X-WebAgentBench: A Multilingual Interactive Web Benchmark for Evaluating Global Agentic System

Peng Wang[1,2,*] **Ruihan Tao**[1,*] **Qiguang Chen**[1] **Mengkang Hu**[3] **Libo Qin**[1,2,†]
[1] School of Computer Science and Engineering, Central South University, China
[2] Key Laboratory of Data Intelligence and Advanced Computing in Provincial Universities, Soochow University, China
[3] The University of Hong Kong, China
wpengxss@gmail.com, aarontao757@gmail.com, lbqin@csu.edu.cn

Abstract

Recently, large language model (LLM)-based agents have achieved significant success in interactive environments, attracting significant academic and industrial attention. Despite these advancements, current research predominantly focuses on English scenarios. In reality, there are over 7,000 languages worldwide, all of which demand access to comparable agentic services. Nevertheless, the development of language agents remains inadequate for meeting the diverse requirements of multilingual agentic applications. To fill this gap, we introduce X-WebAgentBench, a novel multilingual agent benchmark in an interactive web environment, which evaluates the planning and interaction performance of language agents across multiple languages, thereby contributing to the advancement of global agent intelligence. Additionally, we assess the performance of various LLMs and cross-lingual alignment methods, examining their effectiveness in enhancing agents. Our findings reveal that even advanced models like GPT-4o, when combined with cross-lingual techniques, fail to achieve satisfactory results. We hope that X-WebAgentBench can serve as a valuable benchmark for multilingual agent scenario in real-world applications.

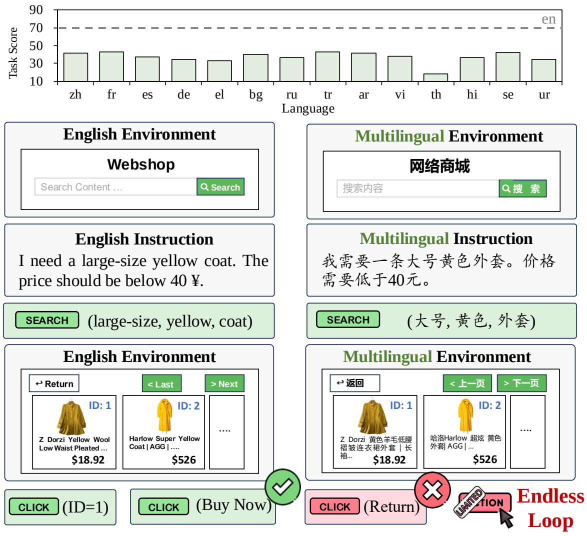


Figure 1: Comparison of performance in English and multilingual settings on GPT-4o: The English task score statistics presented above are derived from Yang et al. (2023) based on the English WebShop benchmark (Yao et al., 2022), while the multilingual task scores are obtained through evaluation on our own benchmark.

1 Introduction

Large language models (LLMs) have demonstrated remarkable success across various natural language processing (NLP) tasks (Zhao et al., 2023; Qin et al., 2024b; Chen et al., 2024a, 2025; Wang et al., 2025), particularly in the development of language agents (Mu et al., 2024; Hu et al., 2024b,c; Shinn et al., 2023). These agentic systems employ various approaches to interact effectively with the real world. Typically, ReAct (Yao et al., 2023) combines reasoning traces with task-specific actions to mitigate error propagation during chain-of-thought reasoning and interaction, enhancing the reliability of decision-making processes. Furthermore, Reflexion (Shinn et al., 2023) incorporates feedback to induce better interaction. Moreover, MetaGPT (Hong et al., 2024) integrates human workflows to decompose complex tasks in software programming, demonstrating efficiency and scalability in collaborative environments.

To better evaluate and understand the capabilities of agents, it emerges a series of research to provide diverse interactive benchmarks to evaluate the ability of language agents in complex environments, such as Mind2Web (Deng et al., 2024), OsWorld (Xie et al., 2024), and WebArena (Zhou et al., 2024). Despite the significant advancements in current language agent benchmarks, most developments remain centered on English-language environments. Actually, with globalization acceler-

[*] Equal Contribution.
[†] Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 19320–19335
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Benchmark	# Language	# Tasks	Interactivity	Multilingual Instruction	Multilingual Environment
MiniWoB++ (Liu et al., 2018)	1	100	✔	✘	✘
RUSS (Xu et al., 2021)	1	80	✔	✘	✘
WebSRC (Chen et al., 2021a)	1	400k	✔	✘	✘
WebShop (Yao et al., 2022)	1	12,087	✔	✘	✘
WebShop-Core (Yao et al., 2023)	1	500	✔	✘	✘
Mind2Web (Deng et al., 2024)	1	2,350	✘	✘	✘
Mind2Web-Live (Pan et al., 2024)	1	542	✔	✘	✘
WebArena (Zhou et al., 2024)	1	821	✔	✘	✘
OmniACT (Kapoor et al., 2025)	1	9,802	✔	✘	✘
X-WebAgentBench	14	2,800	✔	✔	✔

Table 1: Comparison of X-WebAgentBench with previous agent benchmarks. X-WebAgentBench could provide multilingual instruction and multilingual interactive environment.

ating, the real-world application of language agents increasingly involves multilingual contexts (Qin et al., 2025). For example, as illustrated in Figure 1, users engaging in global e-commerce, such as shopping on Amazon, tend to rely on their native languages for searching, selecting, and purchasing products. Yet, when mainstream language agents are employed in these scenarios, a critical issue emerges: current research often overestimates their effectiveness. Specifically, as illustrated in Figure 1, multilingual performance lags behind English-only performance by over 20%. Insights gained from English environments are challenging to generalize to multilingual contexts, which blocks the development of multilingual agents. Moreover, there is a notable absence of standardized benchmarks for evaluating language agents in multilingual scenario. Without such benchmarks, identifying gaps, limitations, and constraints in existing multilingual agentic systems remains challenging, hindering their global development.

Motivated by this, to fill this gap, we introduce X-WebAgentBench, a multilingual interactive website environment benchmark comprising 14 languages, 2,800 instructions, and 589,946 products. In comparison to previous benchmarks (see Table 1), X-WebAgentBench first offers a comprehensive multilingual setting, with a focus on two primary aspects: (1) ***multilingual instruction*** (2) ***multilingual environment*** for language agents. Specifically, *multilingual instruction* aims to evaluate the capabilities to interpret and execute complex commands in various languages based on appropriate actions, ensuring consistent and accurate performance regardless of the linguistic origin of the request. *Multilingual environment*, on the other hand, focuses on the agent's capability to interact seamlessly with web interfaces and systems designed in multiple languages, guided by the rewards provided by X-WebAgentBench.

