Level-Navi Agent: A Framework and benchmark for Web Search Agents

Anonymous submission

Abstract

Large language models (LLMs), adopted to understand human language, drive the development of artificial intelligence (AI) web search agents. Compared to traditional search engines, LLM-powered AI search agents are capable of understanding and responding to complex queries with greater depth, enabling more accurate operations and better context recognition. However, little attention and effort has been paid to the web search, which results in that the capabilities of open-source models have not been uniformly and fairly evaluated. The difficulty lies in lacking three aspects: an unified agent framework, an accurately labeled dataset, and a suitable evaluation metric. To address these issues, we propose a general-purpose and training-free web search agent by level-aware navigation, called Level-Navi Agent, accompanied by a well-annotated dataset (Web24) and a suitable evaluation metric. Level-Navi Agent can think through complex user questions and conduct searches across various levels on the internet to gather information for questions. Meanwhile, we provide a comprehensive evaluation of state-of-the-art LLMs under fair settings. To further facilitate future research, source code will be made publicly available.

1 Introduction

Information gathering is a key step in the interaction between humans and their environment. Search engines are widely used for information acquisition (Brin and Page, 1998). With the development of large language models (LLMs) (Ye et al., 2023) (Achiam et al., 2023), AI search agents based on LLMs have become an emerging and challenging research topic (Nakano et al., 2021). Retrieve-Augmented Generation (RAG) is used to improve the precision of model responses (Ram et al., 2023). Existing methods (Chan et al., 2024) (Siriwardhana et al., 2023) leverage the powerful language capabilities of LLMs to perform retrieval

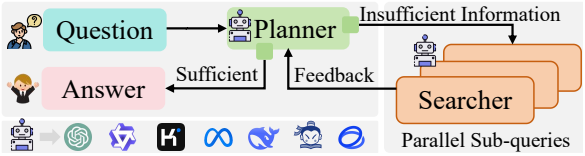

Figure 1: Pipeline of our Level-Navi Agent.

based on user queries and use the retrieved relevant texts to improve the reliability of the model's answers. This advanced ability to understand and analyze questions exceeds that of traditional search engines, driving a revolutionary transformation in AI-powered search (Spatharioti et al., 2023). However, these methods do not further explore how LLMs handle complex questions. The simple text retrieval approach cannot fully align with web search scenarios. And irrelevant texts retrieved have negative impacts on the quality of responses (Asai et al., 2023).

Therefore, refined methods (Chen et al., 2024)(Reddy et al., 2023) are proposed to construct AI search agents. Mindsearch (Chen et al., 2024) employs the concept of Directed Acyclic Graphs to structure the agent's plan, breaking down complex reasoning questions with the aim of simulating the human mind, thereby striving to deliver more comprehensive answers. Infogent (Reddy et al., 2024) utilizes an information aggregation approach to update the retrieved information. Determine whether the retrieved texts meet the required conditions and improve the accuracy of responses by controlling the quality of the information. These methods achieve promising results under detailed process planning. However, the research community still lacks comprehensive studies that can genuinely reveal the true capabilities of various open-source and closed-source LLMs in the web search scenario.

Existing LLM-driven search agents require fine-tuning or rely on high-performance close-source models, making it difficult for researchers to inves-

