

# Task-Aligned Tool Recommendation for Large Language Models

Anonymous ACL submission

## Abstract

By augmenting Large Language Models (LLMs) with external tools, their capacity to solve complex problems has been significantly enhanced. However, despite ongoing advancements in the parsing capabilities of LLMs, incorporating all available tools simultaneously in the prompt remains impractical due to the vast number of external tools. Consequently, it is essential to provide LLMs with a precise set of tools tailored to the specific task, considering both quantity and quality. Current tool retrieval methods primarily focus on refining the ranking list of tools and directly packaging a fixed number of top-ranked tools as the tool set. However, these approaches often fail to equip LLMs with the optimal set of tools prior to execution, since the optimal number of tools for different tasks could be different, resulting in inefficiencies such as redundant or unsuitable tools, which impede immediate access to the most relevant tools. This paper addresses the challenge of recommending precise toolsets for LLMs. We introduce the problem of tool recommendation, define its scope, and propose a novel Precision-driven Tool Recommendation (PTR) approach. PTR captures an initial, concise set of tools by leveraging historical tool bundle usage and dynamically adjusts the tool set by performing tool matching, culminating in a multi-view-based tool addition. Additionally, we present a new dataset, RecTools, and a metric, TRACC, designed to evaluate the effectiveness of tool recommendation for LLMs. We further validate our design choices through comprehensive experiments, demonstrating promising accuracy across two open benchmarks and our RecTools dataset.

## 1 Introduction

Large Language Models (LLMs) have established themselves as powerful intermediaries, demonstrating remarkable impacts across a variety of downstream tasks, including text generation, code debugging, and personalized recommendations (Brown

et al., 2020; Touvron et al., 2023; Nam et al., 2024; Chen et al., 2024; Zhao et al., 2024). However, as these models continue to evolve, they still struggle to solve highly complex problems due to limitations arising from their pre-training data (Mialon et al., 2023; Mallen et al., 2022; Yuan et al., 2023). To expand the potential of LLMs in managing more complex tasks efficiently, recommendations at various levels have been increasingly applied to LLMs. Typically, memory recommendations (Borgeaud et al., 2022) and knowledge-based recommendations (Gao et al., 2023; Hu et al., 2023) enhance consistency and context awareness in ongoing tasks for LLMs, while data augmentation recommendations (Xu et al., 2020) facilitate the inclusion of additional data to augment training. Furthermore, architecture recommendations (Elsken et al., 2019; Fedus et al., 2022) and prompt recommendations (Shin et al., 2020; Pryzant et al., 2023; Liu et al., 2023) optimize efficiency and generate more relevant outputs. Simultaneously, to reduce the cognitive load on LLMs and enhance their complex problem-solving capabilities by enabling actions beyond natural language processing, it is crucial to augment LLMs with recommendations of optimal external tool sets, an aspect currently lacking in existing recommendation frameworks for LLMs. Furthermore, this approach will be helpful to address the challenge of input length limitations encountered when incorporating a large number of external tools into the prompt. Providing LLMs with a precise and dynamically adaptable recommended toolset can help to enhance the effectiveness of LLM’s task-solving ability.

Considering that the capability of LLMs to master and control external tools is instrumental in overcoming some of their fundamental weaknesses, the field of tool retrieval—which aims to identify the top-K most suitable tools for a given query from a vast set of tools—has been increasingly explored. The advent of tool retrieval (Zhuang et al., 2023; Li

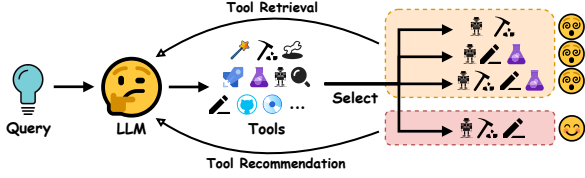


Figure 1: Tool retrieval often provides a broad and variable number of tools with inconsistent quality, whereas tool recommendation delivers a precise, high-quality set of tools directly.

et al., 2023; Tang et al., 2023; Yang et al., 2024) signifies a nuanced evolution, most directly employing term-based methods (Sparck Jones, 1972; Robertson et al., 2009) or semantic-based techniques (Kong et al., 2023; Yuan et al., 2024; Gao et al., 2024). Generally, the primary objective of these methods is to refine the ranked list of tools and subsequently select a fixed number of tools from the top (top-K) (Qu et al., 2024a; Zheng et al., 2024; Qu et al., 2024b). Although such approaches have demonstrated good performance when retrieving a single tool (Patil et al., 2023; Xu et al., 2023) or a small number of tools (generally fewer than three) (Qin et al., 2023; Huang et al., 2023), they remain susceptible to under-selection or over-selection, as illustrated in Figure.1. This limitation may prevent LLMs from addressing the current query or cause them to over-interpret the query, thereby reducing the effectiveness of LLMs in solving complex problems with external tools. Additionally, the validation of these methods often relies on datasets that use a fixed number of tools for each query, meaning that during testing, the number of tools to be used is known in advance—an unrealistic scenario in practical applications where the number of tools needed can vary dynamically. Therefore, recommending a precise and dynamically adjustable set of external tools to LLMs in a single step prior to query execution is increasingly important. This approach not only enhances the thoroughness of problem-solving but also improves efficiency by reducing the need to execute additional tools.

To address these limitations, we first provide a comprehensive explanation of tool recommendation and clearly define the problem ((Appendix.A)), considering the lack of definition and the incompleteness of goals pursued by existing tool retrieval methods. Toward this objective, we propose PTR, a novel model-agnostic **Precision-Driven Tool Recommendation** approach aimed at recommending a precise tool set for LLMs prior to query

execution. By leveraging historical tool bundle usage data to uncover patterns of idiomatic use and dependencies between tools, this method is structured into three main stages: *Tool Bundle Acquisition*, *Functional Coverage Mapping*, and *Multi-view-based Re-ranking*. Initially, using traditional pre-trained language models, we acquire semantic matching information between queries and previously used tool bundles, thereby addressing potential performance issues of these models in zero-shot scenarios for tool recommendation tasks. Subsequently, to evaluate the effectiveness of the selected tool bundle in solving the query, LLMs are prompted to match tools with the specific subproblems they can address and to identify unresolved issues. Based on this, a multi-view-based re-ranking method is employed to select tools that can help resolve the identified issues and complement the existing tool sets. More specifically, to address the unresolved issues, we construct the final ranked list by aggregating three tool lists and ranking each tool based on their frequency of occurrence. The ranked tool list, constructed from multiple views, reduces the randomness associated with selecting tools from the entire available set.

Additionally, we construct a dataset, **RecTools**, tailored to specific queries with recommended tool sets. In contrast to previous tool datasets that standardize the number of tools used for each query (Huang et al., 2023) or employ a small number of tools (Qu et al., 2024a), our tool recommendation set incorporates varying numbers of tools for different queries, with up to ten tools used for a single query. This is achieved through an automated process in which LLMs are prompted to generate specific queries to be addressed by given tool bundles. These queries and tool bundles are subsequently evaluated by prompting LLMs to determine whether the selected tools adequately address the corresponding queries, ensuring that neither excess nor insufficient tools are utilized. Dedicated validation and deduplication steps are implemented to ensure the precision of tool usage, thereby enhancing the quality of the tool recommendation set.

Furthermore, traditional retrieval metrics such as Recall (Zhu, 2004) and Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002), fail to capture the level of precision required for effective tool recommendation. The absence of necessary tools can lead to the failure of LLMs in performing tasks, while the redundancy

of tools may cause LLMs to generate unnecessary responses. This indicates that metrics focusing solely on completeness are inadequate for evaluating tool recommendation tasks. To bridge this gap, we introduce **TRACC**, a novel metric designed to assess tool recommendation performance, considering both the accuracy of the quantity and the quality of the recommended tools. TRACC serves as a reliable indicator of the effectiveness of tool recommendation processes.

To summarize, the main contributions of this work are as follows:

1. We introduce tool recommendation as a novel problem, necessitating the provision of precise tool sets to LLMs for a given query. We propose PTR, an effective tool recommendation approach that leverages historical tool bundle information between queries and tools, resulting in a more accurate and comprehensive final recommended tool list.

2. We present a new dataset, RecTools, and an effective evaluation metric, TRACC, specifically designed to assess tool recommendation for LLMs. This not only addresses gaps in existing tool sets but also advances future research related to tool recommendation.

3. Extensive experiments validate the effectiveness of RecTools and demonstrate the efficacy of PTR in recommending tools for LLMs. The recommended tool sets are both comprehensive and accurate, enhancing the overall performance of LLMs in processing tasks.

## 2 The Precision-driven Tool Recommendation

We introduce a novel approach, Precision-driven Tool Recommendation (PTR), to address the challenges faced by prior research through a three-stage recommendation process: (1) Tool Bundle Acquisition, which involves establishing a potentially useful tool bundle by leveraging past usage patterns across all tool combinations, as opposed to relying solely on instructions for individual tool usage; (2) Functional Coverage Mapping, which entails effectively mapping the tools from the acquired tool bundle to the functionalities of the original query, thereby identifying which tools should be retained and which should be discarded, resulting in any remaining unsolved sub-problems; and (3) Multi-view-based Re-ranking, which involves the effective re-ranking of relevant tools from a large

tool set, tailored to each unsolved sub-problem identified in the second stage, and selecting the top-ranked tool after re-ranking to complete the final recommended toolset. The overview of our approach is illustrated in Figure.2. Please note that all symbols are globally defined in sections A and 2. In the following sections, we present the details of these three PTR recommendation stages.

### 2.1 Tool Bundle Acquisition

To obtain an initiate set of tools, we employ a retriever to capture the relevance between historical tool combinations and the current query. Unlike existing methods that focus on retrieving single tools by analyzing the relationship between a query and individual tools, our approach introduces tool bundle retrieval. By leveraging historical tool combinations, we capture a richer contextual relationship between queries and sets of tools that have been used together effectively in the past. This facilitates a more holistic understanding of tool dependencies and synergies, thereby enhancing the relevance of retrieved tool sets for complex queries. Specifically, Let  $T = \{T_1, T_2, \dots, T_n\}$  be the set of all available tools. Let  $D = \{(Q_i, B_i)\}_{i=1}^M$  represent a set of past queries and their associated tool bundles, where  $Q_i$  is a past query, and  $B_i$  is the corresponding tool bundle used for  $Q_i$ , with  $B_i \subseteq T$ . The collection of unique tool bundles is  $B = \{B_1, B_2, \dots, B_N\}$ . Given a new query  $Q$ , we select a tool bundle  $B_K = \{T_1, \dots, T_z\}$  from  $B$  that is most relevant to  $Q$  through the retriever, which ideally contains tools potentially useful. The subsequent recommendations operate on this obtained tool bundle—either based on sparse representations or dense representations.