Moreover, to investigate the performance and relevant drawbacks of language agents in multilingual scenarios, we conduct comprehensive experiments on X-WebAgentBench, yielding the following key findings: (1) *For larger LLMs, advanced cross-lingual alignment methods can significantly enhance multilingual performance.* (2) *For smaller LLMs, translating multilingual environments into English can effectively mitigate the limitations of their multilingual capabilities.* (3) *The simple integration of existing agent-based and cross-lingual strategies fails to adequately address the challenges posed by multilingual agent systems.*

In summary, our contributions can be summarized as:

- We first highlight the research community's overly optimistic estimates of agent systems, as current research primarily focuses on English-centric environments.
- To the best of our knowledge, we make the first attempt to introduce a novel multilingual agent benchmark (X-WebAgentBench) that integrates multilingual instructions and environments, taking a meaningful step to build a multilingual agentic system.
- Our analysis reveals significant challenges faced by agentic systems in multilingual contexts, particularly in low-resource languages. Further, we provide actionable insights and recommendations to enhance language model performance in these settings.

To facilitate the further research, our code will be available at https://github.com/WPENGxs/X-WebAgentBench.

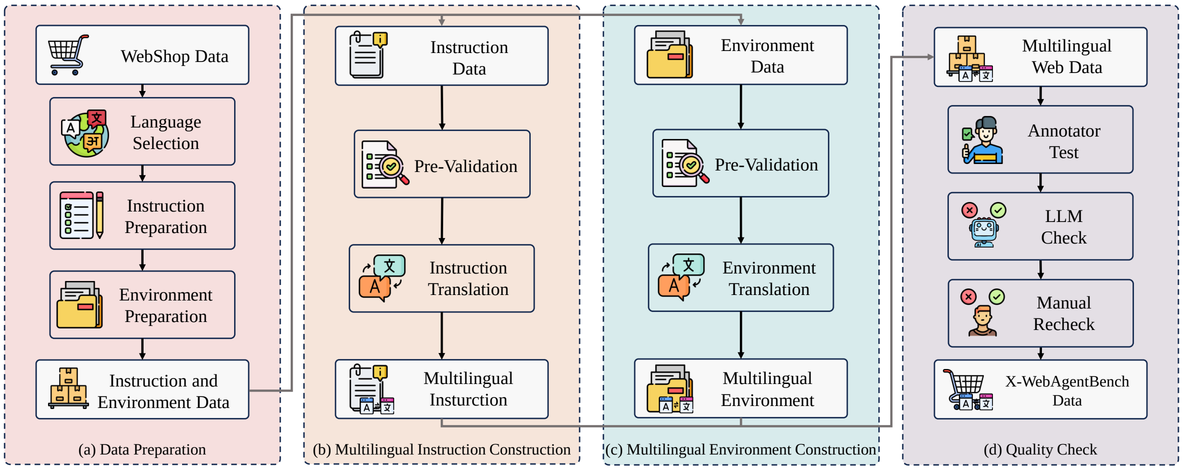


Figure 2: The construction of `X-WebAgentBench` includes four stages: (a) Data Preparation, (b) Multilingual Instruction Construction, (c) Multilingual Environment Construction, and (d) Quality Check. This workflow figure refers to M^{3}CoT (Chen et al., 2024b).

2 `X-WebAgentBench` Construction

This section describes the construction of `X-WebAgentBench`, including Data Preparation (§ 2.1), Multilingual Instruction Construction (§ 2.2), Multilingual Envrionment Construction (§ 2.3), and Quality Check (§ 2.4). Further, we will provide the detailed Data Statistics (§ 2.5).

2.1 Data Preparation

This section introduces the processing of data preparation, which is illustrated in Figure 2 (a).

Language Selection. Following Conneau et al. (2018), we select 14 representative languages to balance resource distribution and maximize coverage across 7 language families. Moreover, this selection also optimally expands the geographical distribution of spoken languages, thereby increasing its global applicability. The detailed language distribution is shown in Table 5 in Appendix A.

Instruction Preparation. Following the mainstream benchmark setup for language agents (Yao et al., 2023), we randomly select 500 informative instructions from the WebShop dataset to facilitate product identification. During selection, we manually remove ambiguous instructions, such as "I need a charger," ensuring that each retained instruction corresponds to a well-defined product search.

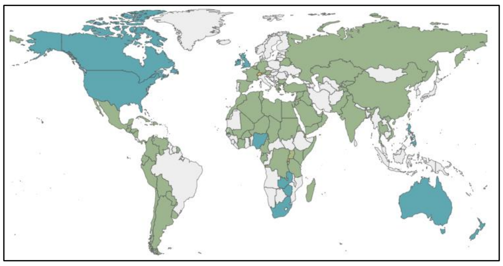
Environment Preparation. To evaluate the interaction capabilities of LLMs effectively, it is crucial to address the challenges presented by lengthy contexts, particularly in extensive environmental product data. To reduce the impact of long contexts for fair assessment, we simplify the multilingual e-commerce environment from WebShop (Yao et al., 2022). Specifically, each instruction involves an average of 211 products, maintaining a sufficient decision space with appropriate contextual length.

2.2 Multilingual Instruction Construction

This section introduces the construction process of multilingual instruction, which is illustrated in Figure 2 (b).

Pre-validation. Due to the high cost of manual translation, we explore the use of automated tools for large-scale instruction translation. To assess translation quality, we translated 50 English instructions into 14 languages using automated methods and manually validated the results. Specifically, we compared two translation tools: prompted GPT-4 and Google Translate. Three experts evaluated whether the translations accurately conveyed the original meaning. The assessment showed that Google Translate consistently achieved accuracy rates above 90% across all languages. In contrast, GPT-4 performed well for high-resource languages but exhibited significant variability in low-resource languages, with an average accuracy of only 74%.

Instruction Translation. Based on the results in pre-validation, we utilize Google Translate API to autonomously translate these instruction data to the selected 14 languages, which also follows previous multilingual benchmark construction strategies (Conneau et al., 2018; Zhang et al., 2023). As a result, we obtain 2,800 multilingual instructions across 14 languages.



(a) Distribution of 15 languages on the world map

Product Category			
Beauty & Personal Care	9012	Cell Phones & Accessories	885
Home & Kitchen	7990	Sports & Outdoors	287
Clothing, Shoes & Jewelry	7920	Health & Household	280
Electronics	6974	Baby Products	166
Grocery & Gourmet Food	5989	Musical Instruments	9
Tools & Home Improvement	1583	Toys & Games	9
Unknown	1035		

(b) Product category distribution in X-WebAgentBench

Figure 3: The distribution of languages and product category in `X-WebAgentBench`, cyan represents English area, and green represents multilingual area in `X-WebAgentBench`.

