Auto-SLURP: A Benchmark Dataset for Evaluating Multi-Agent Frameworks in Smart Personal Assistant

Anonymous ACL submission

Abstract

Recently, multi-agent frameworks based on large language models (LLMs) have been developed rapidly. However, datasets to evaluate these multi-agent frameworks haven't been sufficiently developed. We present Auto-SLURP, a dataset designed to evaluate LLM-based multi-agent frameworks and assess whether they can support smart personal assistants. The dataset is derived from the SLURP dataset, which is originally created to train and test models' natural language understanding capabilities. We evaluate the entire end-to-end process for smart personal assistants, from language understanding to operation execution and also response generation, by relabeling the data and incorporating simulated servers and external services. This benchmark dataset proves sufficiently challenging to test the state-of-the-art multi-agent frameworks. Experiment results show that we are still a few steps away from achieving a reliable and smart personal assistant through multi-agent frameworks.

1 Introduction

Multi-agent frameworks based on LLMs have been developed rapidly in recent years(Li et al., 2023; Hong et al., 2024; Wu et al., 2023; Liu et al., 2024), enabling agents to communicate in the framework and perform various complex tasks. However, datasets to evaluate these multi-agent frameworks are still insufficient. Qin et al. (2023) introduce a tool-use benchmark, ToolBench, while Liu et al. (2023) propose a reasoning and decision-making benchmark AgentBench. However, these benchmarks assess only single aspects of the abilities, and they are somewhat simplistic for multi-agent frameworks, as they are originally designed for LLMs. Abdelnabi et al. (2023), MAgIC(Xu et al., 2023), SOTOPIA(Zhou et al., 2024), and LegalAgentBench(Li et al., 2024) propose benchmarks to evaluate LLM agents in multi-agent scenarios in specific domains such as games, social communications, and Chinese legal contexts. But these benchmarks focus only on the performances of LLM agents, and do not compare the open-source multi-agent frameworks. The need for benchmarks for multi-agent frameworks is urgent.

The intelligent personal assistant is one of the goals that humans have long expected from AI(Edu et al., 2020). In this work, we propose a benchmark, Auto-SLURP, to assess the intelligence of LLM-based multi-agent frameworks in building personal assistants. Auto-SLURP is derived from the existing SLURP dataset(Bastianelli et al., 2020; Liu et al., 2021), a natural language understanding dataset originally created for the development of smart home personal assistants. The Auto-SLURP dataset extends this scope to evaluate the ability to handle end-to-end tasks. We use the queries and intents from the original dataset and re-label the slots to better fit the end-to-end scenario.

Specifically, Auto-SLURP is designed to evaluate the full process of handling a user's query, from language understanding to operation execution and, finally, response generation. To enable this, we introduce simulated servers and integrate external services, which are essential for assessing the ability of LLM-based multi-agent frameworks to perform complex, real-world tasks. These components are critical for testing whether the frameworks not only understand a user's query but also carry out the necessary actions across multiple modules, such as controlling devices, querying APIs, and managing data from various external sources.

Furthermore, Auto-SLURP covers a wide range of tasks across various domains, including calendar management, media playback, information search, transportation coordination, and many others. The diversity of the dataset ensures that it serves as a reliable benchmark for evaluating the usability and performance of multi-agent frameworks. Our experiment results demonstrate that the Auto-SLURP

User	could you please email john saying i'm on leave	
	re-labeled	**original**
Intent	email_sendemail	email_sendemail
Slots	to_person: john, content: i'm on leave	person : john

Table 1: The example of the annotations.