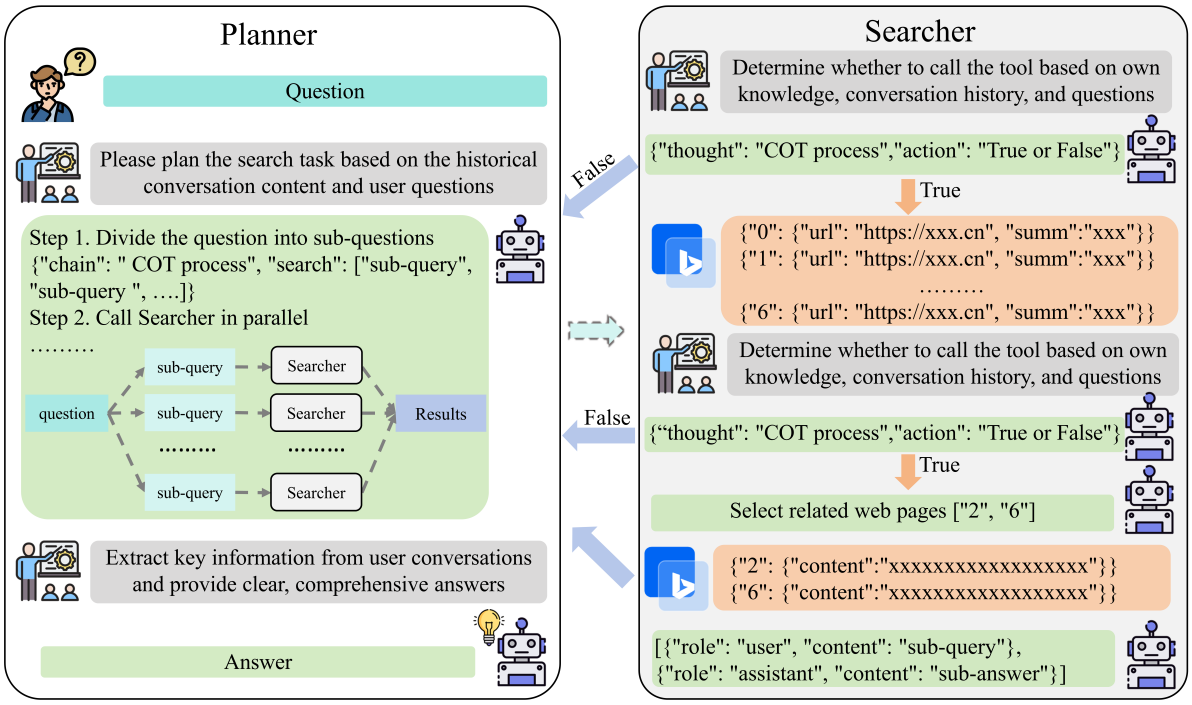

Figure 2: The overall framework of our proposed Level-Navi Agent.

tigate the capabilities of various LLMs due to their costs. Datasets for evaluating the capabilities of LLMs in Chinese scenarios are constructed, such as CMMLU (Li et al., 2023) and AlignBench (Liu et al., 2024). The advent of these datasets shows the demand for model evaluation in real-world Chinese contexts. However, in the field of web search, a suitable Chinese web search dataset for quantitative evaluation is lacking. Meanwhile, we reveal that traditional metrics like F1 and ROUGE (Lin, 2004) do not consider the semantic information across various versions, which poses challenges when comparing the performance of different models.

To address the aforementioned issues, we propose a training-free AI search agent framework for both open-source and close-source models, as illustrated in Fig. 1. Meanwhile, we provide a new Chinese web search dataset and a new evaluation metric to evaluate the performance of LLMs in the Chinese task. Overall, our contributions are as follows.

- **We propose a general-purpose training-free web search agent framework, called Level-Navi Agent.** The question from the user is first analyzed and decomposed by the Planner. Then the sub-questions are provided to the Searcher, which will collect information at different levels. By iterating this step, Level-Navi Agent eventually collects enough information to answer the initial question. Level-Navi Agent does not require training, allowing any open-source LLM to be deployed.

- **We provide a well-annotated benchmark dataset (Web24) for Chinese web search.** Our dataset is capable of a diverse and detailed classification of questions and sources, all sourced entirely from the Chinese internet. Considering the limitations of traditional metrics, we adopted four reasonable metrics to evaluate the ability of different LLMs when execute the Level-Navi Agent. Through our benchmark, the performance of different LLMs for AI web search is clearly presented.

- **We reveal the factors that limit model performance in executing web search agent tasks.** First, we find that the model exhibits an “overconfidence" phenomenon, where it refrains from calling functions for web searches even when it does not know the answer, leading to incorrect responses. Second, the model demonstrates low “task fidelity" during task execution, meaning it fails to fully understand our instructions, resulting in non-compliant answers and poor response quality.