2.3 Multilingual Environment Construction

This section, as shown in Figure 2 (c), presents the construction process of multilingual environment.

Pre-validation. Similarly, to ensure data quality in a multilingual environment, we also introduce a pre-validation stage to achieve translation accuracy. We first select 50 products in environment per language to compare GPT-4 and Google Translate. Specifically, we evaluate translations of titles, categories, clickable buttons, product names, full and short descriptions, and customization options. Each field is assessed against corresponding product data for accuracy. Notably, GPT-4 significantly outperforms other tools, even achieving over 15% higher human annotation accuracy by capturing contextual nuances and producing more precise translations. This step ensures high-quality multilingual data to support the agent's interaction.

Environment Translation. Leveraging pre-validation insights, we standardize all environmental data into a unified JSON format and translate it into 14 languages. In total, we generate 589,946 products to support a robust multilingual environment. Finally, we integrate this product data into an interaction system to construct the multilingual environment in `X-WebAgentBench`.

2.4 Quality Check

Multilingual datasets, created from translations of original English data, always introduce semantic confusion, ambiguity, and cultural differences (Qin et al., 2025; Zhang et al., 2023). For example, in Chinese, both "trousers" and "shorts" are often represented by the same term, illustrating polysemy issues. To address this, we perform quality checks, as shown in Figure 2 (d), to ensure data accuracy.

Onboarding Test. We select 50 test instructions and relevant prodcuts from each of the 14 languages for annotators. To participate in the subsequent recheck task, annotators must achieve an average task score of at least 80 across these items.

LLM Check. To ensure semantic accuracy, we adopt the method of Chen et al. (2024c), comparing translations in multiple languages to the original English using GPT-4. Translations are rated on a 0-to-10 scale, with only those scoring 8 or above being included. This results in approximately 250 instructions and associated products. Notably, most instructions across languages achieve a perfect score of 10, indicating high-quality translations.

Manual Recheck. Subsequently, the final version undergoes manual rechecking to verify overall quality, yielding 200 instructions and corresponding products of high quality for each language. A kappa coefficient of 0.8 confirms the reliability of this process (Landis and Koch, 1977). Details of the recheck process are provided in Appendix B.

2.5 Data Statistics

`X-WebAgentBench` supports 14 languages, includes 2,800 multilingual text instructions (200 per language), and features 589,946 multilingual products across 13 categories, addressing the linguistic needs of most countries (see Figure 3). As shown in Table 6 in Appendix D, the multilingual environment in `X-WebAgentBench` encompass 6 distinct actions, demonstrating a broad decision space. This variety in multilingual scenarios and agent search spaces enables a comprehensive evaluation of the multilingual and interactive capabilities of LLMs.

3 Experiments

3.1 Experimental Setting

We evaluate 5 LLMs in `X-WebAgentBench`, including GPT-4o (Achiam et al., 2023),

Method	zh	fr	es	de	el	bg	ru	tr	ar	vi	th	hi	sw	ur	AVG
Mistral-7B-Instruct (Jiang et al., 2023)															
BaseAgent	**10.80**	3.71	3.36	4.93	2.94	1.95	4.05	**3.73**	**3.02**	3.06	1.88	4.01	1.31	0.33	3.51
BaseAgent + Translate-en	6.08	**7.13**	4.28	**5.34**	**4.25**	**6.65**	**5.83**	3.55	1.73	**3.95**	**2.48**	**6.34**	**3.35**	**1.35**	**4.45**
BaseAgent + Self-Translate-en	1.76	3.57	**4.32**	2.53	2.12	3.48	2.06	0.20	0.29	0.98	0.49	0.71	0.00	0.00	1.61
BaseAgent + CLP	2.90	3.26	2.96	3.30	2.31	2.63	1.00	0.25	1.13	2.33	1.34	0.95	0.00	1.21	1.83
Llama3-8B (Touvron et al., 2023)															
BaseAgent	6.35	**10.66**	3.97	4.88	3.53	7.63	5.81	5.65	3.51	3.80	0.37	2.56	4.47	2.67	4.70
BaseAgent + Translate-en	**11.71**	9.13	**5.44**	**12.67**	**5.21**	**9.77**	**10.65**	**8.12**	**9.02**	**10.61**	**1.66**	**9.52**	**11.58**	**4.30**	**8.53**
BaseAgent + Self-Translate-en	5.04	3.30	1.79	1.79	2.21	1.75	1.99	1.97	1.34	0.83	0.45	1.47	1.21	0.50	1.83
BaseAgent + CLP	0.00	0.00	0.00	0.38	0.00	0.00	0.83	0.00	0.25	0.00	0.00	0.25	0.00	0.00	0.12
Qwen2-7B-Instruct (Yang et al., 2024)															
BaseAgent	**27.64**	**29.47**	**25.02**	10.09	5.59	11.88	**25.55**	14.68	9.49	14.80	6.53	10.38	10.94	**9.07**	15.08
BaseAgent + Translate-en	21.37	28.04	21.30	**23.39**	**22.60**	**23.63**	20.35	**17.47**	**13.90**	**22.62**	6.35	**19.96**	**13.86**	7.62	**18.75**
BaseAgent + Self-Translate-en	19.31	19.32	17.43	10.98	5.93	12.51	16.33	10.30	8.62	16.09	**8.01**	9.63	3.95	6.70	11.79
BaseAgent + CLP	15.54	8.58	2.44	1.03	7.04	9.62	10.86	4.19	6.45	6.51	0.60	1.32	1.27	8.16	5.97
GPT-3.5-turbo (Brown et al., 2020)															
BaseAgent	36.04	35.40	29.61	26.06	27.12	30.17	**31.76**	**29.95**	28.68	23.35	11.48	12.32	33.73	15.62	26.52
BaseAgent + Translate-en	27.87	24.12	13.77	28.90	29.69	31.65	30.26	16.28	14.70	**29.89**	13.40	**27.72**	23.70	5.60	22.68
BaseAgent + Self-Translate-en	22.96	27.87	23.60	22.53	**34.24**	23.74	22.84	26.22	12.02	28.62	12.42	24.93	26.40	8.60	22.64
BaseAgent + CLP	**40.33**	**39.04**	**42.81**	**30.95**	20.98	**34.57**	22.38	27.23	**33.41**	29.53	**16.24**	21.90	**34.16**	**23.58**	**29.79**
GPT-4o (Achiam et al., 2023)															
BaseAgent	**41.65**	42.70	37.31	34.56	32.80	40.10	36.41	**43.18**	41.34	37.93	18.51	36.75	**42.10**	34.15	37.11
BaseAgent + Translate-en	34.80	48.33	25.97	**37.41**	28.92	36.74	**38.00**	33.99	2.18	35.19	12.69	33.47	36.49	2.30	29.03
BaseAgent + Self-Translate-en	20.98	33.39	27.45	18.58	26.63	29.20	29.15	12.75	8.40	26.72	18.33	10.45	26.41	20.42	22.06
BaseAgent + CLP	25.75	**48.39**	**42.70**	36.83	**42.24**	**41.79**	37.04	41.87	**41.40**	**38.44**	**21.79**	**37.68**	40.01	**35.18**	**37.94**

Table 2: Main results for `X-WebAgentBench`. **Bold number** indicates the best performance for that language in the current model and methods. "AVG" presents the average task score in 14 languages.

