**43.85**	**78.94**	**47.00**	**88.00**
Llama-3-70B	ReAct	72.00	50.00	26.31	50.00	41.00	50.00
	DFSDT	74.00	38.00	63.15	61.40	42.00	54.00
	EasyTool	76.00	**64.00**	-	-	46.00	60.00
	DRAFT (Ours) 草稿（ours)	**86.00**	**64.00**	**66.66**	**64.91**	**53.00**	**62.00**
GPT-4o	ReAct	71.00	50.00	28.07	50.00	37.00	50.00
	DFSDT	74.00	61.00	64.91	56.14	41.00	73.00
	EasyTool	79.00	62.00	-	-	45.00	77.00
	DRAFT (Ours) 草稿（ours)	**88.00**	**71.00**	**70.17**	**84.21**	**51.00**	**78.00**

Win% is calculated by comparing each method with ReAct.(是一个相对值)

Evaluation

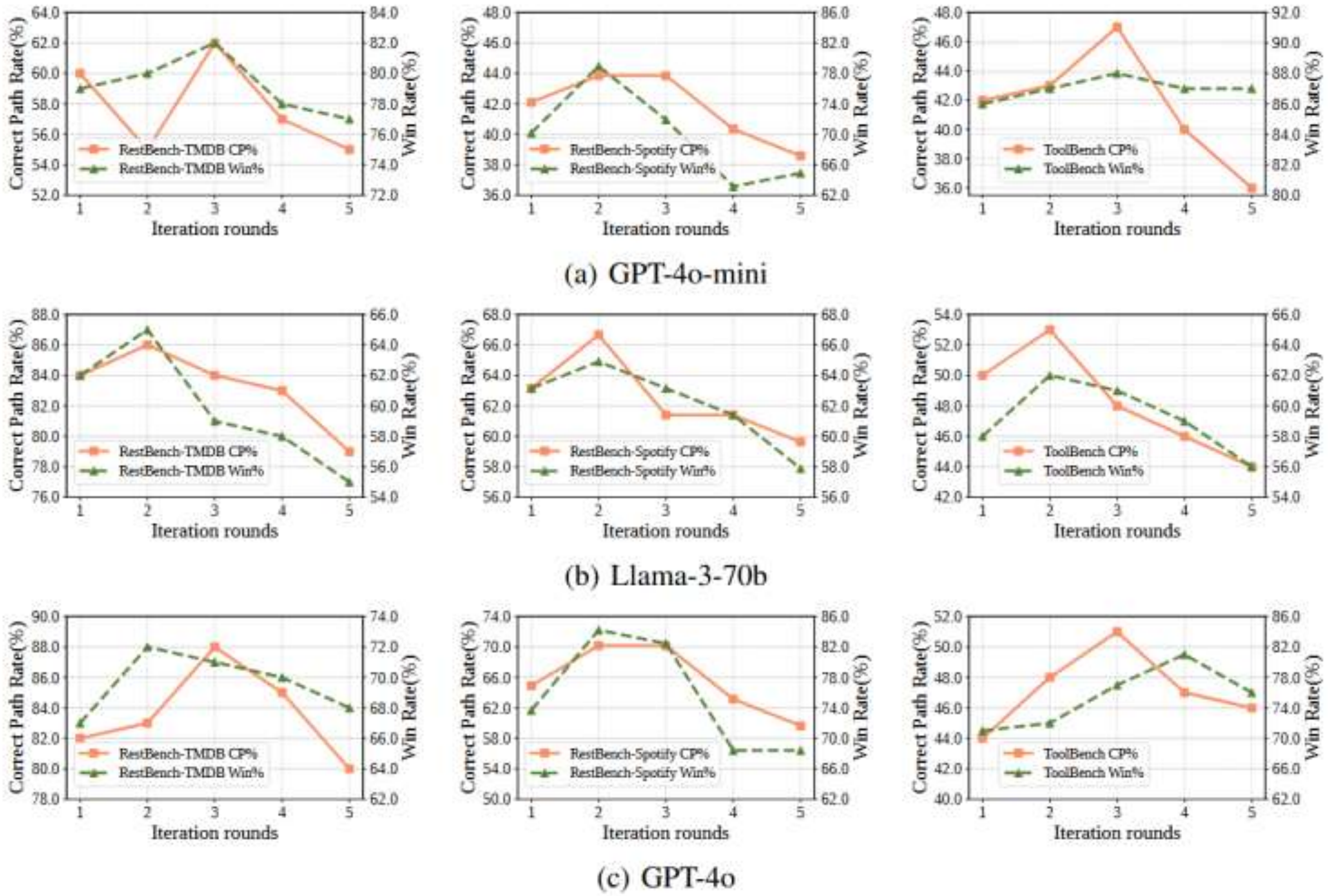


Figure 6: A comparative analysis of performance based on varying numbers of iteration rounds.

However, a **decline** in performance is observed after a certain number of iterations. This decline may be attributed to the introduction of **redundant information** as the number of iterations increases, potentially leading to **overfitting**. Therefore, we implement a **tool-adaptive termination mechanism** to prevent performance degradation and ensure optimal results.