### 2.2 Functional Coverage Mapping

As illustrated in Figure.3, functional coverage mapping presents a structured approach to evaluate and optimize a set of tools in relation to a specific query. By systematically aligning required functionalities with the capabilities of available tools, this method ensures that the toolset comprehensively addresses the user’s needs while minimizing redundancies and identifying any gaps, as each tool may correspond to multiple functionalities. At its core, Functional Coverage Mapping aims to determine whether an initial set of tools  $B_K = \{T_1, T_2, \dots, T_z\}$  adequately fulfills a query  $Q$  with its key functionalities  $F = \{F_1, F_2, \dots, F_m\}$ . Specifically, Functional Cover-

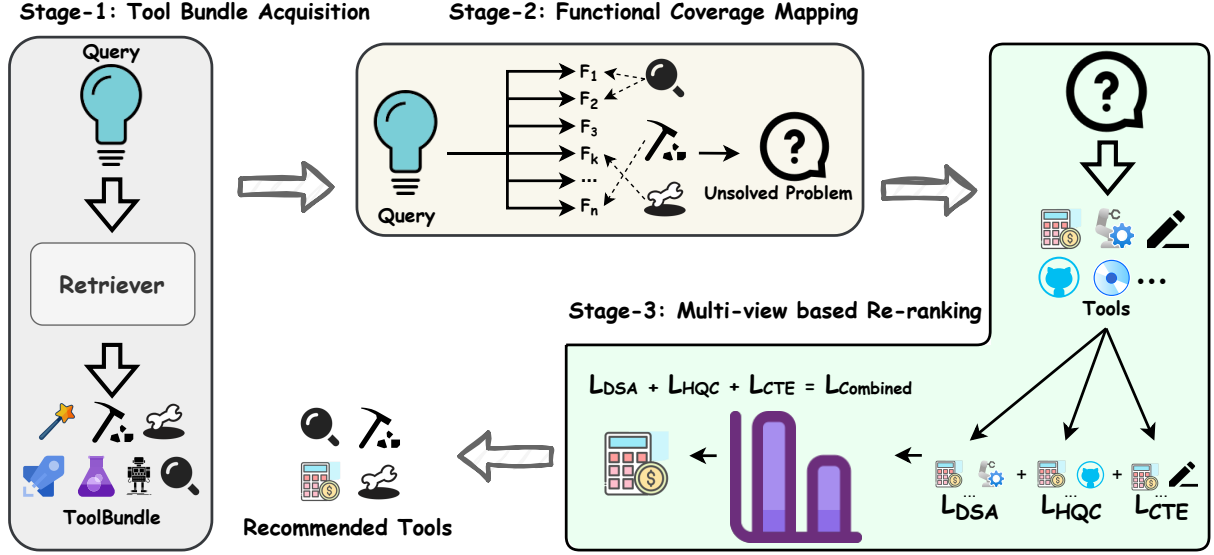


Figure 2: Architecture of the three-stage recommendation framework PTR for tool recommendation.

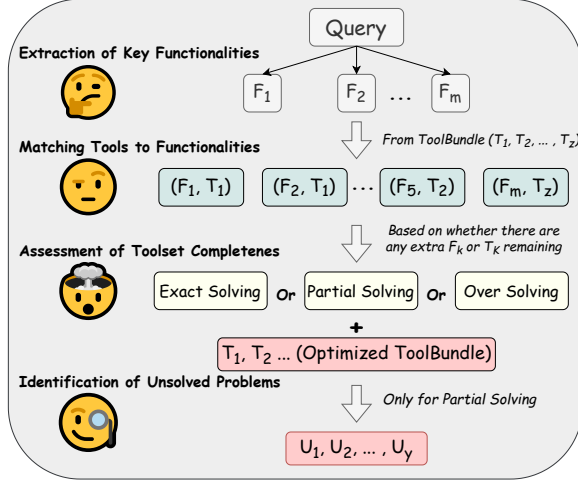


Figure 3: The four stages of Functional Coverage Mapping in PTR.

age Mapping achieves this objective through four steps: *Extraction of Key Requirements*, *Matching Tools to Functionalities*, *Assessment of Toolset Completeness*, and *Identification of Unsolved Problems*, which are described as follows:

**Extraction of Key Functionalities.** The first step involves decomposing the user’s query  $Q$  into a set of discrete and actionable functionalities  $R$ . This extraction ensures a comprehensive understanding of the query that the toolset must address. This extraction is achieved by prompting the language model to identify and enumerate these functionalities directly from the query, ensuring that both explicit and implicit functionalities are captured. **Matching Tools to Functionalities.** Once the key functionalities  $F$  are established, the subsequent phase entails mapping each functionality  $F_i$  to the

tools  $T_j$  within the obtained tool bundle  $B_K$ . This mapping process determines which tools are capable of fulfilling specific functionalities. To achieve this, targeted prompts are employed with the language model, directing it to associate each functionality with the most suitable tool based on tool descriptions.

**Assessment of Toolset Completeness.** With the mapping  $M(F, B_K)$  established, the method evaluates whether the toolset  $B_K$  fully addresses all functionalities  $F$ . This assessment categorizes the toolset into one of three scenarios: (1) Exact Solving: All functionalities are met by all tools without any redundancies; (2) Oversolving: The toolset includes tools that provide functionalities not required by the query; and (3) Partial Solving: Some functionalities remain unfulfilled and some tools remain unused. Based on the identified scenario, the tool bundle is optimized by retaining essential tools and discarding redundant ones. Tools that do not contribute to fulfilling any requirement are removed to streamline the toolset.

**Identification of Unsolved Problems.** In cases of partial solving, the method identifies the remaining unsolved problems directly from the original query  $Q$ . These unsolved problems  $U = \{U_1, U_2, \dots, U_y\}$  are presented in a format that can be directly utilized in the subsequent recommendation stage. To achieve this, the language model is prompted to extract the unmet functionalities without further functional decomposition. This approach ensures that each unsolved problems retains the context of the original query  $Q$ , thereby



facilitating seamless integration with the following re-ranking method. Furthermore, this direct identification allows for straightforward utilization in the following re-ranking process, where each unsolved problem can be addressed individually.

### 2.3 Multi-view based Re-ranking

Addressing the challenge of selecting pertinent tools from an extensive toolset to resolve unresolved problems requires comprehensive consideration. The proposed PTR employs a multifaceted similarity evaluation strategy that integrates three essential dimensions of the unresolved problem  $U_j$ : (1) **Direct Semantic Alignment**, wherein the system quantifies the semantic similarity between the user query and each available tool, ensuring the immediate identification of tools intrinsically aligned with the query’s intent; (2) **Historical Query Correlation**, which involves analyzing past queries that closely resemble the current one to extract tools previously utilized in similar contexts, thereby enriching the current toolset with empirically effective solutions while maintaining uniqueness through aggregation and deduplication; and (3) **Contextual Tool Expansion**, which leverages the most relevant tool identified through direct semantic alignment to retrieve additional tools exhibiting high similarity to this primary tool, thereby uncovering supplementary options that may offer complementary or alternative functionalities beneficial to the user’s query. The multi-view matching process involves obtaining the tool list  $L$  through direct semantic alignment (DSA), historical query correlation (HQC), and contextual tool expansion (CTE), respectively. These three tool lists are then aggregated and ranked according to their frequency of occurrence, with the most frequent tools being selected. After performing the multi-view-based re-ranking for each unsolved problem, the top-ranked tool in each list is selected and added to the final recommended toolset. In some cases, it is also possible that this tool already exists in the toolset acquired from the second-stage recommendation; in such instances, the tool will be ignored. The algorithm for multi-view-based re-ranking is summarized in Algorithm.1.

### 3 Datasets and Metrics

*Datasets.* To verify the effectiveness of PTR, we utilize three datasets for tool recommendation: ToolLens (Qu et al., 2024a), MetaTool (Huang

#### Algorithm 1 Multi-view Based Re-ranking

---

**Require:** Unresolved problem  $U_j$ , Toolset  $T = \{T_1, T_2, \dots, T_n\}$ , Historical queries  $Q = \{Q_1, Q_2, \dots, Q_m\}$ ,  $\text{Select}_K$  represents the function that selects the top  $K$  candidates with the highest similarity,  $\sigma$  indicates the similarity measure.

**Ensure:** Recommended Tool  $\mathcal{T}$ .

- 1: Initialize lists:  $L_{\text{DSA}}, L_{\text{HQC}}, L_{\text{CTE}}$ .  
//Direct Semantic Alignment
- 2:  $L_{\text{DSA}} \leftarrow \text{Select}_K(\{T_i \in T \mid \sigma(U_j, T_i)\})$   
//Historical Query Correlation
- 3:  $L_{\text{HistoricalQuery}} \leftarrow \text{Select}_K(\{Q_i \in Q \mid \sigma(U_j, Q_i)\})$
- 4: **for** each query  $Q_i$  in  $L_{\text{HistoricalQuery}}$  **do**
- 5:   **for** each tool  $T_l$  used in  $Q_i$  **do**
- 6:     Add  $T_l$  to  $L_{\text{HQC}}$
- 7:   **end for**
- 8: **end for**
- 9: Remove duplicates from  $L_{\text{HQC}}$ .  
//Contextual Tool Expansion
- 10: **if**  $L_{\text{DSA}}$  is not empty **then**
- 11:    $T_{\text{primary}} \leftarrow L_{\text{DSA}}[0]$
- 12:    $L_{\text{CTE}} \leftarrow \text{Select}_K(\{T_i \in T \mid \sigma(T_{\text{primary}}, T_i)\})$
- 13: **end if**
- 14: Combine lists:  $L_{\text{Combined}} \leftarrow L_{\text{DSA}} + L_{\text{HQC}} + L_{\text{CTE}}$
- 15: Count frequency of each tool in  $L_{\text{Combined}}$ .
- 16: Rank tools by frequency in descending order.
- 17: Select the top ranked tool as  $\mathcal{T}$ .