GPT-3.5-turbo (Brown et al., 2020), Qwen2-7B-Instruct (Yang et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), and Llama3-8B (Touvron et al., 2023). Following WebShop (Yao et al., 2022), we use the Task Score to evaluate the performance of language agents.

3.2 Baselines

For each LLM we utilize 4 baseline strategies to test the performance in `X-WebAgentBench`, including: (1) *BaseAgent* (Liu et al., 2024a) interacts with `X-WebAgentBench` in local language based on Liu et al. (2024a). (2) *Translate-en* (Shi et al., 2023) translates `X-WebAgentBench` to English with Google Translate. (3) *Self-Translate-en* (Etxaniz et al., 2024) relies on LLM translating `X-WebAgentBench` to English. (4) *CLP* (Qin et al., 2023) uses cross-lingual alignment prompting to enable *BaseAgent* to understand `X-WebAgentBench`, followed by task-specific solver prompting for final solution.

3.3 Results for `X-WebAgentBench`

From the results of `X-WebAgentBench` shown in the Table 2, we have the following observations: ***(1) Cross-lingual self-alignment is only effective for LLMs with advanced multilingual capabilities.*** While original cross-lingual self-alignment methods are applicable to smaller LLMs (Etxaniz et al., 2024; Qin et al., 2023), our findings show that only LLMs with advanced multilingual capabilities (Qin et al., 2025), such as GPT-series, enable self-alignment techniques like *CLP* to outperform other baselines in multilingual settings. In contrast, weak open-source LLMs consistently fail. We attribute it to the limited multilingual capacity in adapting to complex interactive environments of weak LLMs, which prevents cross-lingual self-alignment from functioning properly, often leading to significant performance degradation.

(2) Smaller LLMs ($\leq$ 8B) need external Translation tools for effective cross-lingual alignment. As shown in Table 2, open-source LLMs with fewer than 8B parameters underperform in multilingual settings using *BaseAgent* and other self-alignment strategies. Only methods that translate languages into English using external translation tools, significantly enhance cross-lingual alignment, leading to task score improvements in `X-WebAgentBench`.

(3) Multilingual interactive scenarios remain a significant challenge for current agentic systems. Even GPT-4o, when integrated with advanced *CLP* strategies, fails to produce satisfactory results. This indicates that the persistent performance gap between multilingual and English environments in language agentic systems. The decrease in per-

Model	Action Type	zh	fr	es	de	el	bg	ru	tr	ar	vi	th	hi	sw	ur	AVG
Qwen2-7B-Instruct (Yang et al., 2024)	All	5.70	6.09	5.73	11.43	5.96	5.50	5.73	8.51	5.93	7.00	8.05	6.71	7.28	7.30	6.92
	Search	1.61	1.68	1.18	1.52	1.20	1.20	1.27	1.34	1.40	1.43	2.87	1.62	1.60	1.33	1.52
	Click	4.08	4.41	4.55	9.90	4.76	4.30	4.46	8.17	4.54	5.57	5.19	5.68	5.68	5.97	5.52
GPT-3.5-turbo (Brown et al., 2020)	All	4.70	5.86	5.67	5.25	4.97	5.62	6.15	5.21	5.53	5.64	7.06	6.94	5.81	5.71	5.72
	Search	1.67	1.97	1.57	1.71	1.50	1.64	2.31	1.73	1.49	1.56	3.29	3.09	2.31	1.99	1.99
	Click	3.02	3.89	4.10	3.54	3.46	3.98	3.83	3.48	4.04	4.08	3.77	3.85	3.50	3.72	3.73
GPT-4o (Achiam et al., 2023)	All	5.37	5.59	4.97	5.72	5.38	5.58	5.98	5.00	5.21	5.39	7.90	5.95	5.35	5.29	5.62
	Search	1.32	2.17	1.61	1.65	1.44	1.39	2.39	1.45	1.35	1.42	3.63	2.38	1.79	1.68	1.83
	Click	4.04	3.42	3.36	4.08	3.95	4.20	3.59	3.55	3.85	3.97	4.26	3.56	3.56	3.61	3.79

Table 3: The average action steps of different language in Qwen2-7B-Instruct, GPT-3.5-turbo, and GPT-4o. We fill color scales when all actions are greater than 5, search actions are greater than 1, and click actions are greater than 4. The deeper color means the more actions compared to other languages.

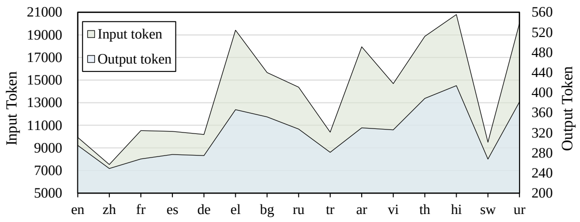


Figure 4: Statistics of average input token and output token for *BaseAgent* method by GPT-3.5-turbo.

formance due to the language gap renders most current agent methods nearly unusable.

4 Analysis

In this section, we analyze the findings from both the multilingual(§ 4.1 and § 4.2) and agentic perspectives(§ 4.3 and Appendix § H) as follows:

4.1 Agentic system illustrates distinct multilingual token cost unfairness

Inspired by the inherent token-cost unfairness in multilingual LLMs (Petrov et al., 2023), we investigate it by measuring the token consumption for both input and output in `X-WebAgentBench`. Interestingly, we identify that, unlike traditional cases, where lower-resource languages face higher token costs, unfairness in agentic systems primarily stems from linguistic characters, not training resources. As illustrated in Figure 4, most non-Latin script languages, especially on Greek (el) and Hindi (hi), consume as least twice as many tokens as english. Conversely, languages like Chinese (zh) and German (de), which share Subject-Verb-Object (SVO) grammar structures, consume fewer tokens and mitigate concerns about token overuse. These findings also emphasize the need and hopeful linguistic-related direction to address token-cost unfairness in multilingual agentic systems.