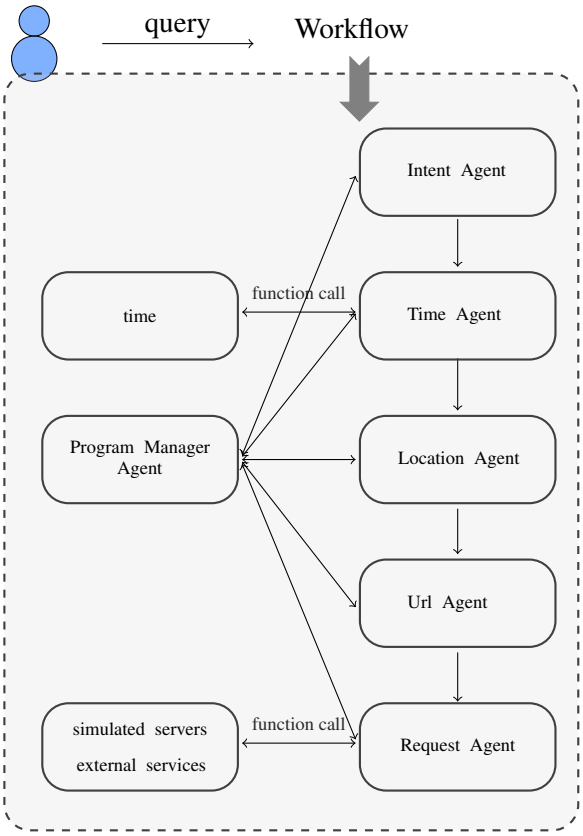

Figure 1: The workflow defined for the Auto-SLURP dataset.

dataset is sufficiently complex to challenge even the most advanced multi-agent frameworks. It also highlights that we are still a few steps away from achieving a fully reliable and smart personal assistant through these frameworks.

2 Related Works

Qin et al. (2023), Chen et al. (2023c), Zhuang et al. (2024), and Ye et al. (2024) provide tool-use benchmarks to evaluate the tool-using capabilities of LLMs. Liu et al. (2023) propose AgentBench to evaluate the reasoning and decision-making abilities of LLMs. However, these benchmarks focus on a single aspect of the abilities of multi-agent systems and are somewhat simplistic for evaluating complex multi-agent frameworks.

Abdelnabi et al. (2023), SOTOPIA(Zhou et al., 2024), AgentSense(Mou et al., 2024), and SocialBench(Chen et al., 2024) create social environments for artificial agents and evaluate their social intelligence. MAgIC(Xu et al., 2023) proposes several games, to assess LLM agents in multi-agent scenarios. Li et al. (2024) propose a comprehensive benchmark to evaluate agents in the Chinese legal domain. Ma et al. (2024) provide AgentBoard, a benchmark covering a range from embodied AI and game agents to web and tool agents. However, these benchmarks focus primarily on the performance of LLM agents, while they are not designed to compare the open-source multi-agent frameworks.

3 Dataset Construction

3.1 Creation of queries and annotations

We make modification to the SLURP dataset, which is collected for the development of smart personal assistants. Personal assistant systems are inherently complex control systems designed to respond to a wide variety of user commands. This dataset is initially released for the purpose of natural language understanding(Weld et al., 2022; Yang et al., 2017). The subtasks for natural language understanding in this dataset include intention detection and slot filling. In traditional methods, intent detection is a classification task, while slot filling is a sequence-to-sequence task. For example, for the user query "play kari jobe for me", the intent is "play_music", and the slot is "artist_name: kari jobe". In the original dataset, the slots are litimed to the entities mentioned in the utterance, while other crucial information is neglected, which would cause the server to fail to execute the user's command.

To adapt SLURP for our specific use case, we decide to use only the queries and their corresponding intents from the original dataset, while re-labeling the slots. Specifically, we add more slots and modify existing ones to cover all the information that needs to be sent to the operating servers. We also ensure that the slots can be generated by LLMs, as LLMs utilize the generation method, rather than the classification method. An example of the modified

	CamelAI	LangGraph	AutoGen	AgentLite
acc	0.21	0.32	0.44	0.46

Table 2: The results of the multi-agent frameworks.

	CamelAI	LangGraph	AutoGen	AgentLite
intent	54%	34%	68%	69%
time	18%	12%	9%	19%
location	-	-	-	7%
url	14%	13%	43%	19%
manager	9%	53%	13%	-
function_call	18%	-	-	-

Table 3: The failure reasons of the frameworks. Because one failure can be caused by multiple reasons, so they do not sum up to 100%.

samples is shown in Table 1, with our re-labeled sample in the middle column, and the original sample in the right column.

The dataset includes a wide range of tasks, from simple actions such as setting calendars or playing music, to more complex activities such as searching for information or managing transportation-related commands. We randomly select 1,000 samples from the training set and 100 samples from the testing set. This subset is considered sufficient for training and testing the performance of LLM-based multi-agent frameworks based on experimental results.