---

**return**  $\mathcal{T}$ .

---

et al., 2023), and a newly constructed dataset, **Rec-Tools**. We randomly select 20% of each dataset to serve as the test data. Both ToolLens and MetaTool focus on multi-tool tasks, leading us to select them as the primary datasets for our experiments. While ToolLens uniquely emphasizes creating queries that are natural, concise, and intentionally multifaceted, MetaTool is a benchmark designed to evaluate whether LLMs possess tool usage awareness and can correctly choose appropriate tools.

*Metrics.* As evaluation metrics for tool recommendation, following previous work focusing on tool retrieval (Gao et al., 2024; Qu et al., 2024b), the widely used retrieval metrics are Recall and NDCG. However, they do not adequately address the requirements for accuracy in both the number of recommended tools and the specific tools recommended, disregarding the impact of differences in size between the tool sets. Therefore, to further tailor the assessment to the challenges of tool recommendation tasks, we introduce a new metric, named **TRACC**. This metric is designed to measure the extent to which the recommended toolset aligns with the ground-truth set in terms of both the accuracy of the number of tools and the accuracy of the tools themselves:

$$\text{TRACC} = \left(1 - \frac{1}{|A \cup B|} \cdot |n_2 - n_1|\right) \cdot \text{ACC}$$

where  $A$  denotes the ground-truth tool set and  $B$  represents the recommended tool set. The cardinalities of  $A$  and  $B$  are denoted by  $n_1$  and  $n_2$ , respectively. And  $|A \cup B|$  signifies the cardinality of the union of  $A$  and  $B$ . ACC represents  $\frac{|A \cap B|}{n_1}$ , where  $|A \cap B|$  indicates the size of their intersection.

## 4 Experiments

### 4.1 Implementation Details

**Baselines.** We considered the following baselines: **Random**, which randomly select from historical tools; **BM25** (Robertson et al., 2009), a classical sparse retrieval method that extends TF-IDF by leveraging term frequency and inverse document frequency of keywords; **Contriever** (Izacard et al., 2021), which utilizes inverse cloze tasks, cropping for positive pair generation, and momentum contrastive training to develop dense retrievers; **SBERT** (Reimers and Gurevych, 2019), a library providing BERT-based sentence embeddings. Specifically, we use all-mpnet-base-v2; **TAS-B** (Hofstätter et al., 2021), the retriever introduces an efficient topic-aware query and balanced margin sampling technique; And **SimCSE** (Gao et al., 2021), a simple contrastive learning framework that greatly advances state-of-the-art sentence embeddings.

Besides, we initially implement the PTR using the open source model open-mistral-7b, due to its cost-effectiveness. Subsequently, we evaluate PTR with the model GPT-3.5-turbo and GPT-4o, to determine its effectiveness when employing a more advanced model. For evaluation metrics, in addition to the specifically designed TRACC metric, we also calculate Recall@K and NDCG@K, reporting these metrics with K set to the size of the ground-truth tool set.

### 4.2 Experimental Results

Table 1 presents the main results of the PTR applied to ToolLens, MetaTool, and RecTools using various models and unsupervised retrievers. Based on these findings, we draw the following observations and conclusions.

We first observe that the MetaTool dataset yields notable performance, whereas other datasets exhibit comparatively standard. This discrepancy can be attributed to the presence of relatively straightforward patterns within the MetaTool dataset, which motivates us the construction of a structurally diversified and high-quality tool-query

dataset. Furthermore, the Random baseline indicates that random sampling of tool bundles leads to relatively poor performance, whereas other unsupervised retrievers outperform the Random baseline, particularly in the ToolLens dataset. This suggests that, although the latter two phases of the PTR can supplement or refine the recommended tool set, employing a targeted bundle in the early stages can enhance PTR performance. Conversely, the SimCSE approach demonstrated a significant improvement over the Random baseline, especially when utilizing GPT-4o as the backbone. Absolute Recall@K improvements of 0.141, 0.111, and 0.117 were observed on the ToolLens, MetaTool, and RecTools datasets, respectively, highlighting the SimCSE method’s capability to leverage tool bundle information for more effective tool recommendation. Despite this advantage, all the methods fall short in the TRACC metric, which is specifically designed for evaluating precision in tool recommendation. This suggests that, although effective for tool retrieval tasks, Recall@K and NDCG@K may not fully satisfy the unique requirements of tool recommendation. Additionally, the results demonstrate that PTR consistently achieves strong performance when utilizing GPT-4o, confirming that PTR remains beneficial for tool recommendation even when employing more capable backbone models.

Overall, PTR exhibits effectiveness across all metrics and datasets, attributable to its implementation of a three-stage recommendation framework. This framework comprises tool bundle acquisition, functional coverage mapping for the deletion or retention of tools, and multi-view-based re-ranking for the addition of tools. By employing this structured approach, PTR dynamically addresses the entirety of the query, thereby facilitating the recommendation of a precise and well-tailored tool set.

### 4.3 Further Analysis

In this section, we conduct an in-depth analysis of the effectiveness for PTR, using the same datasets and evaluation metrics. The results are presented in Table 2.

**w/o Tool Bundle Acquisition.** This variant omits the tool bundle acquisition stage, resulting in queries being exclusively mapped to unresolved problems without any existing recommended tools. The observed decline in performance for this variant further supports the effectiveness of tool bundles in identifying potential rec-

Table 1: Performance comparisons of PTR under different methods within different backbones on ToolLens, MetaTool, and RecTools datasets. “N/A” indicates that this method works alone. The best results are bolded, the best results of each column are denoted as “\*”.

Methods	Framework	ToolLens			MetaTool			RecTools		
		Recall@K	NDCG@K	TRACC	Recall@K	NDCG@K	TRACC	Recall@K	NDCG@K	TRACC
Random	N/A	0.036	0.061	0.034	0.133	0.202	0.133	0.137	0.271	0.097
	+PTR+open-mistral-7b	0.185	0.225	0.145	0.608	0.785	0.505	0.457	0.756	0.235
	+PTR+GPT-3.5-turbo	0.213	0.282	0.172	0.645	0.823	0.543	0.475	0.784	0.288
	+PTR+GPT-4o	<b>0.227</b>	<b>0.303</b>	<b>0.187</b>	<b>0.663</b>	<b>0.843</b>	<b>0.562</b>	<b>0.492</b>	<b>0.802</b>	<b>0.305</b>
BM25	N/A	0.131	0.194	0.125	0.429	0.603	0.429	0.486	0.596	0.382
	+PTR+open-mistral-7b	0.206	0.254	0.162	0.659	0.834	0.554	0.524	0.795	0.355
	+PTR+GPT-3.5-turbo	0.247	0.313	0.193	0.694	0.874	0.593	0.541	0.815	0.408
	+PTR+GPT-4o	<b>0.261</b>	<b>0.331</b>	<b>0.208</b>	<b>0.712</b>	<b>0.892</b>	<b>0.612</b>	<b>0.545</b>	<b>0.810</b>	<b>0.414</b>
Contriever	N/A	0.130	0.190	0.121	0.439	0.672	0.439	0.367	0.786	0.304
	+PTR+open-mistral-7b	0.208	0.256	0.164	0.662	0.837	0.557	0.512	0.773	0.342
	+PTR+GPT-3.5-turbo	0.250	0.316	0.196	0.697	0.877	0.596	0.528	0.792	0.396
	+PTR+GPT-4o	<b>0.264</b>	<b>0.334</b>	<b>0.211</b>	<b>0.715</b>	<b>0.895</b>	<b>0.615</b>	<b>0.559</b>	<b>0.834</b>	<b>0.426</b>
SBERT	N/A	0.251	0.349	0.209	0.495	0.725	0.495	0.496	0.772	0.434
	+PTR+open-mistral-7b	0.272	0.362	0.226	0.682	0.862	0.582	0.538	0.821	0.452
	+PTR+GPT-3.5-turbo	0.308	0.403	0.252	0.723	0.902	0.623	0.555	0.840	0.484
	+PTR+GPT-4o	<b>0.322</b>	<b>0.422</b>	<b>0.268</b>	<b>0.741</b>	<b>0.921</b>	<b>0.642</b>	<b>0.572</b>	<b>0.859</b>	<b>0.501</b>
TAS-B	N/A	0.279	0.381	0.263	0.657	0.897	0.657	0.509	0.841	0.454
	+PTR+open-mistral-7b	0.298	0.398	0.278	0.702	0.882	0.602	0.552	0.854	0.472
	+PTR+GPT-3.5-turbo	0.335	0.438	0.305	0.741	0.922	0.642	0.567	0.872	0.505
	+PTR+GPT-4o	<b>0.352</b>	<b>0.456</b>	<b>0.321</b>	<b>0.759</b>	<b>0.941</b>	<b>0.661</b>	<b>0.583</b>	<b>0.890</b>	<b>0.522</b>
SimCSE	N/A	0.293	0.386	0.279	0.675	0.849	0.675	0.563	0.808	0.523
	+PTR+open-mistral-7b	0.312	0.407	0.291	0.716	0.897	0.631	0.578	0.861	0.542
	+PTR+GPT-3.5-turbo	0.350	0.448	0.319	0.756	0.937	0.671	0.594	0.879	0.575
	+PTR+GPT-4o	<b>0.368*</b>	<b>0.467*</b>	<b>0.336*</b>	<b>0.774*</b>	<b>0.956*</b>	<b>0.690*</b>	<b>0.609*</b>	<b>0.896*</b>	<b>0.591*</b>

Table 2: Ablation study of Tool Bundle Acquisition (w/o Tool Bundle Acquisition).

Methods (w/o Tool Bundle Acquisition)	ToolLens			MetaTool			RecTools		
	Recall@K	NDCG@K	TRACC	Recall@K	NDCG@K	TRACC	Recall@K	NDCG@K	TRACC
PTR + open-mistral-7b	0.221	0.264	0.171	0.695	0.882	0.612	0.532	0.912	0.270
PTR + GPT-3.5-turbo	0.264	0.381	0.208	0.724	0.919	0.656	0.541	0.913	0.430
PTR + GPT-4o	<b>0.283</b>	<b>0.391</b>	<b>0.235</b>	<b>0.745</b>	<b>0.922</b>	<b>0.677</b>	<b>0.581</b>	<b>0.916</b>	<b>0.439</b>

ommended tools, thereby refining the unresolved problems and achieving precise tool recommendations. Moreover, as illustrated in Table 2, the random approach alone is largely ineffective for tool recommendations. However, as presented in Table 1, when combined with functional coverage mapping and multi-view-based re-ranking, the final recommendation performance improves significantly. This underscores the importance of the last two recommendation stages.