4.2 Agentic system tends to overuse actions in languages with less speakers

To assess the impact of multilingualism on action steps, we analyze the *BaseAgent* methods of the top three performing models in Table 3. The results reveal notable variations across languages, with less widely spoken languages, such as German (de), Turkish (tr), Vietnamese (vi), and Thai (th), requiring more steps. For instance, in Qwen-2-7B-Instruct, the number of "click" actions for German and Turkish far exceeds the average of 3-4 clicks for correct interaction. Similarly, in GPT-series, Thai and Hindi exhibit disproportionately high "search" actions, far above the average of 1-2 searches for accurate outcomes. In contrast, Chinese, a widely spoken language[1], demonstrates consistent performance and balanced action distributions across all models. These findings indicate that languages with less speakers are more prone to overuse actions in `X-WebAgentBench`. Therefore, addressing these action imbalances is essential for reducing global inequalities in agentic systems.

4.3 Agentic system performance poorly on long interaction in multilingual scenarios

As shown in Figure 5 (a), GPT-3.5-turbo struggles to generate correct actions that yield sufficient rewards. To address this limitation, we notice that in pure English scenarios, the planning boundary of the model agent typically spans 5-10 steps, beyond which performance degrades (Chen et al., 2024a; Hu et al., 2024a; Jin et al., 2024). Inspired by this, we analyze the reward distribution across all languages, as seen in Figure 5 (b). Notably, most

[1] https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

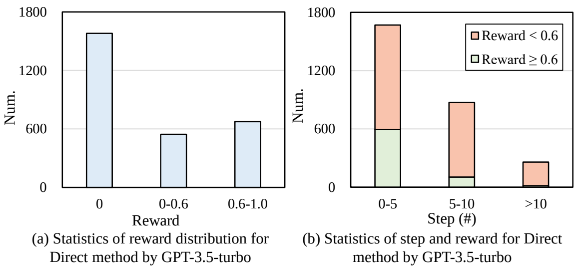


Figure 5: Statistics of action reward for *BaseAgent* method by GPT-3.5-turbo.

Method	de	ru	tr	vi	bg	hi
ReAct + En-ICL	12.81	8.77	15.09	12.06	8.78	2.15
ReAct + De-ICL	19.65	8.21	10.61	9.07	13.50	2.91
ReAct + L-ICL	19.65 ↑	18.58 ↑	1.60 ↓	9.34 ↓	14.65 ↑	5.79 ↑

Table 4: Performance comparison of English, German, and local language ICL demonstration (En-ICL, De-ICL, and L-ICL) on GPT-3.5-turbo."↑" represents the performance of L-ICL is better than En-ICL, and "↓" indicates the opposite.

action rewards are concentrated within 0-5 steps, with a sharp decline thereafter[2]. This indicates that in multilingual scenarios, performance degradation occurs earlier than in English-based tasks. It highlights that current language agents are limited to handling fewer steps and struggle with longer interactions in multilingual scenarios.

5 Exploration

Following Qin et al. (2023), we select 6 representative languages based on language statistics in CommonCrawl, including high- (de, ru), mid- (tr, vi), and low-resource (bg, hi) languages for exploration. Our study examines the effects of multilingual factors (§ 5.1, § 5.2, § 5.3) and agentic factors (§ 5.4, § 5.5, and Appendix F).

5.1 Cross-lingual demonstrations can not introduce effective multilingual alignment

In-context Learning (ICL) with cross-lingual demonstrations has been shown to enhance performance, particularly for multilingual alignment (Shi et al., 2023). Motivated by this, we evaluate ReAct (Yao et al., 2023) using demonstrations in English, German, and a local language (En-, De-, and L-ICL). In this framework, we adapt manually crafted ICL demonstrations to guide the model's next-step actions, with "local language demonstration" referring to cases where the ICL language matches the environment's language. As shown in Table 4, we observe an unexpected result compared to traditional multilingual tasks (Shi et al., 2023; Qin et al., 2025). Specifically, performance of those with local language demonstrations exceeded that of cross-lingual demonstrations (En- and De-ICL) in most languages. This suggests that, unlike local language, current LLMs struggle to achieve robust multilingual alignment in ICL-based agentic scenarios. This also highlights the need for future research to enhance cross-lingual transferability and better leverage multilingual capabilities in LLMs.

[2]Human experts typically require more than 10 action steps in `X-WebAgentBench`.

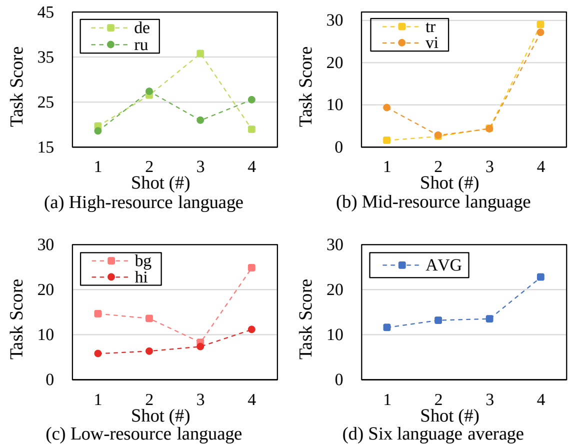


Figure 6: Performance of different shots in multilingual few-shot ReAct prompting.

5.2 The number of demonstrations has a positive effect on performance mainly in mid- and low-resource languages

Further, inspired by Shi et al. (2023); Qin et al. (2024a), we analyze how the number of demonstrations affects the performance of language agents within `X-WebAgentBench`. The results yield the following insights: ***1. Limited benefits for high-resource languages:*** As shown in Figure 6 (a), task score initially improves with additional demonstrations but declines beyond a certain threshold. ***2. Substantial gains for mid- and low-resource languages:*** Figures 6 (b-d) shows that the task score improves steadily as the number of demonstrations increases. It indicates that demonstration effectiveness depends on language resources. Mid- and low-resource languages require more demonstrations to boost their performance.

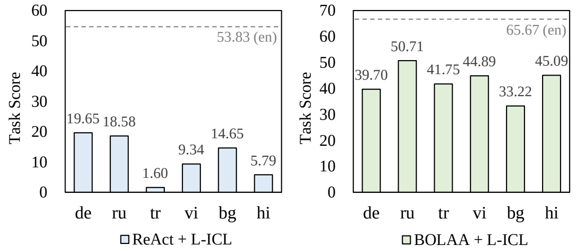


Figure 7: The performance of ReAct and BOLAA with L-ICL. Task score in English is from Liu et al. (2024b).

