Multi-SWE-bench: A Multilingual Benchmark for Issue Resolving

https://multi-swe-bench.github.io

Daoguang Zan, Zhirong Huang, Wei Liu, Hanwu Chen, Shulin Xin, Linhao Zhang, Qi Liu, Aoyan Li, Lu Chen, Xiaojian Zhong, Siyao Liu,
Yongsheng Xiao, Liangqiang Chen, Yuyu Zhang, Jing Su, Tianyu Liu, Rui Long, Ming Ding, Liang Xiang

NEURAL INFORMATION PROCESSING SYSTEMS

Can your LLM fix bugs beyond Python?

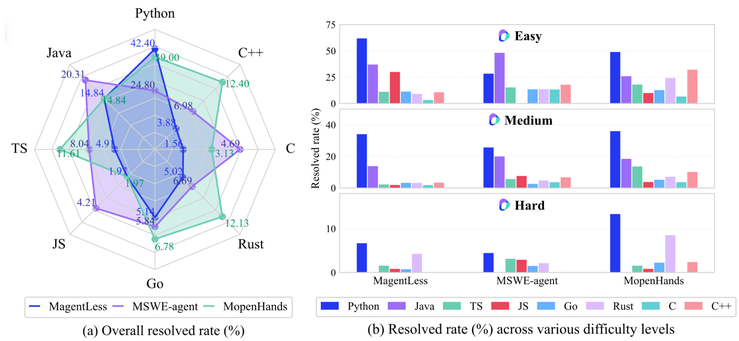
The task of issue resolving aims to modify a codebase to generate a patch that address a given issue. However, most existing benchmarks focus almost exclusively on Python, making them insufficient for evaluating LLMs across different programming languages. To bridge this gap, we introduce a multilingual issue-resolving benchmark, called Multi-SWE-bench, covering 8 widely used programming languages: Python, Java, TypeScript, JavaScript, Go, Rust, C, and C++. It includes a total of 2,132 high-quality instances, carefully curated by 68 expert annotators, ensuring a reliable and accurate evaluation of LLMs on the issue-resolving task. Based on human-annotated results, the issues are further classified into three difficulty levels. We evaluate a series of state-of-the-art models on Multi-SWE-bench, utilizing both procedural and agent-based frameworks for issue resolving.

- 💥 1,632 real-world issues
- ✅ Verified by 68 engineers
- 📦 Dockerized, reproducible, battle-tested
- 🧠 Covers easy, medium, and hard bug fixes
- 📊 Designed to benchmark LLMs as true dev agents

To scale beyond benchmarks, we also launch Multi-SWE-RL —

- 🎮 An open-source RL community to build interactive training environments.
- 🌱 4,723 containerized issue-resolving tasks, 7 languages, and counting.
- 🤝 We invite the community to contribute, expand, and shape the future of software-native RL.

How well do today's top LLMs really perform?



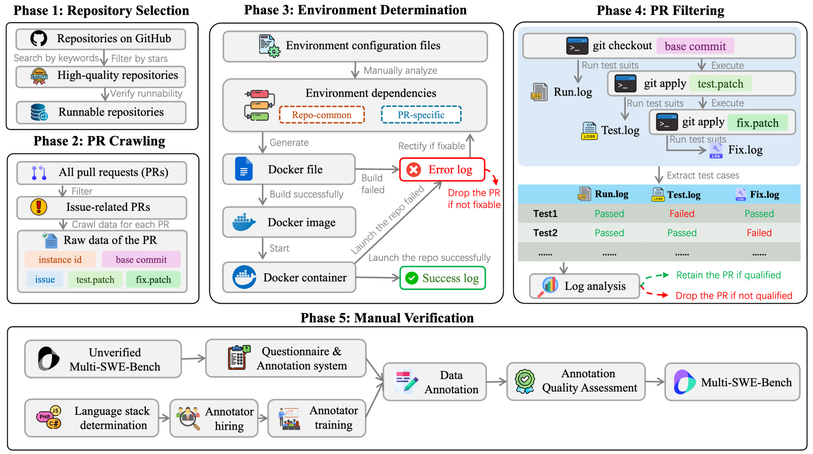
(a) Overall resolved rate (%) (b) Resolved rate (%) across various difficulty levels

We evaluated **9** models (GPT-4o, o1/o3, Claude 3.5/3.7, DeepSeek, Doubao...) across **3** agents: Agentless, SWE-agent, and OpenHands.