Evaluation

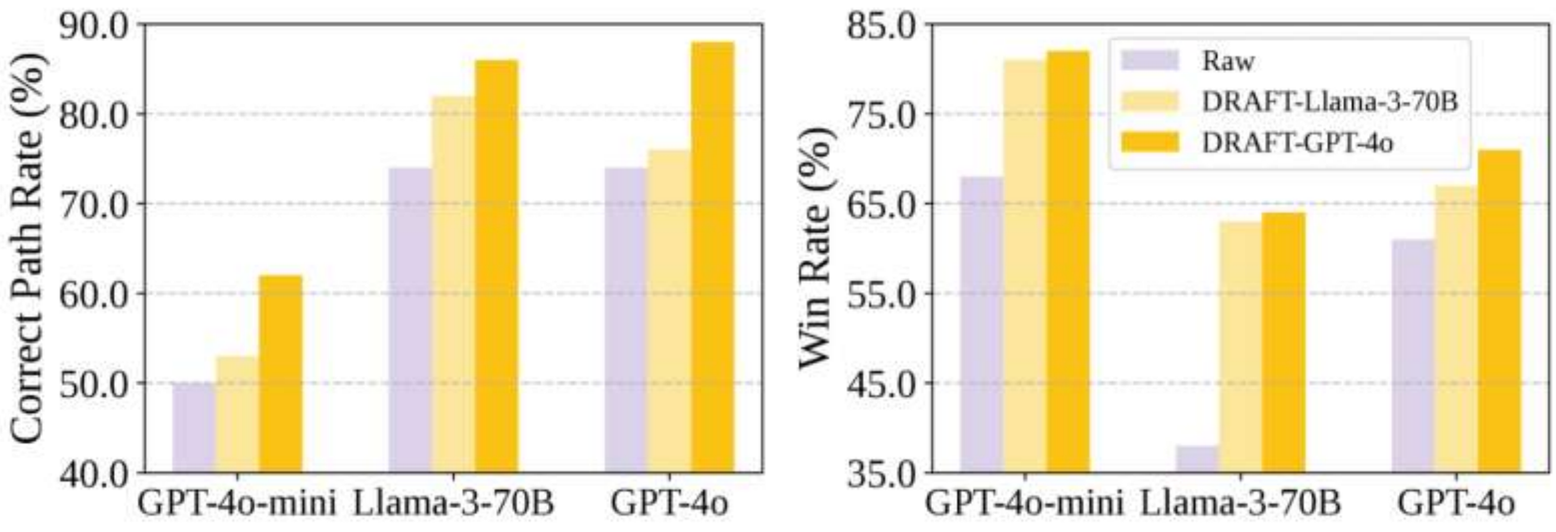


Figure 7: An analysis of cross-model generalization.

use other models as backbones also ensure cross-model generalization!

Evaluation

Table 2: Ablation study of the proposed DRAFT.

Mehtods	TMDB	
	CP%	Win%
DRAFT	**88.00**	**71.00**
w/o diversity	84.00	69.00
w/o adaptive	80.00	68.00

多样性策略和自适应策略的有效性

Table 3: Comparison of tool retrieval performance between raw documentation and our method. We report NDCG@1 and NDCG@10.

Retriever	Documentation	TMDB		Spotify	
		@1	@10	@1	@10
BM25	Raw	24.0	35.0	**43.9**	53.9
	DRAFT	**29.0**	**39.4**	**43.9**	**54.2**
Contriever	Raw	29.0	40.4	45.6	**49.6**
	DRAFT	**31.0**	**44.1**	**47.4**	49.2

修改后的文档提高了工具检索的性能

Evaluation

Table 4: Human evaluation comparing the quality of the raw documentation with ours.

Dataset	Completeness			Conciseness			Accuracy		
RestBench	**Ours** 40%	**Raw** 16%	**Equal** 44%	**Ours** 36%	**Raw** 20%	**Equal** 44%	**Ours** 30%	**Raw** 0%	**Equal** 70%
ToolBench	**Ours** 68%	**Raw** 4%	**Equal** 28%	**Ours** 56%	**Raw** 4%	**Equal** 40%	**Ours** 56%	**Raw** 0%	**Equal** 44%

修改后的工具文档更容易被人类理解

Thinking

- **文章有什么问题，基于这个 paper 还能做什么?**

不足:

- **实验数据集的局限性：**研究主要基于特定工具库（如ToolBench和RestBench）进行验证，缺乏对更复杂或动态环境的测试。
- **评估细节不完整：**论文提到依据“一致性、全面性、简洁性”等标准生成修订建议，但附录中未详细公开评估指标的具体计算方式，只是提示词文档里提到。
- **参数是否合理：**没有探讨计算文档相似度前的参数是否最优。

进一步工作:

- 可以根据工具文档的合理性（由LLMs判断根据生成的工具文档的工具功能是否完善）去完善不太合理的工具。

Thanks

Presenter: Jinhan Xin

2025.05.09