2 Level-Navi Agent

This section introduces the details of our Level-Navi Agent, which mainly contains the Planning Agent and Level-info Agent. The overall structure of our Level-Navi Agent is shown in Fig. 2

2.1 Planning Agent

As a key component in our Level-Navi Agent, the Planner directly affects the overall performance of the web search process. Our Planning Agent plans the trajectory path by chain of thought (Wei et al., 2022) and iterative refinement. One the one hand, when the user inputs a question, our Agent first understands and breaks down the problem through chain of thought. Noticing the difference from the conventional chain of thought steps (Jin et al., 2024; Wang et al., 2023), we let the LLM first think through and determine the information that should be collected next, then generate a list of sub-questions that can be searched in parallel at this stage. The main reason is that in scenarios where complex reasoning is required, a complete plan may seem very clear. However, since we rely on another Agent to gather information, the content obtained each time is not necessarily sufficient or complete, which involves dynamically adjusting the plan every time. To avoid such a complex and redundant process design, we use prompt to enforce that the Planning Agent only giving the list of sub-question that needed to be obtained in the next step. After obtaining feedback from each step, it repeats this process iteratively until the Agent judges that the current information is sufficient to answer the question. We demonstrate the detailed process of our Planning Agent in Algorithm 1.

Our Planning Agent utilizes prompt engineering, making the framework general-purpose and fine-tuning-free. This design ensures compatibility with any open-source or closed-source model.

2.2 Level-Info Agent

As depicted in the right of Fig. 2, the primary task of the Searcher is to obtain relevant information feedback to the Planner by conducting online searches based on the sub-problems received. To enrich the information obtained while enhancing its flexibility, we construct an Agent that dynamically simulates the human information acquisition process through a chain of thought, which we call the Level-Info Agent.

As its name suggests, the Level-Info Agent is capable of dynamically obtaining information at various levels and can return results at any moment, significantly improving the agent's operational speed. Firstly, when faced with an input sub-query, the Level-Info Agent determines whether the information can be answered using its own knowledge. If it can, it returns the result directly; otherwise, it proceeds with a web search. Then, the Agent will call the web search function to return the results of the online search. At this step, the returned materials will only be the summary parts of the web pages. Here, we also have the Agent think and determine whether the current materials obtained can answer the sub-query. If the information is sufficient, it will provide a direct response; otherwise, it will proceed to the next step of opening relevant websites. When performing this step, we also use function calls to let the model select and open relevant websites. After obtaining the information from the web page, it will summarize and respond.

Our Level-Info Agent has up to three levels for providing information feedback, this avoids the need to always read a large number of websites, which consumes a significant amount of tokens, and reduces the cost of search engine APIs.

Algorithm 1 The process of Planning Agent

Input: Q for user's question
Output: R for agent's response
Variables: H denotes the set of history context, M is the collected information.

$H \leftarrow \emptyset$
Add Q to H
while True **do**
  Result $\leftarrow CoT(H)$
  if Result == "no" **then**
    $M \leftarrow \emptyset$
    $Q_{sub} \leftarrow$ Result.sub-question
    parallel for each $q \in Q_{\text{sub}}$ **do**
    $M[q] \leftarrow Searcher(q)$
    end parallel
    Add M to H
  else
    $R \leftarrow$ Result.response
    break
  end if
end while
Return R

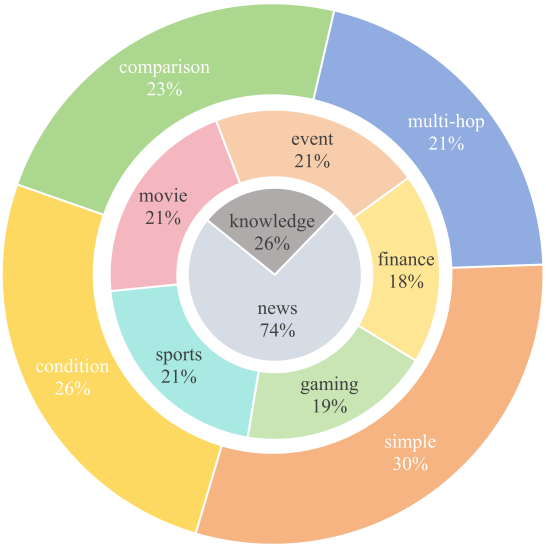

Figure 3: Source, domain and type distribution of Web24 Dateset.