3.2 Collection of the end servers

We simulate the execution servers where user commands can be executed. This simulation allows us to verify whether the commands are processed and performed correctly, ensuring that the overall system functions as expected. In our training set, we have identified 23 distinct domains, and for each domain, we build a dedicated server to handle the relevant operations. Additionally, for certain domains which require further information to complete the query, such as search, weather, and news, we integrate external services, i.e., third-party APIs. Through these API calls, the systems can fetch the required information, ensuring that the user's request is handled efficiently and with up-to-date content.

4 Experiments

4.1 Setup

We compare several representative LLM-based multi-agent frameworks.

CamelAI (Li et al., 2023) introduces a cooperative multi-agent framework that allows communicative agents to autonomously collaborate toward completing tasks through role-playing.

AutoGen (Wu et al., 2023) presents a customizable multi-agent conversation framework that can integrate LLMs, humans, and tools.

LangGraph (2023) is built on top of LangChain (2022) and provides an easy way to create cyclical graphs, which is particularly useful for creating agent runtimes.

AgentLite(Liu et al., 2024) is a lightweight codebase for developing customized LLM agent systems. It enables researchers to easily build prototype applications, as well as integrate and evaluate new reasoning strategies and agent architectures.

For all multi-agent frameworks, we use GPT-4(Achiam et al., 2023) as the LLM. The prompts are created and adjusted during the setup phrase. The temperature is set as 0 to ensure that the LLM's responses are deterministic and fixed.

4.2 Defined workflows

We use the multi-agent frameworks to build systems for smart personal assistant. In the system, a program manager agent serves as the orchestration agent; it processes the user's input query and determines which agent will complete each subtask. We add an intent agent to predict the intent and slots. Additionally, we introduce a time and a location agent to format the time and location parameters, if applicable. If needed, the time agent will also invoke a time function call to provide the current date and time. We adopt a url agent to select the appropriate url from a list of provided urls, and a request agent to execute the tool function call for

USD/query	CamelAI	LangGraph	AutoGen	AgentLite
cost	0.52	0.14	0.80	0.55

Table 4: The costs of the frameworks.

the request. The overall process of the system is illustrated in Figure 1. The workflows for all the multi-agent frameworks are almost the same.

4.3 Evaluation

We use the successful execution rate as the evaluation metric, which measures the percentage of the queries that are completed successfully by the system. This metric assesses the reliability, efficiency, and ability of the framework to perform the intended actions without failure, providing a clear indication of its overall effectiveness. Additionally, we provide an evaluation tool to examine the results, allowing us to automatically measure the performance of the multi-agent frameworks.

5 Experiment Results

5.1 Results analysis

Table 2 presents the results of the multi-agent frameworks. As depicted in Table 2, CamelAI achieves the lowest accuracy score, while AgentLite performs the best. CamelAI's failure can be attributed to its difficulty in selecting the right tool to execute. The main issue with LangGraph is that it only combines the system prompt and all the agents' results into one list as input, without any adjustments. The AutoGen framework separates the prompts for the manager agent and the subtask agents, which improves operational results. The AgentLite framework adopts "think and react" methods in the process, which boosts its success rate. We also test other frameworks, such as AgentVerse(Chen et al., 2023b) and AutoAgents(Chen et al., 2023a), but these frameworks either lack a generalized orchestration policy to support this scenario or do not provide sufficient information for effective implementation. This highlights the complexity of designing a multi-agent framework.

We further calculate the errors caused by each agent and the function call part, and the results are shown in Table 3. From Table 3, it is clear that the main source of failure stems from the intent agent.

The costs of the frameworks are listed in Table 4. As shown, the costs are at the same leval for CamelAI, AutoGen, and AgentLite, but LangGraph has a significantly lower cost. We believe this is because LangGraph only uses the system prompt and all agents' results as input. Therefore, the cost for each query, ranging from 0.5 to 0.8, is reasonable for an advanced multi-agent framework in this scenario.

AutoGen	original	finetuned
acc	0.40	0.62

Table 5: The funetuning results for AutoGen.

5.2 Ablation

According to our analysis, most of the failures are caused by intent agent. To address this, we further finetune a model as the intent agent to see if it can improve the performance of multi-agent frameworks. We choose the open-source Llama 3 model(AI@Meta, 2024) for finetuning. Specifically, we finetune the LLAMA-3 8B model using the training set and use the finetuned version as the intent agent. We report the results on AutoGen framework, and the results are listed in Table 5. Compared to the framework that uses the original LLAMA-3 8B model, the finetuned version shows a performance improvement of 55%. This result demonstrates that improving LLMs can significantly enhance the overall performance of multi-agent frameworks.