**Performance w.r.t to accuracy in quantity.** Furthermore, to evaluate the performance of PTR in terms of tool number precision, we calculate the average length difference between the recommended tool set and the ground truth tool set for each method and backbone. Figure 4 demonstrates the effectiveness of PTR in maintaining consistency in the number of tools. In the MetaTool and ToolLens dataset, which exhibits relatively simple and small patterns, PTR clearly shows its effectiveness. Regarding our RecTools dataset, which has a variable

structure and involves a wide range of tools for each query, the average length difference is effectively controlled within a considerable range, especially when it comes to the Embedding method.

## 5 Related work

Initially, term-based methods such as BM25 (Robertson et al., 2009) and TF-IDF (Sparck Jones, 1972) were employed to measure the similarity between queries and tool documents by identifying exact term matches. Subsequently, with the development of dense retrievers (Karpukhin et al., 2020; Guu et al., 2020; Xiong et al., 2020), the semantic relationships between queries and tool descriptions have been more effectively captured through neural networks. Recently, novel approaches for training retrievers have emerged. For example, Confucius (Gao et al., 2024) selects tools by defining three levels of scenarios, ranging from easy to difficult, to train and deepen the LLM’s understanding of tools. Additionally, execution feedback is itera-

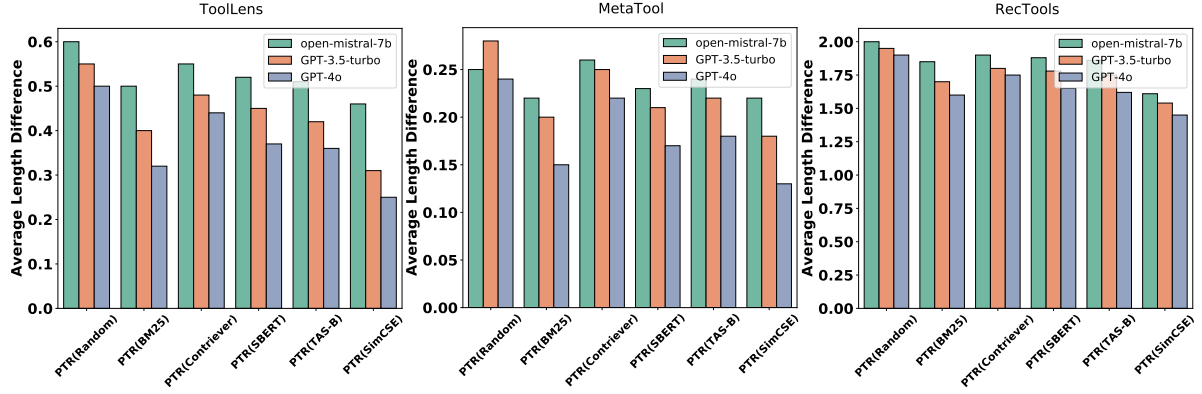


Figure 4: The average length difference between the recommended tool set and the ground truth tool set for each method and backbone.

tively utilized to refine the tool selection process (Wang et al., 2023; Xu et al., 2024). Furthermore, ToolkenGPT (Hao et al., 2024) enhances tool selection by representing each tool as a token ("toolken") and learning an embedding for it, thereby enabling tool calls in the same manner as generating regular word tokens. Moreover, some research has focused on addressing the diversity of retrieval (Carbonell and Goldstein, 1998; Gao and Zhang, 2024b), which can effectively enhance the quality of multiple tools used by query. Despite their comprehensive nature, tool retrieval systems present notable limitations. The inclusion of superfluous tools can introduce noise, thereby interfering with the LLM’s performance and task execution, and these systems are often unable to dynamically adjust the toolset. In this work, we extend our approach beyond getting a rough toolset by ensuring that the tools in the recommended toolset are as accurate as possible in terms of both quality and quantity.

## 6 Conclusions

This study presents a novel challenge, tool recommendation, and offers a precise formalization of the problem. In response, we propose a new approach, PTR, designed to improve the accuracy of tool recommendations, considering both the quantity and the selection of tools. PTR operates through three key stages: tool bundle acquisition, functional coverage mapping, and multi-view-based re-ranking. By dynamically adjusting the tool bundle obtained in the first stage—through the addition or removal of tools—PTR progressively refines the recommended toolset. Extensive experiments and detailed analyses showcase PTR’s effectiveness in addressing diverse query structures requiring multi-

ple tool recommendations. Furthermore, we introduce RecTools, a new dataset, along with TRACC, a comprehensive evaluation metric. Both serve as valuable contributions to the future research in the field of tool recommendation.

## 7 Limitations

Although our proposed framework demonstrates precise and reliable toolset recommendation for LLM agents, several aspects remain open for further study. For example, the effectiveness of our approach is partially shaped by the comprehension abilities of the underlying language models, which may yield minor variations in toolset selection when different models are employed. Moreover, our current implementation is optimized for text-based scenarios, and its extension to incorporate additional modalities, such as visual or structured data, could broaden its practical reach.

## References

- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.



618	Jaime Carbonell and Jade Goldstein. 1998. The use of	674
619	mmr, diversity-based reranking for reordering doc-	675
620	uments and producing summaries. In <i>Proceedings</i>	676
621	<i>of the 21st annual international ACM SIGIR confer-</i>	677
622	<i>ence on Research and development in information</i>	
623	<i>retrieval</i> , pages 335–336.	
624	Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu,	678
625	Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong	679
626	Chen, Xingmei Wang, and 1 others. 2024. When	680
627	large language models meet personalization: Perspec-	681
628	tives of challenges and opportunities. <i>World Wide</i>	682
629	<i>Web</i> , 27(4):42.	683
630	Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter.	684
631	2019. Neural architecture search: A survey. <i>Journal</i>	685
632	<i>of Machine Learning Research</i> , 20(55):1–21.	686
633	William Fedus, Barret Zoph, and Noam Shazeer. 2022.	687
634	Switch transformers: Scaling to trillion parameter	688
635	models with simple and efficient sparsity. <i>Journal of</i>	689
636	<i>Machine Learning Research</i> , 23(120):1–39.	690
637	Hang Gao and Yongfeng Zhang. 2024a. Memory shar-	691
638	ing for large language model based agents. <i>arXiv</i>	692
639	<i>preprint arXiv:2404.09982</i> .	
640	Hang Gao and Yongfeng Zhang. 2024b. Vrsd: Re-	693
641	thinking similarity and diversity for retrieval in large	694
642	language models. <i>arXiv preprint arXiv:2407.04573</i> .	695
643	Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang,	696
644	Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and	697
645	Zhaochun Ren. 2024. Confucius: Iterative tool learn-	698
646	ing from introspection feedback by easy-to-difficult	699
647	curriculum. In <i>Proceedings of the AAAI Conference</i>	700
648	<i>on Artificial Intelligence</i> , volume 38, pages 18030–	701
649	18038.	702
650	T Gao, X Yao, and Danqi Chen. 2021. Simcse: Sim-	703
651	ple contrastive learning of sentence embeddings. In	704
652	<i>EMNLP 2021-2021 Conference on Empirical Meth-</i>	705
653	<i>ods in Natural Language Processing, Proceedings</i> .	706
654	Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia,	707
655	Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen	708
656	Wang. 2023. Retrieval-augmented generation for	709
657	large language models: A survey. <i>arXiv preprint</i>	710
658	<i>arXiv:2312.10997</i> .	
659	Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasu-	711
660	pat, and Mingwei Chang. 2020. Retrieval augmented	712
661	language model pre-training. In <i>International confer-</i>	713
662	<i>ence on machine learning</i> , pages 3929–3938. PMLR.	714
663	Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu.	715
664	2024. Toolkengpt: Augmenting frozen language	716
665	models with massive tools via tool embeddings. <i>Ad-</i>	
666	<i>vances in neural information processing systems</i> , 36.	
667	Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong	717
668	Yang, Jimmy Lin, and Allan Hanbury. 2021. Ef-	718
669	ficiently teaching an effective dense retriever with	719
670	balanced topic aware sampling. In <i>Proceedings of</i>	
671	<i>the 44th International ACM SIGIR Conference on</i>	
672	<i>Research and Development in Information Retrieval</i> ,	
673	pages 113–122.	
	Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang	720
	Nie, and Juanzi Li. 2023. A survey of knowledge	721
	enhanced pre-trained language models. <i>IEEE Trans-</i>	722
	<i>actions on Knowledge and Data Engineering</i> .	723
	Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan	724
	Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan,	725
	Neil Zhenqiang Gong, and 1 others. 2023. Meta-	726
	tool benchmark for large language models: Deciding	727
	whether to use tools and which to use. <i>arXiv preprint</i>	728
	<i>arXiv:2310.03128</i> .	729
	Gautier Izacard, Mathilde Caron, Lucas Hosseini, Se-	730
	bastian Riedel, Piotr Bojanowski, Armand Joulin,	
	and Edouard Grave. 2021. Unsupervised dense in-	
	formation retrieval with contrastive learning. <i>arXiv</i>	
	<i>preprint arXiv:2112.09118</i> .	
	Kalervo Järvelin and Jaana Kekäläinen. 2002. Cu-	
	mulated gain-based evaluation of ir techniques.	
	<i>ACM Transactions on Information Systems (TOIS)</i> ,	
	20(4):422–446.	
	Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick	
	Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and	
	Wen-tau Yih. 2020. Dense passage retrieval for	
	open-domain question answering. <i>arXiv preprint</i>	
	<i>arXiv:2004.04906</i> .	
	Yilun Kong, Jingqing Ruan, Yihong Chen, Bin Zhang,	
	Tianpeng Bao, Shiwei Shi, Guoqing Du, Xiaoru Hu,	
	Hangyu Mao, Ziyue Li, and 1 others. 2023. Tptu-	
	v2: Boosting task planning and tool usage of large	
	language model-based agents in real-world systems.	
	<i>arXiv preprint arXiv:2311.11315</i> .	
	Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio	
	Petroni, Vladimir Karpukhin, Naman Goyal, Hein-	
	rich Küttler, Mike Lewis, Wen-tau Yih, Tim Rock-	
	täschel, and 1 others. 2020. Retrieval-augmented	
	generation for knowledge-intensive nlp tasks. <i>Ad-</i>	
	<i>vances in Neural Information Processing Systems</i> ,	
	33:9459–9474.	
	Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song,	
	Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang,	
	and Yongbin Li. 2023. Api-bank: A comprehensive	
	benchmark for tool-augmented llms. In <i>Proceedings</i>	
	<i>of the 2023 Conference on Empirical Methods in</i>	
	<i>Natural Language Processing</i> , pages 3102–3116.	
	Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning:	
	Optimizing continuous prompts for generation. <i>arXiv</i>	
	<i>preprint arXiv:2101.00190</i> .	
	Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang,	
	Hiroaki Hayashi, and Graham Neubig. 2023. Pre-	
	train, prompt, and predict: A systematic survey of	
	prompting methods in natural language processing.	
	<i>ACM Computing Surveys</i> , 55(9):1–35.	
	Alex Mallen, Akari Asai, Victor Zhong, Rajarshi	
	Das, Hannaneh Hajishirzi, and Daniel Khoshnab.	
	2022. When not to trust language models: Inves-	
	tigating effectiveness and limitations of paramet-	
	ric and non-parametric memories. <i>arXiv preprint</i>	
	<i>arXiv:2212.10511</i> , 7.	



Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. 2024. Gpt4tools: Teaching large language model to use tools via self-instruction. *Advances in Neural Information Processing Systems*, 36.

Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R Fung, Hao Peng, and Heng Ji. 2023. Craft: Customizing llms by creating and retrieving from specialized toolsets. *arXiv preprint arXiv:2309.17428*.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Ren Kan, Dongsheng Li, and Deqing Yang. 2024. Easytool: Enhancing llm-based agents with concise tool instruction. *arXiv preprint arXiv:2401.06201*.

Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. 2024. Let me do it for you: Towards llm empowered recommendation via tool learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1796–1806.

Yuanhang Zheng, Peng Li, Wei Liu, Yang Liu, Jian Luan, and Bin Wang. 2024. Toolrerank: Adaptive and hierarchy-aware reranking for tool retrieval. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16263–16273.

Mu Zhu. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2(30):6.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143.

## Appendix

### A Problem Definition

Tool retrieval, as discussed in previous, involves generating a comprehensive list of tools that are potentially relevant to a user’s query. This approach emphasizes breadth, aiming to maximize the inclusion of pertinent tools. While effective in ensuring extensive coverage, tool retrieval often prioritizes recall over precision, resulting in the inclusion of extraneous tools that may not be essential for the task at hand. Addressing this limitation, we propose a new optimization direction—Tool Recommendation—for LLMs. It aims to ensure that the recommended set of tools aligns closely with the ground-truth set of tools for a task, both in quantity and quality. Specifically, given a user query

with a ground-truth toolset  $(A, B, C)$ , tool recommendation aims to identify precisely  $(A, B, C)$ , avoiding omissions or the inclusion of redundant tools. Here is the definition of the tool recommendation task:

**Definition 1** *Tool Recommendation: Given a comprehensive set of tools  $T = \{T_1, T_2, \dots, T_n\}$  and a query  $Q$ , let  $T_{\text{ground}} \subseteq T$  denote the ground truth toolset that fully satisfies  $Q$ . The objective is to recommend a toolset  $T_{\text{recommend}} = \{T_1, T_2, \dots, T_k\}$  from  $T$  such that  $T_{\text{recommend}} = T_{\text{ground}}$  and the cardinality constraint  $|T_{\text{recommend}}| = |T_{\text{ground}}|$  holds.*

Achieving precision in tool recommendation is pivotal for enhancing the performance and reliability of LLMs. By minimizing the inclusion of irrelevant tools, LLMs can reduce computational overhead, streamline task execution, and improve the overall quality of responses. Addressing precise tool recommendation not only mitigates the drawbacks associated with broad tool retrieval but also paves the way for more sophisticated and user-centric LLM applications. This advancement is essential for deploying LLMs in environments where efficiency, accuracy, and user satisfaction are crucial.

### B Extended Related Work

**Recommendation for LLMs.** Recent research has explored a variety of recommendation techniques to enhance Large Language Models (LLMs), integrating capabilities across multiple dimensions. Data recommendation (Xu et al., 2020; Ouyang et al., 2022) is crucial for selecting relevant datasets to fine-tune models for specific domains, ensuring ongoing performance improvements. Memory recommendation (Borgeaud et al., 2022; Gao and Zhang, 2024a) facilitates the retrieval of relevant past experiences or interactions, improving continuity, consistency, and long-term context in multi-turn conversations. Knowledge base recommendation (Gao et al., 2023; Hu et al., 2023; Petroni et al., 2019; Lewis et al., 2020) enhances factual grounding by retrieving the most pertinent information from external sources, ensuring that model outputs are accurate and up to date. Architecture recommendation (Elsken et al., 2019; Fedus et al., 2022) optimizes model performance by dynamically selecting the most appropriate model components or layers to activate for different tasks, thereby improving efficiency. Lastly, prompt recommendation (Shin et al., 2020; Reynolds and McDonell,

2021; Li and Liang, 2021; Wang et al., 2022; Liu et al., 2023) guides LLMs in utilizing the most effective input prompts, thereby enhancing the quality of generated responses through optimized input-output interactions. Together, these recommendation techniques form a comprehensive framework that enhances the adaptability, efficiency, and task-specific performance of LLMs. However, there remains a lack of research on tool recommendation. In this work, we motivate to seek to provide a clear definition of tool recommendation and proposes an effective recommendation method. Additionally, new datasets and metrics are created to advance research in this area.

## C Details of RecTools

Both existed datasets impose a low upper limit on the number of tools used per query. As the capabilities of LLMs continue to develop, more tools need to be recommended to solve increasingly complex problems, thereby limiting the applicability of these datasets. Additionally, all queries in these two datasets utilize a fixed number of tools, which not only fails to fully simulate the dynamic nature of tool usage in real-world scenarios but also introduces bias in the subsequent testing of the method. Most importantly, since tool recommendation focuses on the precision of the recommended toolset, the test datasets require that each query be exactly solvable by the provided tools (Exact Solving). Using one fewer tool leads to partial solving, while using one additional tool results in oversolving. To validate the effectiveness of the two datasets, we first employ GPT-4o as an evaluator to determine whether the provided toolset can achieve an “Exact Solving” outcome for each query. Subsequently, for each query, we randomly remove one tool from the corresponding toolset and prompt GPT-4o to assess whether the modified toolset can achieve a “Partial Solving” outcome. Queries and their respective toolsets that meet the criteria for both evaluations are considered qualified. The performance of these two datasets is not ideal. Based on these limitations, we constructed a new dataset, **RecTools**, where queries do not have a uniform number of tools and have a high upper limit on the number of tools used (details in Appendix.C). Additionally, RecTools significantly outperforms ToolLens and Metatool in the GPT-4o “Exact Solving” test. The statistics of the three datasets are summarized in Table.3. Specifically, for all (query,

Table 3: Statistics of the experimental datasets.

Feature	ToolLens	MetaTool	RecTools
<b>Tools per Query</b>	1-3	2	1-10
<b>Unified used tool number</b>	✓	✗	✓
<b>Exact Solving Test</b>	6.34%	55.1%	61.3%

tools) pairs involving the use of two and three tools, the success rates of RecTools reached 76% and 89%, respectively.

### C.1 Dataset Construction

To construct our dataset, we utilized tools from the MetaTool (Huang et al., 2023) dataset, along with their corresponding descriptions. Since their objective of tools was to address the issue of overlapping—where a single query could be resolved by multiple tools—MetaTool consolidates groups of tools with similar functionalities into a single tool entity. Besides, those tools and their description come from OpenAI’s plugin list, making them more practical. In our dataset RecTools, there are 10 usage scenarios in total (from 1-10), where the usage scenarios mean the quantitative classification, like two tools be used together, ten tools be used together. Each scenario of tools usage contains 100 examples. In each scenario, there are 20 different tool combinations. In terms of each combination, we randomly select from all possible combinations(i.e.,  $\binom{1}{n}, \binom{2}{n}, \dots, \binom{10}{n}$ ). And for each tool combinations, we generate 5 queries. The prompt is as follows:

```
You are an assistant tasked with
generating user queries that can be
exclusively solved by a specific set of
tools.

**Requirements for the query:**
1. The query must only require the
functionalities of the selected tools.
2. All tools in the selected set must be
necessary to solve the query.
3. The query should not require any
tools outside the selected set.
4. The query should be clear, specific
, and realistic.
5. Each query should address a
different scenario or aspect to ensure
uniqueness. Avoid merely rephrasing
similar ideas; focus on varied use cases
.

Selected Tools:
XX, XXX

Tool Descriptions:
- XX: Search for podcasts and
summarize their content.
```



```

- **XX*: Discover and support
restaurants, shops & services near you.

Generate one unique query that meets the
above requirements.

```

## C.2 Dataset Evaluation

To ensure precision in tool recommendation, it is crucial that the query is addressed entirely by the provided tools. If any tool is missing, the query cannot be fully solved, and if an unnecessary tool is included, the solution becomes redundant or repetitive. We employ GPT-4 as an evaluator to determine whether the provided toolset can achieve an "Exact Solving" outcome for each query. Subsequently, for each query, we randomly remove one tool from the corresponding toolset and prompt GPT-4 to assess whether the modified toolset can achieve a "Partial Solving" outcome. Queries and their respective toolsets that meet the criteria for both evaluations are considered qualified. For the first evaluation, if it achieves "Exact Solving", we give it a score 1, else 0; For the second evaluation, if it achieves "Partial Solving", we give it a score 1, else 0; For the final score, if both of them are 1, then 1; else, 0. The prompt is as follows:

```

Prompt1(Before deletion)
**Query:** "XXX"

**Tools:**
- **XX*: xxxxxx
- **XX*: xxxxxx
- **XX*: xxxxxx

**Classification:** (.Categorize the
solving scenario into one of the
following:
1. **Exact Solving:** All
functionalities are met by all tools
without any redundancies.
2. **Oversolving:** The toolset includes
tools that provide functionalities not
required by the query.
3. **Partial Solving:** Some
functionalities remain unfulfilled and
some tools remain unused.)

-----
Prompt2(After deletion)
**Query:** "XXX"

**Tools after removing one tool:**
- **XX*: xxxxxx
- **XX*: xxxxxx

**Classification:** (.Categorize the
solving scenario into one of the
following:

```

```

1. **Exact Solving:** All
functionalities are met by all tools
without any redundancies.
2. **Oversolving:** The toolset includes
tools that provide functionalities not
required by the query.
3. **Partial Solving:** Some
functionalities remain unfulfilled and
some tools remain unused.)