🔍 What happened? Python? Solid. Java? Slips. TS/JS/C++? Gets messy. C/Rust/Go? Still a nightmare.

Even the best agent (MopenHands) only gets ~12.13% on Rust.

Multi-SWE-bench Construction



Eight widely used programming languages are selected to construct Multi-SWE-bench through five phases. As shown the above figure, the first four phases create a large pool of candidate data for each language, while the fifth phase finalizes the Multi-SWE-bench through manual verification.

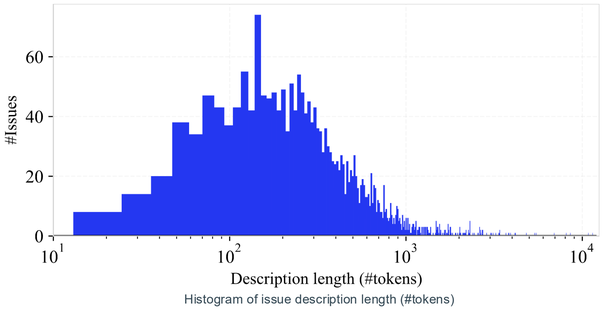
Experimental Results

We compared 9 top LLMs × 3 agents × 8 languages × 3 difficulty levels.

🚨 Even the best models collapse beyond Python.

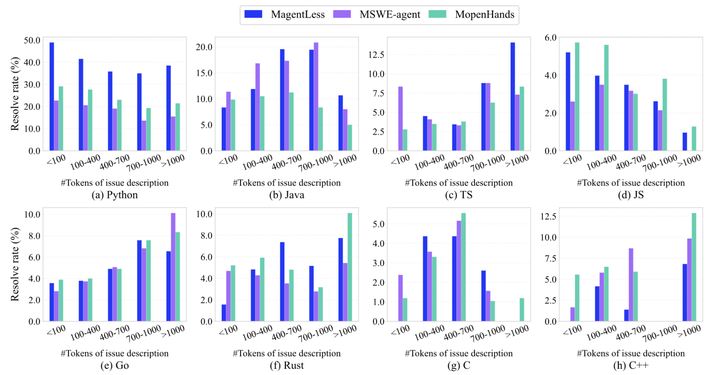
Methods	Models	All	Python	Java	TS	JS	Go	Rust	C	C++
MagentLess	GPT-4o	11.40	36.20	11.72	2.23	1.40	2.80	5.86	1.56	6.98
	OpenAI-o1	16.23	48.20	21.09	5.80	5.06	4.44	7.11	1.56	5.43
	OpenAI-o3-mini-high	13.65	46.40	5.47	0.45	2.81	3.97	7.95	3.91	1.55
	Claude-3.5-Sonnet	13.56	42.40	14.84	4.91	1.97	5.14	5.02	1.56	3.88
	Claude-3.7-Sonnet	14.35	44.60	14.06	3.57	1.97	5.84	5.44	2.34	3.10
	DeepSeek-V3	13.23	41.00	7.03	6.70	3.37	5.37	5.02	3.13	1.55
	DeepSeek-R1	14.40	42.20	22.66	6.25	4.49	3.74	6.69	0.78	3.10
	Qwen2.5-72B-Instruct	8.26	26.80	10.94	4.46	0.84	1.40	2.51	0.78	0.78
	Doubao-1.5-pro	7.83	26.20	5.47	2.23	1.12	2.10	4.18	0.00	0.00
	Doubao-1.5-thinking	15.24	44.80	13.28	7.59	5.62	4.44	7.11	4.69	3.88
	Gemini-2.5-Pro	18.01	49.00	21.88	11.61	8.71	6.07	5.44	9.38	2.33
	Llama-4-Maverick	13.56	37.80	14.84	9.38	5.06	3.50	6.28	3.91	5.43
MSWE-agent	GPT-4o	6.29	18.80	12.50	0.45	0.84	2.34	2.09	1.56	2.33
	OpenAI-o1	11.07	28.80	21.88	4.02	4.21	4.67	4.18	3.91	3.88
	OpenAI-o3-mini-high	10.74	28.60	16.41	4.91	4.21	3.97	5.02	2.34	5.43
	Claude-3.5-Sonnet	11.21	24.80	20.31	8.04	4.21	5.84	6.69	4.69	6.98
	Claude-3.7-Sonnet	17.17	45.80	23.44	11.16	4.78	5.37	6.69	8.59	11.63
	DeepSeek-V3	4.55	4.20	11.72	2.68	2.53	4.44	5.86	2.34	7.75
	DeepSeek-R1	2.95	2.00	9.38	5.80	1.40	2.10	2.09	0.78	6.20
	Qwen2.5-72B-Instruct	2.49	8.60	2.34	0.00	0.56	0.47	0.42	1.56	0.00
	Doubao-1.5-pro	4.88	12.40	7.03	1.79	1.40	2.10	1.67	2.34	6.20
	Doubao-1.5-thinking	10.46	30.60	11.72	7.14	1.69	4.21	3.35	0.78	4.65
	Gemini-2.5-Pro	14.63	27.80	28.91	8.93	7.58	9.81	10.04	9.38	8.53
	Llama-4-Maverick	3.52	2.00	15.63	4.46	2.25	2.80	2.51	0.00	6.98
MopenHands	GPT-4o	8.21	25.60	9.38	0.00	1.97	3.50	3.35	0.00	3.88
	OpenAI-o1	6.10	16.00	3.91	0.45	3.65	3.74	2.51	3.13	3.88
	OpenAI-o3-mini-high	7.55	20.40	10.16	0.45	3.37	2.34	5.02	1.56	6.98
	Claude-3.5-Sonnet	15.24	39.00	14.84	11.61	1.97	6.78	12.13	3.13	12.40
	Claude-3.7-Sonnet	19.32	52.20	21.88	2.23	5.06	7.48	15.90	8.59	14.73
	DeepSeek-V3	8.72	27.80	9.38	1.34	1.12	0.70	4.60	3.13	7.75
	DeepSeek-R1	8.02	26.00	8.59	0.45	2.53	0.00	4.60	2.34	4.65
	Qwen2.5-72B-Instruct	2.02	4.40	3.13	0.00	0.84	1.40	1.67	0.78	2.33
	Doubao-1.5-pro	2.91	8.80	0.78	0.00	1.12	1.64	0.84	0.00	3.10
	Doubao-1.5-thinking	11.49	27.80	10.94	5.36	9.55	6.07	3.35	3.91	5.43
	Gemini-2.5-Pro	21.62	45.80	12.50	22.32	16.29	12.60	14.64	5.47	9.30
	Llama-4-Maverick	7.46	14.40	6.25	8.04	4.78	5.61	5.02	3.13	3.10