Based on all above analysis, we believe that we are still a few steps away from achieving a fully reliable and smart personal assistant. Moreover, the key factors for a successful multi-agent framework include the generalization of orchestration policies, the prompt methods, the reasoning approaches, such as think and react, and the selection of LLMs that suit the scenario.

6 Conclusion

We present Auto-SLURP, a dataset designed to evaluate LLM-based multi-agent frameworks. We assess the end-to-end execution tasks, not just the nature language understanding tasks. By incorporating simulated servers and external services, we evaluate the capacity of the frameworks to complete the entire process. The dataset proves to be sufficiently challenging to test the state-of-the-art multi-agent frameworks.

7 Limitations

The dataset incorporates simulated servers and external services, which may not fully mimic the behavior of real-world systems. This could result in discrepancies between the performance of frameworks in the benchmark and their performance in live applications.

Additionally, the dataset's evaluation is heavily reliant on the performance of LLMs. Variations in the quality and capabilities of LLMs across different versions could influence the outcomes.

References

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schonherr, and Mario Fritz. 2023. Cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation.

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. 2023. Gpt-4 technical report.

AI@Meta. 2024. Llama 3 model card.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. Slurp: A spoken language understanding resource package. *arXiv preprint arXiv:2011.13205*.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F. Karlsson, Jie Fu, and Yemin Shi. 2023a. Autoagents: A framework for automatic agent generation. In *International Joint Conference on Artificial Intelligence*.

Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Xing Gao, Weizhou Shen, Xiaojun Quan, Chenliang Li, Ji Zhang, Fei Huang, et al. 2024. Roleinteract: Evaluating the social interaction of role-playing agents. *arXiv preprint arXiv:2403.13679*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Ya-Ting Lu, Yi-Hsin Hung, Cheng Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*.

Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, et al. 2023c. T-eval: Evaluating the tool utilization capability of large language models step by step. *arXiv preprint arXiv:2312.14033*.

Jide S Edu, Jose M Such, and Guillermo Suarez-Tangil. 2020. Smart home personal assistants: a security and privacy review. *ACM Computing Surveys (CSUR)*, 53(6):1–36.

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 a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

LangChain. 2022. Langchain.

LangGraph. 2023. Langgraph.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Haitao Li, Junjie Chen, Jingli Yang, Qingyao Ai, Wei Jia, Youfeng Liu, Kai Lin, Yueyue Wu, Guozhi Yuan, Yiran Hu, et al. 2024. Legalagentbench: Evaluating llm agents in legal domain. *arXiv preprint arXiv:2412.17259*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing naturalness and flexibility in spoken dialogue interaction: 10th international workshop on spoken dialogue systems*, pages 165–183. Springer.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Liangwei Yang, Zuxin Liu, Juntao Tan, Prafulla K. Choubey, Tian Lan, Jason Wu, Huan Wang, Shelby Heinecke, Caiming Xiong, and Silvio Savarese. 2024. Agentlite: A lightweight library for building and advancing task-oriented llm agent system. *Preprint*, arXiv:2402.15538.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. *arXiv preprint arXiv:2401.13178*.

Xinyi Mou, Jingcong Liang, Jiayu Lin, Xinnong Zhang, Xiawei Liu, Shiyue Yang, Rong Ye, Lei Chen, Haoyu Kuang, Xuanjing Huang, et al. 2024. Agentsense: Benchmarking social intelligence of language agents through interactive scenarios. *arXiv preprint arXiv:2410.19346*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang,

Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys*, 55(8):1–38.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation.

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. 2023. Magic: Benchmarking large language model powered multi-agent in cognition, adaptability, rationality and collaboration. *arXiv preprint arXiv: 2311.08562*.

Xuesong Yang, Yun-Nung Chen, Dilek Hakkani-Tür, Paul Crook, Xiujun Li, Jianfeng Gao, and Li Deng. 2017. End-to-end joint learning of natural language understanding and dialogue manager. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5690–5694. IEEE.

Junjie Ye, Guanyu Li, Songyang Gao, Caishuang Huang, Yilong Wu, Sixian Li, Xiaoran Fan, Shihan Dou, Qi Zhang, Tao Gui, et al. 2024. Tooleyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios. *arXiv preprint arXiv:2401.00741*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2024. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36.