```

The final output of evaluation is like this:

```

{
  "query": "XXX",
  "tools_used": [
    "XX",
    "XX"
  ],
  "first_evaluation": "xxx",
  "second_evaluation_after_deletion": "xxx",
  "score": X
},

```

Listing 1: An full example for evaluation

```

Few-Shot Examples:

**Query:** "I need the latest weather
forecast for New York and a reminder to
carry an umbrella."

**Tools:**
- **WeatherTool*: Provide you with the
latest weather information.
- **ReminderTool*: No description
available.

**Classification:** Exact Solving

**Query:** "Show me the top-rated
restaurants nearby and provide a route
to get there."

**Tools:**
- **RestaurantFinder*: No description
available.
- **RoutePlanner*: No description
available.

**Classification:** Exact Solving

**Query:** "Find me a good book to read
and suggest a nearby coffee shop."

**Tools:**
- **BookRecommender*: No description
available.
- **WeatherTool*: Provide you with the
latest weather information.

**Classification:** Partial Solving

**Query:** "Provide the current exchange
rates and set a reminder to check them
later."

**Tools:**
- **FinanceTool*: Stay informed with
the latest financial updates, real-time

```

1164	insights, and analysis on a wide range	1234
1165	of options, stocks, cryptocurrencies,	1235
1166	and more.	1236
1167	- **ReminderTool**: No description	1237
1168	available.	1238
1169	- **NewsTool**: Stay connected to global	1239
1170	events with our up-to-date news around	1240
1171	the world.	1241
1172		1242
1173	**Classification**: Oversolving	1243
1174		1244
1175	**Query**: "I want to track my fitness	1245
1176	goals and get news updates."	1246
1177		1247
1178	**Tools**:	1248
1179	- **FitnessTracker**: No description	1249
1180	available.	1250
1181	- **NewsTool**: Stay connected to global	1251
1182	events with our up-to-date news around	1252
1183	the world.	1253
1184		1254
1185	**Classification**: Exact Solving	1255
1186		1256
1187	**Query**: "Schedule a meeting and find	1257
1188	the latest sports news."	1258
1189		1259
1190	**Tools**:	1260
1191	- **CalendarTool**: No description	1261
1192	available.	1262
1193	- **NewsTool**: Stay connected to global	1263
1194	events with our up-to-date news around	1264
1195	the world.	1265
1196	- **FinanceTool**: Stay informed with	1266
1197	the latest financial updates, real-time	1267
1198	insights, and analysis on a wide range	1268
1199	of options, stocks, cryptocurrencies,	1269
1200	and more.	1270
1201		1271
1202	**Classification**: Oversolving	1272
1203		1273
1204		1274
1205		1275
1206	**Query**: "Research and select	1276
1207	appropriate investment options for	1277
1208	setting up a trust fund, ensure	1278
1209	compliance with relevant laws, and find	1279
1210	suitable gifts for beneficiaries to	1280
1211	commemorate the establishment of the	1281
1212	trust."	1282
1213		1283
1214	**Tools**:	1284
1215	- **FinanceTool**: Stay informed with	1285
1216	the latest financial updates, real-time	1286
1217	insights, and analysis on a wide range	1287
1218	of options, stocks, cryptocurrencies,	1288
1219	and more.	1289
1220	- **LawTool**: Enables quick search	1290
1221	functionality for relevant laws.	1291
1222	- **GiftTool**: Provide suggestions for	1292
1223	gift selection.	1293
1224		1294
1225	**Classification**: (Respond with only	1295
1226	one of the following exact phrases: "	1296
1227	Exact Solving", "Oversolving", or "	1297
1228	Partial Solving". Do not include any	1298
1229	additional text or explanations.)	1299
1230		1300
1231	First Evaluation: Exact Solving	1301
1232		1302
1233	Few-Shot Examples:	1303

```

1304 - **CalendarTool**: No description
1305 available.
1306 - **NewsTool**: Stay connected to global
1307 events with our up-to-date news around
1308 the world.
1309 - **FinanceTool**: Stay informed with
1310 the latest financial updates, real-time
1311 insights, and analysis on a wide range
1312 of options, stocks, cryptocurrencies,
1313 and more.
1314
1315 **Classification**: Oversolving
1316
1317
1318
1319 **Query**: "Research and select
1320 appropriate investment options for
1321 setting up a trust fund, ensure
1322 compliance with relevant laws, and find
1323 suitable gifts for beneficiaries to
1324 commemorate the establishment of the
1325 trust."
1326
1327 **Tools after removing one tool**:
1328 - **FinanceTool**: Stay informed with
1329 the latest financial updates, real-time
1330 insights, and analysis on a wide range
1331 of options, stocks, cryptocurrencies,
1332 and more.
1333 - **LawTool**: Enables quick search
1334 functionality for relevant laws.
1335
1336 **Classification**: (Respond with only
1337 one of the following exact phrases: "
1338 Exact Solving", "Oversolving", or "
1339 Partial Solving". Do not include any
1340 additional text or explanations.)
1341
1342 Second Evaluation (After Deletion):
1343 Partial Solving
1344 Score for this query: 1
1345
1346 *****
1347 *****
1348
1349 {
1350     "query": "Research and select
1351 appropriate investment options for
1352 setting up a trust fund, ensure
1353 compliance with relevant laws, and
1354 find suitable gifts for
1355 beneficiaries to commemorate the
1356 establishment of the trust.",
1357     "tools_used": [
1358         "FinanceTool",
1359         "LawTool",
1360         "GiftTool"
1361     ],
1362     "first_evaluation": "Exact Solving",
1363     "second_evaluation_after_deletion": "Partial Solving",
1364     "score": 1
1365 }
1366
1367

```

## D Functional Coverage Mapping

### D.1 Extraction of Key Functionalities

```

You are an assistant helping to extract
key requirements from user queries.

Example 1:
User Query:
"I want a website where users can create
accounts, post messages, and follow
other users."

Key Requirements:
- Users can create accounts
- Users can post messages
- Users can follow other users

Example 2:
User Query:
"I need an e-commerce platform that
supports product listings, shopping cart
functionality, payment processing, and
order tracking."

Key Requirements:
- Supports product listings
- Provides shopping cart functionality
- Handles payment processing
- Offers order tracking

Now, given the following user query,
extract the key requirements.

User Query:
XXX

Key Requirements:

```

## D.2 Matching Tools to Functionalities

You are an assistant helping to match tools to requirements, as long as the tool description can roughly provide the needed information for requirements, it does not need to be very specific, ignore the proper nouns.

Available Tools: XX:xxxxx; XX:xxxxxx.

Example 1:  
Requirement:  
"I want to know the latest news about Tesla"

Matched Tools:  
- NewsTool: Stay connected to global events with our up-to-date news around the world.

Example 2:  
Requirement:  
"Please provide me with the current stock price of Apple"

Matched Tools:  
- FinanceTool: Stay informed with the latest financial updates, real-time insights, and analysis on a wide range of options, stocks, cryptocurrencies, and more.

Now, for the following requirement, list

1437 the tools from the available tools that  
 1438 can fulfill it.  
 1439  
 1440 Requirement:  
 1441 XXX  
 1442 XXX  
 1443 XXX  
 1444  
 1445 Matched Tools:

1446

### D.3 Examples

#### Listing 2: An example in ToolLens

1447 You are an assistant helping to extract  
 1448 key requirements from user queries.  
 1449  
 1450 Example 1:  
 1451 User Query:  
 1452 "I want a website where users can create  
 1453 accounts, post messages, and follow  
 1454 other users."  
 1455  
 1456 Key Requirements:  
 1457 - Users can create accounts  
 1458 - Users can post messages  
 1459 - Users can follow other users  
 1460  
 1461 Example 2:  
 1462 User Query:  
 1463 "I need an e-commerce platform that  
 1464 supports product listings, shopping cart  
 1465 functionality, payment processing, and  
 1466 order tracking."  
 1467  
 1468 Key Requirements:  
 1469 - Supports product listings  
 1470 - Provides shopping cart functionality  
 1471 - Handles payment processing  
 1472 - Offers order tracking  
 1473  
 1474 Now, given the following user query,  
 1475 extract the key requirements.  
 1476  
 1477 User Query:  
 1478 "I'm preparing for a marathon in Paris,  
 1479 France."  
 1480 -----  
 1481 Key Requirements:  
 1482 - Marathon preparation  
 1483 - Location: Paris, France  
 1484  
 1485 \*\*\*\*\*  
 1486 \*\*\*\*\*  
 1487  
 1488 You are an assistant helping to match  
 1489 tools to requirements, as long as the  
 1490 tool description can roughly provide the  
 1491 needed information for requirements, it  
 1492 does not need to be very specific, ignore  
 1493 the proper nouns.  
 1494  
 1495 Available Tools:  
 1496 - \*\*Countries\*\*: This gets geo data on a  
 1497 country. Use ISO2 for country\_code.  
 1498 - \*\*Skyscanner\_v2\*\*: Search for a place  
 1499 to get the \*\*entityId\*\* needed in  
 1500 searching the hotel API.  
 1501 - \*\*TimeTable Lookup\*\*: Returns the

nearest airports for a given latitude  
 and longitude  
 Example 1:  
 Requirement:  
 "I want to know the latest news about  
 Tesla"  
 Matched Tools:  
 - NewsTool: Stay connected to global  
 events with our up-to-date news around  
 the world.  
 Example 2:  
 Requirement:  
 "Please provide me with the current  
 stock price of Apple"  
 Matched Tools:  
 - FinanceTool: Stay informed with the  
 latest financial updates, real-time  
 insights, and analysis on a wide range  
 of options, stocks, cryptocurrencies,  
 and more.  
 Now, for the following requirement, list  
 the tools from the available tools that  
 can fulfill it.  
 Requirement:  
 "Marathon preparation"  
 Matched Tools:  
 You are an assistant helping to match  
 tools to requirements, as long as the  
 tool description can roughly provide the  
 needed information for requirements, it  
 does not need to be very specific, ignore  
 the proper nouns.  
 Available Tools:  
 - \*\*Countries\*\*: This gets geo data on a  
 country. Use ISO2 for country\_code.  
 - \*\*Skyscanner\_v2\*\*: Search for a place  
 to get the \*\*entityId\*\* needed in  
 searching the hotel API.  
 - \*\*TimeTable Lookup\*\*: Returns the  
 nearest airports for a given latitude  
 and longitude  
 Example 1:  
 Requirement:  
 "I want to know the latest news about  
 Tesla"  
 Matched Tools:  
 - NewsTool: Stay connected to global  
 events with our up-to-date news around  
 the world.  
 Example 2:  
 Requirement:  
 "Please provide me with the current  
 stock price of Apple"  
 Matched Tools:  
 - FinanceTool: Stay informed with the  
 latest financial updates, real-time