Domain	Simp.	Cond.	Comp.	Multi-hop	All
Finance	23	23	22	22	90
Gaming	28	23	23	17	91
Sports	42	18	21	19	100
Movie	29	33	24	14	100
Event	23	27	22	28	100
All	145	124	112	100	481

Table 1: Distribution of problem domains and types.

3 Benchmark Construction

This section presents the benchmark and evaluation metrics we constructed. The Web24 dataset, which is specifically designed for web search agents, is introduced in Section 3.1. Accordingly, a new evaluation metric which is more suitable for web search agents is introduced in Section 3.2.

3.1 Data Composition

Our Web24 categorizes question-answer pairs into fine-grained divisions based on sources, domains, and types. In the process of evaluating the performance of web search agents, we aim to minimize the influence of the model's internal knowledge to genuinely assess the search capabilities. Therefore, we ensure that the majority of question-answer pairs are sourced from the news, as illustrated in Fig. 3. All cases sourced from the news are entirely from news reports on the Chinese internet before December 2024.

To evaluate the performance of web search agents, The impact of models' internal knowledge to accurately assess their search capabilities should be minimized. To achieve this, we construct our benchmark primarily using question-answer pairs derived from news sources (Fig. 3). All news-based cases are exclusively drawn from Chinese internet news reports published prior to December 2024.

To realistically simulate real-world web search scenarios, we organize our question-answer pairs into five distinct domains: Finance, Gaming, Sports, Movies, and Events. Each domain is carefully designed to represent typical user information needs. Furthermore, to better characterize the structure of these question-answer pairs, we classify all questions into four types: simple, conditional, comparative, and multi-hop. The distribution of these domains and question types is presented in Table 1. The specific characteristics of each question type are detailed below:

- **Simple Questions:** Direct queries seeking a single and factual piece of information. For example, "When was the Chinese national anthem released?"
- **Conditional Questions:** Queries that incorporate specific temporal or situational constraints, requiring the answer to satisfy given conditions. For example, "when was the announcement of the third batch of China's Time-Honored Brands?"
- **Comparative Questions:** Queries that necessitate analyzing and contrasting attributes between two or more distinct entities. For example, "who has a higher career total points, Kobe or LeBron?"
- **Multi-hop Questions:** Complex queries that demand iterative reasoning across multiple information sources, where answering requires chaining together several intermediate search and inference steps. For example, "where is the headquarters of the company of the courier who collected the 1500 billionth package this year?"

3.2 Evaluation Metrics

To comprehensively assess the capabilities of LLMs in performing web search tasks, we consider multiple aspects and use four scoring metrics to evaluate the capabilities holistically. Detailed description of the evaluation metrics are detailed as follows.

Correctness Scores (S_{co}). In order to comprehensively assess the precision of the response compared to ground truth answers, we employ an LLM as an evaluator to assess the consistency and precision of the generated answers compared to ground truth answers (Yang et al., 2024). We use this evaluator to score the responses on a scale of 1 to 10, and then normalize these scores to a range of 0 to 1.

Semantic Similarity Scores (S_{simi}). By using an embedding model (Xiao et al., 2024), we can directly calculate the vectors of discrete tokens mapped to a high-dimensional continuous space, and directly compute the similarity between text vectors through mathematical methods.

Relevance Scores (S_{rele}). This metric primarily examines the model's faithfulness to the task execution trajectory (Es et al., 2024). Based on the responses generated by the LLM, another evaluation LLM will generate multiple questions that are inferred from the responses, and then calculate the semantic similarity between these inferred questions and the originally given questions, then the maximum value is taken as the final score.

Searcher Count (S_c). This metric assesses the ability of LLMs to understand and break down questions. We have counted the number of times the Level-Info Agent is invoked in each task and used the average number of invocations as an evaluation metric.