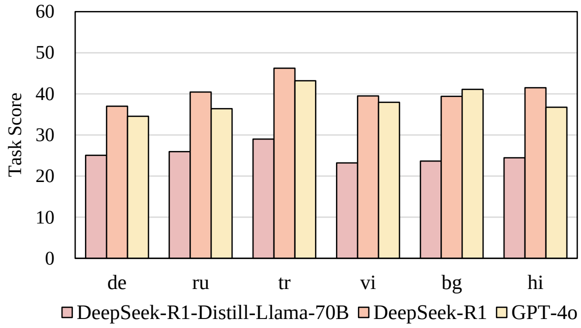


Figure 8: The performance comparison of reasoning models and GPT-4o.

5.3 Multilingual gaps are widespread in current English-centric agentic systems

To comprehensively evaluate the multilingual limitation of current English-centric strong agentic frameworks in `X-WebAgentBench`, we evaluated the multi-agent BOLAA (Liu et al., 2024b) in local language demonstration (BOLAA+L-ICL) using GPT-3.5-turbo and compared it with ReAct. The performance is shown in Figure 7. As observed, BOLAA+L-ICL performs well in `X-WebAgentBench`, demonstrating the performance advantage of the multi-agent approach over the single-agent approach. But it still has a gap with English. This suggests that multilingual gaps are widespread in agentic systems. How to solve the understanding gap between different languages remains an important direction that needs to be explored in the future.

5.4 LLMs overestimate their multilingual abilities rather than using translation tools

Due to agents' high accuracy in English scenarios, an intuitive question arises: ***Will agents autonomously use translation tools to overcome limitations in multilingual comprehension?*** To explore this, we integrated a translation tool into the agent's action list, allowing it to translate web pages into English. Unfortunately, across all 14 languages, the LLMs never utilize the translation function, demonstrating overconfidence in its multilingual capabilities. This underscores the need for further research to enable agentic systems to intelligently utilize translation tools.

5.5 The bottleneck of `X-WebAgentBench` lies in language alignment, not complex logic

To examine whether the bottleneck of `X-WebAgentBench` stems from complex reasoning, we evaluate DeepSeek-R1-Distill-Llama-70B and DeepSeek-R1 (Guo et al., 2025) using *BaseAgent*. As shown in Figure 8, the performance of reasoning models is lower than expected. Notably, DeepSeek-R1 performs similarly to GPT-4o, and the task score for all models are below 50. This suggests that enhancing reasoning alone does not effectively support interaction in a multilingual interactive environment. Therefore, the main bottleneck of `X-WebAgentBench` still lies in language alignment, rather than reasoning logic. Further research into multilingual alignment techniques is essential for LLMs' interaction capability improvements in multilingual environments.

6 Related Work

To evaluate the interaction capabilities of LLMs, researchers focus on benchmarks in interactive environments (Toyama et al., 2021; Sun et al., 2022; Xie et al., 2024). Earlier, MiniWoB++ (Liu et al., 2018) proposes a suite of web-browser tasks with over 100 web interaction environments. WebSRC (Chen et al., 2021b) further integrates web pages and instructions for structural reading comprehension evaluation. AndroidEnv (Toyama et al., 2021) define custom tasks on the Android Operating System, enabling realistic simulation of an Android device. After that, WebShop (Yao et al., 2022) emphasizes interactive tasks based on given instructions in e-commerce environments. Furthermore, Mind2Web (Deng et al., 2024) extends agentic evaluation to 2,000 tasks across 137 websites. WebArena (Zhou et al., 2024) constructs a highly realistic and reproducible environment for language agents, which aims to boost the development of robust agents.

However, current interactive environments are primarily in English, which limits the development of multilingual agents. We are the first attempt to construct a multilingual interactive benchmark, X-WebAgentBench, which quantifies the interaction performance of language agents.

7 Conclusion

To address the multilingual evaluation of language agents, we introduce X-WebAgentBench, a benchmark comprising 14 languages, 2,800 text instructions, and 589,946 products. Through extensive experimentation on X-WebAgentBench, we derive several novel insights into multilingual language agentic systens. We hope that X-WebAgentBench can serve as an effective benchmark, providing valuable reference for future research.

Limitations

We introduce a multilingual interactive web benchmark named X-WebAgentBench, which explores the performance of understanding and interaction in the multilingual scenarios for language agents. Using Google Translate and multiple quality checks can only guarantee the accuracy and reliability of multilingual expressions as much as possible. However, the quality cannot be completely equivalent to that of experienced translators. In the future, we will explore more advanced data construction methods to achieve a more realistic multilingual environment.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62306342. This work was sponsored by the Excellent Young Scientists Fund in HunanProvince (2024JJ4070), the Science and Technology Innovation Program of Hunan Province under Grant 2024RC3024, and CCF-Zhipu Large Model Innovation Fund (NO.CCF-Zhipu202406). This work was also supported by the Key Laboratory of Data Intelligence and Advanced Computing in Provincial Universities, Soochow University. We are grateful for resources from the High Performance Computing Center of Central South University.

References

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Qiguang Chen, Libo Qin, Jiaqi WANG, Jingxuan Zhou, and Wanxiang Che. 2024a. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024b. M^{3}CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, Bangkok, Thailand. Association for Computational Linguistics.

Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021a. WebSRC: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xingyu Chen, Zihan Zhao, Lu Chen, Jiabao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021b. Websrc: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185.

Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Wanxiang Che, Hang Yan, Kai Chen, and Dahua Lin. 2024c. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. *arXiv preprint arXiv:2409.01893*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web.

Advances in Neural Information Processing Systems, 36.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. Do multilingual language models think better in English? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564, Mexico City, Mexico. Association for Computational Linguistics.

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

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. 2024a. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. *arXiv preprint arXiv:2408.09559*.

Mengkang Hu, Yao Mu, Xinmiao Chelsey Yu, Mingyu Ding, Shiguang Wu, Wenqi Shao, Qiguang Chen, Bin Wang, Yu Qiao, and Ping Luo. 2024b. Tree-planner: Efficient close-loop task planning with large language models. In *The Twelfth International Conference on Learning Representations*.

Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jianguang Lou, Qingwei Lin, Ping Luo, Saravan Rajmohan, and Dongmei Zhang. 2024c. Agentgen: Enhancing planning abilities for large language model based agent via environment and task generation. *CoRR*, abs/2408.00764.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842, Bangkok, Thailand. Association for Computational Linguistics.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. 2025. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision*, pages 161–178. Springer.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2024a. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh R N, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, Ran Xu, Phil L Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2024b. BOLAA: BENCHMARKING AND ORCHESTRATING LLM AUTONOMOUS AGENTS. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2024. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, et al. 2024. Webcanvas: Benchmarking web agents in online environments. *arXiv preprint arXiv:2406.12373*.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990. Curran Associates, Inc.

Libo Qin, Qiguang Chen, Hao Fei, Zhi Chen, Min Li, and Wanxiang Che. 2024a. What factors affect multimodal in-context learning? an in-depth exploration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024b. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1).