1502  
 1503  
 1504  
 1505  
 1506  
 1507  
 1508  
 1509  
 1510  
 1511  
 1512  
 1513  
 1514  
 1515  
 1516  
 1517  
 1518  
 1519  
 1520  
 1521  
 1522  
 1523  
 1524  
 1525  
 1526  
 1527  
 1528  
 1529  
 1530  
 1531  
 1532  
 1533  
 1534  
 1535  
 1536  
 1537  
 1538  
 1539  
 1540  
 1541  
 1542  
 1543  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560  
 1561  
 1562  
 1563  
 1564  
 1565  
 1566  
 1567  
 1568  
 1569  
 1570  
 1571



```

1572 insights, and analysis on a wide range
1573 of options, stocks, cryptocurrencies,
1574 and more.
1575
1576 Now, for the following requirement, list
1577 the tools from the available tools that
1578 can fulfill it.
1579
1580 Requirement:
1581 "Location: Paris, France"
1582
1583 Matched Tools:
1584
1585 Tool Matches:
1586 - Requirement: 'Marathon preparation'
1587 matched with Tools: None
1588 - Requirement: 'Location: Paris, France'
1589 matched with Tools: None
1590
1591 Does the toolset exactly solve the query
1592 ? No
1593 Tools to Keep:
1594
1595 Unsolved Problems:
1596 - Marathon preparation
1597 - Location: Paris, France

```

### Listing 3: An example in MetaTool

```

1598 You are an assistant helping to extract
1599 key requirements from user queries.
1600
1601 Example 1:
1602 User Query:
1603 "I want a website where users can create
1604 accounts, post messages, and follow
1605 other users."
1606
1607 Key Requirements:
1608 - Users can create accounts
1609 - Users can post messages
1610 - Users can follow other users
1611
1612 Example 2:
1613 User Query:
1614 "I need an e-commerce platform that
1615 supports product listings, shopping cart
1616 functionality, payment processing, and
1617 order tracking."
1618
1619 Key Requirements:
1620 - Supports product listings
1621 - Provides shopping cart functionality
1622 - Handles payment processing
1623 - Offers order tracking
1624
1625 Now, given the following user query,
1626 extract the key requirements.
1627
1628 User Query:
1629 "I'm looking for a family-friendly
1630 destination in Europe with good weather.
1631 Can you suggest some options and what
1632 the weather will be like during summer?"
1633 -----
1634 Key Requirements Extracted:
1635 - Family-friendly destination in Europe
1636 - Options about Europe
1637 - Information on weather during summer
1638

```

```

*****
*****

You are an assistant helping to match
tools to requirements, as long as the
tool description can roughly provide the
needed information for requirements, it
does not need to be very specific, ignore
the proper nouns.

Available Tools:
- **ResearchFinder**: Tool for searching
  academic papers.
- **WeatherTool**: Provide you with the
  latest weather information.

Example 1:
Requirement:
"I want to know the latest news about
Tesla"

Matched Tools:
- NewsTool: Stay connected to global
  events with our up-to-date news around
  the world.

Example 2:
Requirement:
"Please provide me with the current
stock price of Apple"

Matched Tools:
- FinanceTool: Stay informed with the
  latest financial updates, real-time
  insights, and analysis on a wide range
  of options, stocks, cryptocurrencies,
  and more.

Now, for the following requirement, list
the tools from the available tools that
can fulfill it.

Requirement:
"Family-friendly destination in Europe"

Matched Tools:

You are an AI assistant helping to match
tools to requirements, as long as the
tool description can roughly provide the
needed information for requirements, it
does not need to be very specific, ignore
the proper nouns.

Available Tools:
- **ResearchFinder**: Tool for searching
  academic papers.
- **WeatherTool**: Provide you with the
  latest weather information.

Example 1:
Requirement:
"I want to know the latest news about
Tesla"

Matched Tools:
- NewsTool: Stay connected to global
  events with our up-to-date news around
  the world.

```

```

1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708

```

1709  
1710 Example 2:  
1711 Requirement:  
1712 "Please provide me with the current  
1713 stock price of Apple"  
1714  
1715 Matched Tools:  
1716 - FinanceTool: Stay informed with the  
1717 latest financial updates, real-time  
1718 insights, and analysis on a wide range  
1719 of options, stocks, cryptocurrencies,  
1720 and more.  
1721  
1722 Now, for the following requirement, list  
1723 the tools from the available tools that  
1724 can fulfill it.  
1725  
1726 Requirement:  
1727 "Options about Europe"  
1728  
1729 Matched Tools:  
1730  
1731  
1732 You are an AI assistant helping to match  
1733 tools to requirements, as long as the  
1734 tool description can roughly provide the  
1735 needed information for requirements, it  
1736 does not need to be very specific, ignore  
1737 the proper nouns.  
1738  
1739 Available Tools:  
1740 - \*\*ResearchFinder\*\*: Tool for searching  
1741 academic papers.  
1742 - \*\*WeatherTool\*\*: Provide you with the  
1743 latest weather information.  
1744  
1745 Example 1:  
1746 Requirement:  
1747 "I want to know the latest news about  
1748 Tesla"  
1749  
1750 Matched Tools:  
1751 - NewsTool: Stay connected to global  
1752 events with our up-to-date news around  
1753 the world.  
1754  
1755 Example 2:  
1756 Requirement:  
1757 "Please provide me with the current  
1758 stock price of Apple"  
1759  
1760 Matched Tools:  
1761 - FinanceTool: Stay informed with the  
1762 latest financial updates, real-time  
1763 insights, and analysis on a wide range  
1764 of options, stocks, cryptocurrencies,  
1765 and more.  
1766  
1767 Now, for the following requirement, list  
1768 the tools from the available tools that  
1769 can fulfill it.  
1770  
1771 Requirement:  
1772 "Information on weather during summer"  
1773  
1774 Matched Tools:  
1775 WeatherTool: Provide you with the latest  
1776 weather information.  
1777  
1778 Tool Matches:

- Requirement: 'Family-friendly  
destination in Europe' matched with  
Tools: None  
- Requirement: 'Good weather' matched  
with Tools: None  
- Requirement: 'Information on weather  
during summer' matched with Tools:  
WeatherTool  
Does the toolset exactly solve the query  
? No  
Tools to Keep:  
WeatherTool  
Unsolved Problems:  
- Family-friendly destination in Europe  
- Options about Europe  
- Information on weather during summer

#### Listing 4: An example in RecTools

You are an assistant helping to extract  
key requirements from user queries.  
Example 1:  
User Query:  
"I want a website where users can create  
accounts, post messages, and follow  
other users."  
Key Requirements:  
- Users can create accounts  
- Users can post messages  
- Users can follow other users  
Example 2:  
User Query:  
"I need an e-commerce platform that  
supports product listings, shopping cart  
functionality, payment processing, and  
order tracking."  
Key Requirements:  
- Supports product listings  
- Provides shopping cart functionality  
- Handles payment processing  
- Offers order tracking  
Now, given the following user query,  
extract the key requirements.  
User Query:  
"I want to find a local restaurant with  
a menu that fits my diet plan, book a  
table, get astrology insights on the  
best date for my dinner, and select a  
thoughtful gift for my dining companion  
."  
-----  
Key Requirements Extracted:  
- Find a local restaurant  
- Provide a menu that fits the user's  
diet plan  
- Book a table  
- Offer astrology insights on the best  
date for dinner  
- Select a thoughtful gift for the  
dining companion  
\*\*\*\*\*

1846 \*\*\*\*\*  
1847  
1848 You are an assistant helping to match  
1849 tools to requirements, as long as the  
1850 tool description can roughly provide the  
1851 needed information for requirements, it  
1852 does not need to be very specific, ignore  
1853 the proper nouns.  
1854  
1855 Available Tools:  
1856 - \*\*DietTool\*\*: A tool that simplifies  
1857 calorie counting, tracks diet, and  
1858 provides insights from many restaurants  
1859 and grocery stores. Explore recipe ,  
1860 menus, and cooking tips from millions of  
1861 users, and access recipe consultations  
1862 and ingredient delivery services from  
1863 thousands of stores.  
1864 - \*\*GiftTool\*\*: Provide suggestions for  
1865 gift selection.  
1866 - \*\*HousePurchasingTool\*\*: Tool that  
1867 provide all sorts of information about  
1868 house purchasing  
1869 - \*\*HouseRentingTool\*\*: Tool that  
1870 provide all sorts of information about  
1871 house renting  
1872 - \*\*MemoryTool\*\*: A learning application  
1873 with spaced repetition functionality  
1874 that allows users to create flashcards  
1875 and review them.  
1876 - \*\*RestaurantBookingTool\*\*: Tool for  
1877 booking restaurant  
1878 - \*\*ResumeTool\*\*: Quickly create resumes  
1879 and receive feedback on your resume.  
1880 - \*\*StrologyTool\*\*: Provides strology  
1881 services for you.  
1882 - \*\*local\*\*: Discover and support  
1883 restaurants, shops & services near you.  
1884  
1885 Example 1:  
1886 Requirement:  
1887 "I want to know the latest news about  
1888 Tesla"  
1889  
1890 Matched Tools:  
1891 - NewsTool: Stay connected to global  
1892 events with our up-to-date news around  
1893 the world.  
1894  
1895 Example 2:  
1896 Requirement:  
1897 "Please provide me with the current  
1898 stock price of Apple"  
1899  
1900 Matched Tools:  
1901 - FinanceTool: Stay informed with the  
1902 latest financial updates, real-time  
1903 insights, and analysis on a wide range  
1904 of options, stocks, cryptocurrencies,  
1905 and more.  
1906  
1907 Now, for the following requirement, list  
1908 the tools from the available tools that  
1909 can fulfill it.  
1910  
1911 Requirement:  
1912 "Find a local restaurant"  
1913  
1914 Matched Tools:  
1915

You are an assistant helping to match  
tools to requirements, as long as the  
tool description can roughly provide the  
needed information for requirements, it  
does not need to be very specific, ignore  
the proper nouns.  
  