Ultimately, we express the total score (1-100) as a weighted sum of the aforementioned four metrics. The formula for calculating the total score is as follows:

$$S_{final} = 60 \cdot S_{co} + 15 \cdot S_{simi} + 15 \cdot S_{rele} + 10 \cdot e^{-S_c}. \quad (1)$$

4 Experiments and Analysis

In this section, we will present the experimental results and analysis of our constructed framework, benchmark and evaluation metric. We utilized 16 models to operate Level-Navi Agent, encompassing open-source and closed-source models, which are

a) Open-source. The open-source models we used primarily from the Chinese community, including InternLM series (Cai et al., 2024), GLM-4 (GLM et al., 2024), Qwen series (Bai et al., 2023) and Llama series (AI@Meta, 2024)

b) Closed-source. For closed-source models, we utilized ERNIE-3.5 from Baidu, Moonshot-v1 from Moonshot AI, and GPT-4o (Achiam et al., 2023) from OpenAI. Note that Deepseek-V2.5 (DeepSeek-AI, 2024) is an open-source model, but due to its large parameter size, we call it in the form of an API.

Besides, we also evaluate the Reasoning Model to perform our tasks, including both DeepSeek-R1 (Guo et al., 2025) and its distilled version DeepSeek-R1-Distill-Qwen-32B.

4.1 Experimental Results of Our Agent on the Web24 Dataset

Quantitative results presented in Tables 2 and 3 reveals that Qwen2.5-72B and DeepSeek-V3 demonstrate superior performance. Through systematic analysis of all experimental results, we identify the following critical findings regarding the performance Web Search Agent:

Diminishing Marginal Returns of Model Parameters. Focusing on the scores of the Qwen series models in Table 2, doubling the size from 3B to 14B improved performance by 6 points, but from 14B to 72B, it only gained 3 points. From the results in Table 3, we found that although the performance of closed-source models is quite good, there is not a significant gap with Qwen2.5-72B. From the analysis above, it can be seen that: To further enhance the performance of LLMs in executing web search tasks, researchers should focus on how to obtain higher-quality information, since the diminishing marginal effect of model parameters is quite clear.

Few-shot Prompts Enhance Pass Rates. For all models, we implemented three types of prompt methods: zero-shot, one-shot, and three-shot (Brown et al., 2020). We calculated the number of error responses for each evaluation and compared it with the total number of the dataset to derive the pass rate. The chain of thought method and few-shot prompt combination have been proven effective in previous research (Liang et al., 2023) (Ma et al., 2023), this conclusion is also reflected in our experiments. From Table 2, it can be seen that the three-shot method significantly improved the pass rate of the Agent compared to the zero-shot approach. In general, we recommend providing few-shot prompts when executing agent tasks. This approach is not only simple and cost-effective but also enhances the model's performance in various aspects.

From the results in Table 4, we can observe that the final score of DeepSeek-R1 is slightly lower