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. Meta-gui: Towards multi-modal conversational agents on mobile gui. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6699–6712.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. 2021. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*.

Peng Wang, Wenpeng Lu, Chunlin Lu, Ruoxi Zhou, Min Li, and Libo Qin. 2025. Large language model for medical images: A survey of taxonomy, systematic review, and future trends. *Big Data Mining and Analytics*, 8(2):496–517.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Nancy Xu, Sam Masling, Michael Du, Giovanni Campagna, Larry Heck, James Landay, and Monica Lam. 2021. Grounding open-domain instructions to automate web support tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1022–1032, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pages 20744–20757. Curran Associates, Inc.

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

Wenxuan Zhang, Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Advances in Neural Information Processing Systems*, 36:5484–5505.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*.

Appendix

A Data Preparation Details

Our instructions come from the test items in ReAct. The test instructions of ReAct are randomly selected instructions from the original WebShop. Then we find the corresponding 500 products according to the instructions. Based on the current products, we extract products with the same category from all products in WebShop as the product set of `X-WebAgentBench`.

Language selection in `X-WebAgentBench` is detailed in Table 5: We select 14 languages covering a range of language families such as the Indo-European, Sino-Tibetan, and Afro-Asiatic language families, etc., including Chinese (zh), French (fr), Spanish (es), German (de), Greek (el), Bulgarian (bg), Russian (ru), Turkish (tr), Arabic (ar), Vietnamese (vi), Thai (th), Hindi (hi), Swahili (sw), and Urdu (ur).

Language Family	Language	Abbreviation
Indo-European	French	fr
	Spanish	es
	German	de
	Greek	el
	Bulgarian	bg
	Russian	ru
	Hindi	hi
	Urdu	ur
Sino-Tibetan	Chinese	zh
Turkic	Turkish	tr
Afro-Asiatic	Arabic	ar
Austroasiatic	Vietnamese	vi
Tai-Kaidai	Thai	th
Niger-Congo	Swahili	sw

Table 5: Language families and abbreviations mapping for the 14 `X-WebAgentBench` languages

B Manual Check and Re-Check Details

To ensure the translation quality, for every language, we extract 50 samples from the product data and instruction data each. The manual check involves the accuracy of the translation and appropriate expression with the help of well-trained translators.

As for the re-check process, we extract 20 samples for each language from the 200 instructions scored by GPT-4-turbo as mentioned below. All samples are reviewed by at least two well-trained individuals, and a kappa coefficient of 0.8 indicates the reliability of the process (Landis and Koch, 1977).

For each check, the guideline instructions are as follows:

Task: Ensure that the entries translated from English into various languages are accurate and complete.

[Instruction1]: Open the translated sample in the target language.
[Instruction2]: Identify and extract the content of the“instruction" field and the “ASIN" (Amazon Standard Identification Number) from the translated sample.
[Instruction3]: Locate the corresponding entry in the English dataset using the extracted “ASIN."
[Instruction4]: Compare the “instruction" field from the translated sample with the “instruction" field in the English dataset.
[Instruction5]: Evaluate the translation for accuracy and completeness:

- If the translation is accurate and all essential elements are present, the entry is considered correct.
- If the translation is inaccurate or incomplete, record the “ASIN" and the translation language for further review.

C LLM Check Details

We use GPT-4-turbo to score the translation quality. Specifically, we use the following prompt for scoring:

Original instruction: ...
Translated instruction: ...
Please rate the quality of the translation. Your answer should be a single number instead of

Type	Action	Function
search	search[keyword]	Search item by keyword
click	click[prev/next page]	Jump to the prev/next item page
click	click[back btu]	Back to search page
click	click[item id]	Jump to the item page
click	click[attribute]	View the attribute of the item
click	click[buy]	Buy this item

Table 6: Action list in the `X-WebAgentBench`

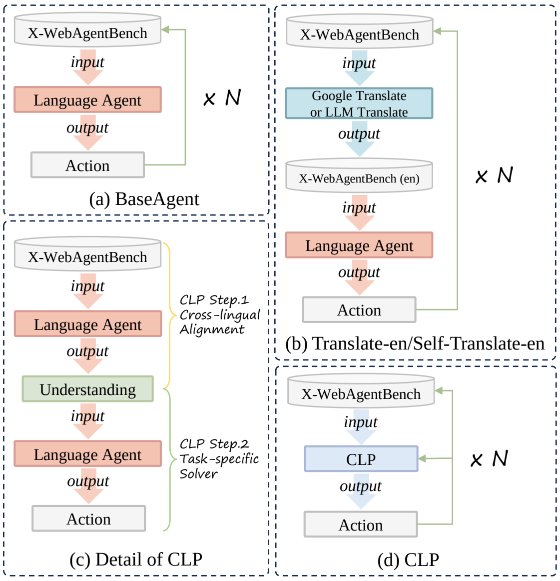


Figure 9: The diagram of these baseline methods. Figure (c) shows the multilingual alignment detail of *CLP*. Language agent in *BaseAgent*, *Translate-en*, and *Self-Translate-en* can read full history when they make a decision. Language agent in *CLP* could not read history, but we input the previous action to the *CLP* next decision as reference.

> any words.
> The highest score is 10 and the lowest is 0.

After scoring, we build up the instruction set using those whose scores are higher than 8.0, with the scale of 200. In fact, except for Swahili and Vietnamese, every instruction in each language has reached the full score. Instructions in majority in Swahili and Vietnamese also reach the full score. Only fewer than 15% get 9 or 8 points.

D Details in `X-WebAgentBench`

The actions within `X-WebAgentBench` are detailed in Table 6. We use *BaseAgent*, *Translate-en*, and *Self-Translate-en*, and *CLP* as baselines to evaluate different LLMs. The detail of these methods is shown in Figure 9.

E Prompt Settings and Format

We use the following prompt format for `X-WebAgentBench` based on AgentBench, and *click* and *search* is multilingual prompt:

> **[Instruction]**
> You are web shopping. I will give you instructions about what to do. You have to follow the instructions.
>
> 1.Every round I will give you an observation and a list of available actions, you have to respond an action based on the state and instruction.
>
> 2.You can use search action if *search* is available. You can click one of the buttons in clickable. An action should be of the following structure:
>
> - *search*[keyword]: Search keywords on the search page. If you are not on the search page, please return to the search page. Such as "*search*[*item*]".
> - *click*[*prev*]/*click*[*next*]: Jump to the previous/next page. Such as "*click*[*prev*]".
> - *click*[*back*]: Back to search page. Such as "*click*[*back*]".
> - *click*[item id]: Click on an item to go to its details page. Such as "*click*[ABCDE12345]".
> - *click*[attribute]: Select item attribute on the details page. Such as "*click*[*attr*]".
> - *click*[*buy*]: Buy this item. Such as "*click*[*buy*]".
>
> 3.If the action is not valid, perform nothing.
>
> 4.Keywords in search are up to you, but the value in click must be a value in the list of

available actions. Remember that your keywords in search should be carefully designed.