Available Tools:  
- \*\*DietTool\*\*: A tool that simplifies  
calorie counting, tracks diet, and  
provides insights from many restaurants  
and grocery stores. Explore recipe ,  
menus, and cooking tips from millions of  
users, and access recipe consultations  
and ingredient delivery services from  
thousands of stores.  
- \*\*GiftTool\*\*: Provide suggestions for  
gift selection.  
- \*\*HousePurchasingTool\*\*: Tool that  
provide all sorts of information about  
house purchasing  
- \*\*HouseRentingTool\*\*: Tool that  
provide all sorts of information about  
house renting  
- \*\*MemoryTool\*\*: A learning application  
with spaced repetition functionality  
that allows users to create flashcards  
and review them.  
- \*\*RestaurantBookingTool\*\*: Tool for  
booking restaurant  
- \*\*ResumeTool\*\*: Quickly create resumes  
and receive feedback on your resume.  
- \*\*StrologyTool\*\*: Provides strology  
services for you.  
- \*\*local\*\*: Discover and support  
restaurants, shops & services near you.  
  
Example 1:  
Requirement:  
"I want to know the latest news about  
Tesla"  
  
Matched Tools:  
- NewsTool: Stay connected to global  
events with our up-to-date news around  
the world.  
  
Example 2:  
Requirement:  
"Please provide me with the current  
stock price of Apple"  
  
Matched Tools:  
- FinanceTool: Stay informed with the  
latest financial updates, real-time  
insights, and analysis on a wide range  
of options, stocks, cryptocurrencies,  
and more.  
  
Now, for the following requirement, list  
the tools from the available tools that  
can fulfill it.  
  
Requirement:  
"Provide a menu that fits the user's  
diet plan"  
  
Matched Tools:  
DietTool: A tool that simplifies calorie  
counting, tracks diet, and provides

1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985

1986	insights from many restaurants and	Requirement:	2056
1987	grocery stores. Explore recipe , menus,	"Book a table"	2057
1988	and cooking tips from millions of users,		2058
1989	and access recipe consultations and	Matched Tools:	2059
1990	ingredient delivery services from		2060
1991	thousands of stores.		2061
1992		You are an AI assistant helping to match	2062
1993	You are an assistant helping to match	tools to requirements, as long as the	2063
1994	tools to requirements, as long as the	tool description can roughly provid the	2064
1995	tool description can roughly provid the	needed information for requirments, it	2065
1996	needed information for requirments, it	does not need to be very specific,ignore	2066
1997	it does not need to be very specific,ignore	the proper nouns.	2067
1998	the proper nouns.		2068
1999		Available Tools:	2069
2000	Available Tools:	- **DietTool**: A tool that simplifies	2070
2001	- **DietTool**: A tool that simplifies	calorie counting, tracks diet, and	2071
2002	calorie counting, tracks diet, and	provides insights from many restaurants	2072
2003	provides insights from many restaurants	and grocery stores. Explore recipe ,	2073
2004	and grocery stores. Explore recipe ,	menus, and cooking tips from millions of	2074
2005	menus, and cooking tips from millions of	users, and access recipe consultations	2075
2006	users, and access recipe consultations	and ingredient delivery services from	2076
2007	and ingredient delivery services from	thousands of stores.	2077
2008	thousands of stores.	- **GiftTool**: Provide suggestions for	2078
2009	- **GiftTool**: Provide suggestions for	gift selection.	2079
2010	gift selection.	- **HousePurchasingTool**: Tool that	2080
2011	- **HousePurchasingTool**: Tool that	provide all sorts of information about	2081
2012	provide all sorts of information about	house purchasing	2082
2013	house purchasing	- **HouseRentingTool**: Tool that	2083
2014	- **HouseRentingTool**: Tool that	provide all sorts of information about	2084
2015	provide all sorts of information about	house renting	2085
2016	house renting	- **MemoryTool**: A learning application	2086
2017	- **MemoryTool**: A learning application	with spaced repetition functionality	2087
2018	with spaced repetition functionality	that allows users to create flashcards	2088
2019	that allows users to create flashcards	and review them.	2089
2020	and review them.	- **RestaurantBookingTool**: Tool for	2090
2021	- **RestaurantBookingTool**: Tool for	booking restaurant	2091
2022	booking restaurant	- **ResumeTool**: Quickly create resumes	2092
2023	- **ResumeTool**: Quickly create resumes	and receive feedback on your resume.	2093
2024	and receive feedback on your resume.	- **StrologyTool**: Povides strology	2094
2025	- **StrologyTool**: Povides strology	services for you.	2095
2026	services for you.	- **local**: Discover and support	2096
2027	- **local**: Discover and support	restaurants, shops & services near you.	2097
2028	restaurants, shops & services near you.		2098
2029		Example 1:	2099
2030	Example 1:	Requirement:	2100
2031	Requirement:	"I want to know the latest news about	2101
2032	"I want to know the latest news about	Tesla"	2102
2033	Tesla"		2103
2034		Matched Tools:	2104
2035	Matched Tools:	- NewsTool: Stay connected to global	2105
2036	- NewsTool: Stay connected to global	events with our up-to-date news around	2106
2037	events with our up-to-date news around	the world.	2107
2038	the world.		2108
2039		Example 2:	2109
2040	Example 2:	Requirement:	2110
2041	Requirement:	"Please provide me with the current	2111
2042	"Please provide me with the current	stock price of Apple"	2112
2043	stock price of Apple"		2113
2044		Matched Tools:	2114
2045	Matched Tools:	- FinanceTool: Stay informed with the	2115
2046	- FinanceTool: Stay informed with the	latest financial updates, real-time	2116
2047	latest financial updates, real-time	insights, and analysis on a wide range	2117
2048	insights, and analysis on a wide range	of options, stocks, cryptocurrencies,	2118
2049	of options, stocks, cryptocurrencies,	and more.	2119
2050	and more.		2120
2051		Now, for the following requirement, list	2121
2052	Now, for the following requirement, list	the tools from the available tools that	2122
2053	the tools from the available tools that	can fulfill it.	2123
2054	can fulfill it.		2124
2055		Requirement:	2125



2126	"Offer astrology insights on the best	Requirement:	2196
2127	date for dinner"	"Select a thoughtful gift for the dining	2197
2128		companion"	2198
2129	Matched Tools:		2199
2130	StrologyTool: Povides strology services	Matched Tools:	2200
2131	for you.	GiftTool: Provide suggestions for gift	2201
2132		selection.	2202
2133	You are an AI assistant helping to match		2203
2134	tools to requirements, as long as the	Tool Matches:	2204
2135	tool description can roughly provid the	- Requirement: 'Find a local restaurant '	2205
2136	needed information for requirments, it	matched with Tools: None	2206
2137	does not need to be very specific,ignore	- Requirement: 'Provide a menu that fits	2207
2138	the proper nouns.	the user's diet plan' matched with	2208
2139		Tools: DietTool	2209
2140	Available Tools:	- Requirement: 'Book a table' matched	2210
2141	- **DietTool**: A tool that simplifies	with Tools: None	2211
2142	calorie counting, tracks diet, and	- Requirement: 'Offer astrology insights	2212
2143	provides insights from many restaurants	on the best date for dinner' matched	2213
2144	and grocery stores. Explore recipe ,	with Tools: StrologyTool	2214
2145	menus, and cooking tips from millions of	- Requirement: 'Select a thoughtful gift	2215
2146	users, and access recipe consultations	for the dining companion' matched with	2216
2147	and ingredient delivery services from	Tools: GiftTool	2217
2148	thousands of stores.		2218
2149	- **GiftTool**: Provide suggestions for	Does the toolset exactly solve the query	2219
2150	gift selection.	? No	2220
2151	- **HousePurchasingTool**: Tool that	Tools to Keep: DietTool, StrologyTool,	2221
2152	provide all sorts of information about	GiftTool	2222
2153	house purchasing		2223
2154	- **HouseRentingTool**: Tool that	Unsolved Problems:	2224
2155	provide all sorts of information about	- Find a local restaurant	2225
2156	house renting	- Book a table	2226
2157	- **MemoryTool**: A learning application		
2158	with spaced repetition functionality		
2159	that allows users to create flashcards		
2160	and review them.		
2161	- **RestaurantBookingTool**: Tool for		
2162	booking restaurant		
2163	- **ResumeTool**: Quickly create resumes		
2164	and receive feedback on your resume.		
2165	- **StrologyTool**: Povides strology		
2166	services for you.		
2167	- **local**: Discover and support		
2168	restaurants, shops & services near you.		
2169			
2170	Example 1:		
2171	Requirement:		
2172	"I want to know the latest news about		
2173	Tesla"		
2174			
2175	Matched Tools:		
2176	- NewsTool: Stay connected to global		
2177	events with our up-to-date news around		
2178	the world.		
2179			
2180	Example 2:		
2181	Requirement:		
2182	"Please provide me with the current		
2183	stock price of Apple"		
2184			
2185	Matched Tools:		
2186	- FinanceTool: Stay informed with the		
2187	latest financial updates, real-time		
2188	insights, and analysis on a wide range		
2189	of options, stocks, cryptocurrencies,		
2190	and more.		
2191			
2192	Now, for the following requirement, list		
2193	the tools from the available tools that		
2194	can fulfill it.		
2195			