Model	Few-shot	S_{final}	S_{co}	S_{rele}	S_{simi}	S_c	Pass rate
Internlm2.5-7B	zero-shot	49.48	0.47	0.81	0.56	2.62	0.92
	one-shot	47.76	0.45	0.79	0.56	2.98	0.91
	three-shot	49.31	0.47	0.8	0.56	2.65	0.95
Internlm2.5-20B	zero-shot	55.02	0.57	0.80	0.57	3.62	0.93
	one-shot	57.70	0.61	0.81	0.58	3.68	0.96
	three-shot	55.43	0.57	0.80	0.57	2.69	0.97
GLM-4-9B	zero-shot	63.25	0.66	0.83	0.67	2.16	0.94
	one-shot	40.82	0.34	0.79	0.54	3.05	0.89
	three-shot	43.43	0.37	0.81	0.56	2.69	0.92
Qwen2.5-3B	zero-shot	60.17	0.62	0.84	0.64	2.56	0.85
	one-shot	54.28	0.54	0.82	0.57	2.27	0.86
	three-shot	60.45	0.63	0.84	0.59	2.12	0.86
Qwen2.5-7B	zero-shot	63.12	0.65	0.85	0.60	1.44	0.99
	one-shot	65.01	0.69	0.84	0.61	1.68	1.00
	three-shot	65.84	0.70	0.84	0.62	1.64	1.00
Qwen2.5-14B	zero-shot	68.34	0.75	0.84	0.61	1.84	0.99
	one-shot	68.45	0.75	0.84	0.61	1.77	1.00
	three-shot	68.39	0.75	0.84	0.61	1.81	1.00
Qwen2.5-32B	zero-shot	68.74	0.76	0.83	0.61	1.87	1.00
	one-shot	69.05	0.76	0.84	0.61	1.77	1.00
	three-shot	68.82	0.76	0.83	0.61	1.82	1.00
Qwen2.5-72B	zero-shot	**69.99**	0.78	0.83	0.60	1.75	1.00
	one-shot	**69.48**	0.77	0.83	0.60	1.70	1.00
	three-shot	**71.30**	0.80	0.83	0.60	1.69	1.00
Llama3.1-8B	zero-shot	37.02	0.30	0.74	0.51	3.60	0.88
	one-shot	34.54	0.28	0.68	0.49	3.97	0.92
	three-shot	32.45	0.27	0.61	0.46	3.89	0.93
Llama3.1-70B	zero-shot	41.56	0.35	0.76	0.54	2.24	0.57
	one-shot	52.28	0.50	0.81	0.60	2.18	0.80
	three-shot	51.02	0.48	0.81	0.61	2.39	0.90

Table 2: Open Source Model Results with GPT-4o Evaluation.

than that of Deepseek-V3. Such results demonstrate the reasoning model is slightly inferior to the regular model in general tasks such as multi-turn conversations and function calling task (Guo et al., 2025). The final score of its distilled model mainly decreased due to a lack of correctness, which reflects that distillation primarily reduces the model's judgment capability. Overall, although reasoning models demonstrate strong capabilities in complex problems and planning, they do not show particular advantages in general tasks. Considering their substantial token consumption, whether to use reasoning models in similar tasks remains a subject of debate.

4.2 Comparison with Other Products

We compare our Agent (Qwen2.5-72B) with established market products, including Chinese LLM services Kimi and Doubao, and GPT-4o. We test 100 web24 dataset examples for answers on their websites. As shown in Fig. 4, the scores for correctness, relevance, and semantic similarity were similar across the three LLM products. Kimi edged out slightly in all metrics, but the differences were not significant. Overall, our Agent matched commercial products in these metrics, such results vali-

Model	Few-shot	S_{final}	S_{co}	S_{rele}	S_{simi}	S_c	Pass rate
Deepseek-V3 *	three-shot	**74.70**	0.84	0.85	0.62	1.49	1.00
ERNIE-3.5	three-shot	72.19	0.80	0.87	0.64	1.87	1.00
Moonshoot-v1	three-shot	70.89	0.77	0.87	0.64	1.59	0.99
GPT-4o	three-shot	71.33	0.79	0.85	0.62	1.67	1.00

* This model is open-source, but due to its large parameter volume, we call it using the API.

Table 3: Close Source Model Results with Qwen2.5-72B Evaluation.

Model	Few-shot	S_{final}	S_{co}	S_{rele}	S_{simi}	S_c	Pass rate
DeepSeek-R1	three-shot	**69.50**	0.78	0.75	0.61	1.47	1.00
DeepSeek-R1-Distill-Qwen-32B	three-shot	63.87	0.68	0.80	0.61	1.65	1.00

Table 4: Reasoning Model Results with GPT-4o Evaluation.

date the effectiveness of our proposed Level-Navi Agent framework.

4.3 Analysis of Metrics

Limitation of Traditional Metrics. Table 5 presents the results of model responses using traditional methods. By comparing and analyzing the performance of LLMs with different parameter sizes on the same task, we discover the increase in model parameters do not universally lead to an improvement in F1 and ROUGE scores and even caused declines. Because traditional F1 and ROUGE methods are based on token matching and do not consider semantic information, more detailed answers tend to deviate from brief standard answers, leading to lower scores. Only the Recall metric benefits from more comprehensive responses. These phenomena all demonstrate the limitations of traditional metrics. Meanwhile, this inspires us to design a metric that is more suitable for evaluating the task under study.