[Response Format]
Thought: $< I\ think... >$
Action: <click[$something$] >

For some models that cannot follow the instructions, we use json to specify the output format:

Your response should use the following json format:
{"Thought":"I think ...", "Action": "$click$[something]"}

For CLP, we use these prompt to align language firstly:

You are an expert in multi-lingual understanding. Please fully understand given observation and give the correct Action.
Please act as an expert in multi-lingual understanding in $language$.
You need fully understand every words, and explain its effection.
Don't response the Action, it is next step task.
Let's understand the observation in English step-by-step!

Then, we use these prompts to get last action:

After understanding, you should act as an expert in reasoning in English. In the end, you should response Action in $language$ according previously understood observation.

Previous Action: $< prev\ action >$

You are web shopping. I will give you instructions about what to do. You have to follow the follow instructions.

1.Every round I will give you an observation and a list of available actions, you have to respond an action based on the state and instruction.

2.You can use search action if search is available. You can click one of the buttons in clickable. An action should be of the following structure:

- $search$[keyword]: Search keywords on the search page. If you are not on the search page, please return to the search page. Such as "$search[item]$".
- $click[prev]$/$click[next]$: Jump to the previous/next page. Such as "$click[prev]$".
- $click[back]$: Back to search page. Such as "$click[back]$".
- $click$[item id]: Click on an item to go to its details page. Such as "$click$[ABCDE12345]".
- $click$[attribute]: Select item attribute on the details page. Such as "$click[attr]$".
- $click[buy]$: Buy this item. Such as "$click[buy]$".

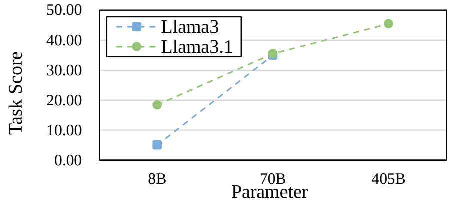
3.If the action is not valid, perform nothing.

4.Keywords in search are up to you, but the value in click must be a value in the list of available actions. Remember that your keywords in search should be carefully designed.

[Response Format]
Thought: $< I\ think... >$
Action: <click[$something$] >

F Multilingual performance follows LLM parameter increase

To assess the influence of LLM parameter scale on performance, we evaluate six languages representing high-, mid-, and low-resource groups. As



(a) Task score of different parameter LLM in X-WebAgent

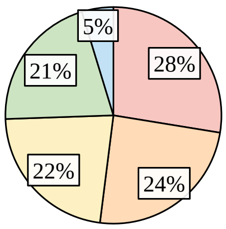
Model	de	ru	tr	vi	bg	hi	AVG
Llama3-8B	4.88	5.81	5.65	3.80	7.63	2.56	5.06
Llama3-70B	30.02	39.32	41.21	34.19	36.66	28.39	34.97
Llama3.1-8B	17.60	21.58	20.09	19.05	14.78	17.32	18.40
Llama3.1-70B	37.02	36.19	38.80	35.27	34.95	30.77	35.50
Llama3.1-405B	45.76	42.27	45.60	48.62	46.33	43.99	45.43

(b) Task score of Llama3 and Llama3.1 for 6 languages

Figure 10: Performance comparison of different-scale LLMs on `X-WebAgentBench`. Subfigure (a) presents the average performance of Llama3 and Llama3.1, while subfigure (b) provides detailed performance across 6 languages.

Model	zh	fr	es	de	el	bg	ru	tr	ar	vi	th	hi	sw	ur	AVG
DeepSeek-R1-Distill-Llama-70B	18.22	30.76	21.84	25.08	22.84	23.66	25.95	29.00	21.58	23.22	17.44	24.48	21.55	18.56	23.16
DeepSeek-R1	34.84	45.28	43.20	37.02	38.61	39.40	40.44	46.25	25.51	39.52	20.47	41.50	26.64	32.90	36.54

Table 7: The performance of DeepSeek-R1-Distill-Llama-70B and DeepSeek-R1 in `X-WebAgentBench`.



(a) The proportion of interaction failures

Error	Num.
No output action	449
Click error	398
Action error	366
Over Search	339
Keyword Loss	77

(b) Number data of error in interaction failures

Figure 11: Statistics of action step (a), reward (b), and interaction failures (c and d). The output of the *BaseAgent* method is from GPT-3.5-turbo.

shown in Figure 10 (a, b), increasing LLM scale consistently improves both task-specific and average scores across all languages, though performance remains limited. These findings highlight a positive correlation between LLM scale and multilingual performance. However, despite enhancing multilingual capabilities, larger LLMs still struggle to address current multilingual challenges effectively.

G Performance of Reasoning Models

We employ the Direct method to evaluate two reasoning large language models, DeepSeek-R1-Distill-Llama-70B and DeepSeek-R1. The final results are presented in Table 7.

H Error analysis of existing agentic system

To understand the factors resulting in language agent errors during interactions, we studied all the error cases in the *BaseAgent* method outputs for 14 languages from the GPT-3.5-turbo logs. The final results are illustrated in Figure 11.

The main errors leading to interaction failures include the following five types: (1) *No output action:* The output does not contain an action, preventing the next step from being executed; (2) *Click error:* The button to be clicked is not on the page, rendering the instruction impossible; (3) *Action error:* Use nonexistent commands on the web page, such as requesting a search action on a product page; (4) *Over search:* Search times exceed the maximum limit; (5) *Keywords loss:* Keywords in the instruction are not extracted.

Therefore, to enhance the decision-making capabilities of language agents on `X-WebAgentBench`, the following abilities matter: (1) Strengthen their understanding abilities in multilingual scenarios, which ultimately output correct actions. (2) Reasonably plan the next instruction, avoiding repetitive and ineffective actions.

I Ethical Considerations

Data Access We obtained our data from the Webshop, which is open-source and freely accessible for academic research, which is in line with our commitment to ethical data usage.

Participant Recruitment We recruit participants from universities, ensuring that all have multilingual abilities. All annotators provided informed consent and were compensated above the local minimum wage. Additionally, the site does not require IRB review.

Dataset Check and Re-check Process Our checking process started with an onboarding test that introduced the task through 50 example questions. Participants were paid \$20 for this initial phase, which was designed to familiarize them with the task. Subsequently, annotators were paid \$15 per hour, averaging approximately 4 human annotation hours per person. In total, 42 experts participated in completing the annotation and recheck tasks.