Resolved rate (%) of different models on Multi-SWE-bench



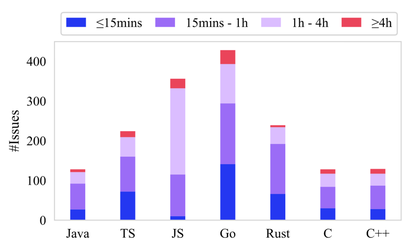
Histogram of issue description length (#tokens)

The above figure illustrates the distribution of issue lengths (in tokens) in Multi-SWE-bench, which follows a power law, with the majority of issue being under 1,000 tokens. To explore the effect of description length, the issues are categorized into 5 intervals: <100, 100-400, 400-700, 700-1000, and >1000 tokens. The absence of a corresponding bars indicates cases where no issues are successfully resolved.



Distribution of estimated time consumption of issues in Multi-SWE-bench

We recorded the estimated time required to resolve each issue, categorized into four buckets: ≤15 minutes, 15 minutes-1 hour, 1-4 hours, and ≥4 hours. We use this time-based annotation to define difficulty levels across all languages: easy (≤15 minutes), medium (15 mins-1 hour), and hard (≥1h).



References

[1] Jimenez C E, Yang J, Wettig A, et al. SWE-bench: Can Language Models Resolve Real-world Github Issues? [C]//The Twelfth International Conference on Learning Representations.

[2] Xia C S, Deng Y, Dunn S, et al. Agentless: Demystifying llm-based software engineering agents[J]. arXiv preprint arXiv:2407.01489, 2024.

[3] Yang J, Jimenez C E, Wettig A, et al. Swe-agent: Agent-computer interfaces enable automated software engineering[J]. Advances in Neural Information Processing Systems, 2024, 37: 50528-50652.

[4] Wang X, Li B, Song Y, et al. OpenHands: An Open Platform for AI Software Developers as Generalist Agents[C]//The Thirteenth International Conference on Learning Representations.