The effectiveness of our metrics. From the quantitative evaluation results in Tables 2 and 3, it can be observed that using our metrics, the performance distribution of various models aligns with empirical knowledge and common sense. In terms of Correctness Scores, the advantage brought by model parameter size is clearly demonstrated. Meanwhile, Semantic Similarity Scores and Relevance Scores consistently reflect the capability differences among models. Through the overall scores, anyone can intuitively discern the performance differences between models. These findings strongly validate the effectiveness of the quantitative evaluation metric proposed in this paper.

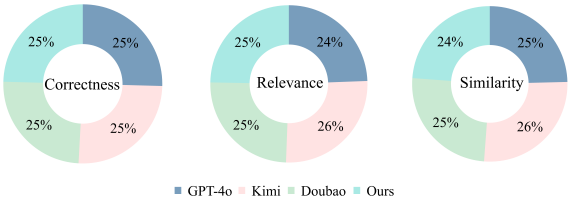

Figure 4: Comparison with other commercial products based on our metrics.

Model	ROUGE	F1	Recall
Internlm2.5-7B	0.12	0.09	0.69
Internlm2.5-20B	0.12	0.09	0.74
Qwen2.5-3B	0.20	0.18	0.71
Qwen2.5-7B	**0.23**	**0.21**	0.74
Qwen2.5-14B	0.19	0.16	0.78
Qwen2.5-32B	0.19	0.16	0.78
Qwen2.5-72B	0.16	0.12	**0.81**

Table 5: Using traditional metrics (Under three-shot method).

Therefore, we believe that using LLM-based evaluation methods (Zhuge et al., 2024) rather than traditional ones in open-ended tasks like web search Q&A can better reflect real-world conditions.

4.4 Error Analysis and Discussion

By error analysis and experiments, we summarize the shortcomings in models and evaluation methods.

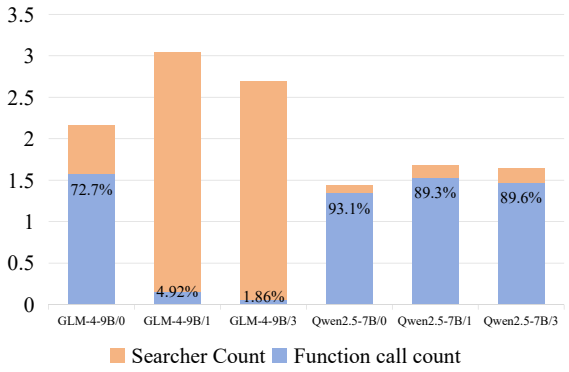

Figure 5: Comparison of Searcher and Function call counts, the percentage at the bottom represent the Searcher count / Function call count(0 for zero-shot; 1 for one-shot; 3 for three-shot).

Model	Few-shot	Values (%)
Llama3.1-8B	zero-shot	25.16
	one-shot	27.86
	three-shot	**23.64**
Llama3.1-70B	zero-shot	2.08
	one-shot	**0.42**
	three-shot	**0.42**

Table 6: Statistics of Non-compliant Responses.

Overconfidence Phenomenon in Web Search Function Usage. The quantitative evaluation results in Table 2 show the scores of GLM-4-9B are lower than expected. We uncovered the reason for this phenomenon by calculating the average Search Count invoked per task and the actual number of Web Search Function calls. In Fig. 5, we compared the gap between GLM-4 and Qwen2.5-7B on the aforementioned metrics. We observed that Qwen2.5-7B's function call rate reached about 90% of the Agent invocations, while GLM-4-9B's ratio dropped from 30% to a single-digit percentage. Given that 70% of the answers in the dataset come from news sources, GLM-4-9B cannot possibly answer correctly without invoking web searches. We refer to this phenomenon as "Overconfidence" (Huang et al., 2024).

The issue indicates that LLMs might overestimate their question-answering abilities during training, overlooking the need for external resources in certain situations (Xiong et al., 2024). To address this overconfidence, we recommend that developers balance positive and negative examples in the training dataset to improve LLMs' function-calling capabilities.

Low Task Fidelity in Conducting Chinese Tasks. When assessing LLM agents, we prioritize whether the model comprehends and adheres to instructions to answer questions. We call it "Task Fidelity", which reflects the model's faithfulness in executing instructions. The Relevance Scores do not take into account the correctness of the model's response, so it can reflect the end-to-end Task Fidelity.

In Table 2, the relevance scores of Llama3.1-8B did not behave as expected compared to other models; instead, they fluctuated significantly. Upon examining the outputs of the Llama 3.1 series models, we found that a considerable portion of the responses did not fully comply with our instructions, with some incorrectly mixing the given instructions with the answers. Table 6 more detailedly reflects this type of non-compliant response. The introduction of the few-shot method did not improve this issue for Llama3.1-8B, only Llama3.1-70B showed improvement.

We view non-compliant responses as indicating low Task Fidelity. The LLM's struggle to grasp the intent of instructions in lengthy Chinese contexts can greatly impair task performance. Developers should focus on ensuring smaller LLMs to maintain the multilingual ability as bigger ones.

5 Conclusion

In this paper, we introduce the Level-Navi Agent and a novel benchmark for web search tasks. The Level-Navi Agent offers an innovative solution to web search challenges through the collaboration of multiple agents and a hierarchical approach to reasoning and searching. Starting from the Chinese open source-community, we employed reasonable metrics to comprehensively assess the performance of various LLMs in executing web search tasks. This analysis sheds light on the true capabilities of current LLMs when performing web search tasks within the Chinese Internet. Meanwhile, we built a benchmark that can be used for web search agents through manual annotation, which may facilitate the evaluation and application of various models. Through data-driven error analysis, we identify the limitations of LLMs in handling web search tasks and provide recommendations for improvement, contributing to the advancement of this field.

Limitations

While our proposed Level-Navi Agent provide a comprehensive framework for evaluating LLM-based web search agents, there are several limitations worth noting: 1) Although we design a hierarchical approach to retrieve web information, the differences in capabilities among various search engines are not reflected. How to construct more fine-grained information retrieval remains a worthwhile research question; 2) Our Web24 dataset covers news information from multiple domains, but it does not include academic papers as information sources. This aspect could be further explored in future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Self-reflective retrieval augmented generation. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, and Mei Li et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117. Proceedings of the Seventh International World Wide Web Conference.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and Agarwal et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, and 3 others. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to refine queries for retrieval augmented generation. In *First Conference on Language Modeling*.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Jiangning Liu, Wenwei Zhang, Kai Chen, and Feng Zhao. 2024. Mindsearch: Mimicking human minds elicits deep ai searcher. *arXiv preprint arXiv:2407.20183*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, and Jiajie Zhang et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. *arXiv preprint arXiv:2310.08395*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Xiaotao Gu, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. Align-Bench: Benchmarking Chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11621–11640, Bangkok, Thailand. Association for Computational Linguistics.

Xilai Ma, Jing Li, and Min Zhang. 2023. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2334–2352, Singapore. Association for Computational Linguistics.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Ouyang Long, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Revanth Gangi Reddy, Yi Ren Fung, Qi Zeng, Manling Li, Ziqi Wang, Paul Sullivan, and Heng Ji. 2023. Smartbook: Ai-assisted situation report generation. *ArXiv*, abs/2303.14337.

Revanth Gangi Reddy, Sagnik Mukherjee, Jeonghwan Kim, Zhenhailong Wang, Dilek Hakkani-Tur, and Heng Ji. 2024. Infogent: An agent-based framework for web information aggregation. *arXiv preprint arXiv:2410.19054*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Sofia Eleni Spatharioti, David M Rothschild, Daniel G Goldstein, and Jake M Hofman. 2023. Comparing traditional and llm-based search for consumer choice: A randomized experiment. *arXiv preprint arXiv:2307.03744*.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, and 1 others. 2024. Crag–comprehensive rag benchmark. *arXiv preprint arXiv:2406.04744*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, and 1 others. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, and 1 others. 2024. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*.